Channel Assignment Algorithms in Cellular Radio Networks

Thesis by

Sanjeev K. Deora

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

1995
(Submitted May 23, 1995)

in memory of

my grandfather Mr. Madan Lal Deora

and

Professor Edward C. Posner

Acknowledgements

I am very fortunate to have had Professor Robert J. McEliece as my advisor. Many of the ideas in this thesis were inspired by him. The techniques that he taught will be very useful to me in future. I am greatly indebted to him for his guidance and encouragement in the past few years. Working with him was an honour, a pleasure and a great learning experience.

I would also like to thank Dr. John Ketchum from GTE Laboratories for some very useful discussions I had with him. These discussions gave me a lot of insight into the real problems facing the cellular industry. I would also like to thank Dr. John Blanchard in the Department of Mathematics at Caltech for the several discussions we had and from which I benefitted a lot. Also, I would like to thank Dr. Kumar N. Sivarajan at the Indian Institute of Sciences, Bangalore, with whom I had some useful discussions and whose work on channel assignment algorithms gave me a lot of ideas.

I also thank the members of the defense committee: Professors R. J. McEliece, P. P. Vaidyanathan, A. Goldsmith, and Drs. J. Blanchard and M. Simon.

I also thank GTE Laboratories Inc., Waltham, as well as Pacific Bell for their financial support.

I would also like to thank some of my friends at Caltech. I thank Ashish Misra and Ranjan Mukhopadhyay. They have been very good friends and have always offered support and advice whenever required. I would also like to thank some of my friends at Keck Graduate House where I stayed for a couple of years: Tyrome Brown, Konrad, Rajit, Hans Michael Mueller and Jessie Xu. Without their friendship and company, life would have been very difficult at Caltech. I would also like to thank Bahadir Erimli, Wei Lin, Hongyu Piao, Akira Takahashi and Zhong Yu. They have been very good friends. Finally, I would like to thank my parents, family members and my friends during my undergraduate years at IIT, Madras.

Abstract

In this thesis, we study and compare the performance of several distributed channel assignment algorithms (CAAs) in a cellular system. The CAA which is used to assign a channel to a new call greatly influences the amount of traffic the system can support. We are interested in the design and analysis of algorithms which perform well, but at the same time are relatively easy to implement. In this thesis, we have analyzed the performance of a very simple CAA which we call the Timid Algorithm, in the limiting case of a large number of channels. We have been able to show that, under a plausible mathematical hypothesis, the algorithm is asymptotically optimal, where "asymptotically" refers to a system with a large number of channels. This is very surprising as there are algorithms of much higher complexity which provably do not have this property.

The Timid Algorithm is asymptotically optimal, but it requires a large number of channels for a satisfactory performance. We looked at some algorithms which retain the simplicity of the Timid algorithm but which can be expected to give a good performance even with a smaller number of channels. We called one such algorithm the Modified DCAA. We present some simulation results which show that this algorithm gives a reasonably good performance even when the number of channels is small. One of the ways to increase the capacity of a cellular system is through the use of microcells. The Modified DCAA, because of its distributed nature and low complexity, is particularly suitable for such microcellular systems.

We also present a method for computing the upper bound on the performance of any CAA in a cellular system with adjacent channel constraints. The method, although computationally intensive, may be useful for determining how close an algorithm's performance is to the optimal performance.

Finally, we discuss ways of obtaining the set of "allowable" states for a system. We also present some "measurement-based" algorithms and compare their performance

with "prediction-based" algorithms.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 A Cellular System

In a cellular system, the service area is divided into a large number of smaller areas called cells. Each cell has a base station. Communication takes place between the mobiles in the cells and the corresponding base stations through frequencies or channels. Since the number of channels is limited, the same channel has to be used in different cells simultaneously to increase the system capacity. However, the cells that use the same channel simultaneously cannot be very close to each other, otherwise the interference among them may be unacceptable. The set of cells which can use the same channel simultaneously while keeping the interference within acceptable limits is specified by the cochannel reuse constraints.

Example. Consider a linear array of three cells shown in Figure 1.1. Suppose the reuse constraint is that the same channel cannot be used in adjacent cells simultaneously. Figure 1.2 shows the state diagram for a single channel. A "1" corresponding to a cell indicates that a channel is being used by a call in the cell, while a "0" indicates that the channel is not being used by any call in the cell. We shall refer to this example throughout most of the thesis.

1	2	3

Figure 1.1: A linear array of three cells

The offered traffic (measured in Erlangs) in a cell is the average number of calls that would be in progress in the cell if all the calls were accepted. Since the number of channels is limited, some calls have to be blocked. The carried traffic is the average number of calls which are in progress at any time when a particular algorithm is used.

The performance of an algorithm is measured by the carried traffic or the blocking probability. The channel assignment algorithm (CAA) which is used to assign a channel to a call greatly influences the amount of traffic the system can support. Our aim is to design algorithms that have a good performance and are also relatively easy to implement. Some algorithms give a good performance by being unfair, i.e., by providing low blocking probabilities in some cells and high blocking probabilities in others. Although the issue of fairness is important, we don't consider it in our thesis.

We will consider only the class of CAAs which either accept a call request or block it. There is no call waiting. Also, calls-in-progress cannot be dropped. Throughout the thesis, we assume that a call remains in the cell in which it originated throughout its duration.

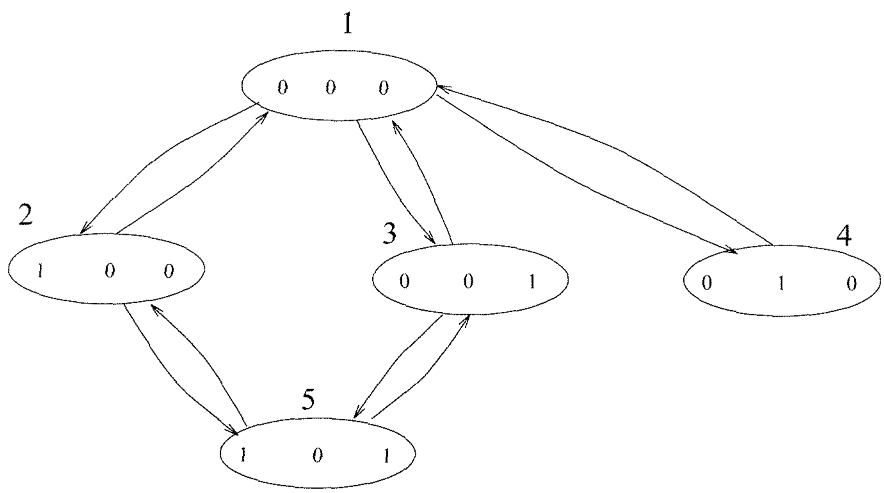


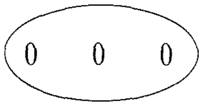
Figure 1.2: The state diagram for a single channel

1.2 Important Questions Regarding a CAA:

Now we state the important questions we face when assigning a channel to a call:

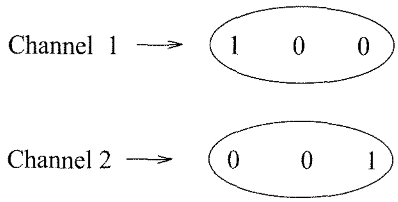
- Should we block it even if there is a channel available to accept the call?

Consider the 3-cell system. Suppose there is only one channel and it is in the all-zero state as shown above. Suppose we know that the offered traffic in cell 1 and



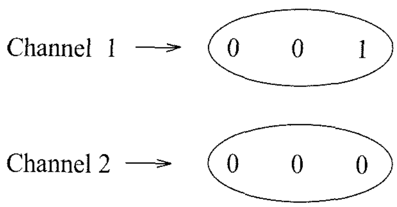
cell 3 is very high. Then if a new call arrives in cell 2, it might be better, from the point of view of maximizing the carried traffic, to block the call although it can be accepted. Whereas such an algorithm definitely increases the carried traffic, it requires a knowledge of the offered traffic which is not always available.

- Should we rearrange the existing calls to see if the new call can be accepted?



Suppose there are two channels in the 3-cell system. Suppose channel 1 is in state 2 and channel 2 is in state 3. A new call arrives in cell 2. If the call in cell 3 is moved to channel 1 or the call in cell 1 is moved to channel 2, then the new call can be accepted. This rearrangement of existing calls to accept a new one apparently increases the carried traffic. However it leads to an increase in the complexity of the algorithm.

- Should we look for the "best" available channel to be given to the call?



Suppose there are two channels in the 3-cell system. Suppose channel 1 is in state 3 and channel 2 is in state 1. A new call arrives in cell 1. If channel 1 is given to the new call, then channel 2 will continue to be in state 1 and hence if the next event is a call arrival in cell 2, the system will be able to accept the new call. However if channel 2 is given to the new call, and the next event is a call arrival in cell 2, the system cannot accept the new call without rearranging the existing ones. Thus channel 1 appears to be a "better" choice than channel 2. However, we should note that although looking for the best available channel apparently increases the carried traffic, it also leads to an increase in the complexity of the algorithm.

The performance as well as the complexity of a CAA depend on how the algorithm answers the three questions given above. We should note that a CAA need not answer the above questions as just "Yes" or "No". The answer may be somewhere in between. For example, there may be a CAA which allows rearrangement of up to one call in order to accommodate a new one.

1.3 The Timid DCAA

One of the ways to meet the increasing demand of cellular phones is through the use of microcells. These allow the use of the same channel by a larger number of users at the same time. However they require the use of low-complexity CAAs as the system size is typically very large because of the large number of cells. Most of the known CAAs which have a good performance are very complex or have some other disadvantages. The complexity of these CAAs increases with the system size.

We propose a very low complexity algorithm. We call it the **Timid Dynamic Channel Assignment Algorithm (TDCAA)**. The algorithm is as follows:

TDCAA: All channels are available for use in all the cells, provided they don't violate the reuse constraints. When a call comes to a cell, a channel is chosen at random from among the channels that are available for use at that time and is given to the call. If no channel is available, the call is blocked. Call rearrangements are not allowed.

Let us see how it answers the questions given above:

- It does not block a call if there is a channel available to accept the call.

- It does not rearrange calls in progress in order to accept a new one.
- It does not look for the "best" available channel before accepting the call. It picks one of the available channels at random.

The idea is that when a call comes to a cell, the interference power received at the corresponding base station at the different frequencies is measured, and from among those frequencies which have interference power less than a threshold (that is, those which are available), one is chosen at random. In this sense, it is a decentralized algorithm and is very suitable for systems with a large number of cells, e.g., microcellular systems.

The TDCAA has none of the features which increase the complexity but which we might think also improve the performance of an algorithm. It is perhaps one of the simplest algorithms one can think of. Yet we have been able to show that the algorithm's performance is quite good. In fact, its asymptotic performance is better than that of some other algorithms of very high complexity.

