Performance Analysis of Wireless Networks Based on Time-Scale Separation: A New Iterative Method

Luis Tello-Oquendo[a,*], Vicent Pla[a], Jorge Martinez-Bauset[a], Vicente Casares-Giner[a]

[a]ITACA. Universitat Politècnica de València. Camino de Vera s/n. 46022 Valencia, Spain

Abstract

The complexity of modern communication networks makes the solution of the Markov chains that model their traffic dynamics, and therefore, the determination of their performance parameters, computationally costly. However, a common characteristic of these networks is that they manage multiple types of traffic flows operating at different time-scales. This time-scale separation can be exploited to substantially reduce the computational cost. Following this approach, we propose a novel solution method named *Absorbing Markov Chains Approximation* (AMCA) based on the transient regime analysis. Briefly, we model the time the system spends in a series of subsets of states by a phase-type distribution and, for each of them, determine the probabilities of finding the system in each state of this subset until absorption. We compare the AMCA performance to that obtained by classical methods and by a recently proposed approach that aims at generalizing the conventional *quasi-stationary approximation*. We find that AMCA has a more predictable behavior, is applicable to a wider range of time-scale separations and achieves higher accuracy for a given computational cost.

Keywords: Cognitive radio networks, performance analysis, PH distribution, time-scale separation, wireless networks

[*]Corresponding author: Telf.: +34963879733

Email addresses: luiteloq@upv.es (Luis Tello-Oquendo), vpla@upv.es (Vicent Pla), jmartinez@upv.es (Jorge Martinez-Bauset), vcasares@upv.es (Vicente Casares-Giner)

1. Introduction

Nowadays, wireless communication networks incorporate sophisticated technology and algorithms to provide a wide range of services. In order to evaluate their performance and to understand the interactions among different components of these rather complex networks, the deployment of analytical models has become a common approach with multiple advantages. Accurate modeling of the wireless network events allows to determine performance parameters like the blocking probability, throughput, average transfer delay, and others [1, 2].

The increasing complexity of wireless networks in terms of size, different configurations, and the interactions among types of traffic flows makes modeling more challenging. From the modeling perspective, we normally encounter two main common characteristics in continuous-time Markov chain (CTMC) models of wireless networks. First, the cardinality of the state-space of their CTMC is large. Second, the multiple types of traffic flows evolve at different time-scales. While the first characteristic usually makes the exact solution of the CTMC computationally intractable, the second one allows us to apply specific solution approaches that exploit the time-scale separation to reduce the computational cost. We can structure the model into subsets of states by using the fact that transitions occur at a fast time-scale in the states belonging to the same subset, while transitions between subsets occur at a slower time-scale. Then, we can approximate the solution of the stationary probability distribution of the complete system by computing separately the stationary distribution of each subset, and then combining them to obtain the stationary distribution of the complete system. Once this is achieved, the performance metrics of the wireless network can be easily computed [3, 4].

The analysis of wireless networks based on time-scale separation has been proposed in recent studies [5, 6, 7, 8, 9, 10, 11, 12]. In many of them, the so-called *quasi-stationary approximation* (QSA) has been shown to be accurate and computationally efficient [6, 9, 10, 11]. However, when the gap between time-scales shortens, the accuracy of the method deteriorates to a point in which the method is no longer useful from a practical perspective.

In [7] a generalization of QSA (called GQSA) has been proposed. It can adjust the accuracy with a parameter called radius (R). In a recent study [13] we showed that, in some network scenarios, the accuracy achieved with GQSA improves as R increases. However, in other scenarios increasing R reduces the accuracy. More importantly, it is difficult to predict the scenarios in which the accuracy can be improved by increasing R.

The main contribution of this paper is a new approximation method applicable to a wide range of time-scale separations, and whose accuracy can be improved by increasing the computational cost. The proposed method is based on an original iterative approach named *Absorbing Markov Chains Approximation* (AMCA). In AMCA, the Markov model of the network is structured in levels and phases. Then, we analyze the transient regime at each level to determine the fraction of time that the system spends at each of its phases until a level change occurs. Once these fractions of time are found for all phases in all levels, a new approximation of the stationary distribution of the complete system is computed. We repeat the procedure until a predefined accuracy is satisfied. This iterative procedure is initialized with the solution obtained by QSA.

To evaluate the proposed method, we used it to analyze two different networks. One is a cognitive radio network (CRN) with two channel sets: one shared by primary and secondary users, and the other dedicated to the secondary users [14, 15]. The other is an integrated service network (ISN), where a single base station serves real-time and non-real-time traffic [16, 17]. We will refer to these two networks as the *test networks*. Note that we selected these test networks to apply the new approximation method to the same scenarios employed by previous approximate methods based on time-scale separation so that a fair comparison is carried out. Specifically, the CRN scenario was employed in [6] and the ISN scenario in [7]. However, the selection of these test networks does not limit the applicability of AMCA in any way.

We carry out two types of analysis in the test networks. First, we evaluate the behavior of AMCA at different time-scale separations. Second, we study the trade-off between accuracy and computational cost. We compare the performance of AMCA with that of QSA, GQSA, and a classical iterative method named *iterative aggregation/disaggregation* (IAD), which is particularly suited to these type of systems [4, Sect. 10.5]. Considering the range of time-scale separations at which we obtain an acceptable accuracy, the results show that AMCA outperforms the other methods.

The rest of the paper is structured as follows. Section 2 details the characteristics of the test networks analyzed and their associated CTMC models. Section 3 describes the quasi-stationary approximation and the related approximation methods based on time-scale separation. In Section 4 we present our approximation method called AMCA. Section 5 shows the numerical evaluation and the results. Finally, Section 6 draws the conclusions.

2. Wireless networks description and modeling

In this section, we detail the main characteristics of the test networks. We describe the performance metrics of interest and define a two-dimensional CTMC model for each network.

2.1. Cognitive radio network

As in [6, 18], we model the primary user (PU) and secondary user (SU) traffic at the session (connection) level and ignore interactions at the packet level (scheduling, buffer management, etc.). We assume an ideal MAC layer for SUs, which allows a perfect sharing of the allocated channels among the active SUs (all active SUs get the same bandwidth portion), introduce zero delay and whose control mechanisms consume zero resources. In addition, we also assume that an active SU can sense the arrival of a PU in the same channel instantaneously and reliably. In this sense, the performance parameters obtained can be considered as an upper bound.

The cognitive radio network has C_1 *primary channels* (PCs) that can be shared by PUs and SUs, and C_2 *secondary channels* (SCs) only for SUs. Let $C =$

$C_1 + C_2$ be the total number of channels in the network. Note that the SCs can be obtained from e.g., unlicensed bands, as proposed in [18]. This assumption is applicable to the *coexistence* deployment scenario for CRNs. Alternatively, as it might be of commercial interest for the primary and secondary networks to *cooperate*, the secondary channels may be obtained based on an agreement with the primary network [19]. A SU in the PCs might be forced to vacate its channel if a PU claims it to initiate a new session. As SUs support *spectrum handover*, a vacated SU can continue with its ongoing communication if a free channel is available. Otherwise, it is *forced to terminate*.

For the sake of mathematical tractability, Poisson arrivals and exponentially distributed service times are assumed. The arrival rate for PU (SU) sessions is λ_1 (λ_2), their service rate is μ_1 (μ_2), and requests consume 1 (1) channel when are accepted.

