An Analytical Performance Evaluation Tool for Wireless Access Points with Opportunistic Scheduling

Ekaterina Evdokimova
Department TELIN
Ghent University
Gent, Belgium
eevdokim@telin.UGent.be

Koen De Turck
Laboratoire des Signaux et
Systèmes, CentraleSupélec
Gif sur Yvette, France
koen.deturck@supelec.fr

Sabine Wittevrongel
Department TELIN
Ghent University
Gent, Belgium
sw@telin.UGent.be

Dieter Fiems
Department TELIN
Ghent University
Gent, Belgium
df@telin.UGent.be

ABSTRACT

This paper introduces a performance analysis tool for a wireless access point serving multiple mobile nodes. The channel conditions vary over time and the opportunistic scheduler in the access point can account for the channel conditions as well as for the number of backlogged packets for the different mobile nodes. The tool allows for fast and accurate calculations of the main performance measures of the system. Calculations are based on a Maclaurin series expansion of the solution of the Markov process that describes the channel conditions and the queue contents for the different mobile nodes. The Maclaurin series expansion only requires the inversion of sparse block triangular matrices, which is considerably faster than directly calculating the solution. The tool provides a user interface for defining the network and channel parameters and can be used for assessing the efficiency of opportunistic schedulers, or for optimizing system parameters and scheduling policies of such schedulers.

CCS Concepts

•Mathematics of computing → Queueing theory; Markov processes; •Networks → Network performance analysis;

Keywords

Opportunistic scheduling, Series expansions.

1. INTRODUCTION

The analytical performance analysis of a wireless access point (AP) with an opportunistic scheduler under varying channel conditions is a challenging task. A Markov model for such an access point needs to track the number of backlogged packets for every mobile user it serves, which quickly results in state-space explosion, rendering a direct solution of the Markov model computationally infeasible. Hence, it is not surprising that most of the existing studies of access points rely on simulation. Only a few authors assess performance of opportunistic schedulers by analytic means [2, 7, 8].

The tool presented in this paper relies on the analytical method presented in [3] and briefly reviewed below. The tool provides a convenient user interface for manual input of the network and channel parameters, and hides the details of the method. As such, this tool allows one to compare the performance of different scheduling policies, system configurations or channel properties in a practical setting. The analytical performance analysis is expected to be helpful for fast and precise performance assessments when designing a new scheduling policy or when choosing the system parameters under given performance requirements.

In the remainder of the paper, we briefly explain the method for assessing the performance in section 2. Then, the implementation of the software tool is discussed and illustrated by an example in section 3 before drawing conclusions in section 4.

2. METHODOLOGY

In this section we first introduce the Markov model of the system, accounting for the buffer behaviour at the wireless AP and the transmission channel variability. Afterwards, we outline the main expressions for its numerical solution.

2.1 System model

We consider an AP serving K users as depicted in Figure 1. Each mobile node has its dedicated buffer, with capacity C_k for node k. Let $\mathcal{C}_k = \{0, \ldots, C_k\}$ and let $\mathcal{C} = \mathcal{C}_1 \times \ldots \times \mathcal{C}_k$. We make the following assumptions:

- There is an exogenous continuous-time Markov process $M(t)$ that modulates the state of the wireless trans-

VALUETOOLS 2015, December 14-16, Berlin, Germany

DOI 10.4108/eai.14-12-2015.2262721

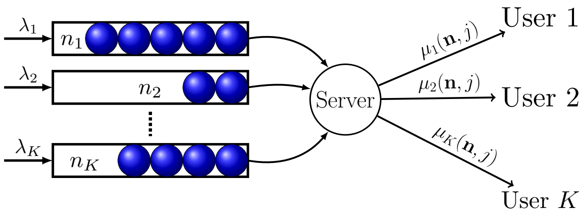


Figure 1: Queueing model for an access point

mission channel. Let $\mathcal{M} = \{1, 2, \ldots, M\}$ be the state space of this Markovian background process and let α_{ij} denote the transition rate from state i to state j, $i \neq j$, $i, j \in \mathcal{M}$.

- Packets for the kth mobile node arrive in accordance with a Poisson process with arrival rate λ_k.

- Let n_k be the number of packets in the kth buffer and let $\mathbf{n}$ be the vector with elements n_k, $k = 1, \ldots, K$. For background state $j \in \mathcal{M}$, and queue content $\mathbf{n} \in \mathcal{C}$, packets leave the kth buffer with rate $\mu_k(\mathbf{n}, j)$.