Asymptotically Optimal Algorithm: Consider a cellular system described by a set of cells, a set of reuse constraints, and some offered traffic per channel (perhaps different in different cells). We say an algorithm is *asymptotically optimal* if, for any given cellular system, as the number of channels becomes large, the carried traffic per channel, obtained by using the algorithm, is at least as large as can be achieved using any other algorithm. (We consider only algorithms that are not allowed to drop calls in progress. We also assume that a call remains in the cell in which it originated throughout its duration.)

Conjecture: The TDCAA is asymptotically optimal.

We can prove the conjecture assuming a certain plausible mathematical hypothesis, and we have strong experimental evidence to support our conjecture in many special cases.

This is very surprising as there are some highly complex algorithms which are provably not asymptotically optimal. For example, the "Greedy" algorithm, which is identical to the TDCAA except that it allows call rearrangements in order to accept a new call, does not have this property.

Example. Consider the linear array of three cells shown in Figure 1.1. Figure 1.2 shows the state diagram for a single channel. Suppose the offered traffic is uniform in all the three cells. Figure 1.3 gives an upper bound on the performance, measured by the carried traffic per channel, for this system for various values of the offered traffic [18]. We simulated the performance of the TDCAA, assuming $n = 10000$ channels, for various values of the offered traffic. Figure 1.3 shows the result. It also shows the performance, in the limit of a large number of channels, of the highly complex GDCAA [19], as well as of a particular FCAA [18]. As is clear, when the number of channels is large, the performance of the TDCAA tracks the upper bound, whereas the performance of the GDCAA does not.

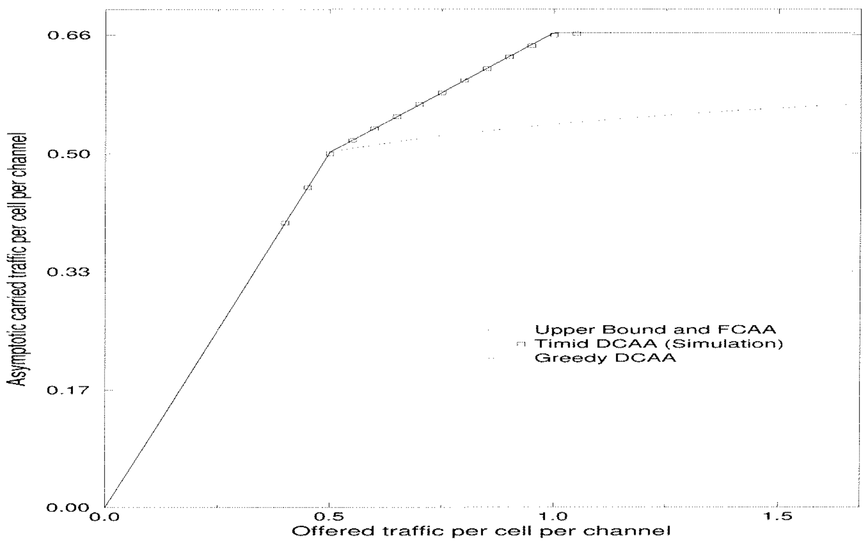


Figure 1.3: Asymptotic performance of various CAAs for a linear array of 3 cells with uniform traffic distribution

1.4 Rest of the Thesis

The main disadvantage with the TDCAA is that a large number of channels are required before it gives a reasonably good performance. One of the reasons for this is that it selects a channel randomly from the set of available channels. We studied another "dynamic" CAA which does away with the random selection from the set of available channels but otherwise retains the simplicity of the TDCAA and which can be expected to give a reasonably good performance even with a small number of channels. We call it the Modified DCAA. We have some simulation results which indicate that the Modified DCAA does indeed approach the performance limits much faster than the TDCAA.

The result regarding the TDCAA has been obtained for cellular systems with only cochannel constraints. However, the use of a channel in a cell imposes restrictions on the use of nearby channels in the neighbouring cells. These constraints are called adjacent channel constraints. Consider a cellular system described by a set of adjacent channel constraints and some offered traffic per channel (perhaps different in different cells). We will show how to compute, by linear programming, an upper bound on the performance of any given cellular system with a given number of channels. We will further show that as the number of channels becomes large, this upper bound approaches a limit. We will also show that the upper bound is asymptotically tight in the sense that as the number of channels becomes large, there are algorithms which achieve the upper bound.

The thesis is organized as follows. In Chapter 2, we prove two theorems, one regarding the equilibrium distribution of two stationary Markov processes whose transition rates satisfy certain conditions and the other regarding the equilibrium distribution of an "almost Markov" stochastic process whose transition rates and equilibrium distribution satisfy certain conditions. These theorems are used to prove the results in Chapter 3. In Chapter 3, we begin with our analysis of the TDCAA. We state the model, the hypothesis, and prove that the algorithm is asymptotically optimal provided the hypothesis is correct. In Chapter 4, we compare the TDCAA with two well known algorithms. We also look at an example where the TDCAA performs better than a well known algorithm of very high complexity and try to understand the reason for this. We also describe the idea of the Modified DCAA and give simulation results that indicate that this algorithm gives a good performance even with a small number of channels. In Chapter 5, we give a method for computing an upper bound on the performance of any given cellular system with a given number of channels and show that the upper bound is tight in the sense that there are algorithms which achieve this bound as the number of channels becomes large. In Chapter 6, we discuss interference, availability and ways of computing the comparability matrix for a cellular system. We also discuss some "measurement-based" CAAs.

Chapter 2 Two Theorems Related to Markov Processes

2.1 Introduction

In this chapter, we prove two theorems related to Markov processes [7], [5]. Throughout this discussion, we assume that the Markov processes we deal with are irreducible, have a finite state space, and have an equilibrium distribution. The first theorem states that if the transition rates of two discrete-space continuous-time stationary Markov processes tend to each other, then their equilibrium distributions also tend to each other. The second theorem states that if the transition rates of a discrete-space continuous-time "almost Markov" process satisfy certain conditions, then the equilibrium distribution of the process is the same as the equilibrium distribution of a Markov process whose transition rates are related to the transition rates of the "almost Markov" process in a simple manner. These theorems turn out to be very useful in proving some of the results in Chapter 3.

2.2 A Theorem About Stationary Markov Processes

In this section, we prove a property about the relation between the equilibrium distribution of two discrete-space, continuous-time, irreducible, stationary Markov processes whose transition rates tend to each other.

Theorem 2.1 Consider two discrete-space, continuous-time, irreducible, stationary Markov processes $X_n(t)$ and $X'_n(t)$ having the same finite state space S. Let $\lambda_{ij}(n)$ be the transition rate from state i to state j for the process $X_n(t)$ and let $\lambda'_{ij}(n)$ be the transition rate from state i to state j for the process $X'_n(t)$. The transition rates for the two systems are functions of the parameter n where n is real. Let $\{\pi_j(n)\}$ be the equilibrium distribution for the process $X_n(t)$ and let $\{\pi'_j(n)\}$ be the equilibrium distribution for the process $X'_n(t)$. Suppose that $\lambda_{ij}(n) = 0$ if and only if $\lambda'_{ij}(n) = 0$ for all $i, j \in S$ and for all n. We also assume that if $\lambda_{ij}(n) = 0$ for any n, then it is zero for all n.

For all $i, j \in S, i \neq j$ such that $\lambda_{ij}(n) \neq 0$, if

$$\lim_{n\to\infty} \frac{\lambda_{ij}(n)}{\lambda'_{ij}(n)} = 1,$$

then for all $j \in S$,

$$\lim_{n\to\infty} \frac{\pi_j(n)}{\pi'_j(n)} = 1.$$

This says that if the transition rates of the two processes tend to each other, then their equilibrium distributions also tend to each other.

In order to prove this theorem, we need several results which we prove before proving the main theorem.

Determinant Properties of a Transition Matrix

Let $a_1, a_2, \ldots$ be variables which can take any non-negative value. Consider a square matrix A with the following properties:

Each diagonal element is a sum of some of these variables. The other entries are either zero or negative of one of these variables. Each column sum is greater than or equal to zero for all non-negative values of these variables.

Let F_l be the set of all such matrices of order l.

Example:

$$G = \begin{pmatrix} a_1 + a_2 + a_3 & 0 & -a_6 & -a_9 \\ -a_1 & a_4 + a_5 & -a_7 & 0 \\ -a_2 & -a_4 & a_6 + a_7 + a_8 & 0 \\ 0 & -a_5 & -a_8 & a_9 \end{pmatrix}.$$

Lemma 2.1 *For all l, the determinant of a matrix $A \in F_l$ always has non-negative coefficients .*

For example, $\det G = a_3a_4a_6a_9 + a_3a_5a_6a_9 + a_3a_5a_7a_9 + a_3a_4a_8a_9 + a_3a_5a_8a_9$. Here the coefficients of all the terms are non-negative.

Proof: We should note that as the column sums are non-negative for all non-negative values of the variables, the variables which appear as elements of a column (excluding the diagonal element) must appear as a term in the diagonal element corresponding to that column. In addition to these, the diagonal element may consist of some other terms with positive coefficients. The sum of these terms is equal to the column sum and we will denote it by Δ_i for column i.

Let a_i denote column i of the matrix A. Then we can write

$$a_i = a'_i + e_i,$$

where e_i is a column vector with Δ_i for entry i and zero for all other entries and a'_i is an appropriate vector such that the above relation is satisfied. We should note that the sum of the entries of a'_i is equal to zero.

We can write

$$A = (a'_1 + e_1, a'_2 + e_2, \ldots, a'_l + e_l),$$

where l is the order of the matrix A.

We will prove the lemma by induction. Suppose the lemma is true for $l \leq k-1$. We will prove it for $l = k$.

If A is of order k, we have

$$\begin{aligned} \det A &= \det(a'_1 + e_1, a'_2 + e_2, \ldots, a'_k + e_k) && (2.1)\\ &= \det(a'_1, a'_2 + e_2, \ldots, a'_k + e_k) + \det(e_1, a'_2 + e_2, \ldots, a'_k + e_k). && (2.2) \end{aligned}$$

Now, the matrix formed by crossing out the first row and column of $(e_1, a'_2+e_2, \ldots, a'_k+e_k)$ is of order $k-1$ and satisfies all the properties for a matrix to be an element of F_{k-1}. Therefore, $\det(e_1, a'_2 + e_2, \ldots, a'_k + e_k)$ consists only of positive terms since it is equal to the product of Δ_1 (consisting only of positive terms) and the determinant of a matrix which belongs to F_{k-1}.

Also,

$$\det(a'_1, a'_2+e_2, \ldots, a'_k+e_k) = \det(a'_1, a'_2, a'_3+e_3, \ldots, a'_k+e_k)+\det(a'_1, e_2, a'_3+e_3, \ldots, a'_k+e_k). \tag{2.3}$$

By an argument similar to the above argument, the second term on the right side of (2.3) also consists only of positive terms. Proceeding in the above manner, we can show that the first term on the right side of (2.3) also consists of only positive terms. We should note that finally we will be left with a matrix of order k whose column sums are zero and hence its determinant is zero.

