Computer Communications 86 (2016) 40–48

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Performance analysis of wireless networks based on time-scale separation: A new iterative method

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

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

ARTICLE INFO

Article history:
Received 22 September 2015
Revised 6 April 2016
Accepted 12 April 2016
Available online 13 April 2016

Keywords:
Cognitive radio networks
Performance analysis
PH distribution
Time-scale separation
Wireless networks

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.

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 [5–12]. In many of them, the so-called *quasi-stationary approximation* (QSA) has been shown to be accurate and computationally efficient [6,9–11]. 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

* Corresponding author. Tel.: +34963879733.
E-mail addresses: luiteloq@upv.es (L. Tello-Oquendo), vpla@upv.es (V. Pla), jmartinez@upv.es (J. Martinez-Bauset), vcasares@upv.es (V. Casares-Giner).

http://dx.doi.org/10.1016/j.comcom.2016.04.004
0140-3664/

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 $\mathcal{S} := \{(i, j) : 0 \le i \le C_1, 0 \le i + j \le C\}$ 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:

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

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

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

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 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, $t \ge 0$.

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, $t \ge 0$. 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