On the efficient solution of a multiserver system with two reattempt orbits

M. Jose Domenech-Benlloch[a,*], Jose Manuel Gimenez-Guzman[b], Vicent Pla[a], Jorge Martinez-Bauset[a], Vicente Casares-Giner[a]

[a] *Dept. Comunicaciones, Universidad Politécnica de Valencia, Cami de Vera s/n, 46022, Valencia, Spain.*
[b] *Dept. Automatica, Universidad de Alcalá, 28871 Alcalá de Henares, Madrid, Spain.*

Abstract

In communication networks that guarantee seamless mobility of users across service areas, reattempts occur as a result of user behavior but also as automatic retries of blocked handovers. A multiserver system with two reattempt orbits is obtained when modeling these networks. However, an exact Markovian model analysis of such systems has proven to be infeasible and resorting to approximate methods is mandatory. To the best of our knowledge all the existing methods are based on computing the steady state probabilities. We propose another approach based on the relative state values that appear in the Howard equations. We compare the proposed method with the most well-known methods appeared in the literature in a wide range of scenarios. The results of the numerical evaluation carried out show that this solution outperforms the previous approaches in terms of both accuracy and computation cost for the most common performance parameters used in retrial systems.

Key words: retrial queue, stochastic process, markov decision process, cellular network.

1. Introduction

The retrial phenomenon appears in multiple situations in telecommunications, computer networking as well as in many other fields. An in-depth study of the bibliography on retrial queues can be found in [1]. In this paper we focus our attention on a generic communication network that guarantees seamless mobility to its customers by means of a cellular architecture. In this type of networks, the network coverage area is divided into services areas, known as cells and customers can move across different cells of the network. When a customer with an active communication moves from one cell to another a so-called handover procedure is executed to allocate the necessary resources in the new cell and release the unused resources in the former cell. Nowadays, perhaps the most widespread and popular example of this type of networks are the telephone cellular networks —2G and 3G— but the current perspective is that in a near future a variety of technologies fitting into this category will be in place, e.g. Mobile IP, IEEE 802.16 [2] —commercially known as WiMAX—, which has recently incorporated mobility into the standard [2] and IEEE 802.20 [3] —Mobile Broadband Wireless Access, MBWA.

The phenomenon of repeated attempts in telecommunication systems has been studied, at least, since the 50's [4]. However the scenario under study in those works is that of a classical telephone network where the effect of reattempts is due to the customer's behavior. In contrast, this paper deals with the case in which reattempts appear not only when a customer is blocked but also when a handover is blocked. An example of technology that enables handover reattempts is GSM [5]. To the best of our knowledge, the first and only paper that has considered the effect on network performance of both types of reattempts simultaneously is [6]. Now, in this paper we refer to the former as redials and to the latter as (automatic) retrials, while we use the concept of reattempt to refer to any of them. Blocked handovers will be automatically retried until a reattempt succeeds or the user moves outside the handover area. In the former case the session will continue without the user noticing any disruption, while in the latter the session will be abruptly terminated. In contrast, persistence of redials depends on the user patience and an eventual abandonment results in session setup failure. Another difference is that the maximum number of unsuccessful automatic retrials is set by the network operator while redials are affected by the randomness of human behavior. Therefore, both types of reattempts have different

*Corresponding author
Email addresses: mdoben@upvnet.upv.es (M. Jose Domenech-Benlloch), josem.gimenez@uah.es (Jose Manuel Gimenez-Guzman), vpla@dcom.upv.es (Vicent Pla), jmartinez@upvnet.upv.es (Jorge Martinez-Bauset), vcasares@dcom.upv.es (Vicente Casares-Giner)

characteristics and as a consequence two separate retrial pools have to be considered in the analysis of the system.

The modeling of repeated attempts has been the subject of numerous investigations [7, 8, 9]. Two functional blocks are typically distinguished in models which consider reattempts: a block that accommodates the servers and possibly a waiting queue, and a block where users that reattempt are accommodated, usually called reattempt orbit. More concretely, the mathematical model of the system under consideration is a multiserver retrial queue. The most important characteristics of the state space of this model are its two infinite dimensions due to the orbits and the non-homegeneity along them as the reattempt rates depend on the number of users in the orbits. It is known that the classical theory [10] is developed for random walks on the semi-strip $\{0, \ldots, C\} \times \mathbb{Z}_+$ with infinitesimal transitions subject to conditions of space-homogeneity. Therefore it is clear that in this case it is necessary to resort to approximate methods, even in a single reattempt orbit case. These methods are usually grouped into three categories: approximations, finite truncated methods and generalized truncated methods [11, 12]. Although all the mentioned categories are in fact approximations, the first category is usually devoted to methods that can be useful only in a certain domain of the system parameters or in special extreme cases [12, Section 2.8], [13]. Therefore we will direct our attention only to finite and generalized truncated methods. The finite truncated methods replace the original infinite state space by a finite one, where steady state probabilities can be computed[14]. On the other hand, generalized truncated methods replace the original infinite state space by another infinite but solvable state space. This last type of methods usually outperform the other two types [15], offering a good tradeoff between precision and computational complexity.

All the approaches presented so far rely on the numerical solution of the steady-state Kolmogorov equations of the Continuous Time Markov Chain (CTMC) that describes the system under consideration. Very recently, however, an alternative approach for evaluating infinite state space Markov processes has been introduced by Leino et al. [16, 17, 18]. The new method, named Value Extrapolation (VE), does not rely on solving the global balance equations, but considers the system in its Markov Decision Process (MDP) setting and solves the expected value from the Howard equations written for a truncated state space. So far VE has been applied to simple retrial systems, being able to obtain very promising results in comparison with the most widespread approximation methods used in the solution of retrial systems [19].

The main objective of this work is to tailor the VE method to a system with two reattempt orbits and compare its performance with the performance of other possible approximate methods. This performance evaluation is done in a cellular network scenario that guarantees seamless mobility to its users. We conclude that VE greatly outperforms the rest of the methods throughout a wide range of scenarios not only in terms of accuracy, but also in terms of computation cost, so its use is highly recommendable.

The rest of the paper is structured as follows. First, we describe the cellular network under study and its associated model, focusing on the reattempt behavior under consideration. In Section 3 we enumerate and explain the main features of the methods we compare VE with. Section 4 is devoted to the description of VE and how it has been applied to the model under study. A numerical study is performed in Section 5 and finally, a summary of the paper and some concluding remarks are given in Section 6.

2. System description and model

We consider a cellular mobile network with a fixed channel allocation scheme and where each cell is served by a different base station, being C the number of resources in the cell. The physical meaning of a unit of resource is dependent on the specific technological implementation of the radio interface. Moreover, and without loss of generality, we consider that each user occupies one resource unit. As shown in Fig. 1 there are two arrival streams: the first one represents new sessions and the second one handovers from adjacent cells. Both arrivals are considered to be Poisson processes with rates λ_n and λ_h respectively, being $\lambda = \lambda_n + \lambda_h$. For the sake of mathematical tractability, the channel holding time is assumed to be exponentially distributed with rate μ. Moreover, in order to keep the analytical model tractable, we have considered exponential distributions for all the random variables that describe time magnitudes. While for some of the involved random variables the statistical features may not be properly captured by an exponential distribution, when it comes to the impact on the performance parameters of interest the exponential approximation is shown to be reasonable in a wide range of scenarios [20, 21].

In general, blocking a new session setup is considered to be less harmful than blocking a handover attempt. When the session under consideration is of streaming type, blocking a handover produces an abrupt termination of the ongoing session, which results more annoying from the user perspective than delaying the initiation of a new session. In the case of an elastic traffic session [22] the effect of tearing down a session is even worse, as the amount of information transmitted so far is rendered completely useless. Therefore we must include an admission control policy to guarentee the prioritization of handovers —and retrials— over new sessions

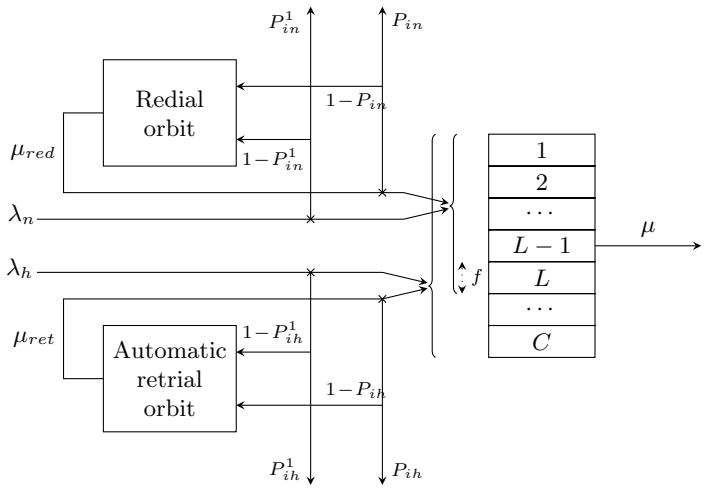

Figure 1: System model.