Hence from (2.1), (2.2) and (2.3) and the above argument, we conclude that for $l = k$, the determinant of a matrix $A \in F_l$ consists only of positive terms. The assumption is true for $l = 1$. Hence it is true for all l.

Lemma 2.2 Consider an irreducible, stationary Markov Process $x(t)$. Let $S = \{1, 2, \ldots, m\}$ be the finite set of states of the system. Let $\lambda_{i,j}$ be the transition rate from state i to state j. Let A be an $m \times m$ matrix defined as follows:

$$a_{ij} = \begin{cases} -\lambda_{ji}, & \text{if } i \neq j; \\ \sum_{j \neq i} \lambda_{ij}, & \text{otherwise.} \end{cases}$$

Whenever we refer to the variables a_{ij}'s, we shall mean the variables a_{ij}'s for $i \neq j$. We can say that the a_{ij}'s are non-negative. Let $\mathbf{B} = (b_{ij})$ be the matrix formed from $\mathbf{A}$ by replacing its first row by an all-one vector. Let B_{ij} be the value of the cofactor corresponding to b_{ij}. We should note that both $\mathbf{A}$ and $\mathbf{B}$ are square matrices of order m. Then,

For all $j \in S$, B_{1j} consists only of positive terms in the variables a_{ij}'s. Similarly, $\det B$ *consists only of positive terms in these variables.*

Proof: Let $\mathbf{P} = (\pi_1, \pi_2, \ldots, \pi_m)$ be the steady state probability distribution vector on the states of the system. We should note that since $x(t)$ is an irreducible, stationary

Markov process defined over a finite state space, the vector $\mathbf{P}$ is unique. Let $\mathbf{e_1}$ be a vector with 1 in the first position and zero everywhere else. Then $\mathbf{P}$ should satisfy the following relation:

$$\mathbf{BP} = \mathbf{e_1}. \tag{2.4}$$

We can write

$$\pi_1 = \frac{B_{11}}{\det B}. \tag{2.5}$$

We should also note that since $\mathbf{P}$ is unique and satisfies (2.4), $\det B$ cannot be zero.

We can easily verify that the matrix obtained by crossing out the first row and column of $\mathbf{B}$ is an element of F_{m-1} and hence from Lemma 2.1 its determinant consists only of positive terms in the variables. Therefore, B_{11} consists only of positive terms.

We will prove now that the determinant of $\mathbf{B}$ also consists only of positive terms. Suppose we represent all the transition rates by different variables. Then each variable will occur only in one column. Therefore in the determinant of the matrix $\mathbf{B}$, no term will consist of a variable raised to a power higher than one. All terms will have the variables raised either to power zero or one.

Suppose that the determinant consists of negative terms also. Take one of them. Let the variables appearing in this term tend to infinity and the remaining variables tend to zero. Then the contribution to the determinant will be dominated by this term and will become negative. However, B_{11} will be positive, whatever be the value of the variables. (We should note that the variables can take only non-negative values.) Hence from (2.5) we see that we will get a negative value for π_1 which is not possible as π_1 is the probability of the system being in a state. Hence our assumption is wrong. Hence $\det B$ consists only of positive terms.

We can go even further and say that each of the cofactors B_{1j} also consists only of positive terms. This is because we can write for all $j \in S$,

$$\pi_j = \frac{B_{1j}}{\det B}. \tag{2.6}$$

Since each variable occurs only in one column, no term in B_{1j} will consist of a variable raised to a power higher than one. All terms will have the variables raised either to power zero or one. Suppose B_{1j} had negative terms. Then take one of those negative terms. Let all the variables which appear in this term tend to infinity and let all the other variables tend to zero. Then B_{1j} will become negative. However, as proved above, since $\det B$ consists only of positive terms, it will always be positive. Hence, from (2.6) we see that we will have a negative value for π_j which is impossible. Hence, B_{1j} also consists only of positive terms.

Hence, for all $j \in S$, B_{1j} as well as $\det B$ consist only of positive terms.

Corollary: If we write

$$\pi_j = \frac{N_j}{D_j},$$

where N_j and D_j are polynomials in the variables a_{ij}'s, then for all $j \in S$, N_j and D_j consist only of positive terms.

∎

Lemma 2.3 Let $f(y_1, y_2, \ldots, y_p)$ be a polynomial with positive coefficients. Let $I_p = \{1, 2, \ldots, p\}$. For $i \in I_p$, let $x_i(n), x_i'(n)$ be variables that are allowed to take only positive values. Here n is real. We will denote $f(x_1(n), x_2(n), \ldots, x_p(n))$ by $f(x(n))$ and $f(x_1'(n), x_2'(n), \ldots, x_p'(n))$ by $f(x'(n))$.
If for all $i \in I_p$,

$$\lim_{n\to\infty} \frac{x_i(n)}{x_i'(n)} = 1, \tag{2.7}$$

then

$$\lim_{n\to\infty} \frac{f(x(n))}{f(x'(n))} = 1.$$

Proof: We can write,

$$x_i'(n) = x_i(n)(1 + \epsilon_i(n)) \qquad \text{for all } i \in I_p. \tag{2.8}$$

Here for all $i \in I_p$,

$$\lim_{n\to\infty} \epsilon_i(n) = 0. \tag{2.9}$$

Suppose $f(x(n))$ has m terms and let $t_k(n)$ denote the k^{th} term in $f(x(n))$. Let $t'_k(n)$ be the corresponding term in $f(x'(n))$. Let the variables occurring in the expression for $t_k(n)$ be $x_{k_1}(n), x_{k_2}(n), \ldots, x_{k_{l_k}}(n)$. Then the variables occurring in the expression for $t'_k(n)$ are $x'_{k_1}(n), x'_{k_2}(n), \ldots, x'_{k_{l_k}}(n)$. Let

$$t_k(n) = a_k \prod_{j=1}^{l_k} x_{k_j}^{p_{k_j}}(n),$$

where a_k is a positive number and p_{k_j}'s are positive integers for all j from 1 to l_k. Then we can write

$$\begin{aligned} t'_k(n) &= a_k \prod_{j=1}^{l_k} {x'_{k_j}}^{p_{k_j}}(n) \\ &= a_k \prod_{i=1}^{l_k} x_{k_i}^{p_{k_i}}(n) \prod_{j=1}^{l_k} (1 + \epsilon_{k_j}(n))^{p_{k_j}} \\ &= t_k(n) \prod_{j=1}^{l_k} (1 + \epsilon_{k_j}(n))^{p_{k_j}} \\ &= t_k(n)(1 + \epsilon'_k(n)). \end{aligned}$$

Here,

$$\epsilon'_k(n) = \prod_{j=1}^{l_k} (1 + \epsilon_{k_j}(n))^{p_{k_j}} - 1,$$

and since from (2.9), for all $i \in I_p$,

$$\lim_{n \to \infty} \epsilon_i(n) = 0,$$

we can say that

$$\lim_{n \to \infty} \epsilon'_k(n) = 0. \tag{2.10}$$

Also we should note that for all $k \in I_m$ and for all positive values of the variables $x_1(n), x_2(n), \ldots, x_p(n)$,

$$\left|\frac{t_k(n)}{f(x(n))}\right| \leq 1. \tag{2.11}$$

Therefore,

$$
\begin{aligned}
f(x'(n)) &= \sum_{k=1}^{m} t'_k(n) \\
&= \sum_{k=1}^{m} (t_k(n)(1+\epsilon'_k(n))) \\
&= (\sum_{k=1}^{m} t_k(n))(1+\sum_{k=1}^{m} \frac{t_k(n)}{\sum_{k=1}^{m} t_k(n)}\epsilon'_k(n)) \\
&= f(x(n))(1+\sum_{k=1}^{m} \frac{t_k(n)}{f(x(n))}\epsilon'_k(n)).
\end{aligned}
$$

Let

$$\epsilon'_{max}(n) = \max(|\epsilon'_k(n)| : k = 1, 2, \ldots, m).$$

Then

$$\lim_{n\to\infty} \epsilon'_{max}(n) = 0. \tag{2.12}$$

Hence we can write

$$
\begin{aligned}
(1-\sum_{k=1}^{m} |\epsilon'_k(n)|) &\le \tfrac{f(x'(n))}{f(x(n))} &\le (1+\sum_{k=1}^{m} |\epsilon'_k(n)|), \\
(1-m|\epsilon'_{max}(n)|) &\le \tfrac{f(x'(n))}{f(x(n))} &\le (1+m|\epsilon'_{max}(n)|).
\end{aligned}
$$

Hence, from (2.12), we have

$$\lim_{n\to\infty} \frac{f(x'(n))}{f(x(n))} = 1.$$

∎

Proof of Theorem 2.1: The proof of Theorem 2.1 now follows from the corollary to Lemma 2.2 and Lemma 2.3.

∎

2.3 A Theorem About Stationary "Almost Markov" Processes

Theorem 2.2 Consider a discrete-space, continuous time stochastic process $x(t)$. Let $I_m = \{1, 2, \ldots, m\}$ be the set of states which the system can be in. Suppose the process $x(t)$ satisfies the following properties:
(1) For all $i, j \in I_m, i \neq j$,

$$\Pr(x(t+h) = j \mid x(t) = i) = \lambda_{ij}(t)h + o(h), \tag{2.13}$$

and for all $i \in I_m$,

$$\Pr(x(t+h) = i \mid x(t) = i) = 1 - \sum_{j \neq i} \lambda_{ij}(t)h + o(h). \tag{2.14}$$

(2) For all $i, j \in I_m, i \neq j$, if $\lambda_{ij}(t) \neq 0$, then for all t,

$$0 < \lambda_L \leq \lambda_{ij}(t) \leq \lambda_U. \tag{2.15}$$

(3) Suppose that for all $i, j \in I_m$, if $\lambda_{ij}(t) \neq 0$ for any t, then it is not zero for all t sufficiently large. For all $\lambda_{ij}(t) \neq 0$ there exist λ_{ij} such that

$$\lim_{t \to \infty} \frac{\lambda_{ij}(t)}{\lambda_{ij}} = 1. \tag{2.16}$$

For i, j such that $\lambda_{ij}(t) = 0$, let

$$\lambda_{ij} = 0.$$

(4) Let $P_i(t) = \Pr(x(t) = i)$ for all $i \in I_m$. For all $i \in I_m$,

$$\lim_{t \to \infty} P_i(t) = P_i. \tag{2.17}$$

The steady-state probability distribution $\{P_i\}$ on I_m of the process $x(t)$ satisfying the above properties is the same as that of a Markov process defined on the state-space

I_m and which has transition rate from i to j given by λ_{ij}.

