Paper

Queuing models for cellular networks with generalised Erlang service distributions

Aruna Jayasuriya

Abstract — Providing seamless handover is one of the major problems in mobile communication environments. Careful dimensioning of the network and the underlying teletraffic analysis plays a major role in determining the various grade of services (GoSs) that can be provided at various network loads for handover users. It has been shown that the channel holding time of a cell, one of the important parameters in any teletraffic analysis, can be accurately modelled by Erlang distributions. This paper focuses on solving queuing systems with generalised Erlang service distributions and exponential arrival distributions. We present the quasi-birth-death (QBD) process, which characterises the queuing models with generalised Erlang service and exponential interarrival distributions. We then use the properties of Erlang distributions and characteristics of channel allocation process of cellular networks to simplify the queues used to model cellular networks. The use of these simplifications provide a significant reduction in computation time required to solve these QBDs.

Keywords — cellular networks, phase-type distributions, generalised Erlang.

1. Introduction

One of the major problems that needs to be addressed in mobile communication networks is the continuity of a service during a handover without any data loss, as the user moves from cell to cell. This is called seamless handover [1]. The blocking probability encountered at handover is an important grade of service parameter for mobile users. It is of utmost importance to carefully dimension the network to provide the guaranteed GoS levels.

The channel holding time of a mobile user is an important parameter in the analysis of communication networks. It was shown in [2] that channel holding time in cellular networks can be accurately modelled as a generalised Erlang phase-type distribution. However the resulting queuing models are not tractable using common matrix manipulation techniques. In this paper we propose to use a simplification technique based on the properties of Erlang distributions and characteristics of channel allocation procedures in cellular networks to create a tractable queuing model for cellular networks.

Rest of the paper is organised as follows. Section 2 briefly describes the proposed QBD processes resulting from the queuing models associated with the cellular network. This work has been presented in [2] and has been included here for completeness. The proposed simplification techniques are presented in the following section. Matrix equations that describe the stationary probabilities of the system are presented in Section 3. Sections 4 and 6 describe the techniques that were used to solve for the blocking probabilities of the system. We conclude the paper staying that the proposed simplification leads to the creation of standard matrix equation from a seemingly un-tractable queuing system. Due to space limitations we only intend to describe the simplification techniques in this paper. Readers who are interested in final results from the study should refer to [2, 3].

2. Queuing model for cellular network channels

A phase-type (PH) distribution of generalised Erlang form with 2 phases has been proposed to model channel holding times in cellular networks [2]. Phase-type distributions can be used to approximate virtually any renewal process, with the dimensionality of the phase-type distribution increasing with the complexity of the particular process being modelled [2, 3]. Furthermore they provide an accurate description of the channel holding time distribution in cellular networks, while retaining the underlying Markovian properties of the distribution. These Markovian properties are essential in generating tractable queuing models for cellular networks. The parameters of this distribution can be estimated from experimental data by using the expectation maximisation (EM) algorithm [2].

References [2] and [3] describe methods used to derive the channel holding time distribution for cellular networks through network simulation models. Use of EM algorithm to approximate the actual distribution with a generalised Erlang distribution with 2 phases is presented in [3] and [4]. Arrivals of new and handover users are modelled with exponential distributions with appropriate parameters.

Assuming an exponential interarrival distribution and a phase-type service distribution, a cellular network with n channels per cell can be modelled as an $M/PH/n/n$ queue [5]. The resulting queuing system may not be tractable even for moderate values of n. Equation (1) shows the rate transition matrix or $\mathbf{Q}$ matrix for an $M/PH/n/n$

queue, which models a cell with n channels. In Eq. (1) it can be observed that $\mathbf{Q}$ is of block tri-diagonal form [5]:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_1 & 0 & 0 & \cdots & & & \\ \mathbf{B}_1 & \mathbf{A}_{11} & \mathbf{A}_{01} & 0 & \cdots & & & \\ 0 & \mathbf{A}_{22} & \mathbf{A}_{12} & \mathbf{A}_{02} & 0 & & & \\ \vdots & \ddots & \ddots & \ddots & & & & \\ \vdots & \vdots & \mathbf{A}_{2i} & \mathbf{A}_{1i} & \mathbf{A}_{0i} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & 0 & \mathbf{A}_{2,n-1} & \mathbf{A}_{1,n-1} & \mathbf{A}_{0,n-1} & \\ & & & 0 & 0 & \mathbf{A}_{2,n} & \mathbf{A}_{1,n} & \end{bmatrix} \quad (1)$$