—and their associated redials— and, therefore assure a certain degree of Quality of Service (QoS). The most widespread technique is to reserve some resources to highest priority flows, being in our case handovers and their associated automatic retrials. This technique can be generalized to a fractional reservation, the so-called Fractional Guard Channel (FGC) admission control policy [23]. The FGC policy is characterized by only one parameter defined by a real number t ($0 \leq t \leq C$). New sessions and redials are accepted with probability 1 when there are less than $L = \lfloor t \rfloor$[1] resources being used and with probability $f = t - L$, when there are exactly L resources in use. If there are more than L busy resources, new sessions and redials are no longer accepted. Handovers and automatic retrials are only rejected when the system is completely occupied. Note that to analyze a system in which there is not an admission control algorithm we must make $t = C$.

When an incoming new session is blocked, according to Fig. 1, it joins the redial orbit with probability $(1 - P_{in}^1)$ or leaves the system with probability P_{in}^1. If a redial is not successful, the session returns to the redial orbit with probability $(1 - P_{in})$, redialing after an exponentially distributed time with rate μ_{red}. Redials are able to access to the same resources as the new sessions. Note that P_{in}^1 and P_{in} model the impatience phenomenon of leaving the system without having been served. Similarly, P_{ih}^1, P_{ih} and μ_{ret} are the analogous parameters for automatic retrials. Making $P_{ih}^1 = 0$, at least one retrial will be performed. In that case, if the system were so loaded that the probability of a successful retrial could be considered negligible, the time elapsed since the first handover attempt until the system finally gives up and the session is dropped will be a sum of X independent and identically distributed exponential random variables of mean μ_{ret}^{-1}. In our model the discrete random variable X follows a geometric distribution with mean $1/P_{ih}$, hence the total time from the first attempt until abandonment is described by an exponential random variable. In the light of the above discussion, our model represents a situation in which the blocked handover requests will keep retrying while the user remains within the handover area, being the sojourn time modeled as an exponential random variable. In cellular networks, this assumption has been shown to have a low impact on the performance measures of interest [24].

There are several performance parameters that are generally used to describe the behavior of this type of cellular systems with retrials and redials. By the one hand, the widely used blocking probabilities for both new sessions (P_b^n) and handovers (P_b^h). On the other hand, the mean number of users redialing (N_{red}) and handovers retrying (N_{ret}) can describe more accurately the reattempt phenomenon in this type of networks.

The considered model can be represented as a tridimensional (k, m, o) CTMC, being the first dimension (k) the number of sessions being served, the second dimension (m) the number of sessions in the redial orbit and the third dimension (o) the number of sessions in the retrial orbit. The state space can be represented by:

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \in \mathbb{Z}_+; o \in \mathbb{Z}_+\}.$$

The transition rates of this model are represented in Table 1. Additionally, in Fig. 2 we can see the transition diagram. The main mathematical features of this queueing model are the fact of having two infinite dimensions —the state space of the model is $\{0, \ldots, C\} \times \mathbb{Z}_+ \times \mathbb{Z}_+$— and the space-heterogeneity along them. This

[1]For a real number x, $\lfloor x \rfloor$ is the largest integer not greater than x.

Table 1: Transition rates of the exact model.

Transition	Condition	Rate
$(k,m,o) \to (k+1,m,o)$	$0 \le k \le L-1$	λ
	$k = L$	$\lambda_h + f\lambda_n$
	$L < k < C$	λ_h
$(k,m,o) \to (k+1,m,o-1)$	$0 \le k \le C-1$	$o\mu_{ret}$
$(k,m,o) \to (k,m,o-1)$	$k = C$	$o\mu_{ret}P_{ih}$
$(k,m,o) \to (k+1,m-1,o)$	$0 \le k \le L-1$	$m\mu_{red}$
	$k = L$	$m\mu_{red}f$
$(k,m,o) \to (k,m-1,o)$	$k = L$	$m\mu_{red}(1-f)P_{in}$
	$L < k \le C$	$m\mu_{red}P_{in}$
$(k,m,o) \to (k-1,m,o)$	$1 \le k \le C$	$k\mu$
$(k,m,o) \to (k,m,o+1)$	$k = C$	$\lambda_h(1-P_{ih}^1)$
$(k,m,o) \to (k,m+1,o)$	$k = L$	$\lambda_n(1-P_{in}^1)(1-f)$
	$L < k \le C$	$\lambda_n(1-P_{in}^1)$

heterogeneity is produced by the retrial and redial rates, which respectively depend on the number of customers in the retrial and the redial orbits.

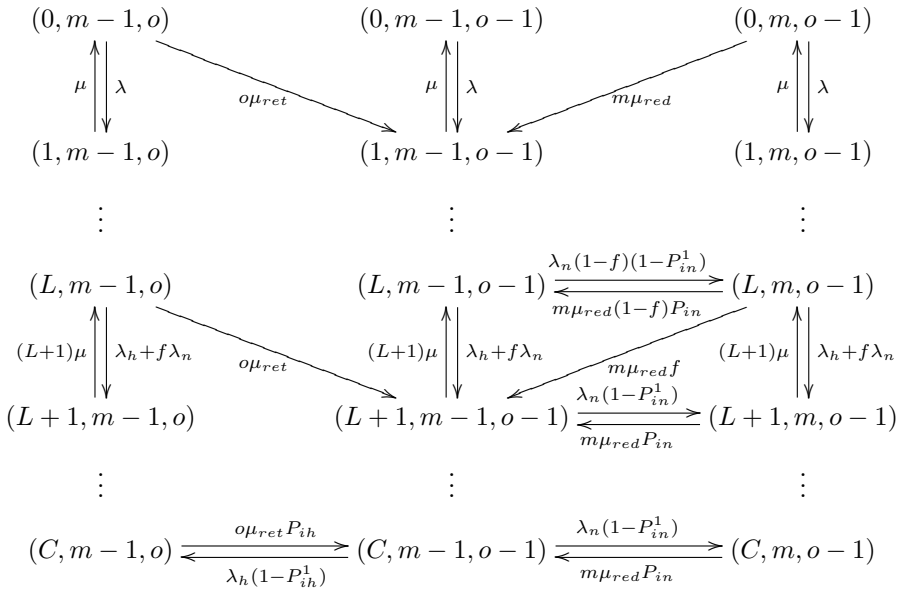

Figure 2: Transition diagram of the exact model.

3. Solving methods

It is known that the classical theory —see, e.g., [10]— is developed for random walks on the semi-strip $\{0,\dots,C\} \times \mathbb{Z}_+$ with infinitesimal transitions subject to conditions of space-homogeneity. When the space-homogeneity condition does not hold the problem of calculating the equilibrium distribution has not been addressed beyond approximate methods [25, 26]. Indeed, if we focus on the simpler case of multiserver retrial queues —with only one retrial orbit— it can be emphasized the absence of closed form solutions for the main performance characteristics when $C > 2$ [11].

Obviously to solve the system under study it will also be necessary to resort to approximate models and numerical methods of solution. Although other approaches exist, for the comparison against VE we have chosen the three most well-known methods that are able to solve the problem under study, being:

1. Double Truncation (DT): this method consists of a simple truncation for each orbit. Although it is expected to have low accuracy, it is also the simplest method possible.
2. Double FM (DFM): this belongs to the family of finite truncated methods and was successfully applied to a system like the one under study in [27].
3. Truncation and generalization (TNR): this method is a hybrid between a finite truncated and a generalized truncated method, as it performs a truncation in the redial orbit and a generalization like the proposed in [28] for the automatic retrial orbit. Note that we can not use a generalization in both orbits simultaneously because the resulting model is not solvable.

3.1. Double truncation (DT)

The easiest and more intuitive method to solve the proposed model lies in the truncation of the infinite dimensions of the state space. The first time this method was applied to approximate the retrial phenomenon was in [29], where it was applied to a single retrial orbit. In our case, it must be applied to both the redial and retrial orbits, truncating them beyond levels Q_n and Q_h respectively and obtaining the next state space:

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \leq Q_n; o \leq Q_h\}.$$

Obviously, by increasing the values of Q_n and/or Q_h the considered state space in the approximation is enlarged and the accuracy of the solution is expected to improve at the expense of a higher computational cost.