2.2 Analytical method

The state space of the Markovian queueing model introduced above is $\mathcal{C} \times \mathcal{M}$. Let $\pi(\mathbf{n}, j)$ be the stationary probability that there are n_k packets waiting for the kth mobile node and that the channel is in state j. The balance equations are easily obtained:

$$\pi(\mathbf{n}, j) \left(\sum_{k=1}^{K} \left(\lambda_k \mathbf{1}_{\{n_k < C_k\}} + \mu_k(\mathbf{n}, j) \mathbf{1}_{\{n_k > 0\}} \right) + \sum_{i=1, i \neq j}^{M} \alpha_{ji} \right)$$
$$= \sum_{k=1}^{K} \pi(\mathbf{n} + \mathbf{e}_k, j) \mu_k(\mathbf{n} + \mathbf{e}_k, j) \mathbf{1}_{\{n_k < C_k\}}$$
$$+ \sum_{k=1}^{K} \pi(\mathbf{n} - \mathbf{e}_k, j) \lambda_k \mathbf{1}_{\{n_k > 0\}} + \sum_{i=1, i \neq j}^{M} \pi(\mathbf{n}, i) \alpha_{ij}, \quad (1)$$

for $\mathbf{n} \in \mathcal{C}$ and $j \in \mathcal{M}$. We here defined $\mathbf{e}_k$ to be the row vector of length K with its kth element set to 1 and all other elements zero, and $\mathbf{1}_{\{X\}}$ denotes the indicator function of the event X. For ease of notation, let $\boldsymbol{\pi}(\mathbf{n}) = [\pi(\mathbf{n}, 1), \ldots, \pi(\mathbf{n}, M)]$, then we get the equivalent set:

$$\boldsymbol{\pi}(\mathbf{n}) \left(\sum_{k=1}^{K} \left(\lambda_k \mathbf{1}_{\{n_k < C_k\}} I_M + M_k(\mathbf{n}) \mathbf{1}_{\{n_k > 0\}} \right) - A \right)$$
$$= \sum_{k=1}^{K} \boldsymbol{\pi}(\mathbf{n} + \mathbf{e}_k) M_k(\mathbf{n} + \mathbf{e}_k) \mathbf{1}_{\{n_k < C_k\}}$$
$$+ \sum_{k=1}^{K} \boldsymbol{\pi}(\mathbf{n} - \mathbf{e}_k) \lambda_k \mathbf{1}_{\{n_k > 0\}}, \quad (2)$$

with $M_k(\mathbf{n})$ the $M \times M$ diagonal matrix with diagonal elements $\mu_k(\mathbf{n}, j)$, with I_M the $M \times M$ identity matrix and with A the generator matrix of $M(t)$.

2.3 Series expansion

The size of the state space of the Markov chain at hand is

$$S = M \prod_{k=1}^{K} (C_k + 1).$$

For a moderate number of buffers, directly solving the balance equation for the stationary vector $\boldsymbol{\pi} = [\pi(\mathbf{n}, j)]_{\mathbf{n} \in \mathcal{C}, j \in \mathcal{M}}$ is computationally demanding. We here show that by considering the Maclaurin series expansion of the stationary solution in some system parameter, we can get an accurate estimate of the performance. Expansions of interest for the access point are discussed in the following subsections.

For a system parameter ϵ, consider a Markov chain with generator matrix $Q = Q^{(0)} + \epsilon Q^{(1)}$ and corresponding stationary distribution vector $\boldsymbol{\pi}^{(\epsilon)}$, such that,

$$\boldsymbol{\pi}^{(\epsilon)} Q = \boldsymbol{\pi}^{(\epsilon)} (Q^{(0)} + \epsilon Q^{(1)}) = 0, \quad \boldsymbol{\pi}^{(\epsilon)} \mathbf{e}' = 1. \quad (3)$$

Here $\mathbf{e}'$ is a column vector of ones. We further assume that the Markov chain is stationary ergodic for all $\epsilon \in [0, \epsilon_m)$.

By Cramer's rule, one easily finds that $\boldsymbol{\pi}^{(\epsilon)}$ is an analytic function of ϵ in an open interval around $\epsilon = 0$. Therefore, let $\boldsymbol{\pi}_n$ be the nth term in the series expansion of $\boldsymbol{\pi}^{(\epsilon)}$,

$$\boldsymbol{\pi}^{(\epsilon)} = \sum_{n=0}^{\infty} \boldsymbol{\pi}_n \, \epsilon^n. \quad (4)$$

Plugging the series expansion (4) in (3) and identifying equal powers of ϵ, we get

$$\boldsymbol{\pi}_0 Q^{(0)} = 0, \quad \boldsymbol{\pi}_{n+1} Q^{(0)} = -\pi_n Q^{(1)}. \quad (5)$$

Complementing the former set of equations with the normalisation condition,

$$\boldsymbol{\pi}_0 \mathbf{e}' = 1, \quad \boldsymbol{\pi}_n \mathbf{e}' = 0, \quad (6)$$

for $n > 0$ allows for recursively calculating the terms $\boldsymbol{\pi}_n$ of the series expansion.