Proof: From (2.17), we can write that for all $i \in I_m$,

$$\lim_{t \to \infty} P_i'(t) = 0. \tag{2.18}$$

We can write for all $j \in I_m$,

$$\begin{aligned} P_j(t+h) &= \sum_{i \in I_m} \Pr(x(t+h) = j \mid x(t) = i) \Pr(x(t) = i) \\ &= (1 - \sum_{i \neq j} \lambda_{ji} h) P_j(t) + \sum_{i \neq j} \lambda_{ij}(t) h P_i(t) + o(h) \quad \text{from (2.13) and (2.14).} \end{aligned}$$

Following the usual process of transferring $P_j(t)$ from the right to the left, dividing by h, and taking the limit as h approaches zero, we get

$$P_j'(t) = -\sum_{i \neq j} \lambda_{ji}(t) P_j(t) + \sum_{i \neq j} \lambda_{ij}(t) P_i(t). \tag{2.19}$$

Consider a $m \times m$ matrix $\mathbf{A}(t)$ defined by

$$a_{ij}(t) = \begin{cases} -\lambda_{ji}(t), & \text{if } i \neq j; \\ \sum_{j \neq i} \lambda_{ij}(t) & \text{otherwise.} \end{cases}$$

Let $\mathbf{B}(t)$ be the matrix obtained by replacing the first row of $\mathbf{A}(t)$ by an all one vector. Let us define two matrices $\mathbf{P}(t)$ and $\mathbf{C}(t)$ as follows :

$$\mathbf{P}(t) = \begin{pmatrix} P_1(t) \\ P_2(t) \\ \vdots \\ P_m(t) \end{pmatrix} \quad \text{and} \quad \mathbf{C}(t) = \begin{pmatrix} 1 \\ -P_2'(t) \\ -P_3'(t) \\ \vdots \\ -P_m'(t) \end{pmatrix}.$$

Then from (2.19) we can write

$$\mathbf{B}(t)\mathbf{P}(t) = \mathbf{C}(t). \tag{2.20}$$

Let $B_{ij}(t)$ denote the cofactor corresponding to the element $b_{ij}(t)$ in $\mathbf{B}(t)$. Let $\mathbf{B}$ be the matrix obtained from $\mathbf{B}(t)$ by replacing $\lambda_{ij}(t)$ by λ_{ij} and let B_{ij} be the cofactor corresponding to the element b_{ij} of $\mathbf{B}$.

Since each $B_{ij}(t)$ is a polynomial in the $\lambda_{ij}(t)$'s, from (2.15) we can say that there exists an M_1 such that for all $i, j \in I_m$,

$$|B_{ij}(t)| < M_1. \tag{2.21}$$

Also, since $\mathbf{B}(t)$ and $\mathbf{B}$ are similar to the matrix discussed in Lemma 2.2 in section 2.2, $\det B(t)$, $\det B$, $B_{1j}(t)$ and B_{1j} (for all $j \in I_m$) consist only of positive terms. We can say the following:

(1) There exists an M_2 such that

$$|\det B(t)| \geq M_2 > 0. \tag{2.22}$$

(2) Since $B_{1j}(t), B_{1j}$ consist only of positive terms, from (2.16) and Lemma 2.3, we have, for all $j \in I_m$,

$$\lim_{t\to\infty} \frac{B_{1j}(t)}{B_{1j}} = 1. \tag{2.23}$$

(3) Since $\det B(t), \det B$ consists only of positive terms, from (2.16) and Lemma 2.3,

$$\lim_{t\to\infty} \frac{\det(B(t))}{\det(B)} = 1. \tag{2.24}$$

Writing $P_j(t)$ in terms of the $\lambda_{ij}(t)$'s and the $P_j'(t)$'s, from (2.20) we get

$$P_j(t) = \frac{(B_{1j}(t) - \sum_{i\neq 1} B_{ij}(t)P_i'(t))}{\det(B(t))}. \tag{2.25}$$

From (2.21) and (2.22), we can say that for all $i, j \in I_m$,

$$\frac{-M_1}{M_2}|P_i'(t)| \leq \frac{B_{ij}(t)P_i'(t)}{\det(B(t))} \leq \frac{M_1}{M_2}|P_i'(t)|.$$

Since

$$\lim_{t\to\infty} P_i'(t) = 0$$

we can say that

$$\lim_{t\to\infty} \frac{B_{ij}(t)P_i'(t)}{\det(B(t))} = 0. \tag{2.26}$$

Therefore,

$$\begin{aligned} P_j &= \lim_{t\to\infty} P_j(t) \\ &= \lim_{t\to\infty} \frac{B_{1j}(t)}{\det(B(t))} && \text{(from (2.25) and (2.26))} \\ &= \frac{B_{1j}}{\det(B)} && \text{(from (2.23) and (2.24)).} \end{aligned}$$

However this is exactly the steady-state probability of being in state j of a Markov process which is defined over the same state space as the process $x(t)$ and whose transition rate from state i to state j is given by λ_{ij}. This completes the proof.

∎

Chapter 3 The Timid DCAA

3.1 Introduction

In this chapter, we shall analyze the performance, in the limiting case of a large number of channels, of a distributed dynamic channel assignment algorithm which we call the Timid Dynamic Channel Assignment Algorithm (TDCAA). The algorithm is as follows:

TDCAA: All channels are available for use in all the cells. When a call comes to a cell, one of the available channels is picked up at random and is given to the call. If there is no available channel, the call is blocked. Call rearrangements are not allowed.

Conjecture : The TDCAA is asymptotically optimal.

We can prove the conjecture assuming a certain plausible mathematical hypothesis, and we have strong experimental evidence to support our conjecture in many special cases.

This is very surprising as there are some highly complex algorithms which are provably not asymptotically optimal. For example, the "Greedy" algorithm, which is identical to the TDCAA except that it allows call rearrangements in order to accept a new call, does not have the property of asymptotic optimality.

3.2 Model and Definitions

We assume that there is a finite set of N cells. The N cells share a common set of n channels. The offered traffic in each cell is described by a simple Poisson birth-death process [12], which is independent from cell to cell. (In practice, however, we should note that handoffs between cells make the traffic among various cells dependent.) Let $I_N = \{1, 2, \ldots, N\}$ be the set of cells in the system. The rate of call request arrivals in cell i is $\lambda_i n$ per second and the rate of call departures is μ per second. Thus the offered traffic intensity in cell i is $r_i = \lambda_i/\mu$ Erlangs per channel. The total offered traffic is $r = \sum_i r_i$ Erlangs per channel. The ratio $p_i = r_i/r$ is the fraction of the total offered traffic present in cell i. The vector $p = (p_1, p_2, \ldots, p_N)$ is called the traffic pattern.

We define the state j of a channel by an N-tuple $a_j = (a_{1,j}, a_{2,j}, \ldots, a_{N,j})$ with $a_{i,j} = 1$ if the channel when it is in state j carries a call in cell i and $a_{i,j} = 0$ if it does not. Although there are 2^N possible channel states, only a subset of these will be allowable, because of the channel "reuse constraints". Let $\Omega = \{1, 2, \ldots, m\}$ denote the set of allowable channel states. The only restriction we impose on the reuse constraints is that the corresponding set of allowable states Ω is closed under the operation of changing a 1 in the state vector a_j corresponding to a $j \in \Omega$ to a 0, i.e., removing a caller from the system.

Let Ω_i be the set of channel states such that only if a channel is in a state $j \in \Omega_i$ that $a_j + e_i \in \Omega$, where e_i is an N-tuple with a 1 in the i^{th} position and a 0 everywhere else. That is, $\Omega_i \subset \Omega$ such that only if a channel is in a state which is in Ω_i, the channel can accept a call in cell i without violating the reuse constraints.

Example: Consider the linear array of three cells shown in Figure 1.1. Suppose the reuse constraint is that the same channel cannot be used in adjacent cells simultaneously. The state diagram for a single channel is shown is Figure 1.2. We have,

$$
\begin{aligned}
\Omega &= \{(0,0,0),(1,0,0),(0,0,1),(0,1,0),(1,0,1)\}, \\
\Omega_1 &= \{(0,0,0),(0,0,1)\}, \\
\Omega_2 &= \{(0,0,0)\}, \\
\Omega_3 &= \{(0,0,0),(1,0,0)\}.
\end{aligned}
$$

We will discuss a method for analyzing the asymptotic performance of the TDCAA i.e., the performance of the TDCAA as the number of channels tends to infinity but the offered traffic per channel in each cell remains constant. In this case, it is clear that as all the channels are treated in the same manner, we can focus our attention on one particular channel, which we call β, and study its behavior. We will refer to the process described by it as $x(t)$. Let S denote the system of n channels and S' denote the system of $(n-1)$ channels excluding channel β. Let π_j be the probability of channel β being in a state $j \in \Omega$. Let S_i' be the set of all channels from S' whose states are in Ω_i at any given time. It should be noted that the elements of S_i' vary with time.

The single-channel system, containing channel β, will be equivalent to the system of n channels in the sense that the carried traffic and the blocking probability in different cells in the single-channel system will be the same as the carried traffic per channel and the blocking probability in the corresponding cells in the system of n channels.

3.3 The Process $x(t)$

We should note that the process $x(t)$ described by the single channel β is not a Markov process. Consider two states $j, k \in \Omega$. Suppose $a_k = a_j + e_i$ for some $i \in I_N$, where e_i is a vector of length N with a 1 in the i^{th} position and 0 everywhere else. The transition rate from state j to state k, a transition caused by an arrival in cell i, is a function of the state of the system S'. The state of S' is not independent of the past history of channel β. Hence the transition rate from j to k, where $a_k = a_j + e_i$, is a function of the past history of the channel. So the process described by it is not Markov. Appendix B explains in detail why the process is not Markov. Since the entire system describes a Markov process, each channel has an equilibrium distribution. In the same Appendix, we prove, using Theorem 2.2 in Chapter 2, that this equilibrium distribution is the same as that of a Markov process in which the transition rate between two states is taken to be the transition rate between the corresponding states of $x(t)$ conditioned only on the present state of the system. From now on, when we refer to the transition rate between two states of $x(t)$, we will mean the transition rate conditioned only on the present state of the system and we will treat the system $x(t)$ as being Markov.

When a channel is in a state $j \in \Omega_i$, let us denote the rate at which calls arrive to it in cell i by $\lambda'_{i,j}$, i.e., the transition rate from state j to state k where $a_k = a_j + e_i$ is $\lambda'_{i,j}$. (We don't talk about $\lambda'_{i,j}$ when a channel is in a state $j \notin \Omega_i$, because in this case, the algorithm does not allow the channel to accept any call in cell i.) The call departure rate is independent of the state of the channel and the cell in which the call is in progress. We define the **apparent traffic** in cell i for the system $x(t)$ when it is in state $j \in \Omega_i$ by $r'_{i,j} = \lambda'_{i,j}/\mu$. Once we know the $\lambda'_{i,j}$'s for all $i \in I_N$ and all $j \in \Omega_i$, we can analyze the performance of the system of n channels by studying the behavior of a system with a single channel β and which has offered traffic in cell $i \in I_N$ when it is in state $j \in \Omega_i$ given by $r'_{i,j} = \lambda'_{i,j}/\mu$.