The stationary probability distribution can be obtained by solving $\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}$ along with the normalization condition. As $\mathbf{Q}$ is a finite matrix this system can be solved by any of the standard methods defined in classical linear algebra. However, we can exploit the block tridiagonal structure of $\mathbf{Q}$ using the algorithm 0 defined in [30], which allows us to reduce the computational cost, although there are other proposals useful for that purpose like [31, 32].

3.2. Double FM (DFM)

As DT, DFM belongs to the family of finite truncated methods [**?**]. These methods consist of replacing the original infinite state space by a finite one. However, DFM is more sophisticated than DT as it introduces in some sense the effect of the truncated states.

In [33] we developed FM, a generalization of the approximation method proposed in [34]. Although developed initially for a single orbit scenario, FM was applied to a system like the one under study in [27]. In this case FM has been applied to both retrial and redial orbits —resulting in DFM—, reducing the state space to a finite set by aggregating all states beyond a given occupancy of the orbits: Q_n (Q_h) defines the occupancy from which the states in the redial (retrial) orbit are aggregated. As in DT, by increasing the values of Q_n and/or Q_h the considered state space in the approximation is enlarged and the accuracy of the solution improves at the expense of a higher computational cost.

Due to that aggregation two new parameters for each orbit are introduced. The parameter M_n denotes the mean number of users in the redial orbit conditioned to those states where there are at least Q_n users in the orbit, i.e. $M_n = E(m|m \geq Q_n)$. The probability that after a successful redial the number of users in the redial orbit does not drop below Q_n is represented by p_n. For the retrial orbit the parameters M_h and p_h are defined analogously.

Therefore the aggregation of states produces the same approximate state space as DT:

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \leq Q_h; o \leq Q_h\}.$$

However, in DFM states of the form $(\cdot, Q_n, \cdot)$ represent the situation where at least Q_n users are in the redial orbit. Likewise the states of the form $(\cdot, \cdot, Q_h)$ represent the situation where there are Q_h or more users in the retrial orbit. The introduction of the effect of the truncated states assures a better accuracy than the obtained by DT.

In order to compute the steady-state probabilities of the system ($\pi(k, m, o)$) the actual values of the parameters M_n, p_n, M_h and p_h should be known. Following the procedure shown in [27] we can express parameters M_n, p_n, M_h and p_h in terms of the steady-state probabilities:

$$p_h = \frac{\sum_{m=0}^{Q_n} \pi(C, m, Q_h)}{\sum_{m=0}^{Q_n} \left[\pi(C, m, Q_h) + \pi(C, m, Q_h - 1)\right]}. \tag{1}$$

$$M_h = \frac{\lambda_h(1 - P_{ih}^1)\left(\sum_{m=0}^{Q_n} \left[(\pi(C, m, Q_h) + \pi(C, m, Q_h - 1)\right]\right)}{\mu_{ret}\left(\sum_{k=0}^{C-1}\sum_{m=0}^{Q_n} \pi(k, m, Q_h) + P_{ih}\sum_{m=0}^{Q_n} \pi(C, m, Q_h)\right)}. \tag{2}$$

$$p_n = \frac{\zeta_1}{\zeta_2} \quad ; \quad M_n = \frac{\lambda_n(1 - P_{in}^1)\zeta_2}{\mu_{red}\zeta_3}, \tag{3}$$

where

$$\zeta_1 = \sum_{k=L+1}^{C}\sum_{o=0}^{Q_h} \pi(k, Q_n, o) + (1 - f)\sum_{o=0}^{Q_h} \pi(L, Q_n, o).$$

$$\zeta_2 = \sum_{k=L+1}^{C}\sum_{o=0}^{Q_h} \left[\pi(k, Q_n - 1, o) + \pi(k, Q_n, o)\right] + (1 - f)\sum_{o=0}^{Q_h} \left[\pi(L, Q_n - 1, o) + \pi(L, Q_n, o)\right].$$

$$\zeta_3 = \sum_{k=0}^{L-1}\sum_{o=0}^{Q_h} \pi(k, Q_n, o) + [f + (1 - f)P_{in}]\sum_{o=0}^{Q_h} \pi(L, Q_n, o) + P_{in}\sum_{k=L+1}^{C}\sum_{o=0}^{Q_h} \pi(k, Q_n, o).$$

The global balance equations, the normalization equation and Eqs. (1)–(3) form a system of simultaneous non-linear equations, which can be solved using —for instance— the iterative procedure sketched next: set $p_n = p_h = 0$, $M_n = Q_n$ and $M_h = Q_h$ and compute the steady-state probabilities using the algorithm defined in [30], now compute M_n, p_n, M_h, p_h using Eqs. (1)–(3) and start again. In all of our numerical experiments we repeated the iterative procedure until the relative difference between two consecutive iterations was less than 10^{-3} for all four parameters.

3.3. Truncation and generalization (TNR)

While the two previous approximations consider a finite truncated method for each retrial orbit, this method considers the use of a generalized truncated method in one of the two orbits. Obviously, we can not use a generalized method for both orbits as the resulting model would not be solvable. For this reason, we have applied a generalized truncated method for the automatic retrial orbit and a finite truncated method for the redial orbit. More specifically, we have used a simple truncation as the finite truncated method for the redial orbit. This way the system considers only the states in which $m \leq Q_n$. On the other hand, the method chosen for the retrial orbit is the method proposed by Neuts and Rao in [28]. This method, which was proved to converge to the original model in [35], is based on the homogenization of the model beyond a given level Q_h, which supposes to restrict the maximum automatic retrial rate, i.e.

$$\mu_{ret}(o) = \begin{cases} o\mu_{ret} & \text{if } o < Q_h \\ Q_h\mu_{ret} & \text{if } o \geq Q_h \end{cases}$$

Therefore, the resulting space state is defined by

$$\mathcal{S} := \{(k, m, o) : k \leq C; m \leq Q_n; o \in \mathbb{Z}_+\}$$

With these two approximations we have to solve a system which state space presents two finite dimensions and an infinite one, being the infinite dimension homogeneous beyond a given level Q_h. So, we can solve the resulting system and obtain the steady state probabilities making use of the matrix-geometric solutions for stochastic models proposed by Neuts in [10].

4. Proposed method: value extrapolation

All the approximate methods described in the previous sections compute the steady state probabilities using the balance equations in order to obtain the desired performance parameters, i.e. they solve the linear system of equations:

$$\pi(s)\sum_{s'} q_{ss'} = \sum_{s'} \pi(s')q_{ss'} \qquad \forall s,$$

along with the normalization condition $\sum_s \pi(s) = 1$, where $q_{ss'}$ represents the transition rate from state s to s'.

Very recently, however, an alternative approach for evaluating infinite state space Markov processes has been introduced by Leino et al. [16, 17, 18]. This approach, named Value Extrapolation (VE), does not rely on the probability of being in a certain state, but in a new metric called relative state values, that appear when we consider the system in its MDP setting. Formally, an MDP can be defined as a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}\}$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P}$ is a state transition function and $\mathcal{R}$ is a revenue function. The state of the system can be controlled by choosing actions a from $\mathcal{A}$, influencing in this way the state transitions. The transition function $\mathcal{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ specifies the transition rate to other states when a certain action is taken at a given state. The first characteristic of VE is the necessity of the definition of a revenue function that must be a function of the system state, i.e., $r(s)$. Following the definition of the revenue function for every state, we will also have a mean revenue rate of the entire process (r), which will be the performance metric we want to compute.

Once defined the MDP framework as well as the revenue function we are in a position to define the relative state values. It is obvious that after performing an action in state s the system will collect a revenue for that action ($r(s)$), but, as the number of transitions increases, the average revenue collected converges to r. The relative state value ($v(s)$) tells how much is the difference between the total revenue incurred when the system starts at state s and the total revenue incurred in a system for which the cost rate at all states is r. If we denote by t_n the time instants in which there is a change in the system state, then

$$v(s) = E\left[\sum_{n=0}^{\infty} \left(r(S(t_n)) - r\right)\Big| S(t_0) = s\right].$$

The equations that relate revenues, relative state values and transition probabilities are the Howard equations defined by:

$$r(s) - r + \sum_{s'} q_{ss'}(v(s') - v(s)) = 0 \quad \forall s.$$

There will be as many Howard equations as number of states, $|\mathcal{S}|$. The number of unknowns will be the $|\mathcal{S}|$ relative state values plus the expected revenue r, i.e, $|\mathcal{S}| + 1$ unknowns. As only the differences in the relative values appear in the Howard equations, we can set $v(\mathbf{0}) = 0$, so we will have a solvable linear system of equations with the same number of equations as unknowns.

