CEAS Technical Report No. 657

PRIORITIZED CHANNEL BORROWING WITHOUT LOCKING: A CHANNEL SHARING STRATEGY FOR CELLULAR COMMUNICATIONS

Hua Jiang and Stephen S. Rappaport
Department of Electrical Engineering
State University of New York
Stony Brook, N. Y. 11794-2350

Abstract

Channel Borrowing Without Locking (CBWL) is a family of channel assignment schemes for cellular communication systems. They allow real-time borrowing of channels from adjacent cells without the need for channel locking in co-channel cells. CBWL with cut-off priority for calls that arise in the cell is presented. This scheme discourages excessive channel lending and borrowing at high traffic load and promotes a more uniform grade of service throughout the service area. An analysis using macrostates and decomposition is devised to evaluate the performance of the scheme. The results are validated by simulation.

The research reported in this paper was supported in part by the U.S. National Science Foundation under Grant No. NCR 9025131 and in part by IST/SDIO under Contract No. N0001485-K0610 and Grant No. N0001491-J4063, administered by the U.S. Office of Naval Research.

1 Introduction

A family of new channel assignment and sharing methods for cellular communication systems has been presented in [1]. The methods are called *channel borrowing without locking* (CBWL). CBWL can be used to enhance traffic capacity of cellular communication systems and to accommodate spatially localized communication traffic overloads (or *"hot spots"*). Variations of the schemes can be considered— but for convenience of presentation and explanation we consider a basic hexagonal layout with base stations (wireless gateways) using omni-directional antennas nominally located at cell centers. The system has a total of C_T channels. With a cluster of size N, the C_T channels are divided into N groups with about $C = C_T/N$ channels in each group. As in fixed channel assignment (FCA), each gateway is assigned a group of channels which are reused at gateways of other cells that are sufficiently distant for the co-channel interference to be tolerable. If all channels of the gateway of a cell are occupied when a new call arrives, channel borrowing is employed according to certain rules.

Channel locking has been suggested in other channel borrowing strategies such as dynamic channel assignment (DCA) [2], [3] and hybrid channel assignment (HCA) [4] to limit co-channel interference. That is, gateways within the required minimum reuse distance from a gateway that borrows a channel cannot use the same channel at the same time. Because of the difficulty in maintaining the reuse distance at the minimum value when channel locking is used, DCA and HCA generally perform less satisfactorily than FCA under high communication traffic loads [2], [4].

In CBWL, a channel can be borrowed only from an *adjacent* gateway. The borrowed channels are used with reduced transmitted power such that the co-channel interference caused by the channel borrowing is no worse than that of non-borrowing scheme. Therefore, *channel locking* is not necessary in CBWL schemes. The borrowed channels can be accessed only in part of the cell. To determine whether a mobile station is in the region that can be served by a borrowed channel, each gateway transmits a signal with the same reduced power as that on a borrowed channel. The signal is called borrowed channel sensing signal (BCSS). If the BCSS is not above some suitable threshold at a mobile station, a borrowed channel cannot be used by the mobile station; otherwise, the mobile station will use a borrowed channel if all its gateway's channels are occupied and any its neighboring gateways has a channel available for lending. Thus, there are two types of new call originations—those that arise in parts of a cell in which a borrowed channel *can* be used if one is available, and those that arise in parts of a cell where borrowed channels *cannot* be used. We denote these as A-type calls and B-type calls, respectively.

Neighboring gateways are identified in the following manner. With respect to the given gateway, choose the first adjacent gateway. The position of the reference adjacent gateway can be arbitrary, but once chosen for a given gateway, all other gateways label their neighbors in a corresponding manner. The remaining five adjacent gateways are numbered sequentially proceeding clockwise from the first. The given gateway is labeled gateway 0. The C channels of a gateway are divided into seven distinct groups. The seven groups are numbered 0, 1, ..., 6. The channels of group 0 are reserved for exclusive use of the given gateway. Channels in each of the other six groups can be lent to neighbors. The *ith* neighbor can only borrow channels in the *ith* group. The number of channels in the *ith* group is denoted C_i, $i = 0, 1, \ldots, 6$. Thus $C = \sum_{k=0}^{6} C_k$. For convenience we consider a symmetrical arrangement with $C_1 = C_2 = \ldots = C_6 = l$. An example of the channel layout structure of CBWL is shown in Figure 1.

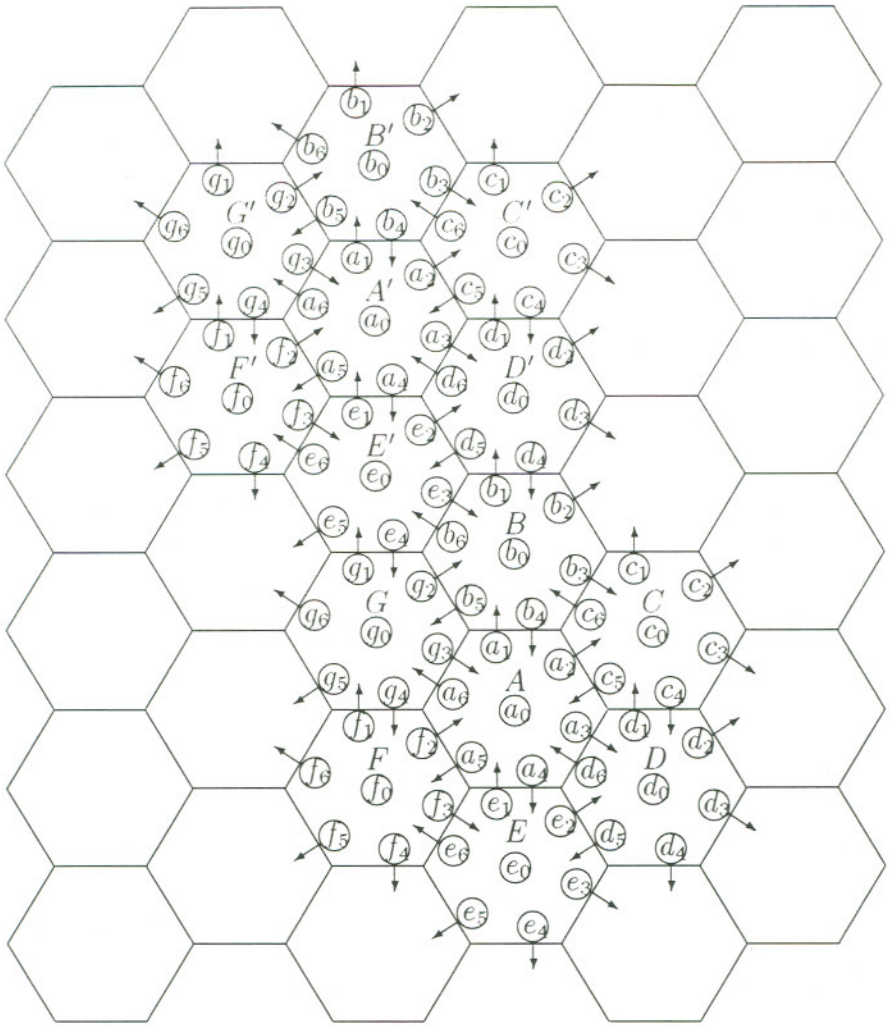

Figure 1: Channel structure of CBWL (cluster size =7).

CBWL with the structure described above has three advantages: 1) In the scheme, a gateway does not need to transmit and receive on all channels of its neighboring gateways. It only needs to access the channels that are assigned to it and the borrowed channels of six groups, one group from one neighbor. Therefore, the transmitter of a gateway only needs to access a total of $C + 6l$ channels instead of $7C$ channels. The cost and complexity of a gateway are reduced. 2) The scheme eliminates the possibility that two co-channel gateways lend the same channel simultaneously to a pair of closely located gateways (which would result in unacceptable co-channel interference). 3) With careful organization, the scheme can ensure that no *adjacent channels* are used in a given cell even though channel borrowing is employed.

As described in [1], *channel rearrangement* can be used in CBWL. With channel rearrangements, if a new B-type call arrival finds all channels of its gateway occupied, the call is *still not necessarily blocked.* If at the same time an A-type call in the cell is using a regular channel, and at least one neighbor can *lend channels to the given gateway*, the A-type call will use a borrowed channel from a neighbor and give its regular channel to the B-type call. In this way, calls that cannot use borrowed channels directly also benefit from the borrowing scheme. The number of calls that can use borrowed channels (directly or indirectly) is increased. For convenience, We call CBWL without channel rearrangement as CBWL/NR and CBWL with channel rearrangement as CBWL/CR.

In this paper we propose and study enhancements of CBWL by a cut-off priority structure that favors calls which arise in the gateway's own cell. It is noted that even *without* cut-off priority, some specific channels at each gateway may be reserved for use only by calls that arise in the gateway's own cell. It may not be unusual in these schemes for any given gateway *simultaneously* to borrow from and to lend channels to its neighbors—even to the same neighbor from which it has borrowed. This is increasingly likely as traffic loading increases. Thus there can be unnecessary borrowing. Since borrowed channel can only be accessed by some fraction of users in a cell while regular channels can be accessed by all users in a cell, unnecessary borrowing can limit the performance of the scheme at high traffic loading. One way to alleviate this problem is to use a cut-off priority structure in which gateways that have more than some number of channels ($m < C$) occupied, will not lend. Thus at high loading some channels will be available only for calls that arise in the cell. As a result, overall performance is improved. Additionally, cut-off priority promotes a more uniform grade of service throughout the cell because it tends to keep channels available for users who are more "distant" from the gateway and who therefore cannot use borrowed channels.

With cut-off priority, if a gateway X receives a channel borrowing request from a neighbor Y, the request *is* or *is not* granted depends on the current channel occupancy of X. X will deny the request,

1. if the total number of occupied channels of gateway X is more than m. Thus gateway X gives a cut-off priority of $C - m$ channels to the calls arising in its cell.

2. if the number of channels that are lent from X to Y is equal to l.

3. if the number of total channels of cell X that are lent to neighbors (including Y) is equal to n.

In Section 2, CBWL/NR and CBWL/CR with cut-off priority is modeled and analyzed. The numerical results from analysis and simulation are given in Section 3.

2 Traffic Analyses of CBWL with Cut-off Priority

For convenience of presentation, we limit our analysis here to the homogeneous case. That is, each gateway has the same number of assigned channels and the same offered traffic. With little modification the method can be extended to non-homogeneous traffic environments – including those with hot spots. For the homogeneous case, it is not necessary distinguish between different cells. An arbitrarily selected cell represents each cell.