For a generic matrix $Q^{(0)}$, there is no gain in computational complexity as one still needs to invert this matrix while solving for the next term in the series expansion. However, for the Markovian queueing model at hand, $Q^{(0)}$ has additional structure, both for the expansion in the arrival rate λ as in the server rate μ. In both cases, $Q^{(0)}$ is a block triangular matrix which reduces the complexity of its inversion.

2.4 Overload-traffic analysis

Assume that the service rate for every queue depends on a rate μ: $\mu_k(\mathbf{n}, j) = \mu \tilde{\mu}_k(\mathbf{n}, j)$. We can now consider the balance equation for $\mu \to 0$. In particular we consider the following series expansion of the steady-state probabilities:

$$\boldsymbol{\pi}(\mathbf{n}) = \sum_{i=0}^{\infty} \boldsymbol{\pi}_i(\mathbf{n}) \, \mu^i. \quad (7)$$

For ease of notation, let $\tilde{M}_k(\mathbf{n}) = \mu^{-1} M_k(\mathbf{n})$. Note that $\tilde{M}_k(\mathbf{n})$ does not depend on μ. Plugging the former expression into equation (2) and comparing terms in μ^i, we get

$$\boldsymbol{\pi}_i(\mathbf{n}) \sum_{k=1}^{K} \lambda_k \mathbf{1}_{\{n_k < C_k\}} - \boldsymbol{\pi}_i(\mathbf{n}) A$$

$$
\begin{aligned}
&= \mathbf{1}_{\{i>0\}} \sum_{k=1}^{K} \boldsymbol{\pi}_{i-1}(\mathbf{n}+\mathbf{e}_k)\tilde{M}_k(\mathbf{n}+\mathbf{e}_k)\,\mathbf{1}_{\{n_k<C_k\}} \\
&\quad - \mathbf{1}_{\{i>0\}}\boldsymbol{\pi}_{i-1}(\mathbf{n}) \sum_{k=1}^{K} \tilde{M}_k(\mathbf{n})\,\mathbf{1}_{\{n_k>0\}} \\
&\quad + \sum_{k=1}^{K} \boldsymbol{\pi}_i(\mathbf{n}-\mathbf{e}_k)\lambda_k\,\mathbf{1}_{\{n_k>0\}}\,. \qquad (8)
\end{aligned}
$$

Plugging $\mathbf{n} = \mathbf{0} = [0, 0, \ldots, 0]$ and $i = 0$ in the former equation and post-multiplying with $\mathbf{e}'$ leads to

$$\boldsymbol{\pi}_0(\mathbf{0})\mathbf{e}' = 0\,, \qquad (9)$$

which implies $\boldsymbol{\pi}_0(\mathbf{0}) = \mathbf{0}$ as the elements of $\boldsymbol{\pi}_0(\mathbf{0})$ are nonnegative. Using the same arguments, one then shows by iteration that for all $\mathbf{n} \in \mathcal{C} \setminus \{\mathbf{c}\}$, we have $\boldsymbol{\pi}_0(\mathbf{n}) = \mathbf{0}$ and $\boldsymbol{\pi}_0(\mathbf{c})A = 0$. Here $\mathbf{c} = [C_1, \ldots, C_K]$ is the vector representing full queues for all mobile nodes. Together with the normalisation condition $\sum_{\mathbf{n}\in\mathcal{C}} \boldsymbol{\pi}_0(\mathbf{n})\mathbf{e}' = 1$, this shows that $\boldsymbol{\pi}_0(\mathbf{c}) = \mathbf{a}$, the steady-state solution of the Markov process $M(t)$.

For the higher-order terms ($i > 0$), we have

$$
\begin{aligned}
&\boldsymbol{\pi}_i(\mathbf{n}) \left(\sum_{k=1}^{K} \lambda_k\,\mathbf{1}_{\{n_k<C_k\}} I_M - A \right) \\
&\quad = \sum_{k=1}^{K} \boldsymbol{\pi}_{i-1}(\mathbf{n}+\mathbf{e}_k)\tilde{M}_k(\mathbf{n}+\mathbf{e}_k)\,\mathbf{1}_{\{n_k<C_k\}} \\
&\quad + \sum_{k=1}^{K} \Big(\boldsymbol{\pi}_i(\mathbf{n}-\mathbf{e}_k)\lambda_k - \boldsymbol{\pi}_{i-1}(\mathbf{n})\tilde{M}_k(\mathbf{n}) \Big)\,\mathbf{1}_{\{n_k>0\}}\,. \qquad (10)
\end{aligned}
$$