We denote by (i, j) the network state, when there are i ongoing PU sessions and j SU sessions. The set of feasible states is $\mathcal{S} := \{(i, j) : 0 \leq i \leq C_1, 0 \leq i + j \leq C\}$ and the cardinality of $\mathcal{S}$ is $|\mathcal{S}| = (C_1/2 + C_2 + 1)(C_1 + 1)$. The state-transition diagram of the network is depicted in Fig. 1. Given the set of feasible states and the transition rates among them, the global balance equations can be defined. Finally, the global balance equations together with the normalization equation can be solved to obtain the steady-state probabilities denoted as $\pi(i, j)$.

The network performance parameters are determined as follows:

$$P_{pu} = \sum_{k=0}^{C_2} \pi(C_1, k) \quad , \quad P_{su} = \sum_{k=C_2}^{C} \pi(C - k, k), \tag{1}$$

$$P_{ft} = \frac{\lambda_1 (P_{su} - \pi(C_1, C_2))}{\lambda_2 (1 - P_{su})}, \tag{2}$$

$$Th_{su} = \sum_{j=1}^{C} \sum_{i=0}^{Z} j\mu_2 \cdot \pi(i, j), \tag{3}$$

where P_{pu} is the PUs blocking probability, which clearly coincides with the one obtained in an Erlang-B loss model with C_1 servers; P_{su} is the SUs blocking probability, i.e., the fraction of SU sessions rejected upon arrival as they find the

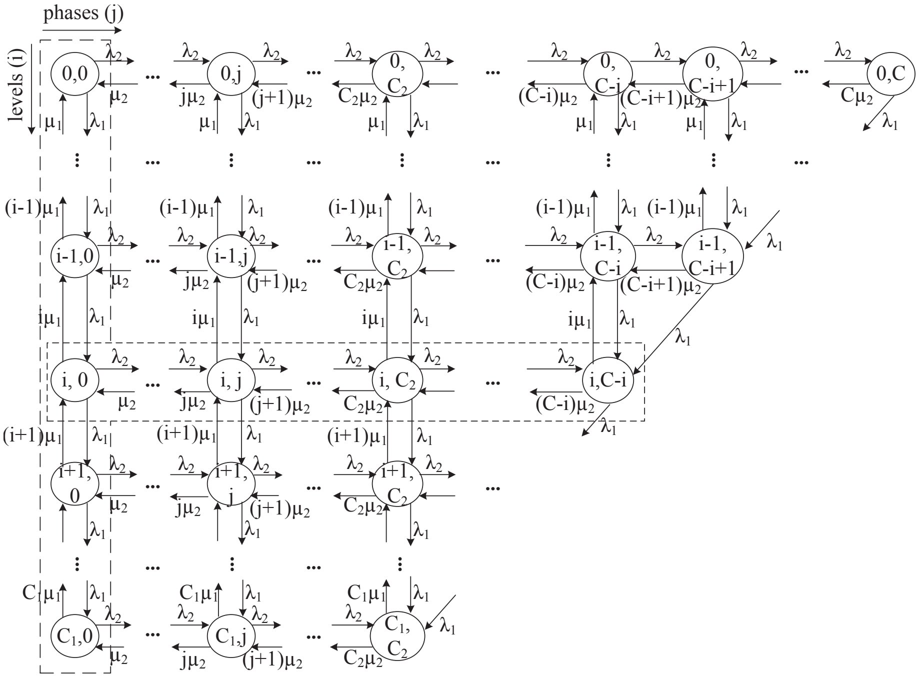


Figure 1: State-transition diagram, Cognitive Radio Network.

network full; P_{ft} is the forced termination probability of the SUs, i.e., the rate of SU sessions forced to terminate divided by the rate of accepted SU sessions; Th_{su} is the SUs throughput, i.e the rate of SU sessions successfully completed and $Z = \min(C_1, C - j)$.

2.2. Integrated service network

We use the same model defined in [7, 17] for an integrated service network, where a single base station serves real-time (RT) and non-real-time (NRT) traffic. We consider that a link with a total capacity of C Mbps is shared among RT and NRT communications.

We assume that all RT calls (sessions) are of the same class and are given strict priority over the NRT traffic. We denote by N_{rt} the maximum number of channels for RT calls. When a RT call arrives, it occupies 1 channel (if available) of rate c bps. Note that a RT call occupies 1 channel during its entire service duration to meet its required QoS; otherwise, it is blocked. We set N_{rt}, such that $N_{rt} \cdot c$ is sufficiently smaller than C to avoid starvation of the NRT traffic. Let $n_{rt}(t)$ be the stochastic process number of RT calls in the network at time t, $t \geq 0$.

The capacity not used by the RT traffic is evenly shared by the NRT flows according to the processor sharing (PS) discipline. Let $n_{nrt}(t)$ be the stochastic process number of NRT flows in the network at time t, $t \geq 0$. Then, $\{(n_{rt}(t), n_{nrt}(t))\}$ is the joint RT and NRT stochastic process. The available capacity for the NRT traffic at time t is given by $C_{nrt}(t) = C - n_{rt}(t) \cdot c$. The bit-rate of each admitted NRT flow at time t is $c_{nrt}(t) = C_{nrt}(t)/n_{nrt}(t)$, and it is updated after any RT or NRT accepted arrivals or departures. To satisfy the QoS of admitted NRT flows, the maximum number of concurrent NRT flows is limited to N_{nrt}. Accordingly, an NRT flow arriving at time t is blocked if $n_{nrt}(t) = N_{nrt}$.For the sake of mathematical tractability, we assume Poisson arrivals for RT and NRT requests with rates λ_{rt} and λ_{nrt} respectively. Also, the service time of each admitted RT request is exponentially distributed with rate μ_{rt}. The size of the flows generated by the NRT sessions are exponentially distributed with mean L (bits). Note that, the service time of NRT flows (transfer delay) depends on the available resources.

We denote by (i, j) the network state, when there are i ongoing RT calls and j NRT flows. Let $\mathcal{S}$ be the set of feasible states as $\mathcal{S} := \{(i, j) : 0 \leq i \leq N_{rt}, 0 \leq i + j \leq N_{rt} + N_{nrt}\}$ and the cardinality of $\mathcal{S}$ is $|\mathcal{S}| = (N_{rt} + 1)(N_{nrt} + 1)$. The state-transition diagram of the network is depicted in Fig. 2.