Assuming an exponential arrival process with the rate λ the matrices $\mathbf{B}_0$, $\mathbf{B}_1$ and $\mathbf{B}_2$ are defined as [2, 5]:

$$\mathbf{B}_0 = \lambda\tau, \quad (2)$$

$$\mathbf{B}_1 = \lambda, \quad (3)$$

$$\mathbf{B}_2 = \mathbf{t}. \quad (4)$$

Where τ is the initial probability vector of the PH service distribution and $\mathbf{t}$ is a column vector partition of the rate transition matrix $\mathbf{Q}_{PH}$ of the PH distribution, given in Eq. (5):

$$\mathbf{Q}_{PH} = \begin{bmatrix} 0 & 0 \\ \mathbf{t} & \mathbf{T} \end{bmatrix}. \quad (5)$$

The construction of matrices $\mathbf{A}_{0i}$, $\mathbf{A}_{1i}$ and $\mathbf{A}_{2i}$ for a general phase type distribution is rather complex and requires lengthy and tedious computations. However, we have identified some properties of generalised Erlang distributions which simplify these calculations to a great extent. The next section of this paper describes the proposed simplifications and algorithms leading to the construction of tractable $\mathbf{Q}$ matrix for the queuing process. We assume the service distribution given by the following equation throughout the rest of the paper:

$$\mathbf{T} = \begin{bmatrix} -\mu_1 & \mu_1 \\ 0 & -\mu_2 \end{bmatrix}. \quad (6)$$

Row sums for any rate transition matrix are zero [5], resulting in,

$$\mathbf{t} = \begin{bmatrix} 0 \\ \mu_2 \end{bmatrix}, \quad (7)$$

where $\mathbf{T}$ and $\mathbf{t}$ are the partition matrices of $\mathbf{Q}_{PH}$ as describes in Eq. (5).

3. $M/PH/n/n$ server with a generalised Erlang service distribution

In the previous section we selected the $M/PH/n/n$ queue to model a cell with n channels. A drawback in using the above model to represent mobile network cells is that the size of the rate transition matrix depends on the number of channels available in a cell. As the size of the block matrix at level i is on order $\binom{i+2}{i}$ for a service distribution of 2 phases [6], the $\mathbf{Q}$ matrix becomes unmanageable for networks with large number of channels available per cell. Future mobile networks intend to provide a large number of channels per cell to support the high data rate services that will be available. Therefore the methods available in [5] cannot be used to find the blocking probabilities experienced by new and handover users in the mobile network environment.

A simplification can be made to the $\mathbf{Q}$ matrix by observing some properties of the Erlang distribution and the behaviour of servers at mobile cells. In servers with Erlang distributions, the users always start the service in the first phase and move on to the next phase with probability 1 once the sojourn in that phase is over. Users who finish the sojourn in the last phase depart the system. When there are m users in the system it is irrelevant which of these m users are at which server and similarly who finishes service first. This leads us to combine all the users in the same service phase to a single server with the service rate equivalent to the combined rate of all the servers. These simplifications allow to represent the system with a reduced state space. The new state space can be defined as follows:

$$\{\text{number of users is phase 1 } (n_1), \\ \text{number of users in phase 2 } (n_2)\}.$$

Using this simplification and a service distribution of two phases, the number of different states possible in the system when there are m users in the system are given in Table 1.

Table 1
Allowed states when there are m users in the system

State	Number of users in phase 1	Number of users in phase 2
1	m	0
2	$m-1$	1
$\vdots$	$\vdots$	$\vdots$
M	1	$m-1$
$m+1$	0	m

This reduces the size of the matrix at level m to $m+1$. The whole system can be arranged into two-dimensional continuous time Markov chain with the following state space:

0	no users in the system
(1,0)	1 user in the system
(0,1)	1 user in the system
⋮	⋮
(n_1, n_2)	$n_1 + n_2$ users in the system with n_1 users in phase 1 and n_2 users in phase 2
⋮	⋮