For $\mathbf{n} \neq \mathbf{c}$, the matrix on the left-hand side is invertible. Hence, we can calculate the probabilities $\boldsymbol{\pi}_i(\mathbf{n})$ in lexicographical order. For $\mathbf{n} = \mathbf{c}$, we get

$$\boldsymbol{\pi}_i(\mathbf{c})A = \sum_{k=1}^{K} \Big(-\boldsymbol{\pi}_i(\mathbf{c}-\mathbf{e}_k)\lambda_k + \boldsymbol{\pi}_{i-1}(\mathbf{c})\tilde{M}_k(\mathbf{c}) \Big)\,, \qquad (11)$$

and the matrix on the left-hand side is not invertible. A solution of this equation takes the form

$$\boldsymbol{\pi}_i(\mathbf{c}) = \sum_{k=1}^{K} \Big(-\boldsymbol{\pi}_i(\mathbf{c}-\mathbf{e}_k)\lambda_k + \boldsymbol{\pi}_{i-1}(\mathbf{c})\tilde{M}_k(\mathbf{c}) \Big)\, A^{\#} + \kappa_i\mathbf{a}\,, \qquad (12)$$

for any κ_i. Here, $A^{\#} = (A+\mathbf{ea})^{-1} - \mathbf{ea}$ is the group inverse of A. Finally, the remaining unknown κ_i follows from the normalisation condition

$$\sum_{\mathbf{n}\in\mathcal{C}} \boldsymbol{\pi}_i(\mathbf{n})\mathbf{e}' = 0\,. \qquad (13)$$

In view of the calculations above, one easily verifies that the numerical complexity of the algorithm is $O(NM^2S)$ as there are S/M blocks, N terms in the recursion and the operations with blocks have complexity $O(M^3)$.

2.5 Light-traffic analysis

Similar arguments can be developed for the case of light-traffic conditions, that is, we set $\lambda_k = \lambda\tilde{\lambda}_k$ and consider an expansion of the form

$$\boldsymbol{\pi}(\mathbf{n}) = \sum_{i=0}^{\infty} \boldsymbol{\pi}_i(\mathbf{n})\,\lambda^i\,.$$

In view of the balance equation, the terms of this series expansion adhere

$$
\begin{aligned}
&\boldsymbol{\pi}_i(\mathbf{n}) \left(\sum_{k=1}^{K} M_k(\mathbf{n})\,\mathbf{1}_{\{n_k>0\}} - A \right) \\
&\quad = \sum_{k=1}^{K} \boldsymbol{\pi}_i(\mathbf{n}+\mathbf{e}_k)M_k(\mathbf{n}+\mathbf{e}_k)\,\mathbf{1}_{\{n_k<C_k\}} \\
&\quad - \mathbf{1}_{\{i>0\}}\boldsymbol{\pi}_{i-1}(\mathbf{n}) \sum_{k=1}^{K} \tilde{\lambda}_k\,\mathbf{1}_{\{n_k<C_k\}} \\
&\quad + \mathbf{1}_{\{i>0\}} \sum_{k=1}^{K} \boldsymbol{\pi}_{i-1}(\mathbf{n}-\mathbf{e}_k)\tilde{\lambda}_k\,\mathbf{1}_{\{n_k>0\}}\,. \qquad (14)
\end{aligned}
$$

For $i = 0$, we can show that $\boldsymbol{\pi}_0(\mathbf{n}) = \mathbf{0}$ for $\mathbf{n} \neq \mathbf{0}$ and $\boldsymbol{\pi}_0(\mathbf{0}) = \mathbf{a}$. For $i > 0$ and $\mathbf{n} \neq \mathbf{0}$, we can recursively calculate all $\boldsymbol{\pi}_i(\mathbf{n})$ in reverse lexicographical order as the matrix on the left-hand side is invertible. For $\mathbf{n} = \mathbf{0}$, we get

$$\boldsymbol{\pi}_i(\mathbf{0}) = (-\sum_{k=1}^{K} \boldsymbol{\pi}_i(\mathbf{e}_k)M_k(\mathbf{e}_k) + \boldsymbol{\pi}_{i-1}(\mathbf{0}) \sum_{k=1}^{K} \tilde{\lambda}_k)A^{\#} + \tilde{\kappa}_i\mathbf{a}\,, \qquad (15)$$

where $\tilde{\kappa}_i$ can be determined from the normalisation condition (13). We refer to [3] for the detailed analysis.

2.6 Modelling of wireless environment

At the wireless AP knowledge of the transmission environment conditions is provided by user nodes reporting their channel state information (CSI). Depending on the system organisation CSI feedback may take different forms including the current signal-to-noise ratio (SNR) level at the receiver, the available service rates, the probability of correct transmission or any other representative value related to the channel quality [5].