To determine performance characteristics, we use a suitable state description of the system and seek solutions for the equilibrium state probabilities.

In this section, we first consider CBWL/NR. We show that no product form solution exists for CBWL/NR scheme with cut-off priority. However, a product form solution can be found for the numbers of channels that are lent to each neighbor. This product form solution is used to reduce the dimensions of vectors and the number of states that must be considered. An algorithm for computation of performance characteristics is devised. A similar analysis for CBWL/CR scheme with cut-off priority is also described.

The usual Markovian assumptions are invoked. New attempts in a cell arise at an average rate λ (new call arrivals per second per cell) according to a Poisson process and call holding times have a negative exponential probability distribution with mean $1/\mu$. In the homogeneous case calls originate uniformly throughout the service area. Let p denote the probability that a new call arrival arises in that part of a cell that can be served by a borrowed channel. To prevent the increase of co-channel interference caused by channel borrowing, the transmitted power on the borrowed channels is smaller than that on regular channels. Thus, usually p is small (a typical value is between 0.1–0.3). We note that channel borrowing requests that are directed to a given gateway from one of its neighbors arise from an overflow process (at the neighbor) Therefore these requests do not strictly conform to a Poisson process [6]. However at the neighbor (i.e., the source of borrowing requests), borrowing requests are randomly split into six parts, only one of which is directed to the given gateway. The random splitting tends to smooth the peakedness of the overflow traffic directed to a given gateway. We model the overflow traffic directed to a given gateway by a Poisson process with intensity λ'. The parameter, λ' will be determined. Our simulation results indicate that this assumption is valid.

2.1 Analysis of CBWL/NR with Cut-off Priority for Calls that Arise in The Cell

2.1.1 Characterization of the State of a Gateway

Let C be the number of channels that are allocated to a gateway. At any given time a gateway is in one of a finite number of states. A state is identified by a vector $\mathbf{I} = (i_0, i_1, i_2, i_3, i_4, i_5, i_6)$. The component i_0 is the number of channels occupied by calls that arise in the cell served by the gateway ($0 \leq i_0 \leq C$). The number of channels at the gateway that are (currently) lent to the kth neighbor is i_k, ($k = 1, 2, \ldots, 6$), where $0 \leq i_k \leq C_k$. Thus, in state $\mathbf{I}$, the total number of occupied channels of the gateway is given by

$$J(\mathbf{I}) \triangleq \sum_{k=0}^{6} i_k \ . \tag{1}$$

The total number of channels that are (currently) lent to all adjacent gateways is

$$L(\mathbf{I}) \triangleq \sum_{k=1}^{6} i_k \ . \tag{2}$$

With cut-off priority, no channels will be lent if the number of the given gateway's channels in use is greater than or equal to m, where $m < C$. The maximum number of channels that a gateway can lend at any given time is

$$L_{max} = \min(\sum_{k=1}^{6} C_k, m, n) \ . \tag{3}$$

For simplicity, we assume that $C_k = l$ for $k = 1, \ldots, 6$. The state variables must satisfy the constraints given in below:

$$\begin{aligned} 0 \leq\ & i_0\ \leq C \\ 0 \leq\ & i_k\ \leq l \qquad k = 1, 2, \ldots, 6 \\ 0 \leq & J(\mathbf{I}) \leq C \\ 0 \leq & L(\mathbf{I}) \leq L_{max} \end{aligned} \tag{4}$$

The set of permissible states are those whose state variables satisfy (4). The set is denote by Ω.

2.1.2 Flow balance equations

We define a function, $O(i_k)$, by

$$O(i_k) \triangleq \begin{cases} 1 & \text{if } 0 \leq i_k < l \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

According to the operation of the structured CBWL scheme, the kth adjacent gateway can possibly borrow a channel from the given gateway only if $O(i_k) = 1$. Furthermore, we define a unit state vector $\delta_{\mathbf{k}}$ as a vector in which $i_k = 1$ and all other elements are 0. Also, let $p(\mathbf{I})$ denote the equilibrium probability of state $\mathbf{I}$.

Define a function, $Z(\mathbf{I})$, such that

$$Z(\mathbf{I}) \triangleq \begin{cases} 1 & \text{if } \mathbf{I} \in \Omega \\ 0 & \text{if } \mathbf{I} \notin \Omega\ . \end{cases} \tag{6}$$

We write the flow balance equations that determine the equilibrium state probabilities of CBWL/NR with cut-off priority. In this case, the flow balance equations for any given state is in one of four possible forms. The forms are different if (a) $J(\mathbf{I}) < m$; (b) $J(\mathbf{I}) = m$; (c) $m < J(\mathbf{I}) < C$; and, (d) $J(\mathbf{I}) = C$.

(a) Flow balance equations for state $\mathbf{I}$, $J(\mathbf{I}) < m$:

The equations are

$$\begin{aligned} &[\lambda + \lambda' \sum_{k=1}^{6} O(i_k) + J(\mathbf{I})\mu] p(\mathbf{I}) = \lambda p(\mathbf{I} - \delta_{\mathbf{0}}) Z(\mathbf{I} - \delta_{\mathbf{0}}) \\ &\quad + \lambda' \sum_{k=1}^{6} O(i_k - 1) p(\mathbf{I} - \delta_{\mathbf{k}}) Z(\mathbf{I} - \delta_{\mathbf{k}}) + \sum_{k=0}^{6} (i_k + 1)\mu p(\mathbf{I} + \delta_{\mathbf{k}}) Z(\mathbf{I} + \delta_{\mathbf{k}}) \\ &\quad (\text{ for any permissible } \mathbf{I} \text{ with } J(\mathbf{I}) < m) \end{aligned} \tag{7a}$$

(b) Flow balance equations for state $\mathbf{I}$, $J(\mathbf{I}) = m$:

The flow balance equations of any permissible state $\mathbf{I}$ with $J(\mathbf{I}) = m$ are similar to (7a) except that no channels can be lent to other gateways. Since there is no probability flow out due to lending channels, the second term on the left side of (7a) is zero. The result is

$$[\lambda + J(\mathbf{I})\mu] = \lambda p(\mathbf{I} - \delta_{\mathbf{0}}) + \lambda' \sum_{k=1}^{6} O(i_k - 1) p(\mathbf{I} - \delta_{\mathbf{k}}) Z(\mathbf{I} - \delta_{\mathbf{k}})$$

$$+\sum_{k=0}^{6}(i_k+1)\mu p(\mathbf{I}+\delta_{\mathbf{k}})Z(\mathbf{I}+\delta_{\mathbf{k}}) \tag{7b}$$

(for any permissible $\mathbf{I}$ with $J(\mathbf{I})=m$).

(c) Flow balance equations for state $\mathbf{I}$, $m < J(\mathbf{I}) < C$:

For any permissible state $\mathbf{I}$ with $J(\mathbf{I}) > m$, the flow rate into $\mathbf{I}$ from the state $\mathbf{I}-\delta_{\mathbf{k}}$ ($k=1, \ldots, 6$) caused by borrowing traffic is zero. Thus,

$$[\lambda + J(\mathbf{I})\mu]p(\mathbf{I}) = \lambda p(\mathbf{I}-\delta_{\mathbf{0}})Z(\mathbf{I}-\delta_{\mathbf{0}}) + \sum_{k=0}^{6}(i_k+1)\mu p(\mathbf{I}+\delta_{\mathbf{k}})Z(\mathbf{I}+\delta_{\mathbf{k}}) \tag{7c}$$

(for any permissible $\mathbf{I}$ with $m < J(\mathbf{I}) < C$).

(d) Flow balance equations for state $\mathbf{I}$, $J(\mathbf{I}) = C$:

For any permissible state $\mathbf{I}$ with $J(\mathbf{I}) = C$, *flow out* arises only from call completions while *flow in* arises only from new call arrivals. Thus,

$$C\mu p(\mathbf{I}) = \lambda p(\mathbf{I}-\delta_{\mathbf{0}})Z(\mathbf{I}-\delta_{\mathbf{0}}) \tag{7d}$$

(for any permissible I with $J(\mathbf{I}) = C$).

Because state transitions from $\mathbf{I}$ to $\mathbf{I}+\delta_{\mathbf{k}}$ ($k=1, \ldots, 6$) are not allowed, on the left side of (7b) and (7c), the probability flow out of state $\mathbf{I}$ due to channel lending is zero. On the right side of (7c), the probability flow into state $\mathbf{I}$ from $\mathbf{I}+\delta_{\mathbf{k}}$ ($k=1, \ldots, 6$) due to channel completions is not zero. Therefore, if $m \leq J(\mathbf{I})$, for each pair of states, $\mathbf{I}$ and $\mathbf{I}+\delta_{\mathbf{k}}$ ($k=1, \ldots, 6$), only one-way transitions (from $\mathbf{I}+\delta_{\mathbf{k}}$ to $\mathbf{I}$ for $k=1, \ldots, 6$) are allowed. Thus, local balance equations are not satisfied for the pair of states. The sufficient conditions that permit a product form solution are not satisfied [7].

To get the necessary condition that permits product form solution, we assume that product form solutions exist, and substitute the solutions into (7). If (7) holds, the assumption is true, otherwise, the solutions are not of product form. We assume

$$p(\mathbf{I}) = \frac{1}{G(\Omega)}\frac{\left(\frac{\lambda}{\mu}\right)^{i_0}}{i_0!}\prod_{k=1}^{6}\frac{\left(\frac{\lambda'}{\mu}\right)^{i_k}}{i_k!}\ , \tag{8}$$

in which, the normalization constant $G(\Omega)$ is the sum of probabilities of all permissible states. The constant, $G(\Omega)$, is given by

$$G(\Omega) = \sum_{\mathbf{I}\in\Omega} p(\mathbf{I})\ . \tag{9}$$

After substitution of (8) into (7c), we got the necessary condition for a product form solution:

$$\frac{1}{G(\Omega)}\frac{(\frac{\lambda}{\mu})^{i_0}}{i_0!}\prod_{k=1}^{6}\frac{(\frac{\lambda'}{\mu})^{i_k}}{i_k!}[\lambda+\mu\sum_{k=0}^{6}i_k] = \frac{\lambda}{G(\Omega)}\frac{(\frac{\lambda}{\mu})^{i_0-1}}{(i_0-1)!}\prod_{k=1}^{6}\frac{(\frac{\lambda'}{\mu})^{i_k}}{i_k!}$$
$$+\frac{\mu}{G(\Omega)}\frac{(\frac{\lambda}{\mu})^{i_0}}{i_0!}6\left(\frac{\lambda'}{\mu}\right)\prod_{k=1}^{6}\frac{(\frac{\lambda'}{\mu})^{i_k}}{i_k!} + \frac{(i_0+1)\mu}{G(\Omega)}\frac{(\frac{\lambda}{\mu})^{i_0+1}}{(i_0+1)!}\prod_{k=1}^{6}\frac{(\frac{\lambda'}{\mu})^{i_k}}{i_k!}\ . \tag{10}$$

With some manipulation, this reduces to

$$6\lambda' = \mu \sum_{k=1}^{6} i_k \ . \tag{11}$$

But, given λ' and μ, (11) is not true for any state. Thus, we have shown that no product form solution exists in this case.