As before, given the set of feasible states and the transition rates among them, the global balance equations can be defined. Finally, the global balance equations together with the normalization equation can be solved to obtain the steady-state probabilities denoted as $\pi(i, j)$. Clearly, the service rate of NRT flows varies according to the number of RT calls in the network as:

$$\mu_{nrt}^{(i)} = \frac{C - i \cdot c}{L}. \tag{4}$$

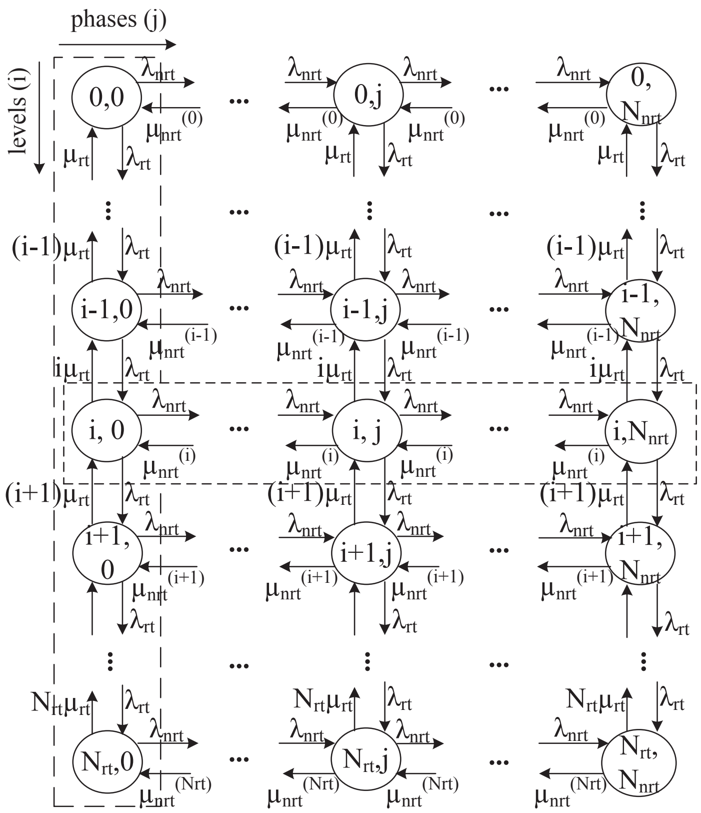


Figure 2: State-transition diagram, Integrated Service Network.

The network performance parameters are determined by

$$P_{nrt} = \sum_{k=0}^{N_{rt}} \pi(k, N_{nrt}), \tag{5}$$

$$E\left[X_{nrt}\right] = \sum_{j=1}^{N_{nrt}} \sum_{i=0}^{N_{rt}} j \cdot \pi(i,j), \tag{6}$$

$$E\left[D_{nrt}\right] = \frac{E\left[X_{nrt}\right]}{\lambda_{nrt}(1 - P_{nrt})}, \tag{7}$$

where P_{nrt} is the blocking probability of NRT flows, $E\left[X_{nrt}\right]$ is the mean number of NRT flows in the network and $E\left[D_{nrt}\right]$ is the average transfer delay of NRT flows. Note that (7) is a direct application of Littles's law.

3. Approximate solution methods

In this section, we describe the approximation methods based on time-scale separation that have appeared in the literature.

If the wireless network model in its entirety is too large or complex to analyze, the state-space may be partitioned into disjoint subsets of states. This partition is made by considering the essence of time-scale separation: the interactions among the states of a subset are strong (high frequency of events) but the interactions among the states of different subsets are weak (low frequency of events). Such models are sometimes referred to as nearly completely decomposable (NCD), nearly uncoupled, or nearly separable [4, 20].

Then, we organized the state-space of the CTMC of each test network into levels and phases (as shown in Fig. 1 for CRN and in Fig. 2 for ISN), so that we call levels the subsets in the y-axis and we call phases the states in the x-axis contained in the same level. The state transitions between states of the same level (phases) very often occur at a higher rate than the transitions between states of different levels, i.e., a high number of phase changes (in the same level) occur before a level change.

Next (Sections 3.1, 3.2, and 3.3) we detail how to compute the approximate steady-state probabilities with each of the approximation methods. With these approximated values, the performance parameters of each test network can be computed using (1)–(3) for the CRN, and (5)–(7) for the ISN.

3.1. Quasi-stationary approximation (QSA)

The simplest approximation based on time-scale separation is the so called quasi-stationary (or, quasi-static) approximation (QSA) [21, 9, 10, 5, 22]. This approximation produces easily computable and accurate results when the separation of the time-scales is large.

We start by obtaining the probability distribution of finding the system at each level, i.e., the slow transitions (PUs in the CRN or RT traffic in the ISN) and denote it by

$$\boldsymbol{\pi} = [\pi(0) \;\; \pi(1) \;\; \cdots \;\; \pi(i) \;\; \cdots \;\; \pi(y)], \tag{8}$$

where y represents the highest level of the CTMC. Then, for each level, we proceed to obtain the conditional probability distributions of finding the system at each phase. This conditional distribution for level i is given as

$$\widehat{\boldsymbol{\pi}}(i) = [\widehat{\pi}(0|i) \;\; \widehat{\pi}(1|i) \;\; \cdots \;\; \widehat{\pi}(j|i) \;\; \cdots \;\; \widehat{\pi}(x|i)], \tag{9}$$

where x represents the highest phase in level i. These are approximate probability distributions because they are computed assuming that when the process enters a level, the time spent there is sufficiently large so that the stationary regime is reached.

Finally, the approximate stationary distribution of the system is computed using (8) and (9) as follows

$$\pi(i,j) \approx \widehat{\pi}(i,j) = \pi(i) \cdot \widehat{\pi}(j|i). \tag{10}$$

3.2. Generalized quasi-stationary approximation (GQSA)

In GQSA [7], the system stationary distribution can be approximated as in QSA, but now a set of adjacent levels is considered for the analysis of level i, rather than just level i. For that, the parameter R indicates the number of adjacent levels to consider. Clearly, R allows to adjust the trade-off between accuracy and computational cost.

The number of levels required at each GQSA step is equal to $2R+1$. Note that QSA can be seen as a special case of GQSA with $R = 0$.

Let $\Omega(i)$ be the set of states contained in level i and its $2R$ closest levels and denote by $\pi_{\Omega(i)}(i,j)$ the stationary distribution of the CTMC restricted to the states in $\Omega(i)$ and the transitions between them. Then, the approximate stationary distribution of the system (i,j) is computed as follows

$$\pi(i,j) \approx \overline{\pi}(i,j) = \pi(i) \cdot \frac{\pi_{\Omega(i)}(i,j)}{\sum_j \pi_{\Omega(i)}(i,j)}. \tag{11}$$

3.3. Iterative aggregation/disaggregation approximation (IAD)

In the IAD method, as with QSA, the idea is to partition the state-space into aggregates (subsets of states), estimate the probability that the system is in a particular aggregate, estimate the conditional probabilities of being in each state of every aggregate, and then combine them to obtain an approximation of the stationary distribution of the complete system [4, Chap.10], [23].

In our test networks, the transition rate matrix $\boldsymbol{Q}$ has the following NCD block structure:

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{U}_1 & & & \\ \boldsymbol{L}_2 & \boldsymbol{D}_2 & \boldsymbol{U}_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \boldsymbol{L}_{n-1} & \boldsymbol{D}_{n-1} & \boldsymbol{U}_{n-1} \\ & & & \boldsymbol{L}_n & \boldsymbol{D}_n \end{bmatrix}.$$

Next we detail our specific implementation of the IAD method:

1. Use the QSA to determine the initial stationary distribution $\boldsymbol{\pi}_i^{(0)}$.

2. Apply the following iteration until the convergence test is met:

$$\boldsymbol{\pi}_i^{(k+1)} = \begin{cases} \boldsymbol{\pi}_{i+1}^{(k)} \boldsymbol{W}_i, & i = 1, \\ \boldsymbol{\pi}_{i-1}^{(k+1)} \boldsymbol{V}_i + \boldsymbol{\pi}_{i+1}^{(k)} \boldsymbol{W}_i, & i = 2, \dots, n-1 \\ \boldsymbol{\pi}_{i-1}^{(k+1)} \boldsymbol{V}_i, & i = n, \end{cases} \tag{12}$$

   where $\boldsymbol{\pi} = [\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_n]$, $\boldsymbol{\pi}_i$ is the sub-vector of $\boldsymbol{\pi}$ that corresponds to level i, $\boldsymbol{\pi}_i^{(k)}$ is the sub-vector value at the k-th iteration, $\boldsymbol{V}_i = -U_{i-1} D_i^{-1}$ and $\boldsymbol{W}_i = -L_{i+1} D_i^{-1}$.