Limiting the number of parameters of the channel, we assume that the characteristics of all transmission channels are identical and channel qualities at the receivers can either vary independently or in correlated manner. We first focus on the Markov process which characterises a single channel. Let H be the number of states of the Markov process $\tilde{H}_k(t)$ for the kth channel and let $\mathbf{g} = [g_1, \ldots, g_H]$ and $\tilde{A}$ be the vector of channel qualities in the different states and the generator matrix of the Markov process $\tilde{H}_k(t)$, respectively. Specifically, in this work the element g_h of the vector $\mathbf{g}$ corresponds to the ratio between the available transmission rate in the hth channel state and the maximally achievable transmission rate. Thus, the value $g_1 = 0$ corresponds to poor channel quality with no transmission, value $g_H = 1$ corresponds to excellent channel conditions associated with the maximum achievable transmission rate. The other elements of $\mathbf{g}$ belong to the interval $(0 \ldots 1)$ and stand for the attenuation levels of the channel quality depending on the SNR. In such a way we represent the channel qualities by means of adjustable coefficients and without being bounded to certain transmission rates or the type of CSI. The channel transition rate matrix $\tilde{A}$ of the finite-state Markov chain (FSMC) for a single user under the assumption of a Rayleigh channel can be defined according to [10]. The additional parameters defining the channel model are: the maximum transmission rate r_t of the system, the average SNR level of the Rayleigh channel SNR_0, and the Doppler spread F_m, which is caused by the users' motion and increases the symbol error probability. However, [10] assumes that channel changes are at discrete time instants. To obtain a corresponding continuous-time Markov model, we introduce an additional rate ε and assume that the channel changes in accordance to [10] on the events of a Poisson process with rate ε . Finally, to obtain the generator matrix A of the FSMC corresponding to the overall channel process $\tilde{M}(t)$ for a multiuser system with K identical independent channels, we apply the Kroneker sum $\oplus$:

$$A = \oplus_{n=1}^{K} \tilde{A}. \tag{16}$$

In order to introduce correlations between the qualities of the channels we have to alter the elements of A according to the provided correlation matrix $\mathbf{R}$.

2.7 Examples of schedulers

For demonstrating purposes, we have implemented several schedulers including purely opportunistic schedulers, some weighted schedulers as well as non-opportunistic schedulers.

A first example of a greedy opportunistic scheduler is the *MaxRate* scheduler. The scheduler serves node $k^*(t)$ at time t, with

$$k^*(t) = \arg\max_{k\in(1\ldots K)} g_k(t)\,,$$

and where $g_k(t)$ stands for the channel quality for user k at time t, which depends on the state of the channel. MaxRate defines a function $k(j)$ such that $\mu_k(\mathbf{n}, j) = 0$ for all $k \neq k(j)$. The scheduler was first considered in [6] for single-cell multiuser communication.

The MaxRate scheduler can be extended such that the AP chooses two users for simultaneous service. That is, such a scheduler always divides the transmission rate between two users with the best channel conditions. This is possible in time division multiple access (TDMA) systems with code division multiple access (CDMA) within each time slot. The scheduler serves nodes $k_1^*(t)$ and $k_2^*(t)$ at time t, with

$$[k_1^*(t), k_2^*(t)] = \arg\max_{[k_1,k_2]\in(1\ldots K), k_1\neq k_2} [g_{k_1}(t) + g_{k_2}(t)]\,.$$

This extension of MaxRate defines functions $k_1(j)$ and $k_2(j)$ such that $\mu_k(\mathbf{n}, j) = 0$ for all $k \notin \{k_1(j), k_2(j)\}$.

In contrast to MaxRate, *MaxWeight* selects the user with the maximum weight, which is calculated as the product of queue length and channel quality, see [1]. MaxWeight selects user $k^*(t)$ at time t, with

$$k^*(t) = \arg\max_{k\in(1\ldots K)} g_k(t) n_k(t)\,.$$

As MaxWeight also accounts for the queue length, there is now a function $k(\mathbf{n}, j)$, such that $\mu_k(\mathbf{n}, j) = 0$ for all $k \neq k(\mathbf{n}, j)$.

Schedulers may also not account for the channel state at all. An example of a non-opportunistic scheduler is one that chooses the longest queue. The scheduler is shown to be stable for dynamic server allocation to parallel queues with randomly varying connectivity in [9] and called the Longest Connected Queue (LCQ). The scheduler serves node $k^*(t)$ at time t, with

$$k^*(t) = \arg\max_{k\in(1\ldots K)} n_k(t)\,.$$

As LCQ accounts for the queue length, there is now a function $k(\mathbf{n})$, such that $\mu_k(\mathbf{n}, j) = 0$ for all $k \neq k(\mathbf{n})$. Notice that also in this case the actual service rate $\mu_{k(\mathbf{n})}(\mathbf{n}, j)$ for node $k(\mathbf{n})$ does depend on the channel condition.