In general, $\lambda'_{i,j}$ is different from λ_i. Let $P'_{i,j}$ be the probability that when a call comes to cell i, it is assigned channel β given that channel β is in a state $j \in \Omega_i$. We can write

$$\lambda'_{i,j} = \lambda_i n P'_{i,j}. \tag{3.1}$$

Let $\nu'_{i,j}(l)$ be the probability that $|S'_i| = l$ given that channel β is in state j. Then we can write

$$\begin{aligned}
P'_{i,j} &= \Pr(\text{A call in cell } i \text{ is assigned channel } \beta \mid \text{channel } \beta \text{ is in a state } j \in \Omega_i) \\
&= \sum_{l=0}^{n-1} \Pr(\text{A call in cell } i \text{ is assigned channel } \beta \mid \text{channel } \beta \text{ is in a state} \\
&\quad j \in \Omega_i, |S'_i| = l) \times \Pr(|S'_i| = l \mid \text{channel } \beta \text{ is in a state } j \in \Omega_i) \\
&= \sum_{l=0}^{n-1} \Pr(\text{A call in cell } i \text{ is assigned channel } \beta \mid \text{channel } \beta \text{ is in a state} \\
&\quad j \in \Omega_i, |S'_i| = l) \times \nu'_{i,j}(l).
\end{aligned} \tag{3.2}$$

Now, when a call comes to cell i, the algorithm picks one of the available channels at random and assigns it to the call. Hence we can say that,

$$\begin{aligned}
\Pr(\text{A call in cell } i \text{ is assigned channel } \beta \quad &| \quad \text{channel } \beta \text{ is in a state } j \in \Omega_i, |S'_i| = l) \\
&= \frac{1}{1+l}.
\end{aligned} \tag{3.3}$$

Therefore, from (3.2) and (3.3), we have

$$P'_{i,j} = \sum_{l=0}^{n-1} \frac{1}{1+l} \nu'_{i,j}(l). \tag{3.4}$$

In general, as is clear from (3.1) and (3.4), $\lambda'_{i,j}$ will be a function of j, i.e.,

$$\lambda'_{i,j} \neq \lambda'_{i,k} \qquad \text{for} \qquad j, k \in \Omega_i \qquad \text{and} \qquad j \neq k.$$

3.4 Hypothesis and the Resulting Reversibility

Hypothesis: We make the following assumption. Let $\nu'_i(l)$ denote the probability that $|S'_i| = l$, i.e., the probability that the number of channels in the system S' which can accept a call in cell i is l. Then we **assume** that, when the TDCAA is used, for all $i \in I_N$, all $j \in \Omega$, and for $l_n = 0, 1, \ldots, n-1$,

$$\lim_{n\to\infty} \frac{\nu'_{i,j}(l_n)}{\nu'_i(l_n)} = 1.$$

In other words we assume that, when the TDCAA is used, as the number of channels tends to infinity, the influence of a particular channel on the state of the rest of the system becomes arbitrarily small.

Then we can write for all $i \in I_N$, all $j \in \Omega$, and for $l_n = 0, 1, \ldots, n-1$,

$$\nu'_{i,j}(l_n) = \nu'_i(l_n)(1 + \xi_{i,j}), \tag{3.5}$$

where

$$\lim_{n\to\infty} \xi_{i,j} = 0.$$

Let

$$\xi_{max} = \max(|\xi_{i,j}| : i \in I_N, j \in \Omega). \tag{3.6}$$

Then we can write

$$\lim_{n\to\infty} \xi_{max} = 0. \tag{3.7}$$

Let us define P'_i as

$$P'_i = \sum_{l=o}^{n-1} \frac{1}{1+l}\nu'_i(l). \tag{3.8}$$

Therefore, from (3.4),(3.5),(3.6) and (3.7), we have,

$$\begin{aligned}
\sum_{l=0}^{n-1} \frac{1}{1+l}\nu'_i(l)(1-|\xi_{i,j}|) &\leq P'_{i,j} &&\leq \sum_{l=0}^{n-1} \frac{1}{1+l}\nu'_i(l)(1+|\xi_{i,j}|), \\
\sum_{l=0}^{n-1} \frac{1}{1+l}\nu'_i(l)(1-|\xi_{max}|) &\leq P'_{i,j} &&\leq \sum_{l=0}^{n-1} \frac{1}{1+l}\nu'_i(l)(1+|\xi_{max}|), \\
P'_i(1-|\xi_{max}|) &\leq P'_{i,j} &&\leq P'_i(1+|\xi_{max}|).
\end{aligned}$$

Hence, for all $i \in I_N$ and all $j \in \Omega_i$,

$$\lim_{n\to\infty} \frac{P'_{i,j}}{P'_i} = 1. \tag{3.9}$$

Now consider two single-channel cellular systems X_n and X'_n with the same set of cells I_N and the same set of allowable states Ω. Suppose that the two systems describe Markov processes. Let the arrival rate in cell i when the system X_n is in a state $j \in \Omega_i$ be $\lambda'_{i,j} = \lambda_i n P'_{i,j}$ and the corresponding rate for X'_n be $\lambda'_i = \lambda_i n P'_i$. The call departure rate is μ for both the systems. We should note that the system X_n is the same as the single-channel system with channel β described earlier. As is clear from (3.9), the two systems X_n and X'_n describe two Markov processes whose transition rates tend to each other as $n \to \infty$. Let $\mathbf{P} = (\pi_1, \pi_2, \ldots, \pi_m)^T$ be the equilibrium distribution vector for the first process and let $\mathbf{P}' = (\pi'_1, \pi'_2, \ldots, \pi'_m)^T$ be the equilibrium distribution vector for the second one. Since X_n is the same as the single-channel system with channel β, this is consistent with the definition of π_j as the probability of channel β being in state j. Then from Theorem 2.1 in Chapter 2, we can say that for all $j \in \Omega$,

$$\pi_j = \pi'_j(1+\psi_j),$$

where

$$\lim_{n\to\infty} \psi_j = 0.$$

Hence,

$$\pi_j = \pi'_j + \zeta_j, \tag{3.10}$$

where

$$\lim_{n\to\infty} \zeta_j = 0.$$

Let ζ be a $m \times 1$ matrix $(\zeta_1, \zeta_2, \ldots, \zeta_m)^T$. Then we can write

$$\mathbf{P} = \mathbf{P}' + \zeta. \tag{3.11}$$

Now from the discussion on reversibility in the Appendix C, it is clear that the process X'_n is reversible.

Thus we have used Theorem 2.1, Theorem 2.2 and the hypothesis to reduce the problem of analyzing the single-channel system containing channel β to that of a system X'_n which describes a reversible process, where the two systems are equivalent in the sense that as $n \to \infty$, the two systems have the same equilibrium distribution. Hence from now on , we will focus our attention on the process X'_n.

3.5 The Process X'_n

Now we will relate the traffic in the cells of the system X'_n to the blocking probabilities in the corresponding cells of the system S.

Let the offered traffic in cell i for the system X'_n be denoted by r'_i. We can write

$$\begin{aligned} r'_i &= \frac{\lambda'_i}{\mu} \\ &= \frac{\lambda_i n P'_i}{\mu} \\ &= r_i n P'_i. \end{aligned} \tag{3.12}$$

For all $i \in I_N$, let

$$P'_i = \frac{1}{n^{\delta_i}}, \qquad 0 < \delta_i < 1. \tag{3.13}$$

Then, from (3.12) and (3.13), we have

$$\begin{aligned} r_i' &= r_i n^{1-\delta_i} \\ &= r_i n^{f_i}, \end{aligned} \tag{3.14}$$

where for all $i \in I_n$,

$$f_i = 1 - \delta_i. \tag{3.15}$$

Let δ , $\mathbf{f}$, and $\mathbf{J}$ be $N \times 1$ matrices defined as follows:

$$\delta = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_N \end{pmatrix}, \qquad \mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}, \qquad \text{and} \qquad \mathbf{J} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}.$$

From (3.15), we have

$$\delta + \mathbf{f} = \mathbf{J}. \tag{3.16}$$

Now we state a very important result relating the traffic r_i' in cell i for the system X_n' to the blocking probability in cell i, denoted by P_{b_i}, for the system S.

Let δ_{th} and P_{th} be arbitrary positive numbers with $0 < P_{\text{th}} < 1$. Let $f_{\text{th}} = 1 - \delta_{\text{th}}$. Let us define sets Γ and Γ' as follows:

$$\Gamma = \{i : i \in \ I_N, \delta_i > \delta_{\text{th}}\},$$

$$\Gamma' = I_N \backslash \Gamma.$$

Lemma A.1 For all $i \in \Gamma$ there exists an n_0 such that for all $n > n_0$,

$$P_{b_i} < P_{\text{th}}.$$

We can also say that for large enough n, for all $i \in \Gamma$,

$$P_{b_i} < P_{\text{th}}.$$

The proof is given in Appendix A.

Let $A = (a_{i,j})$ be the incidence matrix defined by

$$a_{i,j} = \begin{cases} 1, & \text{if channel } \beta \text{ carries a call in cell } i \text{ when it is state } j; \\ 0, & \text{otherwise.} \end{cases}$$

Since the process X'_n is reversible, its equilibrium distribution can be written down in terms of its transition probabilities as follows (see Appendix C):

$$\begin{aligned} \pi'_j &= \prod_{i:a_{i,j}=1} r'_i \pi'_1, \\ &= v_j n^{w_j} \pi'_1 \qquad (3.17) \\ &= v_j n^{w_j} \Big/ \left(\sum_{l \in \Omega} v_l n^{w_l} \right), \qquad (3.18) \end{aligned}$$

where $v_j = \prod_{i:a_{i,j}=1} r_i$, $w_j = \sum_{i:a_{i,j}=1} f_i$ and π'_1 is the probability of the system X'_n being in the all-zero state.

We call w_j the weight of state j. Let $\mathbf{w}$ be an $m \times 1$ matrix defined as follows:

$$\mathbf{w} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix}.$$

We can easily see that

$$\mathbf{w}^T = \mathbf{f}^T \mathbf{A}. \qquad (3.19)$$

Let $w_{\max} = \max(w_j : j \in \Omega)$. Let ϵ_1, ϵ_2 be arbitrary positive numbers.

Now we state another important result relating the weights of the states of X'_n to its probability distribution.

Lemma A.2 There exists an n_1 such that for all $n > n_1$ and for all $j \in \Omega$,

$$\text{if} \qquad (w_{\max} - w_j) > \epsilon_1, \qquad \text{then} \qquad \pi'_j < \epsilon_2.$$

That is, if n is large enough, the above result holds. The proof is given in Appendix A.

Lemma A.2 that as the number of channels tends to infinity, most of the probability lies with states which have their weights in a given neighborhood of the maximum weight. This neighbourhood can be made arbitrarily small and the probability given to the remaining states can also be made arbitrarily small by making ϵ_1 and ϵ_2 sufficiently small. This is because the number of allowable channel states m is independent of n .

3.6 Proof that the Timid DCAA is Asymptotically Optimal under the Hypothesis

We should recall that the equilibrium distribution for the single channel system containing channel β when the Timid DCAA is used is denoted by $\mathbf{P} = (\pi_1, \pi_1, \ldots, \pi_m)^T$. Let the carried traffic vector be denoted by $\kappa = (\kappa_1, \kappa_2 \ldots, \kappa_N)^T$. Let κ_i denote the i^{th} component of κ. Let $(\mathbf{AP})_i$ denote the i^{th} component of the $N \times 1$ vector $\mathbf{AP}$.