However, a finite number of Howard equations are needed to solve the system and, therefore, we need to truncate the state space. Whereas the traditional truncation consists of doing $q_{ss'} = 0 \quad \forall s' \notin \hat{\mathcal{S}}$, VE performs a more efficient truncation. Basically, VE considers the relative state values outside $\hat{\mathcal{S}}$ that appear in the Howard equations as an extrapolation of some relative state values inside $\hat{\mathcal{S}}$. The objective of VE is to find a function $f(s)$ that interpolates some points $(s, v(s))$ for $s \in \hat{\mathcal{S}}$ so that it approximates also $(s, v(s))$ for $s \notin \hat{\mathcal{S}}$. It is important to choose a fitting function that makes the Howard equations remain a closed system of linear equations. The most common fitting functions that acomplish that fact are the polynomials. We can use all $(s, v(s))$-pairs of the state space into the fitting procedure —global fitting— or only a subset ($\mathcal{S}_f$) of them —local fitting. The choice of $\mathcal{S}_f$ will highly depend on the relative state value we want to extrapolate. Note also that function $f(s)$ and set $\mathcal{S}_f$ need to be chosen so that parameters have unambiguous values, i.e. in the case of choosing a polynomial as the fitting function, the number of different points in $\mathcal{S}_f$ has to be equal or greater than the number of coefficients in the polynomial. Note that if the relative values outside $\hat{\mathcal{S}}$ were correctly extrapolated, the results obtained by solving the truncated model would be exact.

4.1. Howard equations of the system

To obtain the Howard equations for a certain state of the system under study, we can classify these states into four different cases depending on the number of sessions being served (k). We next describe such cases and their corresponding Howard equations.

1. $\mathbf{k < L}$: states in which both new sessions and handovers are accepted. The transition rates that go out from these states are represented in Fig. 3. Therefore, the Howard equations related to these states are:

$$r(k,m,o) - r + \lambda[v(k+1,m,o) - v(k,m,o)] + k\mu[v(k-1,m,o) - v(k,m,o)] + \\ + m\mu_{red}[v(k+1,m-1,o) - v(k,m,o)] + o\mu_{ret}[v(k+1,m,o-1) - v(k,m,o)] = 0.$$

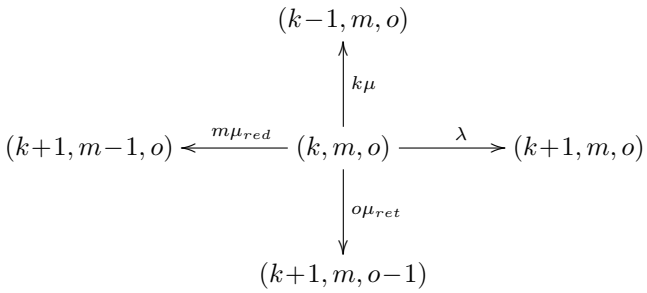

Figure 3: Transition rates when $k < L$.

2. $\mathbf{k = L}$: states in which handovers are accepted but new sessions are only accepted with probability $f = t - L$, being t the parameter that characterizes the FGC admission control policy. Figure 4 represents the transition rates going out from these states, obtaining the next Howard equation:

$$r(L,m,o) - r + (\lambda_h + \lambda_n f)[v(L+1,m,o) - v(L,m,o)] + L\mu[v(L-1,m,o) - v(L,m,o)] + \\ + m\mu_{red} f[v(L+1,m-1,o) - v(L,m,o)] + m\mu_{red}(1-f)P_{in}[v(L,m-1,o) - v(L,m,o)] + \\ + \lambda_n(1-f)(1-P_{in}^1)[v(L,m+1,o) - v(L,m,o)] + o\mu_{ret}[v(L+1,m,o-1) - v(L,m,o)] = 0.$$

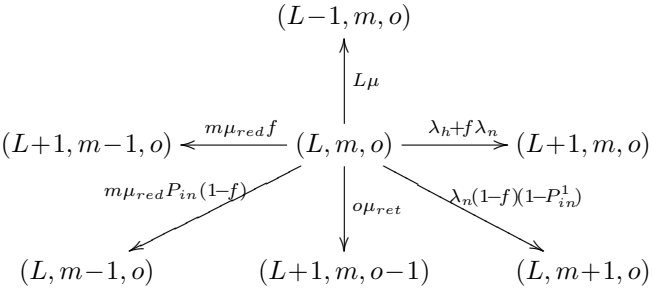

Figure 4: Transition rates when $k = L$.

3. $\mathbf{L < k < C}$: states where handovers are accepted but new sessions are blocked, as shown in Fig. 5. That leads to the Howard equation:

$$r(k,m,o) - r + \lambda_h[v(k+1,m,o) - v(k,m,o)] + k\mu[v(k-1,m,o) - v(k,m,o)] + \\ + m\mu_{red}P_{in}[v(k,m-1,o) - v(k,m,o)] + \lambda_n(1-P_{in}^1)[v(k,m+1,o) - v(k,m,o)] + \\ + o\mu_{ret}[v(k+1,m,o-1) - v(k,m,o)] = 0.$$

4. $\mathbf{k = C}$: states where both new sessions and handovers are blocked, being the transition rates as shown in Fig. 6 and their corresponding Howard equations:

$$r(C,m,o) - r + \lambda_h(1-P_{ih}^1)[v(C,m,o+1) - v(C,m,o)] + C\mu[v(C-1,m,o) - v(C,m,o)] + \\ + m\mu_{red}P_{in}[v(C,m-1,o) - v(C,m,o)] + \lambda_n(1-P_{in}^1)[v(C,m+1,o) - v(C,m,o)] + \\ + o\mu_{ret}P_{ih}[v(C,m,o-1) - v(C,m,o)] = 0.$$

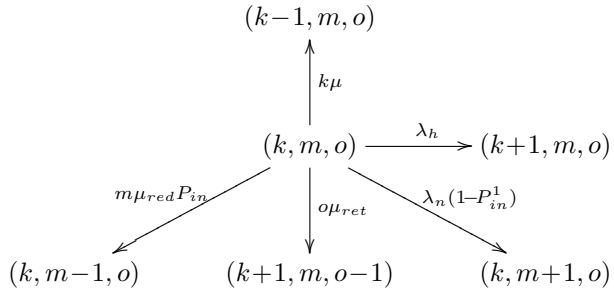

Figure 5: Transition rates when $L < k < C$.

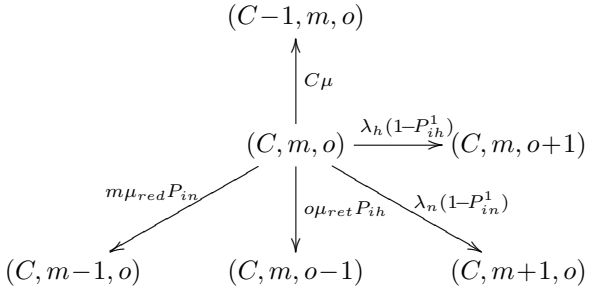

Figure 6: Transition rates when $k = C$.

Table 2: Revenue function definition.

Performance parameters	Symbol	Value
Handover blocking probability	P_b^h	$r(k,m,o) = 1$ for $k = C$, $\forall m$, $\forall o$
		$r(k,m,o) = 0$ otherwise
New session blocking probability	P_b^n	$r(k,m,o) = 1 - f$ for $k = L$, $\forall m$, $\forall o$
		$r(k,m,o) = 1$ for $k \geq L$, $\forall m$, $\forall o$
		$r(k,m) = 0$ otherwise
Mean number of handovers retrying	N_{ret}	$r(k,m,o) = o$ $\forall k$, $\forall m$, $\forall o$
Mean number of users redialing	N_{red}	$r(k,m,o) = m$ $\forall k$, $\forall m$, $\forall o$

4.2. Revenue function

As performance parameters are not computed from the steady state probabilities as usual, it is important to explain more carefully how they are computed. For that purpose we must set the inputs $r(s)$ in the Howard equations properly in order to make that the revenue rate of the entire process r is equal to the performance parameter we want to compute. In a nutshell, r will be the parameter we want to compute if we let $r(s)$ to be the value of that parameter when the system is in state s. Table 2 gives several examples on how $r(s)$ can be set in order to obtain certain performance parameters such as blocking probability of handover requests, blocking probability of new session requests, mean number of handovers in the automatic retrial orbit and mean number of new sessions in the redial orbit.

As an example, for the blocking probability of handover requests we define the revenue function to be one in those states in wich a handover is blocked, i.e. when $r(C,m,o) = 1$, $\forall m, \forall o$, and zero otherwise.