Table 1: Model parameters

Buffer	Number of users K
	Buffer capacity C
	Scheduling policy
Channel	Number of channel states H
	SNR levels, dB
	Average SNR_0, dB
	Doppler spread F_m, Hz
	Transmission rate r_t
	Correlation matrix $\mathbf{R}$

Finally, we mention two schedulers which are inspired by discriminatory (DPS) and generalised processor sharing (GPS) [4], but with weights set to reflect the channel conditions. The service rate then takes one of the following forms:

$$\text{DPS: } \mu_k(\mathbf{n}, j) = \mu\beta_j \frac{g_{jk} n_k}{\sum_{\ell=1}^{K} g_{j\ell} n_\ell}\,,$$

$$\text{GPS: } \mu_k(\mathbf{n}, j) = \mu\beta_j \frac{g_{jk} \mathbf{1}_{\{n_k>0\}}}{\sum_{\ell=1}^{K} g_{j\ell} \mathbf{1}_{\{n_\ell>0\}}}\,,$$

where g_{jk} is the channel quality for user k when the overall channel state is j, and the parameter β_j describes the overall channel condition $\beta_j = (1/K)\sum_{k=1}^{K} g_{jk}$.

3. THE TOOL IMPLEMENTATION

Figure 2 depicts the main components of the performance analysis tool. The graphical user interface (GUI) front end of the tool is implemented in c#, and calls python scripts and c++ functions in order to set up the system and channel models, and to calculate the performance measures, respectively. Another python script which calls LaTeX then formats the results and creates a report in portable document format.

Via the GUI the various parameters of the model and the analysis method are set. The input parameters can be divided into (i) buffer parameters, (ii) channel parameters and (iii) parameters of the series expansion method. The buffer parameters include (i) the number of mobile nodes K, (ii) the buffer capacity C for the nodes and (iii) the scheduling discipline used. Note that the current version of the GUI considers the same capacity C for all nodes. Predefined scheduling disciplines are the MaxRate, MaxRate with simultaneous service of two queues, MaxWeight, LCQ, DPS and GPS schedulers mentioned in section 2.7. The channel parameters include the number of states H of a single channel, and several parameters needed to construct the channel generator matrix A. The main model parameters that one can manually change via the GUI are listed in Table 1.

The model parameters inserted via the GUI are then exploited in the various predefinition blocks. Here, based on the given channel parameters the channel is represented by its transition rate matrix A, and based on the chosen scheduling policy, the buffer parameters and the channel qualities the service rate matrix $\boldsymbol{\mu}$ is constructed. The generator matrix A is formed in the corresponding block according to [10], as explained in section 2.6. In case of correlated channels, the correlations are added based on the given correlation matrix $\mathbf{R}$.

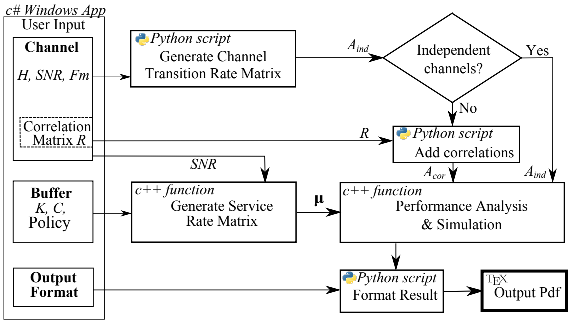


Figure 2: Structure of the performance evaluation tool

Next, the chosen scheduler can be analysed. The performance analysis block calculates the series expansion terms of the steady-state probability vectors for both the overload and light-traffic cases according to (10) and (14). The tool also includes functions for simulation in order to estimate the region of convergence of the analytical results. Via the GUI the user can define several additional parameters for the analysis block, such as the number of terms for the series expansions and the number of samples for the simulation. The user can also change the mode of the performance analysis & simulation block, and manually turn off/on simulation or analytical analysis for light or/and overload regimes. The interface window is shown in Figure 3.

The output report generated by the analysis tool consists of plots of the performance measures, namely of the mean buffer content $E[Q]$ and the blocking probability P_b. The output format can be adjusted for a better visual representation by changing the expansion terms to plot and the range of the axes in the application window.

Additional functional buttons allow saving the results and input parameters in a separate folder, opening the folder with previously stored results, uploading input parameters from any given configuration file and the correlation matrix from any text file, as chosen via the dialogue window. Alternatively, the desired correlation matrix can be inserted manually or randomly generated. In both cases the matrix is checked to be positive semidefinite.

For the sake of demonstration Figure 4 shows the output report for the configuration shown in Figure 3. The output file duplicates the input system information and provides plots with analytical estimations and simulation results of the performance measures in the given ranges.

4. CONCLUSIONS

We have introduced a practical tool for the performance analysis of a multiuser wireless AP operating under varying channel conditions. The tool is implemented as a windows application and presumes the manual input of the AP buffer and channel model parameters through a GUI. The underlying analytical method focuses on the buffer analysis in light and overload traffic regimes. The channel dynamics are modelled by a Markovian exogenous process. To solve the entire continuous-time Markov chain and cope with the prohibitively large size of the state space that is inherent to queueing systems with multiple finite-capacity buffers, series expansion techniques are used. The computational complexity of the suggested method is $O(NM^2S)$, with N the order of the expansions, M the number of channel states and S the size of the state space of the overall Markov chain. As output, the designed tool gives the performance measures of the system under the given conditions, collected in a portable document format. The tool can be used for a theoretical study of the efficiency of a scheduling policy as well as for the performance evaluation of practical systems.