The events, which change the state of the system and rates of leaving the current state at those events are shown in Table 2. Figure 1 shows the transitions listed in Table 2. The first few states illustrating the construction of the rate transition matrix, **Q**, are given in Fig. 2. The rate transition matrix for the $M/PH/n/n$ QBD process can be constructed by observing the transitions between states given in Figs. 1 and 2, and using Table 2 to get the transition rates between different states. In order to obtain the characteristic tri-diagonal form of the **Q** matrix, it is necessary to perform a linear ordering of the states. In this case it is the simple ordering $\{0,(1,0),(0,1),(2,0),(1,1),(0,2),\ldots,(n,0),(n-1,1),\ldots,(1,n-1),(0,n)\}$. We can define levels where level m is the combination of all states when the number of users in the system are m. This ordering allows to generate the **Q** matrix of the form given in Eq. (1).

Table 2
Transition rates between different states

From	To	Rate	Event	Range
0	1,0	λ	Arrival	
(n_1,n_2)	(n_1+1,n_2)	λ	Arrival	$n_1 \geq 1$
(n_1,n_2)	(n_1,n_2-1)	$n_2\mu_2$	Departure	$n_2 \geq 1$
(n_1,n_2)	(n_1-1,n_2+1)	$n_1\mu_1$	Phase change	$n_1,n_2 \geq 1$

Elements of **Q** correspond to the transition rates for all the allowed transitions in the queuing system. A rate of zero means that the particular transition is not allowed in the system. **Q** can be divided into several row levels. Row level i corresponds to the matrices which describe the system when there are i users in the system. Matrices $\mathbf{A}_{0i}$, $\mathbf{A}_{1i}$ and $\mathbf{A}_{2i}$ make up the ith row level of **Q** given in Eq. (1). Similarly the columns can also be divided into different levels resulting column levels $(i-1)$, i and $(i+1)$ for matrices $\mathbf{A}_{0i}$, $\mathbf{A}_{1i}$ and $\mathbf{A}_{2i}$ respectively. Therefore matrix $\mathbf{A}_{2i}$ corresponds to the transitions which result in the number of users in the system being decreased from i to $(i-1)$, namely departures. Similarly the matrix $\mathbf{A}_{1i}$ represents the transitions which do not change the total number of users in the system, which corresponds to phase changes and self transitions in the system. Finally the matrix $\mathbf{A}_{0i}$ corresponds to arrivals which increase the number of users from i to $(i+1)$.

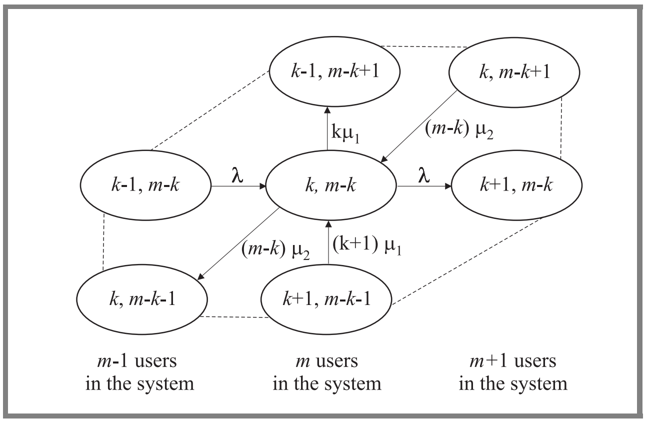

Fig. 1. State transitions for $M/PH/n/n$ QBD process.

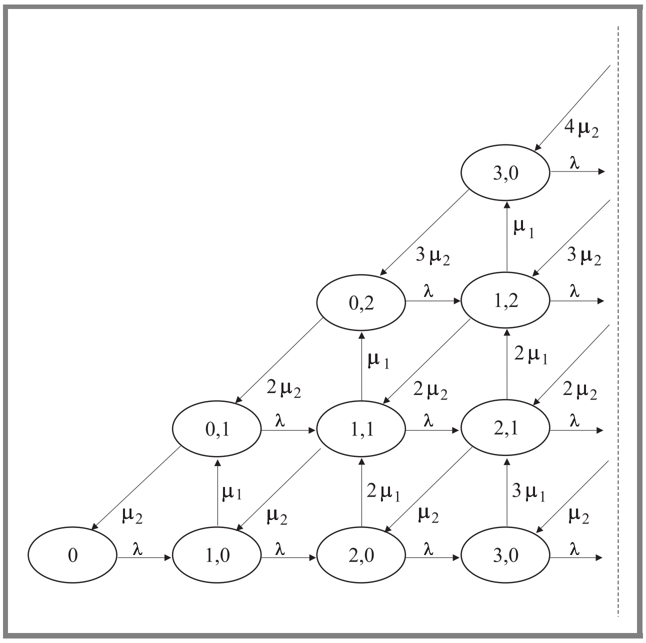

Fig. 2. First few states of the $M/PH/n/n$ QBD process.