3.3.1. Convergence Test

Using the solution obtained by the QSA as $\boldsymbol{\pi}^{(0)}$, the iterative procedure terminates when the following convergence test is met:

$$\hat{e}_r(z^{(k)}) \ = \ \frac{|z^{(k-1)} - z^{(k)}|}{z^{(k)}} \leq \varepsilon\,, \tag{13}$$

where z is one of the performance metrics in $\{P_{su}, P_{ft}, Th_{su}\}$ for the CRN evaluation, or in $\{P_{nrt}, E\left[D_{nrt}\right]\}$ for the ISN evaluation. We iterate until the estimated error for *all* performance parameters of the test network is less than a predefined ε.

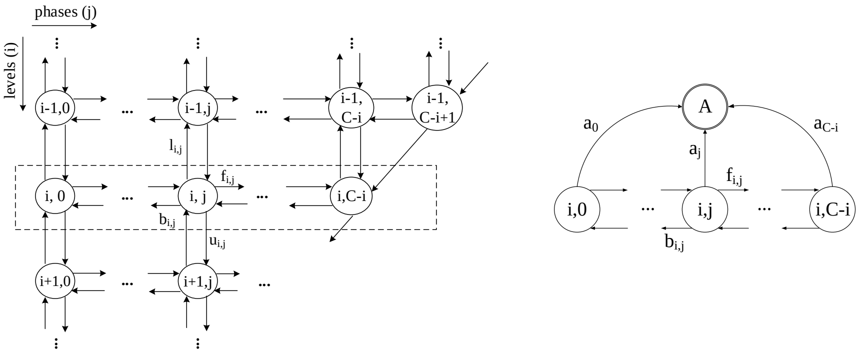
4. Absorbing markov chains approximation

In this section, we present the proposed *Absorbing Markov Chains Approximation* (AMCA), which is an iterative method. As in previous methods, in AMCA we also structure the CTMCs of the test networks in levels and phases. We analyze the system in the transient regime and model the time spent by the system in a level as a phase-type distribution. For each level, we determine the fraction of time the system spends in each of the phases of the level, i.e., since entering the level until departing from it. For doing so, AMCA requires to know the probabilities of finding the system in each phase of the adjacent levels. Thus, an iterative method is devised, that is terminated using the same convergence test detailed in 3.3.1. Next, we detail the procedure, the equations and variables involved in our method, and finally present the AMCA algorithm.

4.1. Approximation method

In the QSA it is assumed that, when the process enters a level, it takes an infinitely long time to leave that level. In our method, we assume that although the sojourn time in a level will be typically large (consistently with the large separation between time-scales), it is finite. Then, we obtain the probability that the process is in phase j of level i, given that the process is in level i, as the fraction of time that the process spends in phase j during a sojourn of the CTMC in level i.

In order to study the sojourn time in a level, we model the states of each level i of the original CTMC as transient states of an absorbing Markov chain with an absorbing state A, where all states outside level i have been lumped together. To illustrate this, in Fig. 3(a) we represent a region of the CRN state-transition



(a) CRN, level i and its adjacent levels

(b) CRN, absorbing Markov chain of level i

Figure 3: Transitions and absorption state

diagram (level i and its adjacent levels) and in Fig. 3(b) the absorbing Markov chain corresponding to level i. As a result, the outgoing transitions from a state (i, j) that in the original CTMC go to a state inside level i (those with rates $b_{i,j}$ and $f_{i,j}$) are directly mapped onto the absorbing Markov chain. In contrast, all transitions from a state (i, j) that in the original CTMC go to a state outside level i (those with rates $l_{i,j}$ and $u_{i,j}$) are aggregated into a single transition in the absorbing Markov chain that leads to the absorbing state (i.e., $a_j = l_{i,j} + u_{i,j}$).

Note that, if we knew the probabilities with which the sojourn time in a level is initiated at each of its phases, then the conditional probabilities obtained by this method would be the exact ones. However, unless the original CTMC has some special structure (for instance, if each level can only be entered by exactly one of its phases), these initial probabilities cannot be obtained without having the stationary distribution of the whole CTMC.

We propose to use the QSA to estimate the initial stationary distribution of the complete system. Then, we obtain the fractions of time spent at each of the phases of each level before absorption. Finally, we combine the estimation of the conditional probabilities of finding the system at the phases of each level, and the probability distribution of finding the system at each level, to determine a new approximation for the stationary distribution of complete system. This way, we obtain a refinement of the initial approximate stationary distribution. The same process can be repeated iteratively to further improve the approximation.

Based on the basic properties of PH distributions (see AppendixA), the iterative procedure described above is defined by the following equations:

$$\boldsymbol{v}_i^{(k-1)} = \pi_{i-1}(\widetilde{\boldsymbol{\pi}}_{i-1}^{(k-1)}\boldsymbol{U}_{i-1}) + \pi_{i+1}(\widetilde{\boldsymbol{\pi}}_{i+1}^{(k-1)}\boldsymbol{L}_{i+1}) \ , \tag{14}$$

$$\boldsymbol{\alpha}_i^{(k-1)} = \left[\boldsymbol{v}_i^{(k-1)}\boldsymbol{e}\right]^{-1} \boldsymbol{v}_i^{(k-1)} \ , \tag{15}$$

$$\widetilde{\boldsymbol{\pi}}_i^{(k)} = \left[\boldsymbol{\alpha}_i^{(k-1)}(-T_i^{-1})\boldsymbol{e}\right]^{-1} \boldsymbol{\alpha}_i^{(k-1)}(-T_i^{-1}) \ , \tag{16}$$

where

- the superscript (k) denotes the iteration number and $\boldsymbol{e}$ is a column vector of ones of appropriate dimension.
- $\boldsymbol{v}_i^{(k-1)}$ is a row vector that contains the input rates to each state of the level i. Its initial value is given by

  $$\boldsymbol{v}_i^{(0)} = \pi_{i-1}(\widetilde{\boldsymbol{\pi}}_{i-1}^{(0)}\boldsymbol{U}_{i-1}) + \pi_{i+1}(\widetilde{\boldsymbol{\pi}}_{i+1}^{(0)}\boldsymbol{L}_{i+1}), \tag{17}$$

  where $\boldsymbol{U}_{i-1}$ is a matrix of suitable dimension with the transition rates from level $i-1$ to level i and $\boldsymbol{L}_{i+1}$ is a matrix of suitable dimension with the transition rates from level $i+1$ to level i.
- $\boldsymbol{\alpha}_i^{(k-1)}$ is the initial probability row vector for level i, i.e., the j-th element of this vector, $\alpha_i^{(k-1)}(j)$, is the probability that the process enters through phase j when it visits level i. Its initial value is given by

  $$\boldsymbol{\alpha}_i^{(0)} = \left[\boldsymbol{v}_i^{(0)}\boldsymbol{e}\right]^{-1} \boldsymbol{v}_i^{(0)}. \tag{18}$$

- $\widetilde{\boldsymbol{\pi}}_i^{(k)}$ is a row vector containing the fractions of time the process spends in each phase of level i before absorption, e.g., the j-th element of this vector, $\widetilde{\pi}_i^{(k)}(j)$, is the fraction of time the process spends in the phase j of level i before absorption. Its initial value is given by QSA

$$\widetilde{\boldsymbol{\pi}}_i^{(0)} = \widehat{\boldsymbol{\pi}}(j|i), \tag{19}$$

where $\widehat{\boldsymbol{\pi}}(j|i)$ is the distribution of probabilities of

- CRN: finding j ongoing SU sessions in an M/M/(C$-i$)/(C$-i$) system with only SUs.
- ISN: finding j NRT flows in an M/M/1/N-PS system with only NRT traffic.