Acknowledgement

This research has been partly funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office.

5. REFERENCES

[1] M. Andrews, K. Kumaran, K. Ramanan, A. Stolyar, R. Vijayakumar, and P. Whiting. Scheduling in a queuing system with asynchronously varying service rates. *Probability in the Engineering and Informational Sciences*, 18(02):191–217, 2004.

[2] B. Bellalta, A. Faridi, J. Barceló, V. Daza, and M. Oliver. Queueing analysis in multiuser multi-packet transmission systems using spatial multiplexing. *arXiv preprint arXiv:1207.3506*, 2012.

[3] E. Evdokimova, K. De Turck, S. Wittevrongel, and D. Fiems. Efficient performance evaluation of wireless networks with varying channel conditions. In *Analytical and Stochastic Modelling Techniques and Applications*, pages 59–72. Springer, 2015.

[4] D. Fiems and K. De Turck. A series expansion approach for finite-capacity processor sharing queues. In *Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools*, pages 118–125. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013.

[5] A. Goldsmith. *Wireless communications*. Cambridge university press, 2005.

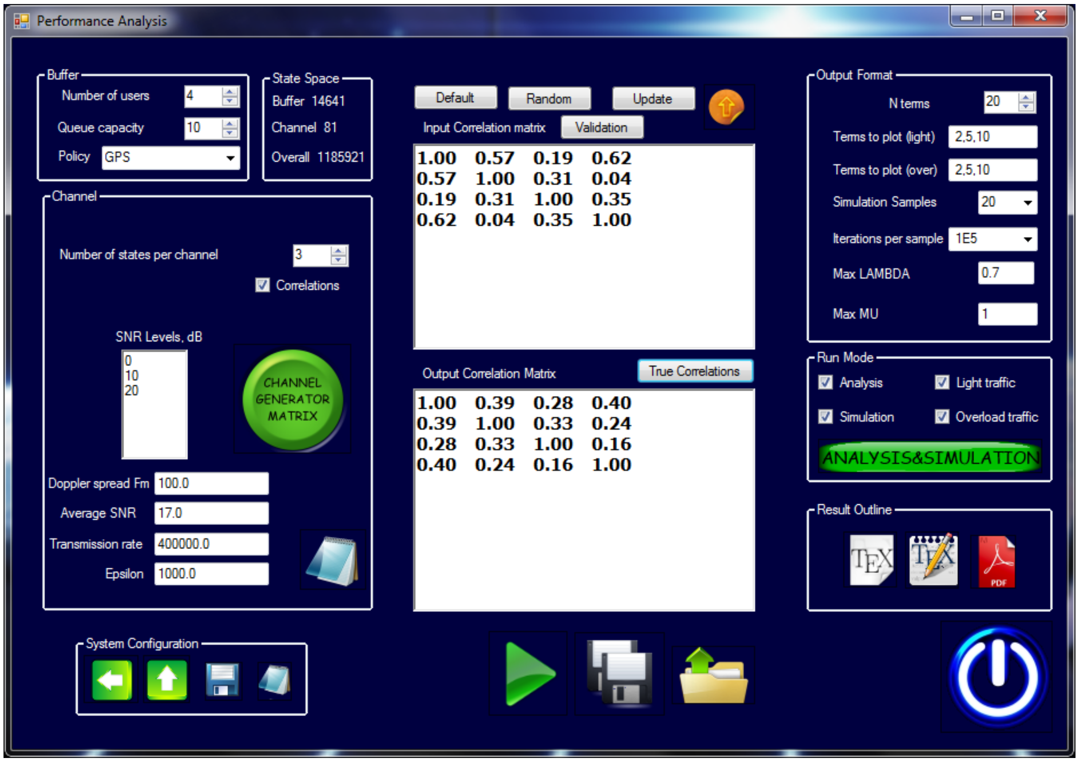
[6] R. Knopp and P. A. Humblet. Information capacity and power control in single-cell multiuser communications. In *Communications, 1995. ICC'95 Seattle,'Gateway to Globalization', 1995 IEEE International Conference on*, volume 1, pages 331–335. IEEE, 1995.

[7] L. Lei, C. Lin, J. Cai, and X. Shen. Performance analysis of wireless opportunistic schedulers using stochastic petri nets. *Wireless Communications, IEEE Transactions on*, 8(4):2076–2087, 2009.