3.1. Creating matrix $\mathbf{A}_{2i}$

It was explained earlier that the **Q** matrix represents all the allowed transitions in the system. Furthermore, as matrix $\mathbf{A}_{2i}$ is a sub matrix of **Q**, matrix $\mathbf{A}_{2i}$ corresponds to the all the departures from the system with i users. Assume that the number of users in phases 1 and 2 are given by n_1 and n_2 respectively. Then the allowed transitions in matrix $\mathbf{A}_{2i}$ are from states of the form (n_1, n_2) to states of the form (n_1, n_2-1). In other words, the transitions that leave the number of users in phase 1 unchanged while decrease the number of users in phase 2 by 1. The transition rate for these transitions are the combined rate of all n_2 users in phase 2, which results in $n_2\mu_2$ as the appropriate transition rate. To construct matrix $\mathbf{A}_{2i}$ it is necessary to identify all the allowed departures from the system with i users and then calculate the effective rate for these departures. The following algorithm is used to achieve this by identifying all the departures from the system with i users. Firstly we define matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ as follows:

$$\mathbf{S}_1 = \begin{bmatrix} i & 0 \\ i-1 & 1 \\ i-2 & 2 \\ \vdots & \vdots \\ 1 & i-1 \\ 0 & i \end{bmatrix} \quad \mathbf{S}_2 = \begin{bmatrix} i-1 & 0 \\ i-2 & 1 \\ \vdots & \vdots \\ 1 & i-2 \\ 0 & i-1 \end{bmatrix}$$

Matrix $\mathbf{S}_i$ contains the linearly ordered set of all possible states with i users in the system, while $\mathbf{S}_2$ contains the state space for system with $i-1$ users. In matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ the first and second columns correspond to the number of users in phase 1 and phase 2 respectively. The two state spaces $\mathbf{S}_1$ and $\mathbf{S}_2$ correspond to the state spaces along rows and columns of $\mathbf{A}_{2i}$ respectively. For example, the transition resulted by a move from first row of $\mathbf{S}_1$, $\{i,0\}$, to first row of $\mathbf{S}_2$, $\{i-1,0\}$, corresponds to the element $\{1,1\}$ of matrix $\mathbf{A}_{2i}$. We use the variable *row* to represent rows of matrix $\mathbf{A}_{2i}$ and variable *column* to represent columns of $\mathbf{A}_{2i}$. The algorithm then loops through the elements of $\mathbf{S}_1$ and $\mathbf{S}_2$ (effectively compares the transitions formed by moving from a state in $\mathbf{S}_1$ to a state in $\mathbf{S}_2$) and identify which transitions are allowed in the system. For allowed transitions the transition rate is created and it is entered into the position $\{row,\ column\}$ of $\mathbf{A}_{2i}$. All other transitions are given a value of zero to represent that they are not allowed in this system. The algorithm described is given below:

For *row* $= 1$ to $i+1$
    For *column* $= 1$ to i
        If $\mathbf{S}_2\ (column, 1) == \mathbf{S}_1(row, 1)$
        AND $\mathbf{S}_2(column, 2) == \mathbf{S}_1(row, 2) - 1$
            $\mathbf{A}_{2i}(row, column) = \mathbf{S}_1(column, 2)\mu_2$
        Else
            $\mathbf{A}_{2i}(row, column) = 0.0$
        End If
    End *column* loop
End *row* loop

Using the above algorithm $\mathbf{A}_{2i}$ (of size $(i+1)\times i$) is calculated and presented in Eq. (8):

$$\mathbf{A}_{2i} = \begin{bmatrix} 0 & \cdots & \cdots & \vdots \\ \mu_2 & & & \vdots \\ \vdots & 2\mu_2 & & \vdots \\ \vdots & & \ddots & \vdots \\ \vdots & \cdots & \cdots & i\mu_2 \end{bmatrix} \tag{8}$$