- π_i is the probability of finding the system at level i. It is the probability of finding i PUs in the CRN or i ongoing RT sessions in the ISN. It is computed using simple recursions since their corresponding CTMC are one-dimensional birth-and-death processes.

Finally, the steady-state probability distribution can be approximated as

$$\pi(i,j) \approx \widetilde{\pi}^{(k)}(i,j) = \pi_i \cdot \widetilde{\pi}_i^{(k)}(j)\,. \tag{20}$$

To compute the approximate values of the performance parameters, we use (1)–(3) for the CRN, and (5)–(7) for the ISN, with the distribution of probabilities defined in (20). Finally, the proposed iterative method may be halted once the predefined convergence test defined in Section 3.3.1 is satisfied.

Algorithm 1 summarizes the procedure used to conduct the performance evaluation of the test networks with AMCA.

5. Numerical evaluation and results

We perform two types of analysis. First, we evaluate the behavior of the approximation methods when the separation of time-scales varies. Second, we study the trade-off between accuracy and computational cost. The results of these analyses are presented in Sections 5.1 and 5.2 respectively.

As a baseline for our study, we implemented the exact solution of the CTMC associated with each test network, in order to evaluate the error of the approximation methods. For the sake of comparison, we used test network sizes that

Algorithm 1 Iterative Absorbing Markov Chains Approximation Method

```
1: Set $\pi^{(0)} \approx \widehat{\pi}(i,j) = \pi(i)\cdot\widehat{\pi}(j|i)$ as the initial approximation to the $\boldsymbol{\pi}$ solution
   computed by QSA. Set $k = 1$.
2:    a) Compute the vector of input rates to each state of the level $i$, $\boldsymbol{v}_i^{(k-1)}$,
         using (14).
      b) Compute the initial probability vector of level $i$, $\boldsymbol{\alpha}_i^{(k-1)}$, using (15).
      c) Compute the conditional probabilities vector, $\widetilde{\boldsymbol{\pi}}_i^{(k)}$, using (16).
3: Compute the new approximate steady-state probability distribution
   $\pi(i,j) \approx \widetilde{\pi}^{(k)}(i,j)$ using (20).
4: Compute the performance metrics of each network: using (1)–(3) for CRN,
   and (5)–(7) for ISN.
5: Apply the convergence test to iterations $k$ and $k-1$, using (13):
      If satisfactory, then stop. Otherwise, set $k = k + 1$ and go to step 2.
```

allowed the computation of the exact solution with reasonable execution time and memory requirements. In addition, we implemented GQSA and the IAD method to validate AMCA and to compare its performance with that of the other methods in terms of accuracy and computational cost.

The accuracy of the methods is measured as the relative error (e_r) of each performance parameter. For instance, the relative error of the SUs blocking probability in the CRN is computed as

$$e_r(P_{su}) = \frac{|P_{su}^E - P_{su}^A|}{P_{su}^E}, \tag{21}$$

where P_{su}^E is the exact SUs blocking probability and P_{su}^A is the approximate SUs blocking probability. Note that (21) is the (exact) relative error whereas $\hat{e}_r(P_{su}^{(k)})$, as defined in (13), is an estimation of it.

We evaluate the performance of the test networks for different sizes (number of channels available for each type of user or flow) and different load conditions. For the SUs in the CRN, we analyze their blocking probability, forced termination probability and throughput. We consider the following values for the number of primary channels: $C_1 = \{70, 80, 90, 100, 120, 140\}$. For each of them, we consider the following values for the number of secondary channels: $C_2 = \{10, 20, 40, 60\}$.

For the NRT traffic in the ISN, we determine its blocking probability and the average transfer delay. Keeping $c = 64\,\text{kb/s}$ and $L = 500\,\text{kB}$ constant, we consider the following values for the total link capacity of the network: $C = \{1.92, 7.68, 10\}\,\text{Mbps}$, which are a similar to the ones used in [7].

We set the service rates to $1\,\text{s}^{-1}$, and then we adjust the arrival rates to obtain two load conditions: low (L) and high (H), which correspond to blocking probabilities of $1 \cdot 10^{-3}$ and $5 \cdot 10^{-2}$ respectively. Combining the two load conditions for each user type or traffic category, we obtain four different configurations:

LL low load condition for PUs (RT traffic), and low load condition for SUs (NRT traffic).

LH low load condition for PUs (RT traffic), and high load condition for SUs (NRT traffic).

HL high load condition for PUs (RT traffic), and low load condition for SUs (NRT traffic).

HH high load condition for PUs (RT traffic), and high load condition for SUs (NRT traffic).

Next we present the results obtained for the two types of analysis.

5.1. Behavior of the approximation methods when the separation of time scales varies

We analyze the behavior of the approximation methods as a function of the time-scale separation. For that, we first configure the test networks to a specific load condition (LL, LH, HL or HH). Then, we use an accelerating factor f, $10^{-5} \leq f \leq 10^5$, to equally accelerate or decelerate both the arrival and service rates of the components with high priority in the networks (PUs in

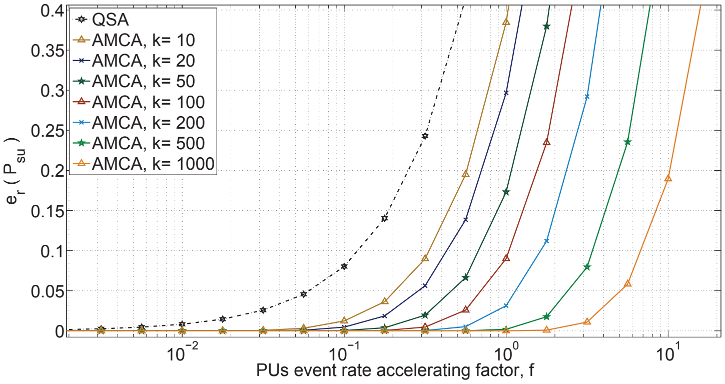


Figure 4: Relative error for the SUs blocking probability in LH load condition: $\lambda_1 = 110.90\,\mathrm{s}^{-1}$, $\mu_1 = 1\,\mathrm{s}^{-1}$, $C_1 = 140$; $\lambda_2 = 87.69\,\mathrm{s}^{-1}$, $\mu_2 = 1\,\mathrm{s}^{-1}$, $C_2 = 60$; k denotes the number of iterations performed.

the CRN or RT traffic in the ISN), while keeping the offered traffic constant. For instance, in the CRN, for each value of f the PU arrival and service rates are obtained as $\lambda_1(f) = f \cdot \lambda_1$ and $\mu_1(f) = f \cdot \mu_1$. Note that the offered traffic $\lambda_1(f)/\mu_1(f) = \lambda_1/\mu_1$ is independent of f. As f approaches 0 the event rate of high priority users gets lower. Therefore, the behavior of the systems gets better aligned with the hypothesis underlying all approximation methods considered here: high priority users are nearly static from the perspective of low priority users. As a consequence, it is expected that the accuracy of all approximation methods improves when f decreases toward 0, and conversely, degrades when f grows.