We can write

$$\begin{aligned} \kappa &= \mathbf{AP} \\ &= \mathbf{AP}' + \mathbf{A}\zeta \qquad\qquad \text{(from (3.11))}. \end{aligned}$$

Since offered traffic in any cell is greater than or equal to the carried traffic in that cell, we have, for all $i \in I_n$,

$$r_i \geq \kappa_i = (\mathbf{AP})_i.$$

Also, $(r_i - (\mathbf{AP})_i)$ gives the blocked traffic per channel in cell i when this algorithm is used. Let Δ be the total blocked traffic per channel when the distribution is $\mathbf{P}$.

Then from Lemma A.3 in Appendix A,

$$\begin{aligned}\Delta &= \sum_{i=1}^{N}(r_i - (\mathbf{AP})_i) &(3.20)\\ &= \mathbf{J}^T[\mathbf{r} - \mathbf{AP}] &(3.21)\\ &\leq \mathbf{f}^T[\mathbf{r} - \mathbf{AP}'] + \delta_{\text{th}} r + P_{\text{th}} r - \mathbf{f}^T[\mathbf{A}\zeta], &(3.22)\end{aligned}$$

where $r = \sum_{i \in I_N} r_i$ is the total offered traffic per channel.

Consider a distribution $\mathbf{P}''$ on Ω due to some other algorithm A . Let the carried traffic when this algorithm is used be given by an $N \times 1$ matrix κ'' . Let κ_i'' denote the i^{th} component of κ''. Let $(\mathbf{AP}'')_i$ denote the i^{th} component of the $N \times 1$ vector $\mathbf{AP}''$.

Since offered traffic in any cell is greater than or equal to the carried traffic in that cell, $\mathbf{P}''$ also must satisfy the following relation for all $i \in I_N$:

$$r_i \geq \kappa_i'' = (\mathbf{AP}'')_i.$$

Also, $(r_i - (\mathbf{AP}'')_i)$ gives the total blocked traffic in cell i when the algorithm A is used. Let Δ'' be the total blocked traffic per channel when algorithm A is used. We have

$$\begin{aligned}\Delta'' &= \sum_{i=1}^{N}(r_i - (\mathbf{AP}'')_i)\\ &= \mathbf{J}^T[\mathbf{r} - \mathbf{AP}'']. &(3.23)\end{aligned}$$

From (3.21), (3.23) and (3.22), we have

$$\begin{aligned}\Delta - \Delta'' &= \mathbf{J}^T[\mathbf{r} - \mathbf{AP}] - \mathbf{J}^T[\mathbf{r} - \mathbf{AP}'']\\ &\leq \mathbf{f}^T[\mathbf{r} - \mathbf{AP}'] + \delta_{\text{th}} r + P_{\text{th}} r - \mathbf{J}^T[\mathbf{r} - \mathbf{AP}''] - \mathbf{f}^T[\mathbf{A}\zeta]. &(3.24)\end{aligned}$$

We should note that $f_i \leq 1$ for all $i \in I_N$. Therefore,

$$\mathbf{f}^T[\mathbf{r} - \mathbf{AP}''] \leq \mathbf{J}^T[\mathbf{r} - \mathbf{AP}''].\tag{3.25}$$

Hence, from (3.24) and (3.25), we have

$$\begin{aligned}\Delta - \Delta'' &\leq \mathbf{f}^T[\mathbf{r} - \mathbf{AP}'] - \mathbf{f}^T[\mathbf{r} - \mathbf{AP}''] + \delta_{\text{th}}r + P_{\text{th}}r - \mathbf{f}^T[\mathbf{A}\zeta] \\ &= -\mathbf{f}^T\mathbf{AP}' + \mathbf{f}^T\mathbf{AP}'' + \delta_{\text{th}}r + P_{\text{th}}r - \mathbf{f}^T[\mathbf{A}\zeta].\end{aligned}\tag{3.26}$$

From Lemma A.4 in Appendix A, we have

$$\mathbf{f}^T\mathbf{AP}'' \leq w_{\max},\tag{3.27}$$

and from Lemma A.5 in Appendix A,

$$\mathbf{f}^T\mathbf{AP}' \geq (w_{\max} - \epsilon_1)(1 - m\epsilon_2).\tag{3.28}$$

Therefore, from (3.26), (3.27) and (3.28), we have

$$\begin{aligned}\Delta - \Delta'' &\leq w_{\max} - (w_{\max} - \epsilon_1)(1 - m\epsilon_2) + \delta_{\text{th}}r + P_{\text{th}}r - \mathbf{f}^T[\mathbf{A}\zeta] \\ &= w_{\max}m\epsilon_2 + \epsilon_1(1 - m\epsilon_2) + \delta_{\text{th}}r + P_{\text{th}}r - \mathbf{f}^T[\mathbf{A}\zeta].\end{aligned}$$

Now ϵ_1 , ϵ_2 , δ_{th}, P_{th} are arbitrary positive numbers which can be made as small as we please. Also the vector ζ tends to $(0, 0, \ldots, 0)$ as n tends to infinity. Therefore, $\Delta - \Delta'' \to 0$ as $n \to \infty$.

Hence, we conclude that the **TDCAA is asymptotically optimal under the stated hypothesis**.

Example: Consider the linear array of three cells whose state diagram is given in Figure 1.2. Assume that the offered traffic is uniform in all the three cells. Figure 1.3 shows the upper bound [18] on the performance, measured by the carried traffic per channel, for this system for different values of total offered traffic. Assuming $n =$

10000 channels, we simulated the performance of the TDCAA for several values of the total offered traffic. The simulation results are shown in Figure 1.3. The simulation strongly supports our conjecture that the TDCAA is asymptotically optimal.

3.7 Conclusion

In this chapter, we analyzed the performance of the TDCAA, in the limiting case of a large number of channels. We showed, under a plausible mathematical hypothesis, that the performance of the algorithm, measured by the carried traffic per channel, is at least as good as that of any other algorithm, in the limiting case of a large number of channels. This is very surprising, as there are algorithms of very high complexity which provably do not have this property. We believe that the TDCAA requires a large number of channels before it achieves the upper bound. One of the main reasons for this is that it selects a channel randomly from the set of available channels. We would like to comment here that this property of the TDCAA was chosen more because it made it easy to analyze the asymptotic performance of the algorithm than to make the algorithm simple. As will be described in Chapter 4, there is a slight modification of the TDCAA which does away with the random selection of a channel from the set of available channels, but which continues to retain the simplicity of the TDCAA. This modified algorithm can be expected to give a good performance even with a small number of channels.

Chapter 4 Some Channel Assignment Algorithms

4.1 Introduction

In this chapter, we compare the performance of three different types of algorithms including the TDCAA. We will show that the TDCAA performs better than a very complex algorithm when we have uniform traffic distribution in a linear array of three cells and try to understand why this is the case. Although we conjecture that the TDCAA is asymptotically optimal, the number of channels required before its performance becomes "good" is very large. In this chapter, we introduce another algorithm which we call the "Modified" DCAA. The advantage with this algorithm is that while retaining the simplicity of the TDCAA, we can expect it to give a "good" performance even with a small number of channels.

4.2 Comparison of CAAs

(1) Fixed Channel Assignment Algorithm (FCAA): In FCAA, each cell is given a fixed number of channels. When a call comes to a cell, if one of the channels which has been assigned to it is free, the call is accepted. Otherwise the call is blocked. It has been shown that for any given traffic distribution, there exists a fixed channel assignment which is asymptotically optimal [18]. However the problem with FCAA is that we require a knowledge of the traffic distribution for the algorithm to be asymptotically optimal. The traffic distribution may not always be available, and it may change with time.

(2) Greedy Dynamic Channel Assignment Algorithm (GDCAA): It is also known as the *maximum packing algorithm* [6]. In GDCAA, all channels are available for use in all the cells. When a call comes to a cell, the call is accepted if there is a rearrangement of the calls in progress which allows that call to be accepted without causing any of the calls which were in progress to be dropped. If there is no such rearrangement of the calls in progress which will allow that new call to be accepted, the new call is blocked. From the way the algorithm works, we should expect GDCAA to give a near optimal performance. It can be shown that asymptotically, the algorithm does track the carried traffic upper limit up to the first break point. This means that if there is an algorithm which asymptotically does not result in any blocking in any cell for a given traffic distribution, then using GDCAA will also result in no blocking. Moreover the algorithm does not require a knowledge of the traffic distribution, a major drawback of the FCAA. The main problem with GDCAA is that the complexity of the algorithm is very high. It increases with the number of cells and the number of channels. This makes it impractical.

(3) Timid Dynamic Channel Assignment Algorithm (TDCAA): TDCAA is perhaps one of the simplest algorithms one can think of. Moreover, our conjecture is that it is asymptotically optimal.

Example: A linear array of three cells

Consider the linear array of three cells whose state diagram is given in Figure 1.2. Assume that the offered traffic is uniform in all the three cells. Figure 1.3 shows the upper bound [18] on the performance, measured by the carried traffic per channel, for this system for different values of the total offered traffic per channel. Assuming $n = 10000$ channels, we simulated the performance of the TDCAA for several values of the total offered traffic. Figure 1.3 shows the simulation result. It also shows the performance, in the limit of a large number of channels, of the GDCAA [19] and a particular FCAA [18]. As is clear, when the number of channels is large, the performance of the TDCAA tracks the upper bound, whereas the performance of the GDCAA does not.

We will now try to explain why, when the number of channels is large, the TDCAA gives a better performance than the GDCAA in the case of a linear array of 3 cells with uniform traffic distribution.

Why greed doesn't pay

We shall consider the case when the offered traffic is one Erlang per channel in all the three cells. In this case, we know from simulation and theory that the TDCAA is better than GDCAA [19]. Using the ATP satisfied by independent Poisson arrivals [18], we can say that all the channels will either be in state 4 or state 5. In order to maximize the carried traffic, we would like all the channels to be in state 5 as then each channel will carry two calls.