2.1.3 Macrostates

Although a product-form equilibrium state-distribution does not exist, we will find that some component variables of I do have product form solution, which is used to reduce the dimensions of vectors and the number of states that must be considered. The complexity of numerical computation is reduced.

We define a macrostate $\mathbf{I_m} = (u, v)$ as

$$\mathbf{I_m} = \{u = i_0, v = \sum_{k=1}^{6} i_k : \mathbf{I} \in \Omega\} \ . \tag{12}$$

The original states (each charactered by a 7-dimensional vector) are grouped into two-dimension macro states. Note that, u represents the number of channels that are occupied by the calls that arise in the given cell and v represents the number of channels that are lent to adjacent gateways. The two-dimensional state transition diagram for an example with $n = 8$ and $m = 6$ is shown in Figure 2.

To determine the probability flow balance equations for the macro-state description, we must determine the transition rates in (u, v).

The transition rates from any state for which $u > 0$, $v > 0$ and $u + v < m$ consist of four parts. Two of them are transition rates of u: that due to new call arrivals and that due to the completion of the calls in the given cell (the horizontal arrows in Figure 2.). The other two are transition rates of v: that due to channel borrowing demands from neighbors, and that due to the returning of loan channels (the vertical arrows in Figure 2). Since u and i_0 are identical, the transition rates from (u, v) to $(u+1, v)$ and $(u-1, v)$ is λ and $u\mu$ respectively. The transition rate of (u, v) to $(u, v-1)$ is the returning rate of borrowed channels. If v channels are lent, the rate is just $v\mu$. The transition rate from (u, v) to $(u, v+1)$ is channel lending rate, which is denoted as $\rho(v)$. The rate will be determined.

If a neighbor has borrowed less than l channels from the given gateway, the channel lending rate of the given gateway to this neighbor is λ'. If the neighbor has borrowed l channels from the given gateway, it can not borrow any more channels from that gateway. The channel lending rate of the given gateway to the neighbor becomes zero. Given v, there can be many different sequences of i_k's ($k = 1, \ldots, 6$) for which $\sum_{k=1}^{6} i_k = v$. Each sequence may have a different channel lending rate. Therefore, given v, we define $\rho(v)$ as an *average* channel lending rate.

To define $\rho(v)$ mathematically, we use a ν-vector

$$\mathbf{I}_\nu \triangleq (i_1, \ldots, i_\nu) \ , \qquad \nu = 1, 2, \ldots, 6, \tag{13}$$

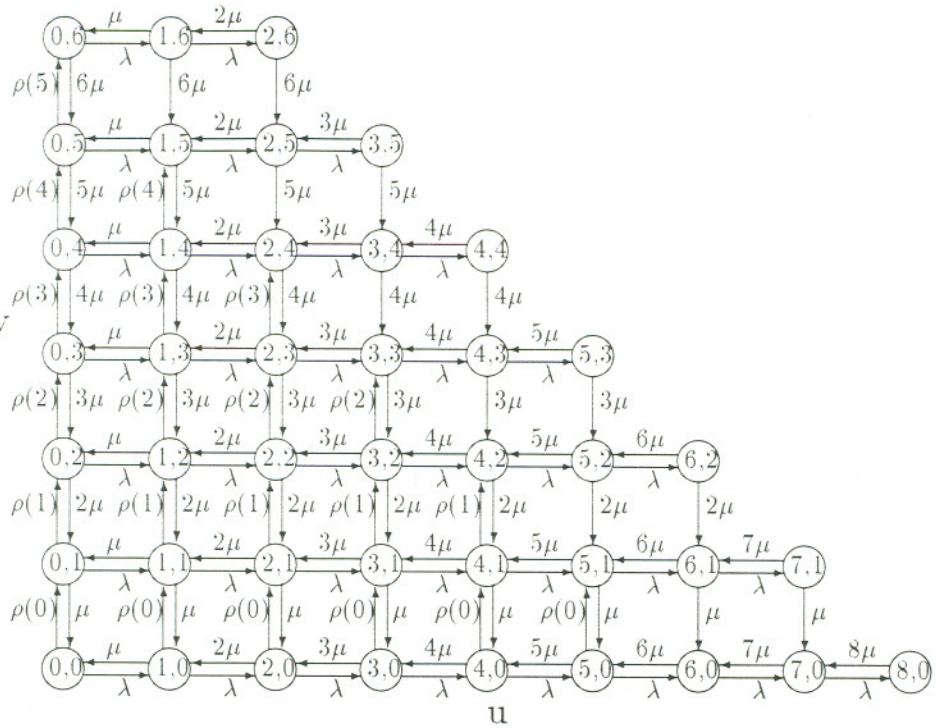

Figure 2: State-transition diagram of two-dimensional macro-state (u, v) for a CBWL scheme with $C = 8$ and $m = 6$.

in which, the components of the ν-vector, $\mathbf{I}_\nu$, represent the numbers of channels that are lent individually to the first through the νth adjacent gateways. Thus, we must have

$$\mathbf{I} = (i_0, \mathbf{I}_\nu, i_{\nu+1}, \ldots, i_6) \in \Omega \,, \qquad (\nu = 1, 2, \ldots, 6). \tag{14}$$

The six-vector $\mathbf{I_6}$ represents the numbers of channels that are lent to each neighbor. Then, we define $S(x, 6)$ as the set of $\mathbf{I_6}$ whose components sum to x. That is,

$$S(x, 6) \triangleq \{\mathbf{I_6} = (i_1, i_2, i_3, i_4, i_5, i_6) : (i_0, \mathbf{I_6}) \in \Omega, \sum_{k=1}^{6} i_k = x\} \,. \tag{15}$$

Now, we can define the average channel lending rate

$$\rho(v) \triangleq \frac{\sum\limits_{\mathbf{I_6} \in S(v,6)} \Pr(\mathbf{I_6}) \lambda' \sum\limits_{k=1}^{6} O(i_k)}{\sum\limits_{\mathbf{I_6} \in S(v,6)} \Pr(\mathbf{I_6})} \,. \qquad v = 0, \ldots, L_{max}. \tag{16}$$

The calculation of $\rho(v)$ is given in Appendix A.

2.1.4 Probability Flow Balance Equations (For the macro-state)

All permissible states of (u, v) must satisfy the following constraints,

$$0 \leq \ u \ \leq C$$

$$
\begin{aligned}
0 \leq \quad v \quad &\leq L_{max} \\
0 \leq u + v &\leq C \ .
\end{aligned}
\tag{17}
$$

which are transliterated from (4).

Denote Ψ as the space of (u, v) constrained by (17). Then we can define an indicating function

$$
z(u,v) = \begin{cases} 1 & \text{if } (u,v) \in \Psi \\ 0 & \text{if } (u,v) \notin \Psi \end{cases}
\tag{18}
$$

Define $p(u, v)$ as the equilibrium distribution of the state (u, v). The macro-state balance equations can be written as

$$
\begin{aligned}
&[\lambda + \rho(v) + u + v]p(u,v) = \lambda p(u-1,v)z(u-1,v) + \rho(v-1)p(u,v-1)z(u,v-1) \\
&\qquad +(u+1)\mu p(u+1,v)z(u+1,v) + (v+1)\mu p(u,v+1)z(u,v+1) \\
&\qquad\qquad (\text{for } 0 \leq u+v < m) \\
&(\lambda + m\mu)p(u,v) = \lambda p(u-1,v)z(u-1,v) + \rho(v-1)p(u,v-1)z(u,v-1) \\
&\qquad +(u+1)\mu p(u+1,v)z(u+1,v) + (v+1)\mu p(u,v+1)z(u,v+1) \\
&\qquad\qquad (\text{for } u+v = m) \\
&[\lambda + (u+v)\mu]\, p(u,v) = \lambda p(u-1,v)z(u-1,v) \\
&\qquad +(u+1)\mu p(u+1,v)z(u+1,v) + (v+1)\mu p(u,v+1)z(u,v+1) \\
&\qquad\qquad (\text{for } m < u+v < C) \\
&C\mu p(u,v) = \lambda p(u-1,v)z(u-1,v) \\
&\qquad\qquad (\text{for } u+v = C)
\end{aligned}
\tag{19}
$$

These macro-state balance equations correspond to equations (7) in the original state description. If $u + v \geq m$, only the transition from state $(u, v+1)$ to (u, v) is allowed. The transition from state (u, v) to state $(u, v+1)$ is prohibited (see Figure 2), and no local balance can be found in the pair of states. The two-dimensional model does not support product form solution. Gauss-Seidel iteration was used to solve for the two dimensional probabilities, [8]. The method starts with a guess of a group of arbitrary $p(u, v)$'s in (19) to compute a group of new $p(u, v)$'s. From the new $p(u, v)$'s, another iteration is started. The procedure is continued until the two groups of $p(u, v)$'s from two successive iterations agree to the desired precision.

2.1.5 Determination of Blocking Probabilities

Important performance measures can be expressed in terms of the state probabilities, $p(u, v)$'s. The probability that all channels are occupied can be calculated from

$$
p_c = \sum_{v=0}^{L_{max}} p(C-v, v) \ .
\tag{20}
$$

Denote the probability that a borrowing request from a specific adjacent gateway is denied by a given gateway as p_f. A borrowing request from a specific neighbor will be denied by the given gateway if any of the following three events is true at the time that the borrowing request arises.

Event E_1: The number of occupied channels of the given gateway is more than m.

Event E_2: The total number of channels that have been lent to all neighbors is equal to the maximum possible number, L_{max}.

Event E_3: The neighbor has already borrowed its maximum allowable channel quota, (l channels).

We denote the probability that a borrowing request from a specific adjacent gateway is denied by the given gateway as p_f, which is the probability of the union of the events. That is,

$$p_f = \Pr\{E_1 \bigcup E_2 \bigcup E_3\} = \Pr(E_1) + \Pr(E_2\bar{E_1}) + \Pr(E_3\bar{E_2}\bar{E_1}) \tag{21}$$

in which an overbar denotes the complementary event.