In Figs. 4–7 we show the relative error of the blocking probability against the accelerating factor f for LH and LL load conditions. With regard to the other performance metrics and load conditions, the behavior of the approximation methods is qualitatively similar, but for conciseness their results are not shown.

We can quantify the validity range of an approximation in the time-scale domain as the maximum value of f for which a certain accuracy is met. Figures 4 and 5 show that AMCA can extend the validity range of QSA at the expense of higher computational cost.

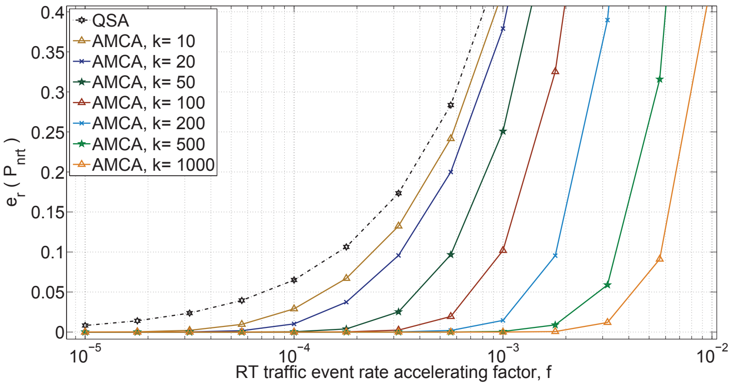


Figure 5: Relative error for the NRT flows blocking probability in LL load condition: $\lambda_{rt} = 75.24\,\mathrm{s}^{-1}$, $\mu_{rt} = 1\,\mathrm{s}^{-1}$, $N_{rt} = 100$; $\lambda_{nrt} = 1.27\,\mathrm{s}^{-1}$, $N_{nrt} = 140$; $C = 10$ Mbps, $c = 64$ kbps, $L = 4$ Mb; k denotes the number of iterations performed.

In Figs. 6 and 7 we compare AMCA with GQSA and IAD in terms of accuracy at different time-scales. These results were obtained by the following procedure. We measured the time to execute GQSA with a given radius R ($\mathrm{GQSA_R}$). Then, for IAD and AMCA we performed the maximum number of iterations such that computation time not higher than computation time of $\mathrm{GQSA_R}$. These results are labeled as $\mathrm{AMCA_R}$ and $\mathrm{IAD_R}$. For instance, the curve for $\mathrm{AMCA_1}$ represents the result obtained iterating AMCA while the computation time not exceeding that of $\mathrm{GQSA_1}$.

The following observations can be made from Figs. 6 and 7:

- As expected, with all approximation methods, when the accelerating factor f decreases ($f \to 0$) the approximate values of all evaluated performance parameters tend to their exact values.

- Increasing the radius in GQSA not always ensures a reduction of the relative error [13]. Figure 6 illustrates this behavior; as can be seen for $f > 10^{-1}$, $\mathrm{GQSA_1}$ has better accuracy than $\mathrm{GQSA_2}$ and $\mathrm{GQSA_3}$.

- AMCA outperforms GQSA and IAD in terms of validity range, i.e., with the same computation time AMCA is able to achieve a validity range

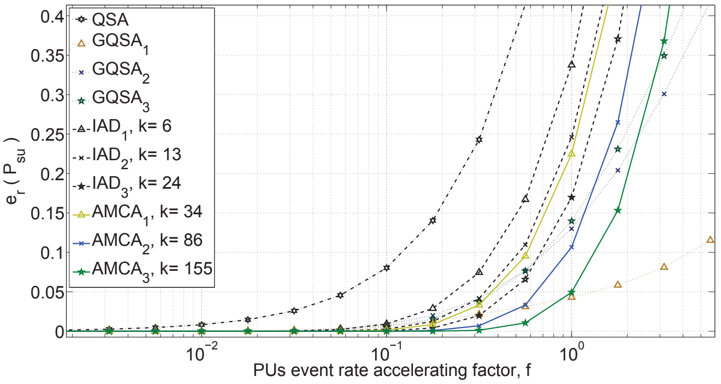


Figure 6: Relative error for the SUs blocking probability in LH load condition: $\lambda_1 = 110.90\,\text{s}^{-1}$, $\mu_1 = 1\,\text{s}^{-1}$, $C_1 = 140$; $\lambda_2 = 87.69\,\text{s}^{-1}$, $\mu_2 = 1$ s^{-1}, $C_2 = 60$; k denotes the number of iterations performed.

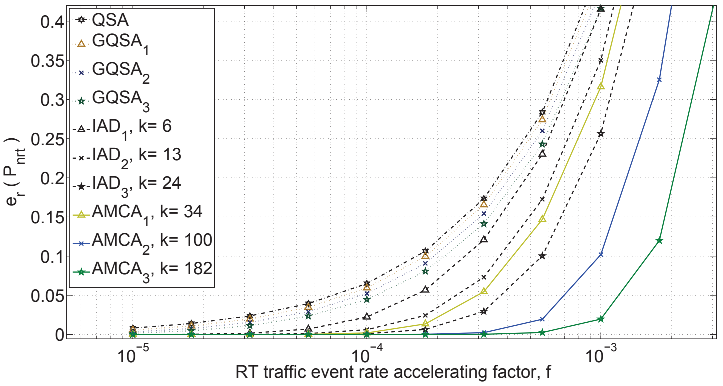


Figure 7: Relative error for the NRT flows blocking probability in LL load condition: $\lambda_{rt} = 75.24\,\text{s}^{-1}$, $\mu_{rt} = 1\,\text{s}^{-1}$, $N_{rt} = 100$; $\lambda_{nrt} = 1.27\,\text{s}^{-1}$, $N_{nrt} = 140$; $C = 10$ Mbps, $c = 64$ kbps, $L = 4$ Mb; k denotes the number of iterations performed.

wider than that of GQSA and IAD method. For instance, see in Fig.7 the curves of QSA, GQSA$_3$, IAD$_3$ and AMCA$_3$; for a relative error lower than 0.05, AMCA is able to achieve a validity range that is approximately 18 times wider than that of QSA, whereas GQSA and IAD method are able to achieve a validity range of approximately 2 and 6 times wider than that of QSA, respectively. A similar behavior was observed for all load

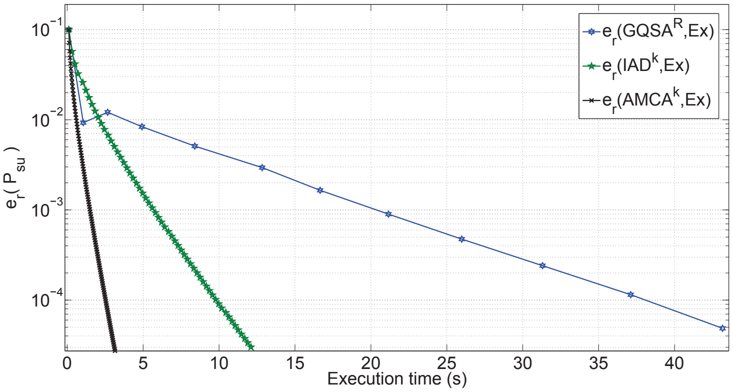


Figure 8: Relative error for the SUs blocking probability in LH load condition: $\lambda_1 = 13.90\,\mathrm{s}^{-1}$, $\mu_1 = 0.13\,\mathrm{s}^{-1}$, $C_1 = 140$; $\lambda_2 = 87.69\,\mathrm{s}^{-1}$, $\mu_2 = 1\,\mathrm{s}^{-1}$, $C_2 = 60$.

conditions and network sizes.