4.3. Polynomial fitting and solution

Note that in the system under study the number of states is infinite because both m and o can take any value in $\mathbb{Z}_+$, thus some truncation is needed. We have made a truncation similar to DT and DFM, obtaining a

truncated state space defined by:

$$\hat{\mathcal{S}} := \{s = (k, m, o) : k \leq C; m \leq Q_n; o \leq Q_h\}.$$

Therefore, in the system under study, we have truncated the state space beyond a value of Q_n (Q_h) for the occupancy of the redial (automatic retrial) orbit. However, in the Howard equations of the truncated state space appear the relative state value of some states that do not belong to the truncated state space, being $v(C, m, Q_h + 1)$ $\forall m$ and $v(k, Q_n + 1, o)$ for $k \geq L$ and $\forall o$. Therefore, we must extrapolate these two sets of states to obtain a closed system of equations. We have used a $(p_h - 1)$-th degree polynomial that interpolates the p_h points in $\{(j, v_j)|v_j = v(C, m, j), \forall m, Q_h - p_h < j \leq Q_h\}$ to extrapolate $v(C, m, Q_h + 1)$. To extrapolate $v(k, Q_n + 1, o)$ for $k \geq L$ we interpolate the p_n points in $\{(i, v_i)|v_i = v(k, i, o), k \geq L, Q_n - p_n < i \leq Q_n, \forall o\}$. Note that including value extrapolation neither increase the computational cost nor increase the number of Howard equations, remaining in $|\hat{\mathcal{S}}| = (C + 1) \times (Q_n + 1) \times (Q_h + 1)$.

After some algebra, and using the Lagrange basis to reduce the complexity of the procedure, we obtain a simple closed-form expression for the extrapolated value of both sets

$$v(C, m, Q_h + 1)^{(p_h)} = \sum_{j=0}^{p_h - 1} (-1)^j \binom{p_h}{j+1} v(C, m, Q_h - j), \forall m,$$

and

$$v(k, Q_n + 1, o)^{(p_n)} = \sum_{i=0}^{p_n - 1} (-1)^i \binom{p_n}{i+1} v(k, Q_n - i, o), \text{ for } k \geq L \text{ and } \forall o.$$

As stated above, including VE for the truncated states into the original Howard equations we obtain a closed system of linear equations, which can be expressed in its matrix form for simplicity reasons. Therefore the system can be seen as $\mathbf{xT} = \mathbf{b}$, where $\mathbf{x}$ is a vector with the $(C + 1) \times (Q_n + 1) \times (Q_h + 1)$ unknowns (r and the relative state values $v(s)$), $\mathbf{b}$ are the negative relative state values for the different states and matrix $\mathbf{T}$ represents the matrix of coefficients.

Unfortunately, unlike DT, DFM and TNR methods, in order to solve the system $\mathbf{xT} = \mathbf{b}$, we cannot utilize methodologies that make use of the block tridiagonal structure [30, 31] to effectively solve such systems. So we must use a general procedure to solve the linear system of equations, such as Gauss-Seidel, Gauss-Jordan or LU factorization methods.

A drawback of VE is that it is only able to compute one performance parameter each time we solve the system. Notwithstanding we can overcome this drawback in the following way. In a general manner, the solution of the system $\mathbf{xT} = \mathbf{b}$ can be obtained using the inverse matrix of $\mathbf{T}$ by doing $\mathbf{x} = \mathbf{bT^{-1}}$. Note also that choosing a different performance parameter to solve will only affect to the values in $\mathbf{b}$. Therefore, computing a second performance parameter will only increase the computation expenses by the cost of the product $\mathbf{bT^{-1}}$, as the rest of the process —specially the computation of the inverse matrix $\mathbf{T^{-1}}$— is solved only once. Similarly, we can compute several performance parameters with a marginal increase in the computation cost using LU factorization, as the first part of the procedure —the factorization, which is supposed to be the most computation consumption part— is done only once for the matrix $\mathbf{T}$.

5. Results and discussion

In this section a number of numerical examples are presented with the purpose of illustrating the capabilities and versatility of our model and the analysis methodology. The numerical analysis is also aimed at assessing a comparison between the proposed methodology and previous approaches not only in terms of accuracy but also in terms of computation cost.

For the numerical experiments a basic configuration is used and then the different parameters are varied, normally a single variation is introduced in each experiment. Thus, unless otherwise indicated, the value of the parameters will be those of the basic configuration: $C = 10$, $t = 9$, $\mu = 1$, $P_{ih}^1 = P_{in}^1 = 0$, $P_{ih} = P_{in} = 0.2$ and $\mu_{red} = \mu_{ret} = 1$. The values of λ_n and λ_h have been modified by means of the offered traffic $\rho = \lambda / C\mu$, being $\lambda = \lambda_n + \lambda_h$ and taking $\lambda_h = 2\lambda_n$ in all cases.

5.1. VE performance

5.1.1. Accuracy

The objective of this section is to study the performance of different extrapolation polynomials in a wide range of scenarios. In Table 3 we show four numbers in each cell which correspond, from top to bottom, to the

"exact" values of P_b^n, P_b^h, N_{red} and N_{ret}. Obviously, as stated in Section 3, for the system under study we are not able to compute the exact values of the most common performance parameters. For this reason, the first step is to assume that the exact value can be obtained choosing increasing and sufficiently high values of the truncation level. More specifically, we ran all methods presented in Section 3 and VE until the value of all the performance parameters under study had stabilized up to the 8th decimal digit. It must be noted that, due to the introduction of the impatience phenomenon modeled by P_{in}^1, P_{in}, P_{ih}^1 and P_{ih}, we will be able to consider values of $\rho > 1$.

Table 3: Exact values.

	$\mu_{red} = 1$ $\mu_{ret} = 1$	$\mu_{red} = 2$ $\mu_{ret} = 0.5$	$\mu_{red} = 0.5$ $\mu_{ret} = 2$	$\mu_{red} = 0.5$ $\mu_{ret} = 0.5$
$\rho = 0.4$	0.0188	0.0190	0.0187	0.0184
	0.0041	0.0041	0.0042	0.0040
	0.0311	0.0177	0.0568	0.0555
	0.0124	0.0234	0.0069	0.0225
$\rho = 0.6$	0.1394	0.1392	0.1385	0.1364
	0.0434	0.0425	0.0440	0.0415
	0.4046	0.2282	0.7292	0.7097
	0.2091	0.3835	0.1159	0.3722
$\rho = 0.8$	0.4067	0.4029	0.4066	0.4113
	0.1654	0.1613	0.1676	0.1642
	2.0402	1.1027	3.7475	3.7703
	1.1788	2.1649	0.6497	2.1943
$\rho = 1.0$	0.6873	0.6833	0.6876	0.7034
	0.3481	0.3436	0.3493	0.3547
	5.8427	3.0394	11.119	11.551
	3.5960	6.7727	1.9276	7.0124
$\rho = 1.2$	0.8443	0.8438	0.8420	0.8539
	0.5047	0.5043	0.5016	0.5119
	10.986	5.6097	21.344	22.115
	7.2880	14.115	3.7957	14.403

Just the same as in the case of a single retrial orbit [15], the general trend is that the higher value of the truncation level, the lower the relative error is. This is due to the fact that the system under consideration becomes more similar to the exact model as truncation level increases. However, in the system under study, there are two different truncation levels that must be specified, namely Q_n and Q_h. The purpose will be to determine the pair (Q_n, Q_h) that makes the cardinality of the problem $((C+1) \times (Q_n+1) \times (Q_h+1))$ as small as possible while a certain accuracy criterion is met. To fulfill these requirements we must define a direction of search to determine the desired (Q_n, Q_h) pair. To avoid an exhaustive search we have followed the next algorithm in order to determine (Q_n, Q_h), similar to the one proposed in [36].

Algorithm: Calculation of the pair (Q_n, Q_h).

Step 0. Let $Q_n = 0$ and $Q_h = 0$.

Step 1. Increase successively Q_n and Q_h following the diagonal, i.e. $Q_n \to Q_n+1$ and $Q_h \to Q_h+1$, until it is satisfied that

$$\frac{|\Psi^{approx} - \Psi^{exact}|}{\Psi^{exact}} < 10^{-4},$$

being Ψ any of the values $\{P_b^n, P_b^h, N_{ret}, N_{red}\}$.
Let (Q_N, Q_H) be the final values.

Step 2. Starting with (Q_N, Q_H), decrease successively Q_n ($Q_n \rightarrow Q_n - 1$) while

$$\frac{|\Psi^{approx} - \Psi^{exact}|}{\Psi^{exact}} < 10^{-4}.$$

Let $W_1 = (a+1)(Q_H+1)$, where a is the final value of Q_n.