The probability of event E_1 is the probability that the given gateway in any state with $u + v \geq m$. Thus,

$$\Pr(E_1) = \sum_{v=0}^{L_{max}} \sum_{u=m-v}^{C-v} p(u, v) \ . \tag{22}$$

The event of $E_2\bar{E}_1$ includes all permissible states with $v = L_{max}$ and $u + v < m$. That is,

$$\Pr(E_2\bar{E}_1) = \sum_{u=0}^{m-1-L_{max}} p(u, L_{max}) \ . \tag{23}$$

The probability of $E_3\bar{E}_2\bar{E}_1$ can be found as follows. For the homogeneous case, if s neighbors have borrowed l channels, the chance that the given adjacent gateway is among the s gateways is $s/6$. Therefore, if the probability that s neighbors borrow l channels given that v (($l \leq v < L_{max}$) channels are lent is known, we have

$$\Pr(E_3\bar{E}_2\bar{E}_1) = \sum_{v=l}^{L_{max}-1} \sum_{s=1}^{6} \Pr(s \text{ gateway borrow } l \text{ channels} \mid v \text{ channels are lent}) \cdot \frac{s}{6}\Pr(v)\Pr(0 \leq u + v < m|v) \ . \tag{24}$$

From equation (A.2) in Appendix A, we know that the probability that s neighbors have borrowed l channels given that v channels are lent is $b(6-s, v)/b(v)$. The probability $\Pr(v)\Pr(0 \leq u + v < m|v)$ appearing in (24) is given by $\sum_{u=0}^{m-v-1} p(u, v)/\Pr(v)$. Thus,

$$\Pr(E_3\bar{E}_2\bar{E}_1) = \sum_{v=l}^{L_{max}-1} \sum_{s=1}^{6} \frac{b(6-s, v)}{b(v)} \frac{s}{6} \sum_{u=0}^{m-v-1} p(u, v) \ . \tag{25}$$

Then, from (21), we have

$$p_f = \sum_{v=0}^{L_{max}} \sum_{u=m-v}^{C-v} p(u, v) + \sum_{u=0}^{m-1-L_{max}} p(u, L_{max}) + \sum_{v=l}^{L_{max}-1} \sum_{s=1}^{6} \frac{b(6-s, v)}{b(v)} \frac{s}{6} \sum_{u=0}^{m-v-1} p(u, v) \ . \tag{26}$$

Average rate of borrowing requests from a neighbor λ'

In the CBWL schemes, requests that cannot be served on a channel of a given gateway may be served on a channel borrowed from a neighbor. Clearly borrowing requests from a specific neighbor will be more frequent when that neighbor's *own* C channels are more heavily used. Thus the states of adjacent gateways are coupled. In principle one could define an overall *system* state as a concatenation of gateway states, for all gateways in the system. However this approach is not fruitful because the number of states that must be considered is analytically unmanageable. We retreat from this more rigorous approach and instead account for the coupling between gateways by considering the *average* rate of borrowing and lending between gateways. We found that this approach permits the construction of an analytically tractable model. Theoretical performance characteristics were calculated and were then compared with those obtained by Monte Carlo simulation. The results compared favorably and are discussed in section 3.

By considering the coupling between adjacent gateways as the average rate of borrowing requests from adjacent gateways and including the rate into the model for state analysis of a given gateway, the state probabilities of the given gateway can be determined completely without the knowledge of the states of other gateways. This has been shown in our previous analysis [1]. Therefore, if λ', the average rate of borrowing request from an adjacent gateway has been determined, the needed coupling between adjacent gateways is accounted for.

The average rate of borrowing requests from a neighbor can be calculated as in CBWL/NR scheme without cut-off priority [1].

$$\lambda' = \sum_{k=1}^{6} \lambda'(k) = \frac{\lambda p p_c}{6} \sum_{k=0}^{5} p_f^k = \frac{\lambda p_c}{6} \frac{(1 - p_f^6)}{(1 - p_f)} \ . \tag{27}$$

Because p_c and p_f in (27) depend on λ, the equation is actually an implicit equation that can be used to obtain p_f, p_c and λ' simultaneously. An iterative procedure was used as outlined below:

The iterative procedure

Step 1 The procedure starts with an arbitrary guess of λ'.

Step 2 Use the last updated λ' in convolution algorithm to calculate $e_t(x)$ from (A.11) and then find $b(t, v)$ from (A.13).

Step 3 Find $\rho(v)$ from (A.8).

Step 4 Use $\rho(v)$ to construct balance equations (19). Then solve the equations to find $\Pr(u, v)$.

Step 5 Find p_c and p_f from (20) and (26).

Step 6 The average rate of borrowing requests λ' from a neighbor is updated using (27).

Step 7 Step 2–6 of the procedure is continued until the absolute value of the difference between λ''s from the two consecutive iterations agree with the desired number of significant figures.

The blocking probabilities

The blocking probability is different for calls in different positions of a cell. For a B-type call, the blocking probability β_{NR} is equal to p_c. An A-type call is blocked if all of the regular channel are occupied and no channel can be borrowed from any neighbor. Its blocking probability α_{NR} is given by

$$\alpha_{NR} = p_c p_f^6 \ . \tag{28}$$

The average blocking probability B_{NR} is

$$B_{NR} = p(\alpha_{NR}) + (1-p)(\beta_{NR}) = p_c[pp_f^6 + (1-p)] \ . \tag{29}$$

2.2 Analysis of CBWL/CR with Cut-off Priority for Calls That Arise in The Cell

The seven-dimensional state vector $\mathbf{I} = (i_0, i_1, i_2, i_3, i_4, i_5, i_6)$, which is used to analyze the CBWL/NR scheme cannot completely describe the CBWL/CR scheme. In CBWL/CR, A-type and B-type calls must be distinguished. Let i_a and i_b denote the numbers of a gateway's channels that are used by A-type and B-type calls respectively. With channel rearrangement, an A-type call will use a borrowed channel and its regular channel is released to be given to a new B-type call. Therefore, the preliminary condition of channel rearrangement is that $i_a > 0$. To determine whether a gateway can use channel rearrangement, the number of channels that are used by A-type calls should be included into state variables. Thus, in CBWL/CR, we use an eight-dimensional state vector,

$$\mathbf{I_r} = (i_a, i_b, i_1, i_2, i_3, i_4, i_5, i_6) \ , \tag{30}$$

to characterize the state of a gateway.

2.2.1 State Aggregation and Decomposition Method

Due to channel rearrangement, the distribution of $\mathbf{I_r}$ is not in product form. We can construct the probability flow balance equations of $\mathbf{I_r}$ and solve them to find the state probabilities. However, the number of states is prohibitively large (millions for a system with $C = 24$ and l=4). Instead, we will use a state aggregation and decomposition method and use the results from the analysis of CBWL/NR to expedite the calculation.

With channel rearrangement, an A-type call borrows a channel from a neighbor and give its regular channel to a new B-type call. Thus, channel rearrangement reduces i_a by one and adds one to i_b but $i_a + i_b$ remains unchanged. If we let $i_0 = i_a + i_b$, the effect of channel rearrangement is cancelled by the aggregation of states. Then $\mathbf{I_r}$ becomes $\mathbf{I}$, which has a product form solution. We can use the convolution algorithm to find the distributions of $\mathbf{I}$. From the distributions, we can find p_c (20) and p_f (26). However, because we have not distinguished i_a and i_b in $\mathbf{I}$, the distribution cannot give us the information about channel rearrangement. Further analysis is needed.

If all channels of a gateway are occupied and the number of channels that are occupied by A-type calls is zero, A new B-type call arrival cannot use channel rearrangement and is blocked. We denote the probability as p_a. The probability can be expressed in terms of conditional probabilities given that v channels are lent

$$p_a = \sum_{v=0}^{L_{max}} \Pr(i_a = 0, i_b = C - v | \sum_{k=1}^{6} i_k = v) \Pr(v) \tag{31}$$

in which $\Pr(v)$ is the probability that v channels are lent. From $p(u, v)$, which is solved from (19), we can find marginal distribution $\Pr(v)$ by

$$\Pr(v) = \sum_{u=0}^{C-v} p(u, v) \ . \tag{32}$$

2.2.2 Conditional Probability, $\Pr\{i_a = o, i_b = C - v | \sum_{k=1}^{6} i_k = v\}$

In [1], we have shown that for CBWL/CR without cut-off priority, a decomposition method can be used to calculate this conditional probability. With the decomposition method, we can divide all state space into $L_{max} + 1$ subspaces, each of which corresponds to a fixed value of $v = \sum_{k=1}^{6} i_k$ ($v = 0, 1, \ldots, L_{max}$). The conditional distribution, $\Pr\{i_a, i_b | v\}$ can be calculated separately for each fixed v as if these subspaces were "independent" from one another. For CBWL/CR without cut-off priority, the decomposition method can produce exact solutions. Since the distribution of $\mathbf{I}$ is in product form, i_0 can be separated from other variables, and the interactions between i_a and i_b do not depend on other i_k's ($k \geq 1$). This is not true for CBWL/CR with cut-off priority. In this latter case, the distribution of $\mathbf{I}$ is not in product form, and i_0 cannot be completely separated from the other variables. Nevertheless, we can still use decomposition method approximately. Since the calls that arise in the given cell usually occur much often than borrowed requests ($\lambda \gg \lambda'$), the interactions between i_a and i_b are much stronger than the interactions between i_0 and other i_k's ($k \geq 1$). We can calculate $\Pr\{i_a, i_b | v\}$ separately for each fixed v and *omit* the interactions between i_0 and other i_k's ($k \geq 1$) as if those interactions do not exist. The agreement between results of simulation and analysis validates this approximation.