5.2. Trade-off between accuracy and computational cost

In this section, we analyze the trade-off between accuracy and computational cost. Figures 8 (CRN) and 9 (ISN) illustrate the evolution of the relative error of the blocking probability with the execution time. To obtain these results, we set f such that the relative error obtained by QSA for the studied parameter is 10%. Recall that the stationary distribution obtained by QSA is used for the initial values of IAD and AMCA.

It is worth nothing that GQSA is not an iterative method in the sense that it can be executed for any radius value, $R = n$, without having previously obtained the results for $R = 0, 1, \ldots, n-1$. However, there is no available method that allows to find the appropriate n to achieve a given accuracy. Although it does not always occur (for example in Fig. 8 the accuracy decreases from GQSA_1 to GQSA_2), it is expected that the obtained accuracy tend to improve when R is increased. Thus, in our comparative study we increase R until a predefined convergence test (estimated relative error) is met, roughly mimicking the operation of the other two iterative methods.

We can observe in Figs. 8 and 9 how the accuracy of each method evolves

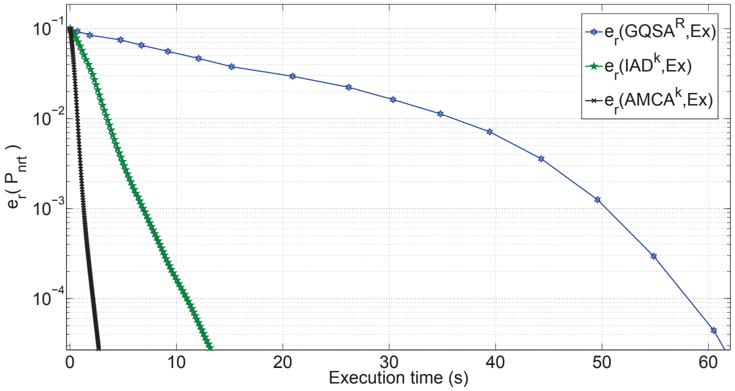


Figure 9: Relative error for the NRT flows blocking probability in LL load condition: $\lambda_{rt} = 1.25 \cdot 10^{-2}\,\mathrm{s}^{-1}$, $\mu_{rt} = 1.65 \cdot 10^{-4}\,\mathrm{s}^{-1}$, $N_{rt} = 100$; $\lambda_{nrt} = 1.27\,\mathrm{s}^{-1}$, $N_{nrt} = 140$; $C = 10$ Mbps, $c = 64$ kbps, $L = 4$ Mb.

as the number of iterations, and hence the computation time, increases. Note that the execution time to obtain a determined accuracy with AMCA is lower than that of GQSA and IAD.

Tables 1 and 2 show the relative error estimations ($\hat{e}_r$), the execution times of all the approximation methods, and (for comparative effects) the required time to obtain the exact system stationary distribution for the CRN and the ISN test networks respectively. To obtain these results, we have considered scenarios where the initial (exact) relative errors obtained by QSA were 20% and 40%, and with different load configurations. These two values represent scenarios in which the separation between time-scales is not long enough so that QSA cannot provide a sufficiently accurate approximation and, as a consequence, an enhanced method is required.

In practice, where the exact value of the error is not available, a stop criterion is needed for the normal use of the approximation methods. Here the estimated relative error ($\hat{e}_r$) is used in the stop criterion.

The threshold that $\hat{e}_r$ must fall below for the procedure to stop is chosen heuristically such that: i) the (exact) relative error obtained after the iterative procedure stops must be $e_r \leq 10^{-2}$; ii) it is unique for each method and for all

Table 1: Relative Error Analysis - Cognitive Radio Network

Load Config	$e_r^{(0)}$ (%)	$\hat{e}_r^{(k)}$			Execution Time (s)			
		GQSA	IAD	AMCA	GQSA	IAD	AMCA	Exact
LL	20	9.1e-4	9.9e-6	9.9e-6	184.9	134.5	10.4	209.1
LL	40	0.01	2.1e-5	9.9e-6	197.9	209.8	43.7	210.6
LH	20	6.6e-4	9.9e-6	9.9e-6	88.1	25.5	3.9	207.5
LH	40	9.8e-4	9.9e-6	9.9e-6	14.3	55.7	8.1	208.8
HL	20	7.1e-4	9.9e-6	3.9e-6	91.2	79.0	0.8	207.3
HL	40	6.3e-4	9.9e-6	9.9e-6	132.5	187.1	55.0	209.5
HH	20	6.8e-4	9.6e-6	9.8e-6	54.8	30.3	4.4	207.5
HH	40	7.0e-4	9.9e-6	9.7e-6	100.4	65.0	9.2	212.1

Table 2: Relative Error Analysis - Integrated Service Network

Load Config	$e_r^{(0)}$ (%)	$\hat{e}_r^{(k)}$			Execution Time (s)			
		GQSA	IAD	AMCA	GQSA	IAD	AMCA	Exact
LL	20	9.6e-4	9.7e-6	9.9e-6	53.4	25.7	3.1	118.8
LL	40	7.4e-4	9.9e-6	9.9e-6	61.9	46.8	6.3	120.0
LH	20	3.5e-4	4.9e-6	9.5e-6	96.1	23.5	3.4	118.5
LH	40	6.4e-3	1.1e-4	9.9e-6	108.9	118.4	35.2	118.8
HL	20	7.7e-4	9.9e-6	4.3e-6	20.5	12.1	1.2	118.8
HL	40	5.8e-4	9.6e-6	8.8e-6	34.6	24.7	2.3	118.8
HH	20	9.7e-4	7.6e-6	5.0e-6	38.1	15.8	2.8	118.8
HH	40	3.6e-4	9.7e-6	9.5e-6	77.6	69.1	9.2	118.9

the studied configurations. Note however that $\hat{e}_r$ is not necessarily the same for all the methods; the $\hat{e}_r$ values used for GQSA, IAD and AMCA are: 10^{-3}, 10^{-5} and 10^{-5}, respectively. In addition, the iterative procedure is halted by time, i.e., we established a maximum execution time so that, the iterative procedure stops when the time required to meet the convergence test is larger than the maximum execution time.

We observe that AMCA converges in all the evaluated scenarios with significantly lower execution times than those of GQSA and IAD. In the CRN case, AMCA is between 2 and 114 times faster than GQSA, and between 3 and 99 times faster than IAD. A similar behavior was observed in the ISN case: AMCA is between 3 and 28 times faster than GQSA and between 3 and 11 times faster than IAD. Note that there are a couple of scenarios (see Table 1, row LL - 40% and Table 2, row LH - 40%) in which GQSA and IAD were halted by time. Although in such cases converge could have been achieved if more iterations had been performed, it would be of no practical interest, since the benefit with respect to the exact solution (in terms of execution time) will be marginal or non-existent.

6. Conclusion

We have presented a novel approximation method for the performance evaluation of wireless networks that is based on time-scale separation. The proposed method, which is iterative in nature, permits trading-off computational effort in exchange of an increased accuracy. We applied the new method to two different types of wireless networks and we compared the performance of our method with that of a recently published generalization of QSA (GQSA) and also with a classical method known as iterative aggregation/disaggregation (IAD). Numerical results show that our method outperforms GQSA and IAD by providing the same accuracy with a substantially lower computational cost.