Step 3. Starting with (Q_N, Q_H), decrease successively Q_h ($Q_h \rightarrow Q_h - 1$) while

$$\frac{|\Psi^{approx} - \Psi^{exact}|}{\Psi^{exact}} < 10^{-4}.$$

Let $W_2 = (Q_N+1)(b+1)$, where b is the final value of Q_h.

Step 4. If $W_1 < W_2$ then the result of the algorithm is (a, Q_H). Otherwise, the result is (Q_N, b).

In short, we increase (Q_n, Q_h) along the diagonal until we obtain a system that fulfills the desired accuracy and later we decrease both parameters separately following descendent directions of the coordinate axis and finally take the best solution in terms of the cardinality of the problem. The rationale behind this last horizontal or vertical movement is the fact that, generally, $Q_n \neq Q_h$, and this cannot be accomplished only with the diagonal movement, so the solution with this last movement improves the initial diagonal movement.

Table 4: Minimum complexity (Ω) to obtain relative errors lower than 10^{-4} in P_b^n/P_b^h.

$\{\mu_{red}, \mu_{ret}\}$	ρ	VE1	VE2	VE3	VE4	VE5	VE6
{1,1}	0.4	25/30	**12**/**12**	16/16	25/25	36/36	49/49
	0.8	144/144	49/72	64/72	49/**35**	**36**/36	49/49
	1.2	484/506	342/342	240/**36**	**98**/120	121/132	99/120
{2,0.5}	0.4	20/25	**12**/**12**	16/16	25/25	36/36	49/49
	0.8	130/90	45/55	56/64	**36**/**30**	**36**/36	49/49
	1.2	-/-	432/336	280/170	**99**/**136**	126/144	135/168
{0.5,2}	0.4	20/25	**12**/**12**	16/16	25/25	36/36	49/49
	0.8	160/160	66/110	80/100	56/49	**36**/**42**	49/49
	1.2	-/-	-/-	400/-	154/189	**144**/**187**	162/198
{0.5,0.5}	0.4	25/30	**9**/**9**	16/16	25/25	36/36	49/49
	0.8	224/160	100/121	90/100	48/**35**	**36**/36	49/49
	1.2	-/-	-/-	-/-	**168**/280	195/**196**	441/378

In Tables 4 and 5 we show the minimum complexity of the problem needed to fulfill a relative error lower than 10^{-4} for different performance parameters, for different loads (ρ) and reattempt rates ($\{\mu_{red}, \mu_{ret}\}$) and for different orders of the extrapolation polynomial. More specifically, Table 4 refers to parameters P_b^n and P_b^h and Table 5 is its equivalent for parameters N_{red} and N_{ret}. Note that VEx denotes the use of an extrapolation polynomial of order x for both orbits ($p_n = p_h = x+1$). Note also that the numbers shown in each cell represent the product $(Q_n+1) \times (Q_h+1)$ which defines the complexity and it is denoted by Ω, although the cardinality of the problem should also include the factor $(C+1)$. Notwithstanding, we have omitted this factor as it is common to all cases. Therefore, the best order for the extrapolation polynomial will be the one that has the lowest Ω, which is in bold in the table. Moreover, we denote by "-" those cases in which the computer could not obtain a result because of lack of memory[2].

From the results in Tables 4 and 5 we can conclude that there is not a clear choice in the order of the extrapolation polynomial that can get the lowest Ω in all cases. Neither the lowest nor the highest orders offer the best results. When the load is not high ($\rho = 0.4$), VE2 offers the lowest complexities, due to the fact that VE3-VE6 offer the result of the minimum Ω they require to work —e.g. to extrapolate with VE4 at least $Q_n = Q_h = 4$ is needed and therefore the minimum Ω required to use VE4 is $(4+1) \times (4+1) = 25$. When the

[2]Results have been obtained using Matlab running in an Intel Core 2 Quad Q6600 with 4GB RAM memory.

Table 5: Minimum complexity (Ω) to obtain relative errors lower than 10^{-4} in N_{red}/N_{ret}.

$\{\mu_{red}, \mu_{ret}\}$	ρ	VE1	VE2	VE3	VE4	VE5	VE6
{1,1}	0.4	36/35	**12**/**15**	16/16	25/25	36/36	49/49
	0.8	117/132	110/90	72/48	**35**/**25**	42/36	49/49
	1.2	380/**81**	255/272	160/180	196/156	**64**/90	121/121
{2,0.5}	0.4	30/30	**12**/**16**	16/**16**	25/25	36/36	49/49
	0.8	121/144	99/55	64/48	**35**/**25**	42/36	49/49
	1.2	-/-	**81**/-	221/220	204/**64**	160/84	98/117
{0.5,2}	0.4	36/35	**12**/**15**	16/16	25/25	36/36	49/49
	0.8	171/56	105/120	48/60	**45**/**36**	48/49	49/49
	1.2	-/-	-/-	-/275	230/**120**	240/162	**140**/144
{0.5,0.5}	0.4	36/36	**12**/**12**	16/16	25/25	36/36	49/49
	0.8	180/182	168/132	**42**/70	45/**25**	48/36	49/49
	1.2	-/-	-/-	375/425	396/336	**285**/**168**	506/506

retrial orbits are more heavily loaded, VE4 is a good choice, as it offers low values of Ω and requires less points for the extrapolation than, for example, VE5. Moreover, in many cases in which VE4 is not the best polynomial, the Ω that is able to get is not far from the optimal. Therefore, hereafter we will use the polynomial of order 4 (VE4) and we will simply denote it as VE. Regarding the results obtained for the different performance parameters, we can conclude that results are quite similar in all of them at least qualitative and, in many cases, also quantitatively.

5.1.2. Computation cost

Initially, a drawback of VE is that it is only able to compute a performance parameter each time the system is solved. However, as sketched in Section 4, by solving several performance parameters simultaneously the computation cost is not expected to increase linearly with the number of parameters. In Fig. 7 we show in dotted lines the computation time needed to obtain a different number (g) of performance parameters. Moreover, as the different curves are very close to each other, we also show in solid lines the relative value of the time with regard to the curve $g = 1$. Observing Fig. 7 it follows that for a fixed value of Ω the computation cost growth is negligible as we compute more performance parameters, having cost growths lower than 3% when we compute four performance parameters instead of only one. However, it must also be noted that the total computing time is, in all cases, very low. Finally, it is worth noting that the computation cost in VE does not depend on the order of the extrapolation polynomial as it does not affect the size of the resulting system.

5.2. Comparison among different methods

5.2.1. Accuracy

The objective of this section is to compare the performance of VE with the methods described in Section 3 (DT, DFM and TNR), which are based on the traditional approach of solving the steady state probabilities using the balance equations for computing the performance parameters of interest.

In Table 6 we show the minimum values of Ω needed to obtain a relative error lower than 10^{-4} for N_{red}. The results for the rest of performance parameters have been omitted as N_{red} is usually the worst case for all methods and results are found to be qualitatively equivalent for all performance parameters. We have studied a wide range of scenarios, modifying not only the offered traffic $\rho = \lambda/C\mu$, but also the retrial rates and the proportion between these rates. We show in bold the best results, i.e. those that offer the minimum complexity Ω. Results show that VE clearly outperforms classical methods as it needs a much lower value of Ω to achieve the desired accuracy in all the scenarios under study. Moreover, and what is probably more important, there are some scenarios where VE is the only method that is able to get a result due to the complexity of those scenarios, produced by having low reattempt rates. Comparing the rest of the methods among them we conclude that DT is the worst method, as it could be expected. The comparison between DFM and TNR is not so clear because the number of cases where one outperforms the other is not far from the opposite.

Tables 7 and 8 show the comparison when other system parameters are modified. In these cases for fixed values of $\rho = 0.8$ and $\mu_{red} = \mu_{ret} = 1$ we have taken different values for the abandoning probability $P_i = P_{in} = P_{ih}$ in Table 7, and for the system capacity, C, in Table 8.