Given v, only i_a and i_b are unknown variables. The state-transition diagram of i_a and i_b for $C - v = 8$ is shown in Figure 3. In this diagram, $\lambda_1 = p\lambda$ is the arrival rate of A-type calls, $\lambda_2 = (1-p)\lambda$ is the arrival rate of B-type calls. When a gateway is in a state (i_a, i_b) with $i_a + i_b = C - v$ and $i_a > 0$, if a B-type call arrives, and if its channel borrowing request is not denied by neighbors, channel rearrangement is used. As the result of channel rearrangement, an A-type call is transferred to a borrowed channel and the released channel is given to the B-type call. Thus, the gateway's state is changed to $(i_a - 1, i_b + 1)$. Denote λ_3 as this transition rate. The probability that a borrowing request is accepted by neighbors is $1 - p_f^6$. Thus,

$$\lambda_3 = \lambda(1-p)(1-p_f^6) \ . \tag{33}$$

Denote $p_v(i_a, i_b)$ as the equilibrium distribution of (i_a, i_b) given that v channels are lent. From the diagram, we can obtain the probability flow balance equation of $p_v(i_a, i_b)$ and solve the equations to find $p_v(i_a, i_b)$. Since the dimensions are reduced, the number of states is decreased greatly and computation time is saved. We substitute $p_v(0, C-v)$ in (31) to find p_a. To use (31), $L_{max}+1$ groups of flow balance equations for v from 0 to L_{max} must be solved. However, the number of groups of equations to be solved can be reduced greatly. Because the probability that a gateway lends a lot of channels is very small and the contribution of these small $\Pr(v)$ to p_a can be omitted. In our calculations, when $\Pr(v)$ from (32) is less than a desired precision, the group of equations that corresponds to that v is not necessary to be solved.

2.2.3 Average Rate of Borrowing Requests, λ'

The average rate of borrowing request from a gateway to a neighbor, λ', in CBWL/CR scheme arise from both A-type and B-type calls. Totally,

$$\lambda' = \lambda \frac{1}{6} \frac{1 - p_f^6}{1 - p_f} \left[p p_c + (1-p)(p_c - p_a) \right] \ . \tag{34}$$

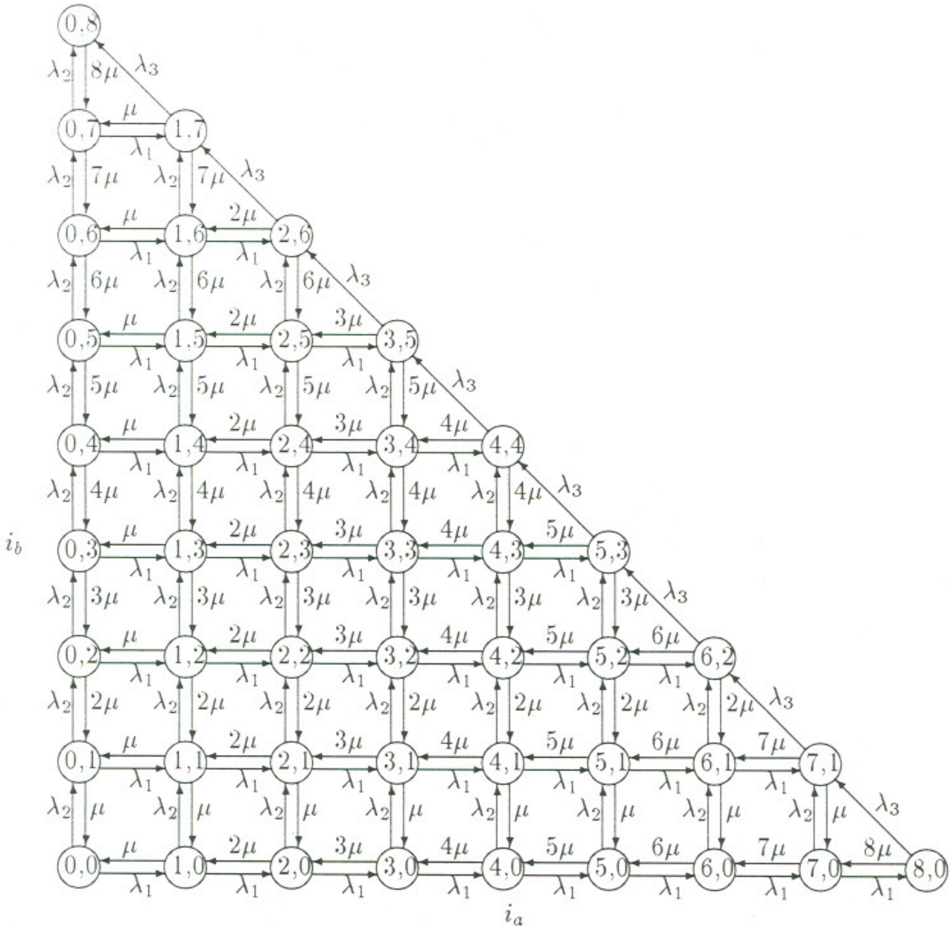

Figure 3: State-transition diagram of (i_a, i_b) $(C - v = 8)$.

Because p_c, p_a and p_f in (34) depend on λ, the equation is actually an implicit equation that can be used to obtain p_f, p_c and λ' simultaneously. An iterative procedure was used as outlined below.

Iterative procedure

Step 1 The procedure starts with an arbitrary guess of λ'.

Step 2 Use the last updated λ' in convolution algorithm to calculate $e_t(x)$ from (A.17) and then find $b(t, v)$ from (A.13).

Step 3 Find $\rho(v)$ from (A.8).

Step 4 Use $\rho(v)$ to construct balance equations (19). Then solve the equations to find $\Pr(u, v)$.

Step 5 Find p_c, p_f and $\Pr(v)$ from (20), (26) and (32).

Step 6 For any v, with $\Pr(v)$ large enough, $v = 0, \ldots, L_{max}$, solve probability balance equations of $P_v(i_a, i_b)$.

Step 7 Calculate p_a with equation (31).

Step 8 Update λ' using Equation (34).

Step 9 Step 2–8 of the procedure is continued until the difference of λ''s from the two consecutive iterations agree with the desired number of significant figures.

2.2.4 Blocking Probabilities

Once p_c, p_f and p_a are found, we can find blocking probabilities. First, the blocking probability experienced by an A-type call, α_{CR}, is the same as (28). With channel rearrangement, B-type calls can use borrowed channel indirectly. A new B-type call arrival will be blocked if All channels of the given gateway are occupied and the gateway cannot make channel rearrangement or if neighbors cannot lend any channels to the given gateway. Thus,

$$\beta_{CR} = p_a + (p_c - p_a)p_f^6 \ . \tag{35}$$

The overall blocking probability in a gateway is

$$B_{CR} = p\alpha_{CR} + (1 - p)\beta_{CR} \ . \tag{36}$$

The algorithms described in this section can be extended to the nonhomogeneous case, i.e., each cell may has different traffic rate and different number of assigned channels. The extension is described in [1].

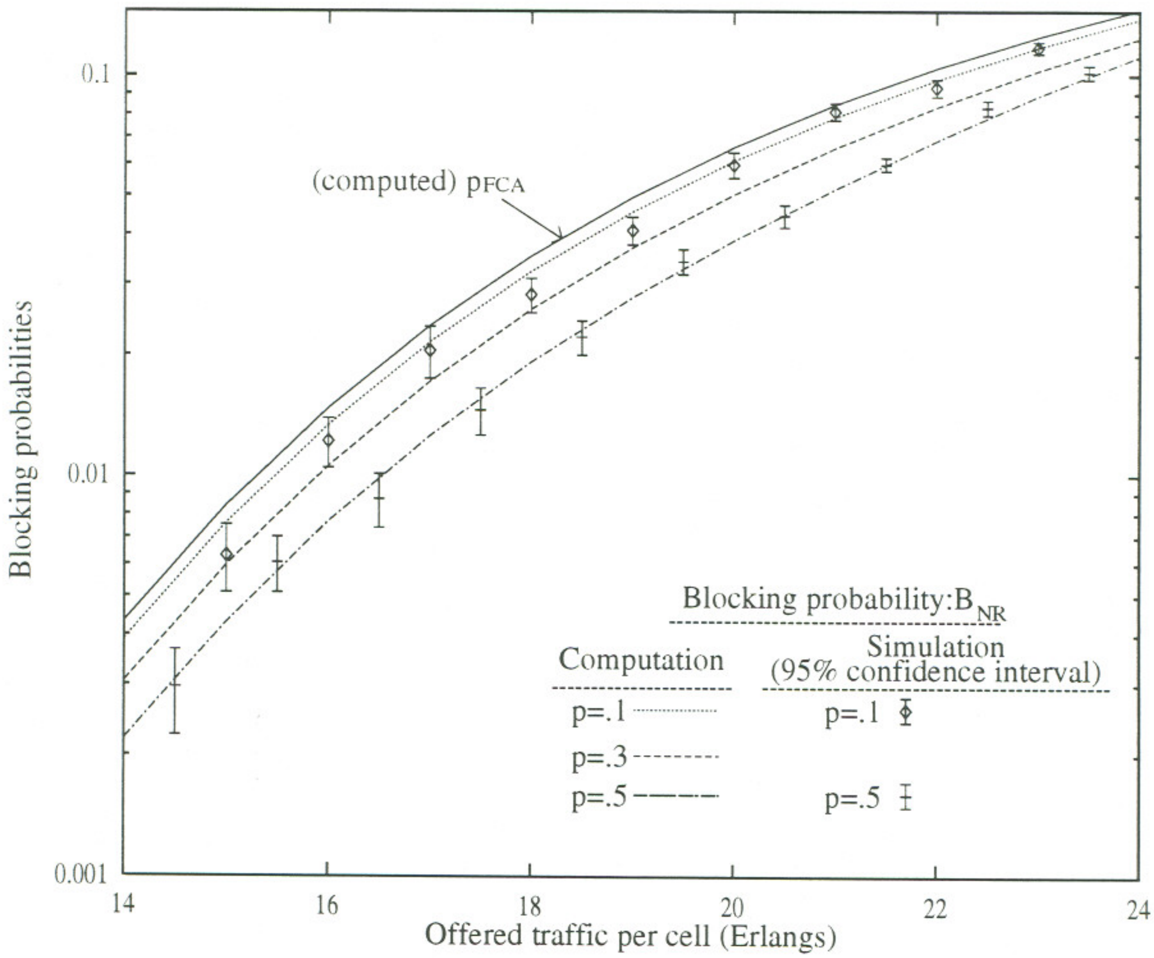

Figure 4: Performance of CBWL/NR with cut-off priority ($C = 24, m = 20, l = 4$).

3 Numerical Results and Discussion

As an example, we studied CBWL/NR and CBWL/CR for a cellular system with 24 channels in each gateway. We first consider homogeneous system. For simplicity, we assume in the homogeneous case, that the system has a very large (essentially infinite) number of cells. Thus we do not need to distinguish the boundary cells and the internal cells.

Figure 4 shows blocking probability, B_{NR} obtained by numerical computation plotted against offered traffic for CBWL/NR. Simulation confidence intervals of 95% are also shown. Figure 5 is a similar plot for CBWL/CR.