3.2. Creating matrix $\mathbf{A}_{0i}$

The concept used to derive the algorithm described in the previous section can also be used to derive an algorithm to calculate $\mathbf{A}_{0i}$. Matrix $\mathbf{A}_{0i}$ represents the arrivals into a system with i users. Using the previous notations the allowed transitions in this matrix are from states of the form (n_1, n_2) to states of the form (n_1+1, n_2). Alternatively, the transitions that increase the number of users in phase 1 by 1, while leaving the number of users in phase 2 unchanged. As the arrival rate into the system is independent of number of users in the system, all transitions have rate λ.
We define matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ as follows:

$$\mathbf{S}_1 = \begin{bmatrix} i & 0 \\ i-1 & 1 \\ \vdots & \vdots \\ 1 & i-1 \\ 0 & i \end{bmatrix} \quad \mathbf{S}_2 = \begin{bmatrix} i+1 & 0 \\ i & 1 \\ i-1 & 2 \\ \vdots & \vdots \\ 1 & i \\ 0 & i+1 \end{bmatrix}$$

The notation used here is the same as explained earlier in Section 3.1. Matrix $\mathbf{S}_1$ contains the linearly ordered set of all possible states with i users in the system while $\mathbf{S}_2$ contains the state space for system with $i+1$ users. Then the following algorithm can be used to identify the allowed transitions in $\mathbf{A}_{0i}$. As explained in Section 3.1 this algorithm loops through all the possible transitions in the system by traversing through the state spaces of $\mathbf{S}_1$ and $\mathbf{S}_2$ and finds the transitions allowed for this particular queuing model:

For *row* $= 1$ to $i+1$
    For *column* $= 1$ to $i+2$
        If $\mathbf{S}_2\ (column, 1) == \mathbf{S}_1(row, 1) + 1$
        AND $\mathbf{S}_2(column, 2) == \mathbf{S}_1(row, 2)$
            $\mathbf{A}_{0i}(row, column) = \lambda$
        Else
            $\mathbf{A}_{0i}(row, column) = 0.0$
        End If
    End *column* loop
End *row* loop

Using this algorithm matrix $\mathbf{A}_{0i}$ (of size $(i+1)\times(i+2)$) can be given as follows:

$$\mathbf{A}_{0i} = \begin{bmatrix} \lambda & 0 & \cdots & \vdots \\ 0 & \ddots & & \vdots \\ \vdots & 0 & \lambda & 0 \end{bmatrix} \tag{9}$$

3.3. Creating matrix $\mathbf{A}_{1i}$

Matrix $\mathbf{A}_{1i}$ represents all the transitions which do not change the number of users, i, in the system, namely the phase changes and the self-transitions. Using the previous notations, the allowed transitions in this matrix are from states (n_1, n_2) to $(n_1 - 1, n_2 + 1)$ (phase changes) and from (n_1, n_2) to (n_1, n_2) (self-transitions). The rate for the phase transitions are given by the combined rate of $n_1$users in the first phase before the transitions. The self-transition rates can be found easily by observing that all the row sums are zero for any **Q** matrix [5].

The following algorithm can be used to create all the elements except the diagonal elements of $\mathbf{A}_{1i}$. The diagonal elements are initially given zero in this algorithm. Then the correct diagonal elements become the negative of row sums of the **Q** matrix as the row sums are zero for any rate transition matrix. We define matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ as follows to obtain all the elements except the diagonal elements of $\mathbf{A}_{1i}$:

$$\mathbf{S}_1 = \mathbf{S}_2 = \begin{bmatrix} i & 0 \\ i-1 & 1 \\ \vdots & \vdots \\ 1 & i-1 \\ 0 & i \end{bmatrix}$$

Matrix $\mathbf{S}_1$ and $\mathbf{S}_2$ contain the linearly ordered set of all possible states with i users in the system. Then we can use the following procedure to derive $\mathbf{A}_{1i}$:

For *row* $= 1$ to $i+1$
    For *column* $= 1$ to $i+1$
        If $\mathbf{S}_2$ $(column, 1) == \mathbf{S}_1(row, 1) - 1$
        AND $\mathbf{S}_2(column, 2) == \mathbf{S}_1(row, 2) + 1$
            $\mathbf{A}_{1i}(row, column) = \mathbf{S}_1(row, 1)\mu_1$
        Else
            $\mathbf{A}_{1i}(row, column) = 0.0$
        End If
    End *column* loop
End *row* loop

The diagonal elements are given by:

$$-\left(\sum_{row} \mathbf{A}_{0i} + \sum_{row} \mathbf{A}_{1i} + \sum_{row} \mathbf{A}_{2i}\right),$$

where $\sum_{row}$ represents the row sum of the particular matrix. $\mathbf{A}_{1i}$ (of size $i+1) \times (i+1)$) is given in Eq. (10):

$$\mathbf{A}_{2i} = \begin{bmatrix} -\zeta_0 & i\mu_1 & 0 & \cdots & & \cdots & \vdots \\ 0 & -\zeta_1 & (i-1)\mu_1 & & & & \\ \vdots & 0 & \ddots & \ddots & & & \\ & & & -\zeta_k & (i-k)\mu_1 & & \vdots \\ & & & & \ddots & \ddots & 0 \\ & & & & & -\zeta_{i-1} & \mu_1 \\ \vdots & \cdots & & & & & \zeta_i \end{bmatrix} \tag{10}$$

where for $k = 0, 1, \ldots, i$

$$\zeta_k = \lambda + (i-k)\,\mu_1 + k\mu_2\,. \tag{11}$$

However, due to the extra boundary condition present when the number of users in the system is equal to the number of servers, ζ_k's for $k = 0, 1, \ldots, n$ in the matrix $\mathbf{A}_{1n}$ have to be modified as follows:

$$\zeta_k = (i-k)\,\mu_1 + k\mu_2\,. \tag{12}$$

4. Stationary probabilities of the system

Sections 3.1–3.3 explained how to create all the sub matrices required to generate the rate transition matrix representing the $M/PH/n/n$ QBD process with generalised Erlang service distribution. The objective of this study is to analyse the performance of a cellular network, focusing on the blocking probabilities for handover and new users as the main performance analysis parameter. To calculate the blocking probabilities, it is necessary to find the stationary probabilities of the system. In other words the probability of having i users in the system, with i ranging from 0 to n, need to be found. The phase-type service distribution introduces $i+1$ *substates* within each of these i states. In this section we will present methods to calculate the stationary probabilities corresponding to all the sub states and then show how we can combine these sub state stationary probabilities to calculate the probability of finding i users in the system at any instance.

The relationship between the stationary or equilibrium probabilities and the rate transition matrix for a time-homogeneous continuous time Markov chain (CTMC) is given by Eq. (13). It has been shown in [5] that Eq. (13) is valid for a wide variety of systems involving phase-type distributions. The stationary distribution vector, $\mathbf{x} = [x_1, x_2, \ldots, x_{S_{n+1}}]$ of the $M/PH/n/n$ queuing system satisfies the following equations:

$$\mathbf{xQ} = 0, \qquad x_i \geq 0, \quad \sum_{i=1}^{S_{n+1}} x_i = 1\,, \tag{13}$$

$$\mathbf{xP} = \mathbf{x}, \qquad x_i \geq 0, \quad \sum_{i=1}^{S_{n+1}} x_i = 1\,. \tag{14}$$

Where $\mathbf{P} = \mathbf{Q} + \mathbf{I}$ is a stochastic matrix and **I** is an identity matrix. The blocking probability of the system (i.e., the probability that a new user joining the system finds all the channels occupied) is given by the following equation[1]:

$$P_{block} = \sum_{j=S_n+1}^{S_{n+1}} x_j\,. \tag{15}$$

[1]State space of a queue with a single-phase service distribution can be expressed as the number of users at service. With a 2 phase generalised Erlang service distribution the state space of the resulting QBD process is 2 dimensional and can be expressed as {number of users in service phase 1, number of users in service phase 2}. Therefore the probability of having j users in the system is calculated by adding all the sub states such that,

number of users in phase 1 + number of users in phase 2 = j.

For small values of n, Eq. (14) can be solved by finding an eigenvector of $\mathbf{P}^T$ associated with unit eigenvalue and then normalising it such that the sum of the entries of the normalised eigenvector equals one. This method becomes computationally very expensive even for moderate values of n.