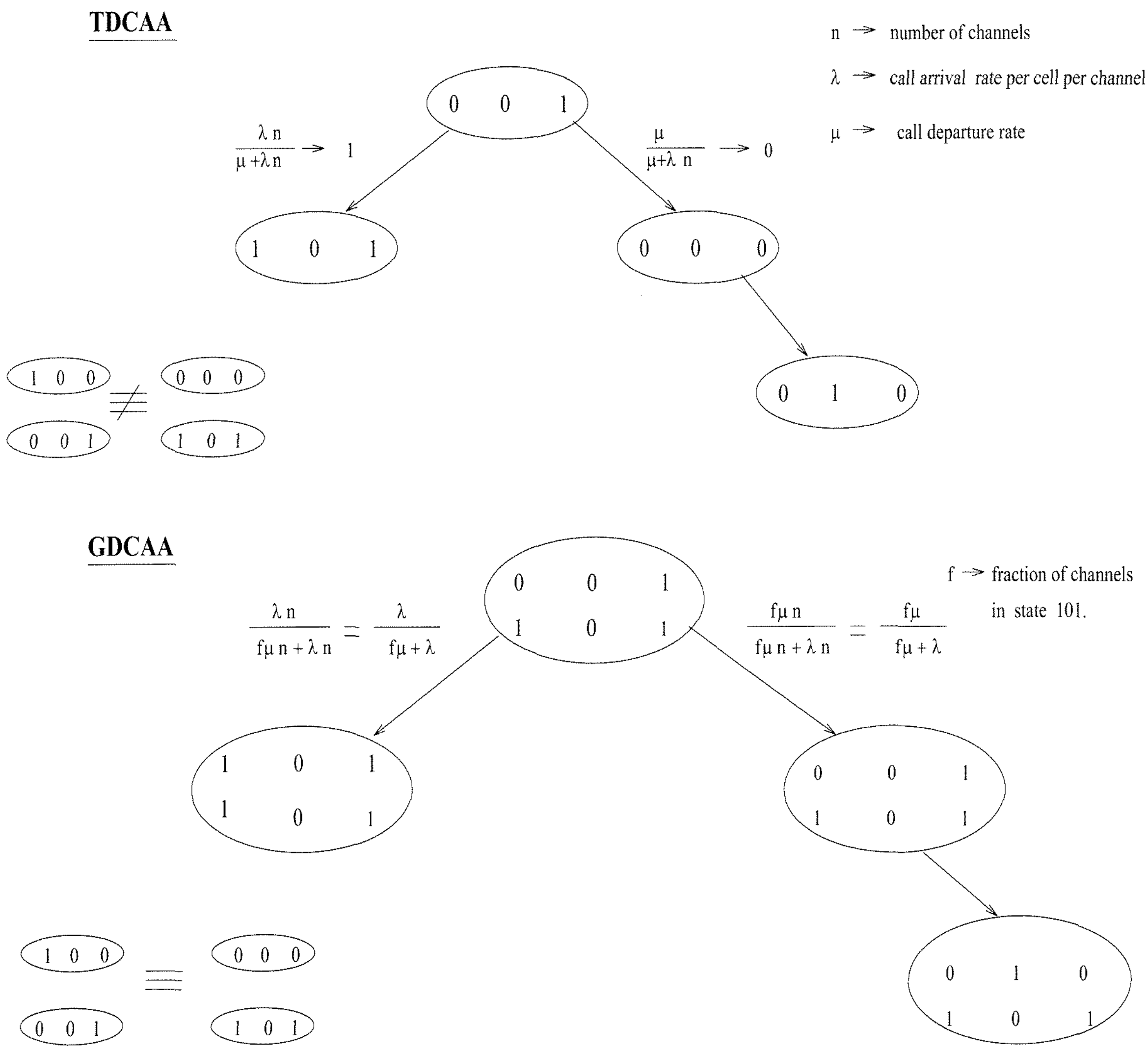


Figure 4.1: TDCAA versus GDCAA

Consider the TDCAA. Suppose a channel was in state 5 and a call departs from it taking it to state 3. We assume that all the other channels are in state 5 or state 4. The probability that this channel goes to state 1 is approximately $\mu/(\mu + \lambda n)$. (Here λ and μ are the call arrival rates and departure rates per channel per cell. In this case, we have $\lambda = \mu$.). See Figure 4.1. This probability tends to zero as n becomes large. Hence the probability of the channel going to state 4 can be made very small by increasing n. On the other hand, the probability of it going back to state 5 is approximately $\lambda n/(\lambda n + \mu)$. This probability tends to 1 as n becomes large.

Now we will examine what happens when we use GDCAA. A channel in state 5 goes to state 3 after the departure of a call from it. Here also we assume that the remaining channels are either in state 5 or state 4. Suppose kn channels are in state 5. Then the probability that a departure in cell 3 (from any channel) occurs before an arrival occurs in cell 1 is approximately $k\mu n/(k\mu n + \lambda n)$ (see Figure 4.1) i.e., $k\mu/(k\mu + \lambda)$, which is a non-zero constant. Hence, under the operation of GDCAA, the probability that a channel in state 3 goes to state 1 does not tend to zero. This is because, when GDCAA is used, a channel in state 3 and another in state 2 is equivalent to having a channel in state 5 and another in state 1 as it allows call rearrangements. Once a channel reaches state 1, the probability of it going to state 4 from there is non-zero. Similarly, the probability of a channel going from state 3 to state 5 is also non-zero.

Suppose a channel is in state 4. One can easily verify that irrespective of whether we use GDCAA or TDCAA, the probability of it going to state 5 does not tend to zero with the increase in the number of channels.

Thus we see that, under TDCAA, once a channel is caught in a state which has a higher rank (one carrying a larger number of calls) than another state, it becomes very difficult for it to go to a state of lower rank (one carrying a lower number of calls). On the other hand, if the channel is in a low-rank state, the probability of it going to the high-rank state does not tend to zero. Under the GDCAA, however, the probabilities for a channel in a high-rank state to go to a low-rank state and vice versa do not tend to zero.

GDCAA, in trying to be optimal in the short run, fails to be optimal in the long run. On the other hand, TDCAA turns out to be optimal in the long run.

4.3 Modified DCAA

We have proved under a reasonable mathematical hypothesis that the TDCAA is asymptotically optimal and the simulations we have carried out have only strengthened our conjecture. Although the TDCAA is apparently asymptotically optimal, the number of channels required before the carried traffic per channel comes very close to the upper bound for a given offered traffic and a given set of allowable states is very high, as some of the simulation results show. In this section we propose a modification of the TDCAA, which we call the Modified DCAA. The basic idea is that instead of picking up an available channel at random, the algorithm looks for an available channel in a certain order. The advantage is that with hardly any increase in complexity, the algorithm can be expected to approach the upper limit very fast. We were able to confirm this through simulation on very regular cellular systems.

Consider a linear array of cells numbered $1, 2, \ldots$ (Figure 4.2(a)). Suppose the reuse constraint is that the same channel cannot be used simultaneously in adjacent cells. We have inefficient use of the spectrum if the same channel is being used in cell numbers 3, 6 and 9 (Figure 4.2(b)). We have only three calls in progress whereas if the channel was being used in cell numbers 3, 5, 7 and 9 (Figure 4.2(c)), we would have had 4 calls in progress. The main idea is to avoid the use, as far as possible, of the same channel by cells numbered i and $i+3$. Hence we propose the following algorithm:

Algorithm: Consider any regular cellular system where the cells can be partitioned into k groups $c_1, c_2, \ldots, c_k$ such that cells in the same group can use the same channel simultaneously without violating the reuse constraints. We divide the number of channels also into k groups $a_1, a_2, \ldots, a_k$. When a call comes to a cell in group c_i, the algorithm first looks for a channel from the group a_i, then from group $a_{i+1 \text{ (modulo } k)}$, then from group $a_{i+2(\text{modulo} k)}$, and so on.

	2	3	4	5	6	7	8	9	10	

(a)

	0	1	0	0	1	0	0	1	0	

(b)

	0	1	0	1	0	1	0	1	0	

(c)

Figure 4.2: (a) A linear array of cells (b) Inefficient channel utilization (c) Efficient channel utilization

A little thought will indicate that if we have uniform traffic and only cochannel reuse constraints, the algorithm, without increasing the complexity much, tries to avoid configurations which might lead to inefficient spectrum utilization.

For example, for a linear array of cells where the reuse constraint is that the same channel cannot be used in adjacent cells simultaneously, we can divide the cells into two groups: the odd-numbered cells and the even-numbered cells and then apply the above algorithm.

The advantage with such an algorithm is that we can expect it to give a good performance when the traffic is not very high, even without a knowledge of the offered traffic in different cells.

Simulation Results: We have simulation results which show that the Modified DCAA performs quite well. Figure 4.3 shows the number of channels required by different algorithms for blocking to be less than 1% for a circular array of 30 cells. These results were obtained through simulation. The simulation results show that the Modified DCAA performs quite well, at least in this particular case.

We should note that the results shown in Figure 4.3 do not contradict what we said earlier regarding the relative performance of the TDCAA and GDCAA. TDCAA performs better than GDCAA when we have a large number of channels and the traffic is very heavy (that is, beyond the first break point [18]). Upto the first break point, both are optimal when the number of channels is large. However, when the number of channels is small, the GDCAA does perform better than the TDCAA up to the first break point, as the simulation results show.

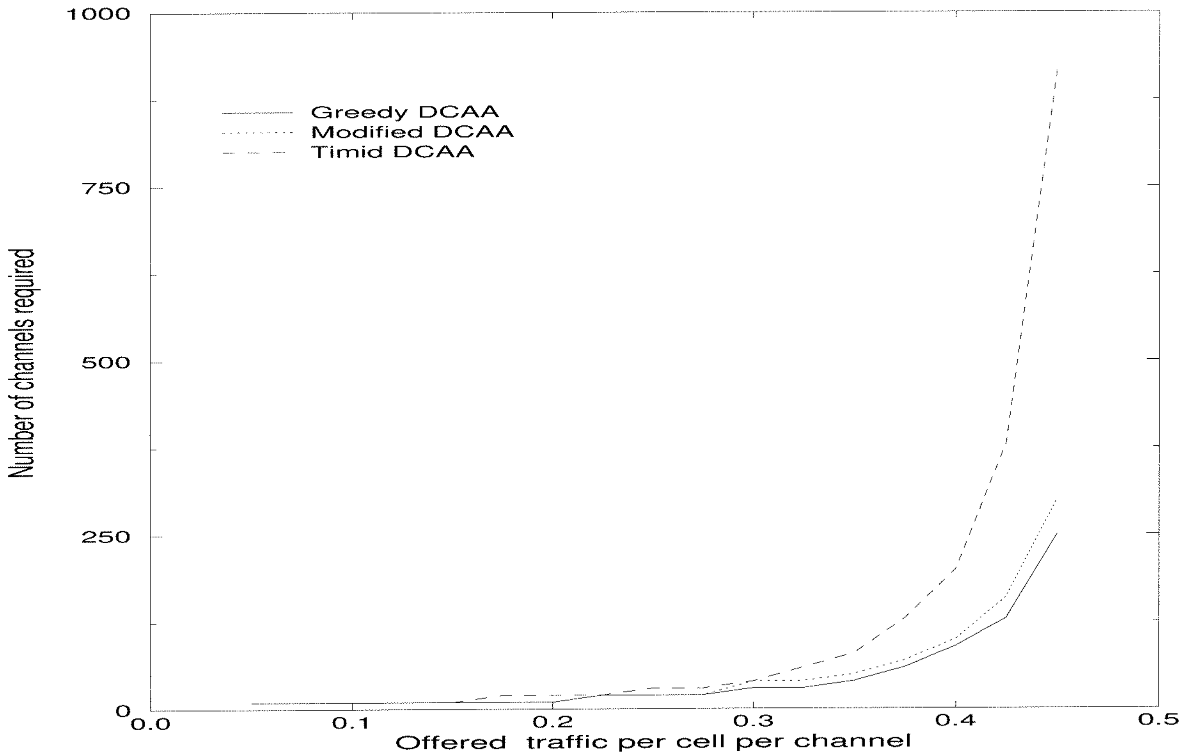


Figure 4.3: Number of channels required for average blocking probability of less than 1 percent for a circular array of 30 cells with uniform traffic distribution

Modified DCAA taking adjacent channel constraints into account and for inhomogeneous traffic:

We should note that the Modified DCAA given applies only to cellular systems with cochannel reuse constraints and can be expected to perform well when the traffic distribution is uniform. The main problem comes when we are considering adjacent channel constraints and when the traffic is not uniform. In this case, we have not been able to come up with a satisfactory order in which the different cells should look for an available channel. One way to get the order of the channels for the different cells is described below. Some simulation results are also given.