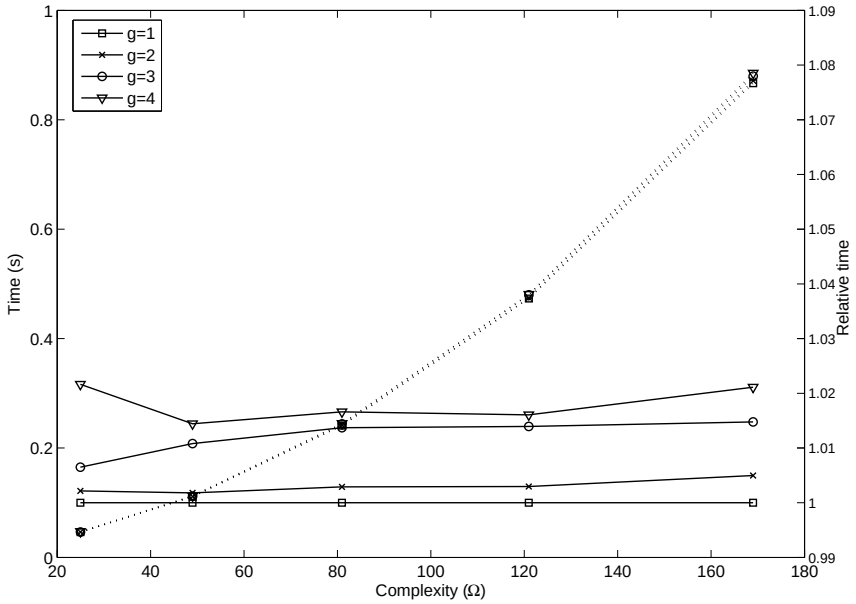

Figure 7: Computation cost when solving g performance parameters simultaneously.

In the case of considering different values for P_i we observe that VE outperforms the others methods in all cases. We can also observe that the required Ω decreases as P_i increases, since the number of users retrying decreases as we increase the abandoning probability P_i, and therefore we require a lower value of Q_n/Q_h to obtain a good accuracy. Moreover, VE is the only method that is able to solve the scenario in which we do not consider the impatience phenomenon, i.e. where $P_i = 0$. Note also that the complexity required by VE method is constant for values of $P_i \geq 0.4$. We can conclude from this observation that VE is being limited by the order of the extrapolation polynomial chosen —fixed to 4 as stated in the previous section— and not by the system configuration.

In the case of modifying the value of C, we have also modified t accordingly using $t = C - 1$ in order to assure some handover priorization. The results show that VE also outperform the rest of the methods under study in all the studied cases.

5.2.2. Computation cost

Although it is shown that VE clearly outperforms the other methods in terms of accuracy, it is also interesting to study their associated computation cost. In Table 9 we plot the time needed to achieve a relative error of 10^{-4} for N_{red} using the different methods under study. Results should be interpreted carefully, because computation cost highly depend on the algorithm used to solve the resulting system of equations. More concretely, in order to compute matrix $\mathbf{R}$ that appear in TNR we have used the logarithmic reduction algorithm as proposed in [26, Section 8.4], using a precision of 10^{-6} for the iterative procedure. Moreover, for solving the systems obtained with the DT, DFM and TNR methods we have made use of the efficient algorithm described in [30] that takes advantage of the block-tridiagonal structure that presents the infinitesimal generator. Unfortunately, the linear system of equations obtained in VE has no longer such a block-tridiagonal structure, and therefore we must use a more general algorithm. More concretely, we have used LU factorization. Moreover, for VE we show the time needed to compute four performance parameters simultaneously, i.e. $g = 4$. Table 9 shows that VE is faster than the other methods under study.

From a practical perspective, it is perhaps more interesting to consider accuracy along with computation time. Figure 8 shows a joint representation of both parameters. As the figure shows, VE yields much higher accuracy than any other method for a given computation time. In Fig. 9 we show a similar result but now in terms of the number of floating point operations (flops) instead of the computation time.

Therefore, the application of the VE approach can be strongly recommended, especially in those cases where computation time is a concern. However it can be seen that in the system under study the computation times needed for any of the methods are not very high from a human point of view. For that reason, the time results should be compared qualitatively, as the time units may be different from just seconds when we solve more complex systems or when we have to solve the basic retrial system several times —for example to balance the

Table 6: Minimum complexity (Ω) to obtain relative errors lower than 10^{-4} in N_{red}.

$\{\mu_{red}, \mu_{ret}\}$	ρ	DT	DFM	TNR	VE
{1,1}	0.4	64	48	48	**25**
	0.6	143	72	91	**25**
	0.8	324	208	180	**35**
	1.0	550	360	400	**110**
	1.2	930	324	651	**196**
{2,0.5}	0.4	56	49	48	**25**
	0.6	132	100	99	**25**
	0.8	304	176	182	**35**
	1.0	522	378	196	**108**
	1.2	-	-	640	**204**
{0.5,2}	0.4	63	40	45	**25**
	0.6	150	112	90	**25**
	0.8	325	242	225	**45**
	1.0	-	-	-	**72**
	1.2	-	-	-	**230**
{0.5,0.5}	0.4	63	56	54	**25**
	0.6	180	126	135	**25**
	0.8	528	352	240	**45**
	1.0	-	-	-	**195**
	1.2	-	-	-	**396**

Table 7: Minimum complexity (Ω) to obtain relative errors lower than 10^{-4} in N_{red}.

P_i	DT	DFM	TNR	VE
0.0	-	-	-	**95**
0.2	324	208	180	**35**
0.4	196	100	126	**25**
0.6	156	99	108	**25**
0.8	121	90	88	**25**
1.0	110	90	80	**25**

incoming handover rate to the outgoing handover rate, as shown in [37]—.

6. Conclusions

In mobile communication systems like cellular networks, Mobile IP or the recently defined IEEE 802.16e and IEEE 802.20 networks, mobile operators must guarantee seamless mobility to its customers. In these networks, repeated attempts occur due to user redials when their session establishments are blocked and also due to automatic retries when a handover fails. The impact of both phenomena plays an important role in the system performance and, therefore, it should not be ignored. The Markovian model describing such a complex network is a multiserver retrial system that presents space-heterogeneity along two infinite dimensions. However, when the number of servers is higher than two the absence of closed-form solutions for the main performance characteristics can be emphasized, so it is mandatory to develop approximate methods. To the best of our knowdledge, all the methods studied in the literature to solve these systems are based on their steady state probabilities. In this paper we propose an alternative method based on a different metric: the relative state values and the Howard equations that relate them. This method performs an efficient truncation of the state space, because the relative state values just outside the truncated state space are extrapolated using some known relative state values.

We have compared the proposed method with the most well-known approaches appeared in the literature so far. The results show that the proposed method greatly outperforms previous approaches not only in terms of

Table 8: Minimum complexity (Ω) to obtain relative errors lower than 10^{-4} in N_{red}.

C	DT	DFM	TNR	VE
5	195	132	135	**25**
10	324	208	180	**35**
15	378	252	189	**48**
20	462	300	264	**56**
25	-	294	-	**56**

Table 9: Time (s) needed to obtain relative errors lower than 10^{-4} in N_{red}.

$\{\mu_{red}, \mu_{ret}\}$	ρ	DT	DFM	TNR	VE
{1,1}	0.4	0.086	0.125	0.106	**0.047**
	0.6	0.339	0.310	0.435	**0.047**
	0.8	1.255	3.710	1.368	**0.098**
	1.0	3.577	15.33	4.798	**0.448**
	1.2	8.875	11.36	11.17	**1.111**
{2,0.5}	0.4	0.085	0.128	0.125	**0.047**
	0.6	0.276	0.604	0.368	**0.047**
	0.8	1.059	2.479	0.998	**0.114**
	1.0	2.535	17.55	1.513	**0.248**
	1.2	-	-	5.709	**0.669**
{0.5,2}	0.4	0.106	0.113	0.139	**0.047**
	0.6	0.398	0.758	0.530	**0.047**
	0.8	1.819	4.425	2.794	**0.246**
	1.0	-	-	-	**0.256**
	1.2	-	-	-	**12.44**
{0.5,0.5}	0.4	0.104	0.167	0.150	**0.047**
	0.6	0.500	0.975	0.732	**0.047**
	0.8	3.282	14.47	3.790	**0.246**
	1.0	-	-	-	**1.583**
	1.2	-	-	-	**9.908**

accuracy, but also in terms of computation cost. Moreover, we have shown that in some scenarios the proposed method is the only one that is able to guarantee a certain accuracy. For all those reasons the proposed method is highly recommendable to solve this type of systems.

Acknowledgements

This work was supported by the Spanish Government (30% PGE) and the European Commission (70% FEDER) through projects TSI2007-66869-C02-02 and TIN2008-06739-C04-02/TSI.

References

[1] Artalejo J R. Accessible Bibliography on Retrial Queues. Mathematical and Computer Modelling ; 30 (1999) 1–6.

[2] IEEE 802.16 standard. http://www.ieee802.org/16/pubs/80216e.html October 2008.

[3] Bolton W, Xiao Y, Guizani M. IEEE 802.20: mobile broadband wireless access. IEEE Wireless Communications 14 (2007) 84–95.

[4] Cohen J W. Basic problems of telephone traffic theory and the influence of repeated calls. Philips Telecommunication Review 18 (1957) 49–100.