5. Stochastic complementation

Stochastic complementation provides a computationally cheaper mechanism to solve Eq. (14) by decoupling an irreducible large Markov chain into smaller irreducible Markov chains [7, 8]. Reference [8] states that an irreducible large Markov chain $\mathbf{P}$ with m states can be uncoupled into k smaller chains, say $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_k$, containing $r_1, r_2, \ldots, r_k$ states, where $\sum_{i=1}^{k} r_i = m$. It further states that it is possible to determine the solutions for these smaller chains, $\mathbf{s}_i s$, completely independent of each other. Therefore instead of solving the larger matrix a number of smaller matrices can be solved. Solutions of these smaller chains can then be combined appropriately to generate the solution for the larger system.

A divide-and-conquer approach can be used to systematically simplify the system until the individual matrices become small enough to solve directly. The $S_{n+1} \times S_{n+1}$ matrix $\mathbf{P}$ can be partitioned roughly in half as given in Eq. (16):

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \qquad (16)$$

From this partition two *stochastic complements* of $\mathbf{P}$, $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$ can be derived as follows:

$$\mathbf{S}_{11} = \mathbf{P}_{11} + \mathbf{P}_{12}(\mathbf{I} - \mathbf{P}_{22})^{-1}\mathbf{P}_{21}\,, \qquad (17)$$

$$\mathbf{S}_{21} = \mathbf{P}_{22} + \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{P}_{21}\,. \qquad (18)$$

These are each irreducible stochastic matrices of order approximately $S_{n+1}/2$, and the combination of $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$ is equivalent to the irreducible Markov chain $\mathbf{P}$ [8]. The most time consuming operation in calculating $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$ is the inversion of matrices of size $S_{n+1}/2$, which is far simpler than solving for eigen vectors of a square matrix of size S_{n+1}.

If $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$ are small enough to be solved directly (through eigen vectors) then the solutions of $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$, $\mathbf{s}_1, \mathbf{s}_2$, can be combined to generate the solution for $\mathbf{P}$, in terms of the *coupling factors*, $\varepsilon_1 \varepsilon_2$ between the two matrices [8]. Where

$$\varepsilon_1 = \frac{\mathbf{s}_2\mathbf{P}_{21}\mathbf{e}}{\mathbf{s}_1\mathbf{P}_{12}\mathbf{e} + \mathbf{s}_2\mathbf{P}_{21}\mathbf{e}}\,, \qquad (19)$$

$$\varepsilon_2 = \frac{\mathbf{s}_1\mathbf{P}_{12}\mathbf{e}}{\mathbf{s}_1\mathbf{P}_{12}\mathbf{e} + \mathbf{s}_2\mathbf{P}_{21}\mathbf{e}}\,. \qquad (20)$$

Then the solution for $\mathbf{P}$, x, is given by

$$\mathbf{x} = \begin{bmatrix} \varepsilon_1\mathbf{s}_1 & \varepsilon_2\mathbf{s}_2 \end{bmatrix}\,. \qquad (21)$$

Where $\mathbf{s}_1$ and $\mathbf{s}_2$ are the stationary distribution vectors for $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$ respectively, and $\mathbf{e}$ is a column vector of ones. If the matrices $\mathbf{S}_{11}$ and $\mathbf{S}_{22}$ are too large to solve for $\mathbf{s}_1$ and $\mathbf{s}_2$ directly, they are roughly partitioned in half again to get four stochastic complements, $(\mathbf{S}_{11})_{11}, (\mathbf{S}_{11})_{22}, (\mathbf{S}_{22})_{11} (\mathbf{S}_{22})_{22}$. Size of these stochastic complements is approximately $s_{n+1}/4$. This process can be continued until the resulting stochastic complements are small enough to be solved using a direct method.

Once the stochastic complements are solved using a direct method coupling factors for that level can be found using Eqs. (19) and (20). Then the coupling factors and solution for that level are combined to get the solutions for the matrices at the level above using Eq. (21). This combination process at each level continues until the solution for $\mathbf{P}$ is obtained.