To simulate the system with a large (theoretically infinite) number of cells, we used a 37 cell configuration with each cell having six adjacent neighboring cells. We can imagine that the 37 cells cover a *ball* such that the boundary cells on one side are adjacent to the cells on the other side. In each run, we generate *about* 2000 call arrivals in each cell and determine the fraction of blocking calls for each cell. Since the cells are statistically the same, in one simulation run, 37 blocking probabilities can be found, as well as the mean, variance and confidence intervals for those blocking probabilities. The simulation was written in the *Simscript* simulation language, and was executed on a Sun workstation. From the figures, we can see that the results of analysis are close to those obtained by simulation. The results displayed in Figure 4 and 5 compare the performance of FCA (for which $p = 0$) to the CBWL scheme with $p = 0.1$, 0.3 and 0.5. It is seen that with $p = 0.5$ CBWL/NR can reduce the blocking probability about 50% while CBWL/CR can reduce the blocking probability by order of magnitude. Thus CBWL/CR can improve the performance significantly.

Figure 6 depicts blocking probability of A-type calls, α_{CR}, blocking probability of B-type calls, β_{CR} and overall blocking probabilities, B_{CR} for a CBWL/CR, also shown is the blocking probability of fixed channel assignment, B_{FCA}. Like CBWL/CR *without* cut-off priority, blocking probabilities are ordered by $B_{FCA} > \beta_{CR} > B_{CR} > \alpha_{CR}$. Figure 7 shows a comparison of blocking probabilities of CBWL/CR with and without cut-off priority ($m = 22$ and $m = 24$ for $C = 24$). It is seen that cut-off priority can reduce the *difference* in blocking probabilities of A-type calls and B-type calls. The overall blocking probability is also improved.

Figure 8 compares the overall blocking probabilities of FCA, CBWL/NR and CBWL/CR. Cut-off priority with $m = 22$ was used for the CBWL schemes. The figure indicates that for CBWL with channel rearrangement, the performance of the system is significantly enhanced.

Table 1 shows a comparison of offered traffic that can be accommodated at a 2% blocking probability for CBWL/CR with and without cut-off priority. Specifically, it tabulates the percentage increase (in offered traffic) in comparison with the corresponding FCA scheme. When the fraction of A-type calls, p, is increased, the offered traffic of CBWL/CR is increased. For $p = 0.5$, the offered traffic is increased about 34.2%. The offered traffic that can be accommodated increases very rapidly with increasing p for $p < 0.5$. When p is greater than 0.5, the increase is slowed. Thus increasing p beyond $p > 0.5$ helps little to improve system performance for CBWL/CR with cut-off priority. The effect is not a severe limitation for CBWL/CR, because co-channel interference usually requires small p for CBWL. The performance of CBWL/CR with cut-off priority is better for small p ($p < .4$) while the performance of CBWL/CR without cut-off priority is better for larger p. For the usual range of small p, cut-off priority can improve the performance of system.

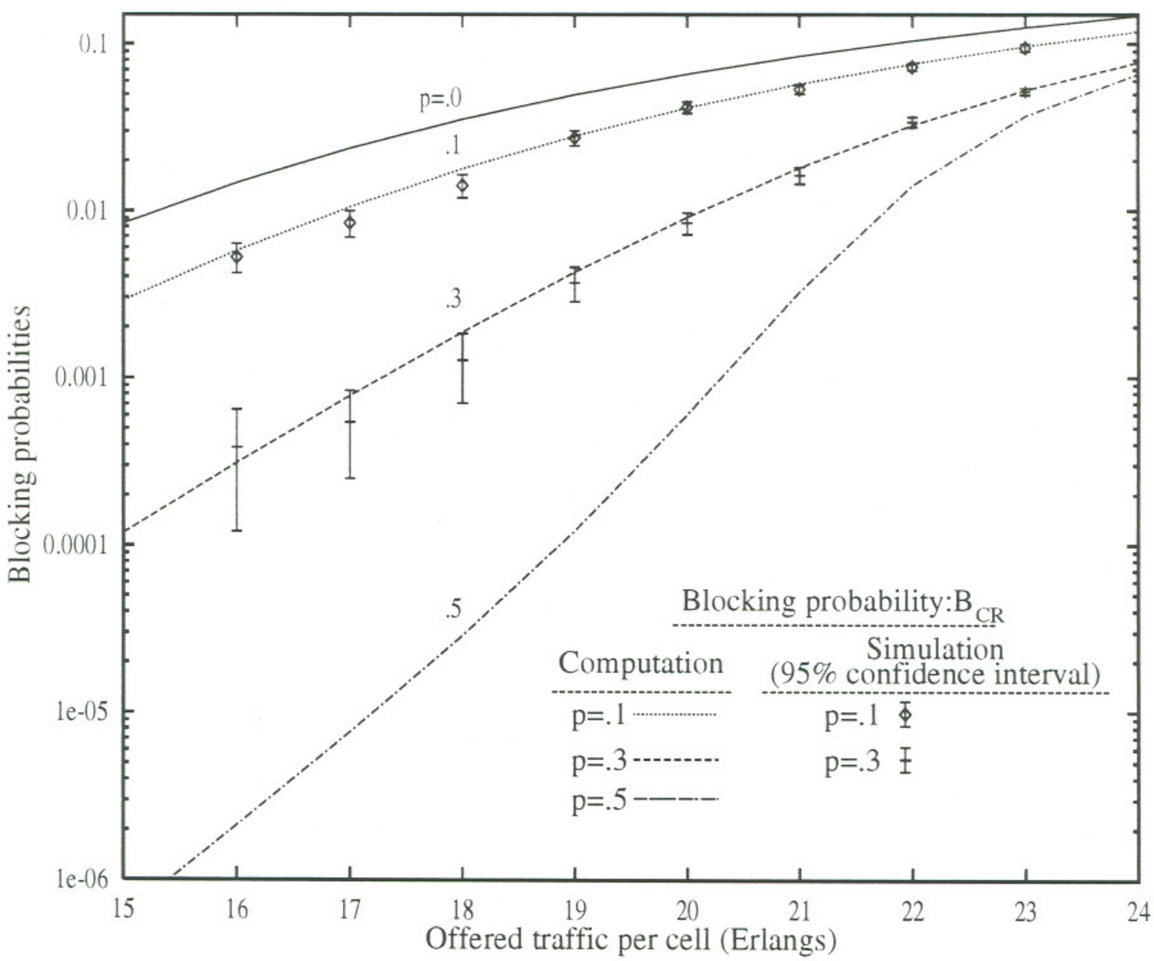

Figure 5: Performance of CBWL/CR with cut-off priority ($C = 24, m = 20, l = 4$).

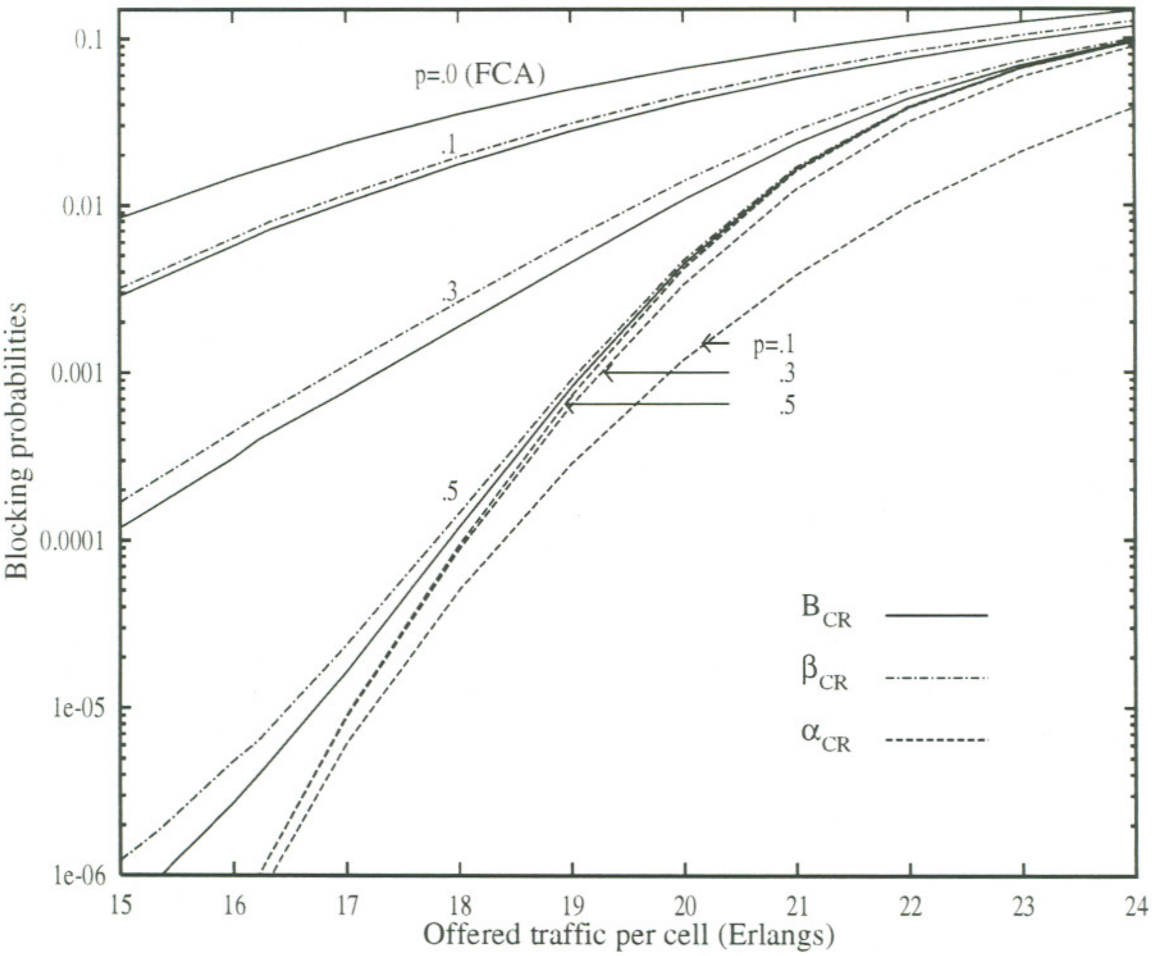

Figure 6: Blocking probabilities for CBWL/CR ($C = 24, m = 20, l = 4$).