AppendixA. Phase-type (PH) distribution

Consider a CTMC on a finite state-space $\mathcal{S} = \{0, 1, 2, \dots, m\}$ where one state is absorbing and the remaining m states are transient. The random variable defined as the time to absorption is said to have a continuous PH distribution [24].

A PH distribution is uniquely given by the pair $(\boldsymbol{\alpha}; \boldsymbol{T})$, where $\boldsymbol{\alpha}$ is a m-dimensional row vector that defines the probabilities that the system starts at any of the transient states and meet $\sum_{i=0}^{m} \alpha_i = 1$; while T is a $m \times m$ matrix referred to as the *PH generator* that contains the transition rates between the transient states.

The infinitesimal generator for the CTMC can be written in block-matrix form as $\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{t} \\ \boldsymbol{0} & 0 \end{bmatrix}$. Here, $\boldsymbol{0}$ is a $1 \times m$ row vector of zeros. The elements of the column vector $\boldsymbol{t} = [t_1, t_2, \ldots, t_m]'$ are the transition rates from the transient states to the absorbing state. The $m \times m$ sub-stochastic matrix $\boldsymbol{T}$ meets $\boldsymbol{t} = -\boldsymbol{T}\boldsymbol{e}$, where $\boldsymbol{e}$ is a column vector of ones of appropriate dimension.

It is known that $-(\boldsymbol{T}^{-1})_{ij}$ is the expected total time spent in phase j during the time until absorption, conditioned on the system starting at phase i [25, Theorem 2.4.3]. The elements of $-\boldsymbol{T}^{-1}$ are used to obtain the fractions of time the system spends at each of the m states until absorption. The interested reader is referred to [24, 25, 26] for further details and a comprehensive theoretical treatment of PH distributions.

Acknowledgments

This research has been supported in part by the Ministry of Economy and Competitiveness of Spain under Grants TIN2013-47272-C2-1-R and TEC2015-71932-REDT. The research of L. Tello-Oquendo was supported in part by Programa de Ayudas de Investigación y Desarrollo (PAID) of the Universitat Politècnica de València.

References

[1] N. Tadayon, S. Aissa, Modeling and analysis framework for multi-interface multi-channel cognitive radio networks, Wireless Communications, IEEE Transactions on 14 (2) (2015) 935–947. `doi:10.1109/TWC.2014.2362535`.

[2] W. Zhang, M. Suresh, R. Stoleru, H. Chenji, On modeling the coexistence of 802.11 and 802.15.4 networks for performance tuning, Wireless Communications, IEEE Transactions on 13 (10) (2014) 5855–5866. `doi:10.1109/TWC.2014.2326151`.

[3] G. G. Yin, Q. Zhang, Discrete-time Markov chains: two-time-scale methods and applications, Vol. 55, Springer, 2006.

[4] W. J. Stewart, Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling, Princeton University Press, 2009.

[5] E. Wong, C. Foh, Analysis of cognitive radio spectrum access with finite user population, IEEE Communications Letters 13 (5) (2009) 294–296.

[6] J. Martinez-Bauset, V. Pla, J. Vidal, L. Guijarro, Approximate analysis of cognitive radio systems using time-scale separation and its accuracy, IEEE Communications Letters 17 (1) (2013) 35–38.

[7] Y. Huang, K. Ko, M. Zukerman, A generalized quasi-stationary approximation for analysis of an integrated service system, IEEE Communications Letters 16 (11) (2012) 1884–1887.

[8] L. Jiao, E. Song, V. Pla, F. Li, Capacity upper bound of channel assembling in cognitive radio networks with quasistationary primary user activities, Vehicular Technology, IEEE Transactions on 62 (4) (2013) 1849–1855. `doi:10.1109/TVT.2012.2236115`.

[9] S. Liu, J. Virtamo, Performance analysis of wireless data systems with a finite population of mobile users, in: Proceedings of the 19th International Teletraffic Congress ITC 19, 2005, pp. 1295–1304.

[10] O. J. Boxma, A. F. Gabor, R. Núñez-Queija, H.-P. Tan, Performance analysis of admission control for integrated services with minimum rate guarantees, in: Proceedings of NGI'06, 2006, pp. 41–47.

[11] L. Jiao, F. Y. Li, V. Pla, Modeling and performance analysis of channel assembling in multichannel cognitive radio networks with spectrum adaptation, Vehicular Technology, IEEE Transactions on 61 (6) (2012) 2686–2697.

[12] L. Jiao, I. Balapuwaduge, F. Li, V. Pla, On the performance of channel assembling and fragmentation in cognitive radio networks, Wireless Communications, IEEE Transactions on 13 (10) (2014) 5661–5675. doi:10.1109/TWC.2014.2322057.

[13] L. Tello-Oquendo, V. Pla, J. Martinez-Bauset, Approximate analysis of wireless systems based on time-scale decomposition, in: Wireless Days (WD), 2013 IFIP, 2013, pp. 1–6. doi:10.1109/WD.2013.6686513.

[14] C. Politis, Managing the radio spectrum, Vehicular Technology Magazine 4 (1) (2009) 20–26.

[15] Y.-C. Liang, K.-C. Chen, G. Li, P. Mahonen, Cognitive radio networking and communications: an overview, Vehicular Technology, IEEE Transactions on 60 (7) (2011) 3386–3407. doi:10.1109/TVT.2011.2158673.

[16] J. W. Roberts, Internet traffic, QoS, and pricing, Proceedings of the IEEE 92 (9) (2004) 1389–1399.

[17] W. Song, H. Jiang, W. Zhuang, X. Shen, Resource management for QoS support in cellular/WLAN interworking, Network, IEEE 19 (5) (2005) 12–18.

[18] H. Al-Mahdi, M. A. Kalil, F. Liers, A. Mitschele-Thiel, Increasing spectrum capacity for ad hoc networks using cognitive radios: an analytical model, IEEE Communications Letters 13 (9) (2009) 676–678. doi:10.1109/LCOMM.2009.090103.

[19] J. Peha, Sharing spectrum through spectrum policy reform and cognitive radio, Proceedings of the IEEE 97 (4) (2009) 708–719.

[20] P. Courtois, Decomposability, instabilities, and saturation in multiprogramming systems, Communications of the ACM 18 (7) (1975) 371–377.

[21] F. Hubner, P. Tran-Gia, Quasi-stationary analysis of a finite capacity asynchronous multiplexer with modulated deterministic input, ITC-13, Copenhagen.

[22] V. Alexiades, A. D. Solomon, Mathematical modeling of melting and freezing processes, Taylor & Francis, 1993.

[23] D. P. Heyman, M. J. Goldsmith, Comparisons between aggregation/disaggregation and a direct algorithm for computing the stationary probabilities of a markov chain, ORSA Journal on Computing 7 (1) (1995) 101–108. doi:10.1287/ijoc.7.1.101.

[24] M. Neuts, Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach, The Johns Hopkins University Press, 1981.

[25] G. Latouche, V. Ramaswami, Introduction to Matrix Analytic Methods in Stochastic Modeling, ASA-SIAM, 1999.

[26] A. S. Alfa, Queueing Theory for Telecommunications: Discrete Time Modelling of a Single Node System, Springer, 2010.