Once the stationary probabilities of the system, $\mathbf{x}$, have been obtained such that $\mathbf{x}.\mathbf{e} = 1$, where $\mathbf{e}$ is a column vector of 1's, we can find the probability of having i users in the system $p(i)$ as follows:

$$p(0) = x(1)$$

$$p(i) = \sum_{j=S_i+1}^{S_i+1+i} x(i) \quad \text{for} \quad i \geq 1\,. \qquad (22)$$

If the system does not distinguish between new and handover users no priority will be given to one class over the other. In such a system the blocking probability for any user, P_{block}, will be given by

$$P_{\text{block}} = p(n) \qquad (23)$$

and the average load of the system, L, at this blocking probability is given by:

$$L = \sum_{i=1}^{n} i p(i)\,. \qquad (24)$$

6. Conclusions

Accurate methods have been derived to model cellular networks in recent times [9, 10]. However due to their complexity, these model do not result in tractable queuing systems. In [2] we proposed to model the channel holding time in cellular networks with a 2 phase generalised Erlang distribution. We showed that this distribution accurately approximates the distribution of channel holding time in a cellular network. In this paper we used some properties of the generalised Erlang distribution to derive a simplified queuing system to model the collective channels of a cell in cellular networks. These simplifications enabled us to derive a tractable queuing model from a seemingly untractable QBD. We then presented the methods used to calculate the blocking probabilities for handover and new users. Due to space limitations of the paper, we only presented the simplification techniques we used in this study that can be applied to similar problems. Interested readers should refer to [2] and [3] for final results.

References

[1] E. D. Re, R. Fantacci, and G. Giamabene, "Handover and dynamic channel allocation techniques in mobile cellular networks", *IEEE Trans. Veh. Techn.*, vol. 44, no. 2, pp. 397–405, 1995.

[2] A. Jayasuriya, J. Asenstorfer, and D. Green, "Modelling service time distributions in cellular networks using phase-type service distributions", in *Proc. Int. Conf. Commun.*, Helsinki, Finland, June 2001, vol. 2, pp. 440–444.

[3] A. Jayasuriya, "Improved handover performance through mobility predictions". Ph.D. thesis, University of South Australia, 2001; URL – http://www.itr.unisa.edu.au/∼aruna/papers/thesis.pdf

[4] S. Asmussen, O. Nerman, and M. Olsson, "Fitting phase-type distributions via the EM algorithm", *Scand. J. Stat.*, vol. 23, pp. 419–441, 1996.

[5] M. Neuts, *Matrix-Geometric Solutions in Stochastic Models*. John Hopkins, 1981.

[6] V. Naoumov, U. Krieger, and D. Wagner, "Analysis of a multiserver delay-loss system with a general Markovian arrival process", in *Proc. 1st Int. Conf. Matrix-Analyt. Meth. (MAM) Stoch. Mod.*, Flint, USA, Aug. 1995, pp. 44–66.

[7] C. Meyer, "Stochastic complementation, uncoupling Markov chains and the theory of nearly reducible systems", *SIAM Rev.*, vol. 31, no. 2, pp. 240–272, 1989.

[8] A. Jayasuriya and K. Dogancay, "Stochastic complementation applied to the analysis of blocking probabilities in cellular networks", in *Proc. IASTED Int. Conf., Wirel. & Opt. Commun.*, Banff, Alberta, Canada, July 2002, pp. 607–610.

[9] P. Orlik and S. Rappaport, "A model for teletraffic performance and channel holding time characterization in wireless cellular communication with general session and dwell time distributions", *IEEE J. Selec. Areas Commun.*, vol. 16, pp. 788–803, 1998.

[10] Y. Fang and I. Chlamtac, "Teletraffic analysis and mobility modelling of PCS networks", *IEEE Trans. Commun.*, vol. 47, pp. 1062–1072, 1999.

Aruna Jayasuriya received his bachelor of engineering from University of Adelaide in 1997 and Ph.D. from University of South Australia in 2001. Currently he is research fellow with Institute for Telecommunication Research of University of South Australia. His research interests include traffic modelling in cellular and IP networks, mobility prediction mechanisms for cellular and wireless LAN networks, policy- based quality of service control and routing in ad hoc networks. He has published numerous papers in the above areas.
e-mail: Aruna.Jayasuriya@unisa.edu.au
Institute for Telecommunications Research
University of South Australia
Mawson Lakes, SA 5095, Australia