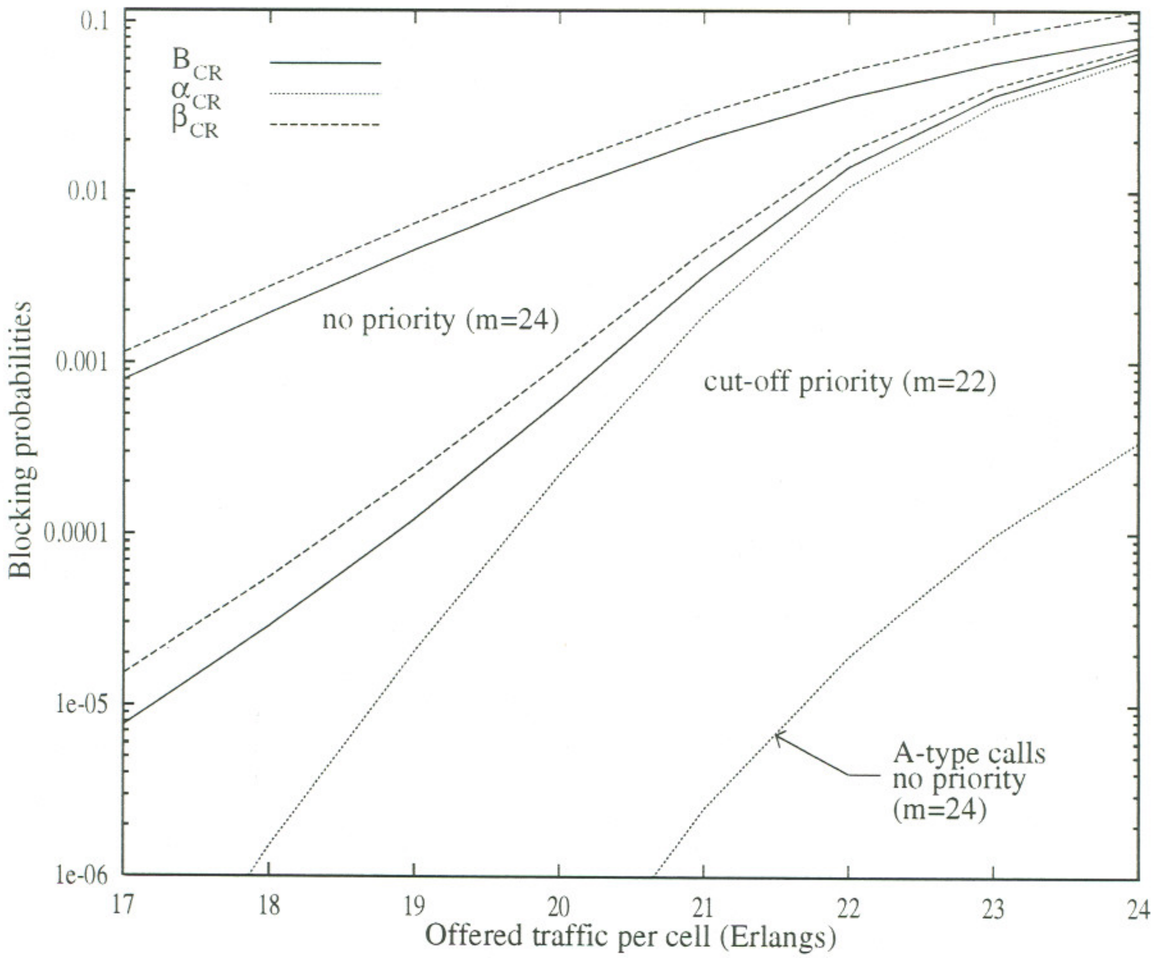

Figure 7: Blocking probabilities for CBWL/CR with and without cut-off priori ty ($C = 24$).

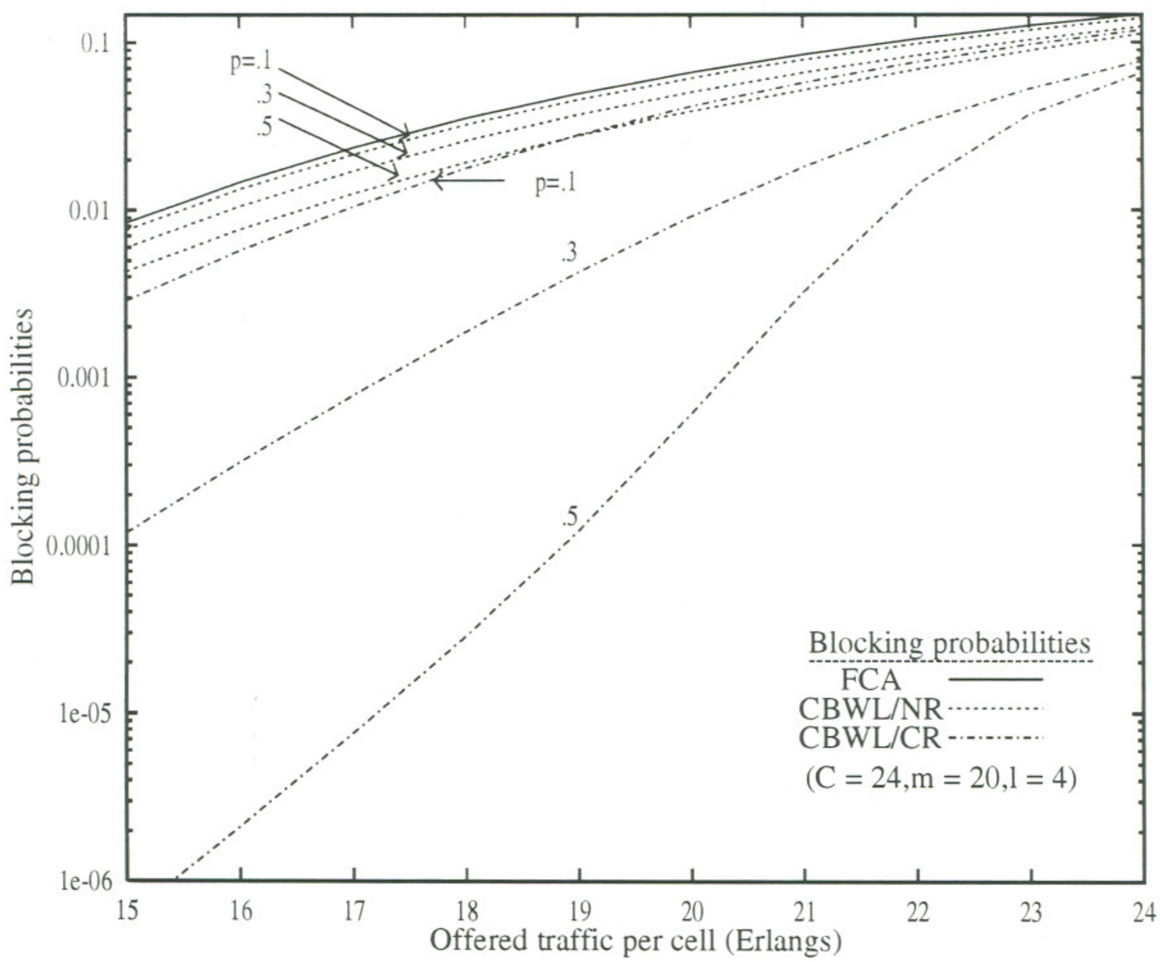

Figure 8: Comparison of CBWL performance with cut-off priority ($m = 22$).

Table 1: The offered traffic per cell of CBWL/CR for 2% blocking probability, $B_{CR} = .02$.

p	with cut-off priority $C = 24, m = 22, l = 4$				no cut-off priority $C = m = 24, l = 3$	
	offered traffic (Erlang)	percent increase	β_{CR}	α_{CR}	offered traffic (Erlang)	percent increase
0.0	16.63	0.0%	.0200	.0000	16.63	0.0%
0.1	18.25	9.7%	.0223	.0000	18.22	9.6%
0.2	19.81	19.0%	.0250	.0001	19.74	18.7%
0.3	21.16	27.2%	.0280	.0013	21.00	26.3%
0.4	22.01	32.4%	.0281	.0082	21.97	32.1%
0.5	22.31	34.2%	.0238	.0162	22.72	36.6%
0.6	22.39	34.6%	.0210	.0195	23.18	39.3%
0.7	22.40	34.7%	.0201	.0199	23.37	40.5%
0.8	22.40	34.7%	.0200	.0200	23.42	40.8%

Table 1 also shows β_{CR} and α_{CR} given $B_{CR} = .02$. It is seen that β_{CR} and α_{CR} are closer than those in CBWL without cut-off priority. We notice that β_{CR} has a peak at about $p = 0.4$. The effect can be explained as follows. When p is increased from 0, for a fixed $B_{CR} = 0.02$, the offered traffic that a cell can accommodate is increased. The increasing of offered traffic causes p_c and p_f to increase, this causes increase of β_{CR}. But, when p is increased, the fraction of A-type calls is increased and the probability that a B-type call cannot use channel rearrangement, p_a [in (31), the probability that all channels of a gateway are occupied, and no channel is occupied by A-type of calls) becomes relatively small. The decrease of p_a causes the decrease of β_{CR}. When p is greater than 0.4, the rate of decrease is greater than the rate of increase, β_{CR} thus is decreased. When p continue to increase, β_{CR} approximates to B_{CR} and α_{CR}.

The effect of the cut-off priority m on blocking probability is shown in Figure 9. The blocking probabilities for $m = 21$, 22 and 23 are better than that for $m = C = 24$ (without cut-off priority). This is because too much lending to neighboring cells reduces the availability of regular channels to the users in the given cell. Since borrowed channels can only be accessed by a fraction of users, unlimited channel lending will hinder performance of CBWL/CR. Therefore, an appropriate borrow limit should be chosen In this example, $m = 22$ gives best performance.

For the nonhomogeneous case, an example system with a single hot spot was considered. In Figure 10, the system has 37 cells with $C = 24$ and $m = 22$. The central hot spot cell has 1.5 times the offered traffic of its surrounding 3 tiers of cells. In CBWL/CR, the blocking probability of the central hot spot cell and any other cell is significantly reduced in comparison with FCA ($p = 0$).

4 CONCLUSION

Our analysis and simulation shows that CBWL/NR and CBWL/CR *with cut-off priority* can reduce the difference of blocking probabilities of calls that arise in different location. The overall blocking probability of CBWL with cut-off priority (with a carefully selected m) is also better than the corresponding schemes without cut-off priority in both homogeneous and hot spot cases.