[8] M. M. Rashid, M. J. Hossain, E. Hossain, and V. K. Bhargava. Opportunistic spectrum scheduling for multiuser cognitive radio: a queueing analysis. *Wireless Communications, IEEE Transactions on*, 8(10):5259–5269, 2009.

[9] L. Tassiulas and A. Ephremides. Dynamic server allocation to parallel queues with randomly varying connectivity. *Information Theory, IEEE Transactions on*, 39(2):466–478, 1993.

[10] H. S. Wang and N. Moayeri. Finite-state markov channel-a useful model for radio communication channels. *Vehicular Technology, IEEE Transactions on*, 44(1):163–171, 1995.



Performance Analysis

Buffer

Number of users 4

Queue capacity 10

Policy GPS

State Space

Buffer 14641

Channel 81

Overall 1185921

Input Correlation matrix

1.00	0.57	0.19	0.62
0.57	1.00	0.31	0.04
0.19	0.31	1.00	0.35
0.62	0.04	0.35	1.00

Output Format

N terms 20

Terms to plot (light) 2,5,10

Terms to plot (over) 2,5,10

Simulation Samples 20

Iterations per sample 1E5

Max LAMBDA 0.7

Max MU 1

Channel

Number of states per channel 3

☑ Correlations

SNR Levels, dB
0
10
20

Doppler spread Fm 100.0

Average SNR 17.0

Transmission rate 400000.0

Epsilon 1000.0

Output Correlation Matrix

1.00	0.39	0.28	0.40
0.39	1.00	0.33	0.24
0.28	0.33	1.00	0.16
0.40	0.24	0.16	1.00

Run Mode

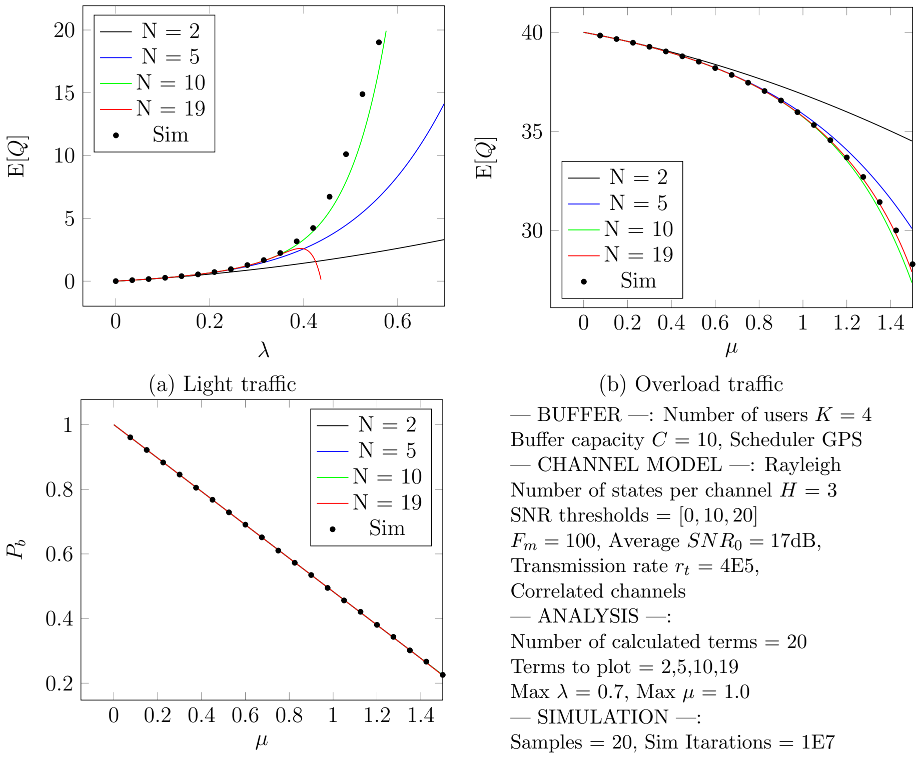
☑ Analysis ☑ Light traffic

☑ Simulation ☑ Overload traffic

Result Outline

System Configuration

Figure 3: The graphical user interface allows for setting up the buffer and channel parameters



(a) Light traffic

(b) Overload traffic

— BUFFER —: Number of users $K = 4$
Buffer capacity $C = 10$, Scheduler GPS
— CHANNEL MODEL —: Rayleigh
Number of states per channel $H = 3$
SNR thresholds $= [0, 10, 20]$
$F_m = 100$, Average $SNR_0 = 17$dB,
Transmission rate $r_t = 4E5$,
Correlated channels
— ANALYSIS —:
Number of calculated terms = 20
Terms to plot = 2,5,10,19
Max $\lambda = 0.7$, Max $\mu = 1.0$
— SIMULATION —:
Samples = 20, Sim Itarations = 1E7

(a) Overload traffic

(b) System Info

Figure 4: The output report generated by the tool