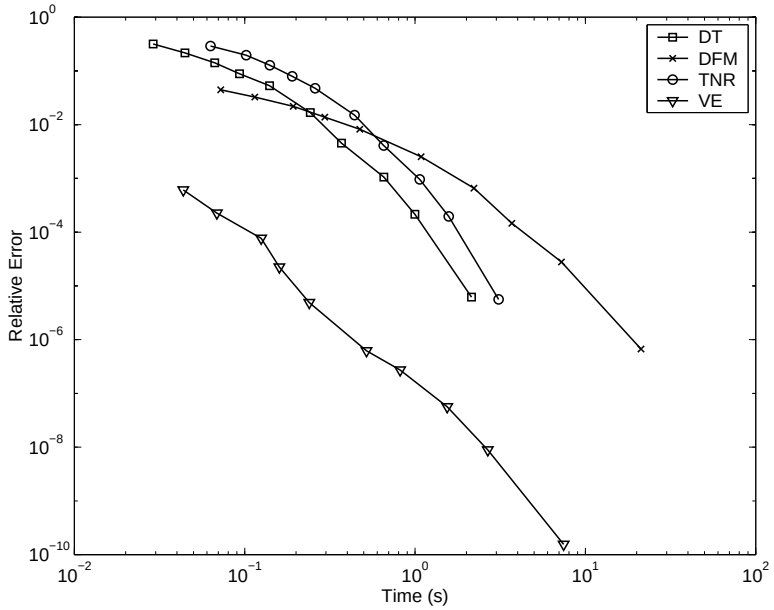

Figure 8: Computation time for different methods.

[5] Mouly M, Pautet M B. The GSM system for mobile communications. Published by the authors 1992.

[6] Onur E, Deliç H, Ersoy C, Çağlayan M U. Measurement-based replanning of cell capacities in GSM networks. Computer Networks 39 (2002) 749–767.

[7] Almasi B, Roszik J, Sztrik J. Homogeneous Finite-Source Retrial Queues with Server Subject to Breakdowns and Repairs. Mathematical and Computer Modelling 42 (2005) 673–682.

[8] Gharbi N, Dutheillet C, Ioualalen M. Colored stochastic Petri nets for modelling and analysis of multiclass retrial systems. Mathematical and Computer Modelling 49 (2009) 1436–1448.

[9] Boualem M, Djellab N, Aïssani D. Stochastic inequalities for M/G/1 retrial queues with vacations and constant retrial policy. Mathematical and Computer Modelling 50 (2009) 207–212.

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

[11] Artalejo J R, Gómez-Corral A. Retrial queueing systems, a computational approach. Springer, 2008.

[12] Falin G, Templeton J. Retrial Queues. Chapman & Hall 1997.

[13] Greenberg BS, Wolff R W. An upper bound on the performance of queues with returning customers. Journal of Applied Probability 24 (1987) 466-–475.

[14] Stepanov S N. Markov models with retrials: tje calculation methods of stationary performance measures based on the concept of truncation. Mathematical and Computer Modelling, 30 (1999), 207–228.

[15] Domenech-Benlloch MJ, Gimenez-Guzman JM, Pla V, Martinez-Bauset J, Casares-Giner V. Generalized Truncated Methods for an Efficient Solution of Retrial Systems. Mathematical Problems in Engineering 2008, Article ID 183089.

[16] Leino J, Penttinen A, Virtamo J. Flow level performance analysis of wireless data networks: A case study. In: Proceedings of IEEE ICC 2006; 3:961–966.

[17] Leino J, Virtamo J. An approximative method for calculating performance measures of Markov processes. In: VALUETOOLS 2006.

[18] Leino J, Virtamo J. Determining the moments of queue-length distribution of discriminatory processor-sharing systems with phase-type service requirements. In Proceedings of 3rd EuroNGI Conference on Next Generation Internet Networks 2007. p. 205–208.

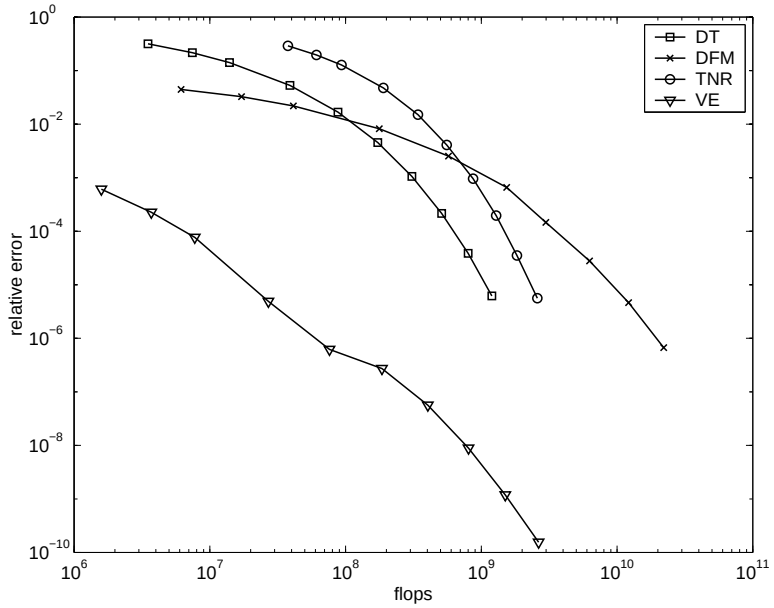

Figure 9: Number of floating point operations (flops) for different methods.

[19] Gimenez-Guzman JM, Domenech-Benlloch MJ, Pla V, Casares-Giner V, Martinez-Bauset J. Value Extrapolation Technique to Solve Retrial Queues: a Comparative Perspective. ETRI Journal 30 (2008) 492–494.

[20] Orlik P, Rappaport S. On the Handoff Arrival Process in Cellular Communications. Wireless Networks Journal (WINET) 7 (2001) 147–157.

[21] Khan F, Zeghlache D. Effect of Cell Residence Time Distribution on the Performance of Cellular Mobile Networks. In: Proceedings of IEEE VTC 1997. p. 949–953.

[22] Bonald T, Roberts J. Congestion at flow level and the impact of user behaviour. Computer Networks 42 (2003) 521–536.

[23] Ramjee R, Nagarajan R, Towsley D. On optimal call admission control in cellular networks. Wireless Networks Journal 3 (1997) 29-–41.

[24] Pla V, Casares-Giner V. Effect of the handoff area sojourn time distribution on the performance of cellular networks. In: Proceedings of IEEE MWCN 2002. p. 401–405.

[25] Bright L, Taylor P G. Calculating the equilibrium distribution of level dependent quasi-birth-and-death processes. Communications in Statistics-Stochastic Models 11 (1995) 497–525.

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

[27] Gimenez-Guzman JM, Domenech-Benlloch MJ, Pla V, Casares-Giner V, Martinez-Bauset J. Guaranteeing Seamless Mobility with User Redials and Automatic Handover Retrials. Journal of Universal Computer Science 14 (2008) 1597–1624.

[28] Neuts M, Rao B. Numerical investigation of a multiserver retrial model. Queueing systems 7 (1990)169–190.

[29] Wilkinson R I. Theories for toll traffic engineering in the USA. The Bell System Technical Journal 35 (1956) 421-–514.

[30] Servi L D. Algorithmic solutions to two-dimensional birth-death processes with application to capacity planning. Telecommunication Systems 21 (2002) 205–212.

[31] Gaver D, Jacobs P, Latouche G. Finite birth-and-death models in randomly changing environments. Advances in Applied Probability 16 (1984) 715–731.

[32] Ye J, Li S-Q. Folding algorithm: a computational method for finite QBD processes with leveldependent transitions. IEEE Transactions on Communications 42 (1994) 625-–639.

[33] Domenech-Benlloch MJ, Gimenez-Guzman JM, Martinez-Bauset J, Casares-Giner V. Efficient and accurate methodology for solving multiserver retrial systems. IEE Electronic Letters 41 (2005) 967–969.

[34] Ajmone Marsan M, De Carolis G, Leonardi E, Lo Cigno R, Meo M. Efficient estimation of call blocking probabilities in cellular mobile telephony networks with customer retrials. IEEE Journal on Selected Areas in Communications 19 (2001) 332–346.

[35] Anisimov VV, Artalejo JR. Approximation of multiserver retrial queues by means of generalized truncated models. Top 10 (2002) 51-–66.

[36] Artalejo J R, Pla V. On the impact of customer balking, impatience and retrials in telecommunication systems. Computers & Mathematics with Applications 57 (2009) 217 – 229.

[37] Ajmone Marsan M, De Carolis G, Leonardi E, Lo Cigno R, Meo M. How many cells should be considered to accurately predict the performance of cellular networks? In: Proceedings European Wireless 1999.