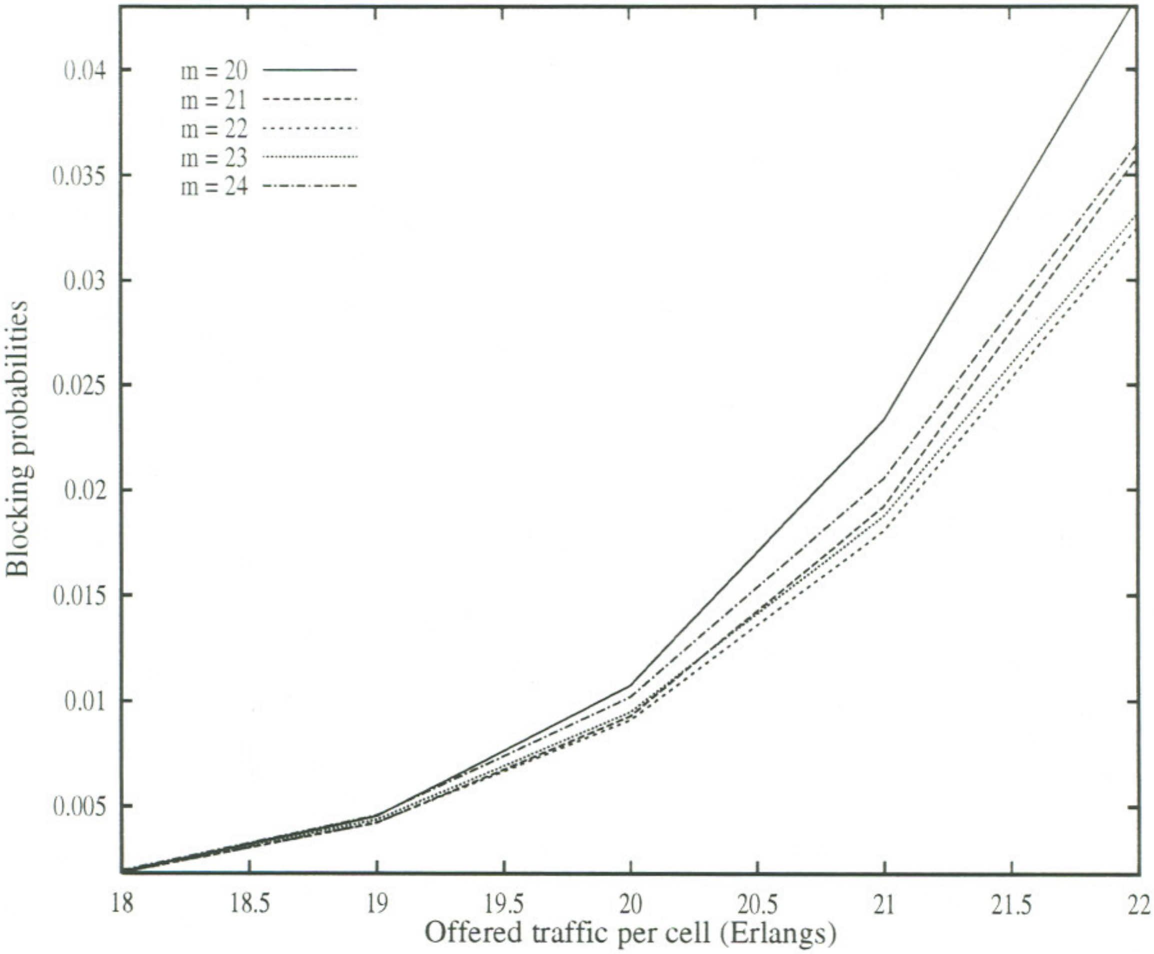

Figure 9: Performance of CBWL/CR under different m ($C = 24, p = .3, l = 4$).

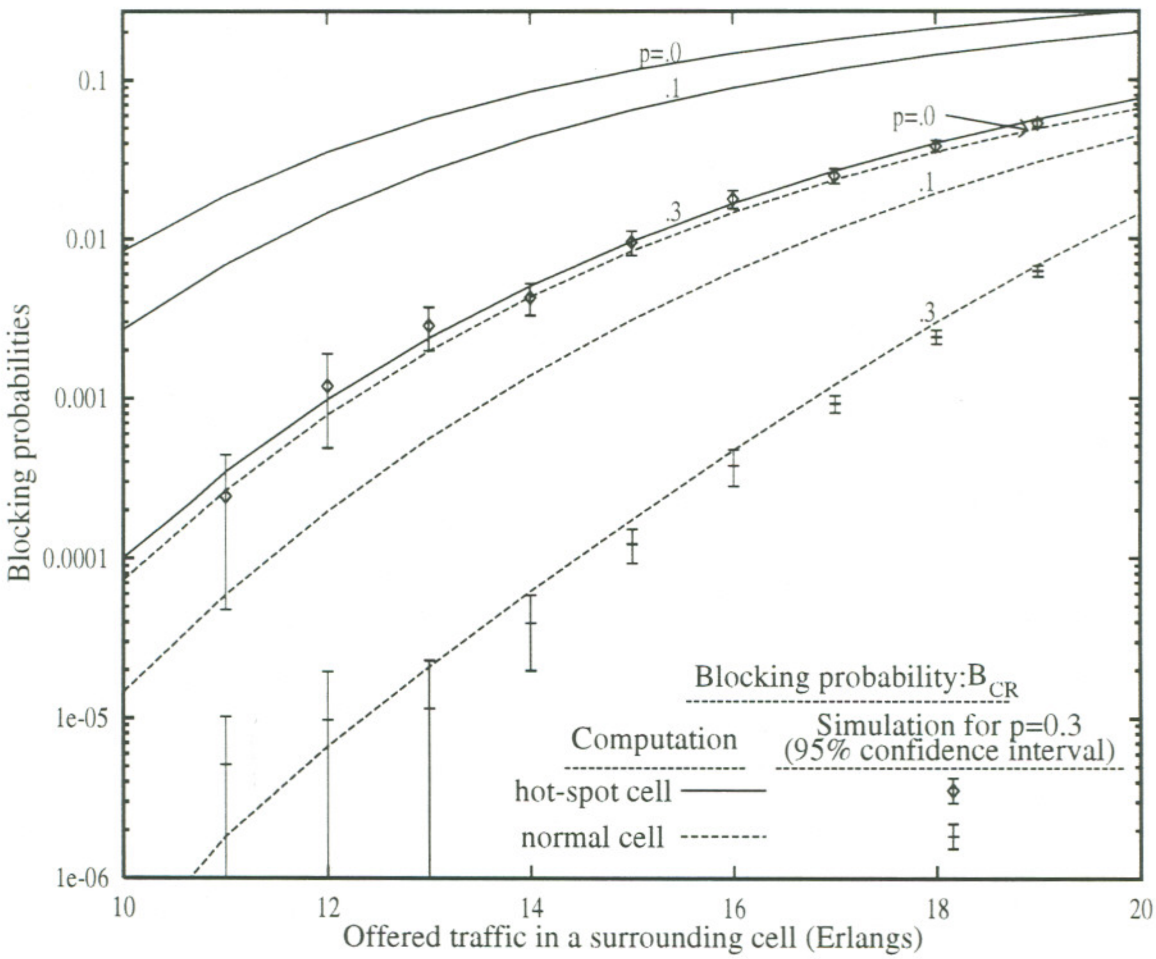

Figure 10: Blocking probabilities of CBWL/CR in hot-spot traffic ($C = 24, m = 22, l = 4$).

Appendix A. Calculation of $\rho(v)$

To find $\rho(v)$ in (16), we need the distribution, $\Pr(\mathbf{I_6})$, in which $\mathbf{I_6} \in S(v,6)$. From (15), the probability is $\Pr(i_1, i_2, i_3, i_4, i_5, i_6 | \sum_{k=1}^{6} i_k = v)$. Given that v channels are lent, the v channels are shared by six streams of borrowing requests from neighbors. The analytical structure of this problem is essentially the same as that in which several types of customers share a finite group of servers. It has been shown that the state probabilities can be expressed in product form [8]. Therefore, a convolution algorithm can be devised to find the distribution $\Pr(i_1, i_2, i_3, i_4, i_5, i_6 | \sum_{k=1}^{6} i_k = v)$ effectively.

We define a function

$$f(x) \triangleq \frac{\left(\frac{\lambda'}{\mu}\right)^x}{x!} \qquad x = 0, 1, \ldots, l\ . \tag{A.1}$$

Given v channels being lent, the distribution of number of channels lent to each adjacent gateway has following product form

$$\Pr(\mathbf{I_6} \in S(v,6)) = \Pr(i_1, i_2, i_3, i_4, i_5, i_6 | \sum_{k=1}^{6} i_k = v) = \frac{1}{b(v)} \prod_{k=1}^{6} f(i_k) \tag{A.2}$$

in which, $b(v)$ is the normalization constant. Since the sum of the conditional probabilities given by (A.2) must add to unity, the constant is given by:

$$b(v) = \sum_{\mathbf{I_6} \in S(v,6)} \prod_{k=1}^{6} f(i_k)\ . \tag{A.3}$$

Buzen's convolution algorithm can be used to calculate $b(v)$ recursively, [9]. The algorithm considerably reduces the computational effort needed to find $b(v)$, which is necessary for determining $\Pr(\mathbf{I_6})$ using (A.2). However, when we calculate $\rho(v)$ from (16), many numerical operations are still required with all possible $\mathbf{I_6} \in S(v,6)$. We will devise a modified convolution algorithm which not only can be used to find $b(v)$ but also can be used to find $\rho(v)$ efficiently.

Using (5) in (16), we note that the channel lending rate $\lambda' \sum_{k=1}^{6} O(i_k)$ for any given $\mathbf{I_6} \in S(6,v)$ is one of seven possible values: 0, λ', …, $6\lambda'$. If $\mathbf{I_6}$ contains exactly $(6-t)$ l's, its corresponding lending rate is $t\lambda'$. Denote S_{tv} ($t = 0, 1, \ldots, 6$) as the set of $\mathbf{I_6} \in S(6,v)$ which has exactly $6-t$ components that are equal to l. The set, S_{tv}, consists of all ways that exactly $6-t$ adjacent gateways (without regard to which gateways they are) have borrowed l channels from the given gateway. Thus,

$$S(v,6) = \bigcup_{t=0}^{6} S_{tv}\ . \tag{A.4}$$

Define

$$b(t,v) \triangleq \sum_{\mathbf{I_6} \in S_{tv}} \prod_{k=1}^{6} f(i_k)\ . \tag{A.5}$$

If $v < (6-t)l$, in any case, no more than $6-t-1$ components of $\mathbf{I_6}$ can be exactly l. If $v \geq 6l - t$, more than $6-t$ components of $\mathbf{I_6}$ must be exactly l. Thus,

$$b(t,v) = 0 \qquad \text{if } v < (6-t)l, \text{ or } v \geq 6l - t \tag{A.6}$$