Estimate the traffic pattern. If we have no idea about the traffic intensity in the different cells, assume that the traffic is uniform in all the cells. Using the algorithm in [22], divide the available channels among the cells so that the adjacent and cochannel constraints are not violated and the number of channels each cell gets is roughly proportional to the traffic intensity in that cell. Now when a call comes to a cell, it first looks for an available channel from among the channels which it has been allocated and then from among the rest.

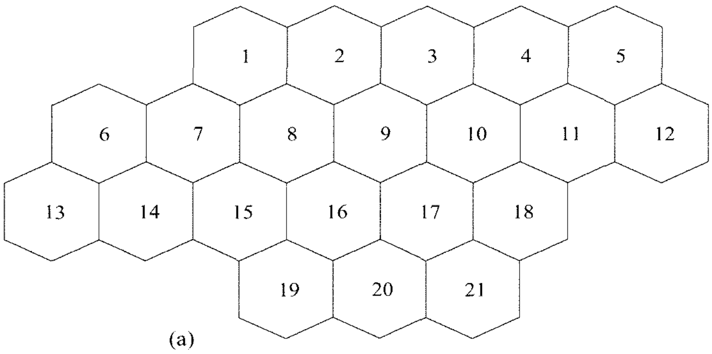
We will briefly describe two dynamic algorithms [22], with which we compare the performance of the Modified DCAA:

• **Simple** : The algorithm assigns the least available frequency to an incoming call.

• **Maxavail** : When a call comes to a cell, for each available channel, the algorithm computes the systemwide channel availability which is the sum of available channels in the different cells provided this channel is given to the new call. It picks the one with the maximum systemwide channel availability.

We have simulation results for the Modified DCAA as well as the algorithms described above. Consider a hexagonal system containing 21 cells. The 21-cell system as well as the traffic pattern for inhomogeneous traffic is given in Figure 4.4. Figure 4.5 shows the simulation results when the traffic is inhomogeneous. The total number of channels is 447. We consider two cases of Modified DCAA : (1) The channels are allocated equally among the cells (i.e., we have no idea about the traffic in the different cells) (2) The channels are allocated according to the traffic pattern. As the figure shows, if we know the traffic pattern, then the performance of the algorithm is actually better than that of the very complex Maxavail Algorithm described above. However if we do not know the traffic distribution, then the performance is not so good, although it is better that that of FCAA.

Figure 4.6 gives the simulation results for the case when the traffic is uniform. The number of channels is 444. In this case, the performance is slightly worse than



(a)

$$
\begin{array}{ccccccccccccccccccccc}
5 & 2 & 1 & 1 & 0 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
2 & 5 & 2 & 1 & 1 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 5 & 2 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 2 & 5 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 1 & 1 & 2 & 5 & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 5 & 2 & 1 & 1 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
2 & 1 & 1 & 0 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 0 & 1 & 2 & 2 & 1 & 1 & 0 & 1 & 1 & 0 \\
2 & 2 & 1 & 1 & 0 & 1 & 2 & 5 & 2 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 2 & 1 & 1 & 1 & 1 & 2 & 5 & 2 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 1 & 1 \\
1 & 1 & 2 & 2 & 1 & 0 & 1 & 1 & 2 & 5 & 2 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\
0 & 1 & 1 & 2 & 2 & 0 & 0 & 1 & 1 & 2 & 5 & 2 & 0 & 0 & 0 & 1 & 1 & 2 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 2 & 0 & 0 & 0 & 1 & 1 & 2 & 5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 2 & 1 & 1 & 0 & 0 & 0 & 0 & 5 & 2 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 0 & 0 & 0 & 2 & 5 & 2 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 2 & 2 & 1 & 1 & 0 & 0 & 1 & 2 & 5 & 2 & 1 & 1 & 2 & 1 & 1 \\
1 & 1 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 1 & 1 & 2 & 5 & 2 & 1 & 2 & 2 & 1 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & 1 & 1 & 2 & 5 & 2 & 1 & 2 & 2 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 0 & 0 & 1 & 1 & 2 & 5 & 1 & 1 & 2 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 5 & 2 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 2 & 5 & 2 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 2 & 5
\end{array}
$$

(b)

Cell number	p(i)
1	0.0166
2	0.0519
3	0.0166
4	0.0166
5	0.0166
6	0.0312
7	0.0374
8	0.1081
9	0.1601
10	0.0582
11	0.0270
12	0.0312
13	0.0644
14	0.0312
15	0.0748
16	0.1185
17	0.0582
18	0.0166
19	0.0208
20	0.0270
21	0.0166

(c)

Figure 4.4: (a) A 21-cell hexagonal system (b) The compatibility matrix (c) Inhomogeneous traffic pattern: $p(i)$ indicates the fraction of the total offered traffic in cell i

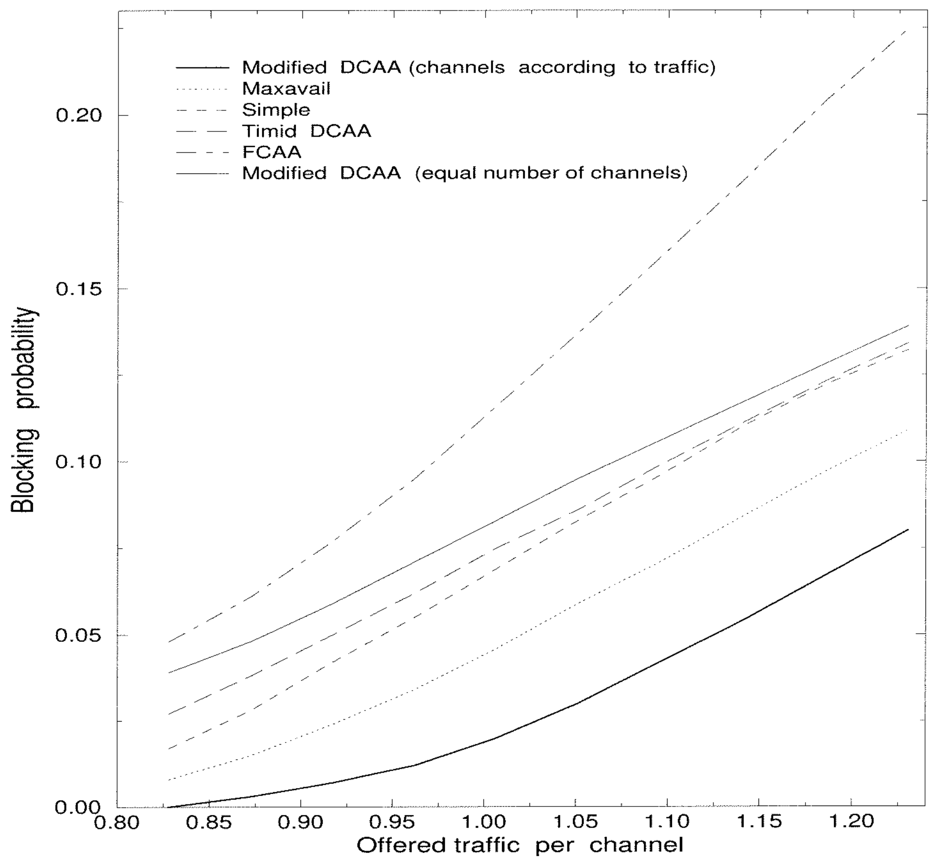


Figure 4.5: A 21-cell hexagonal system: Non-uniform traffic distribution

that of the Maxavail algorithm but is better than that of the other algorithms.

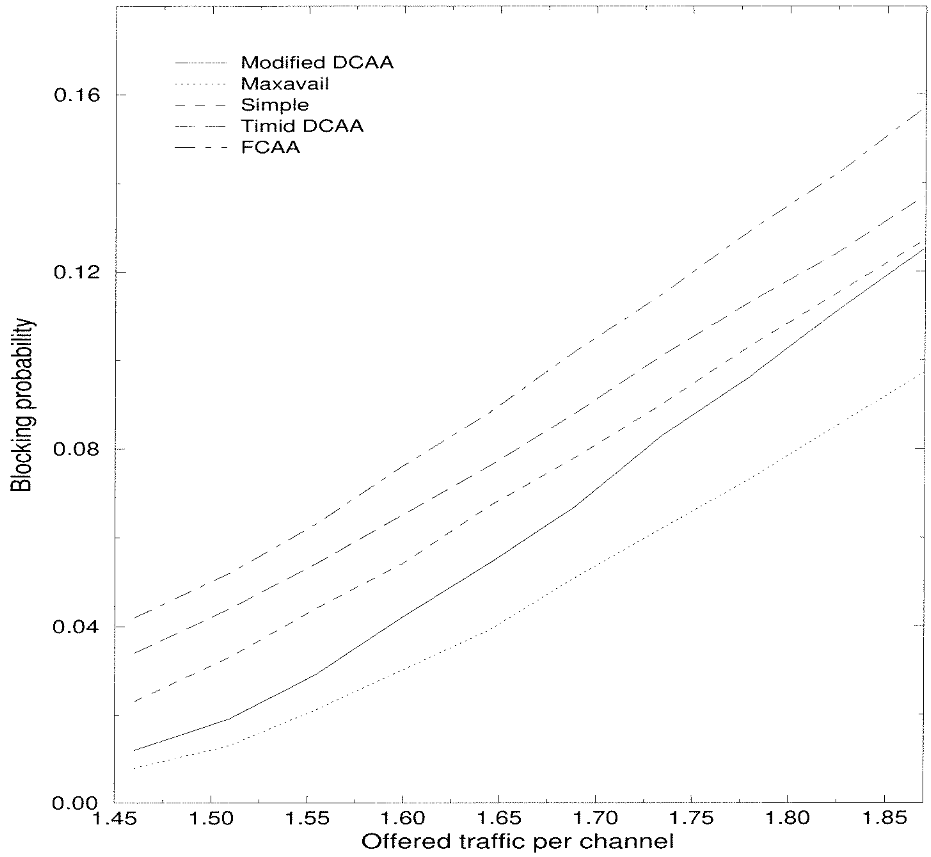


Figure 4.6: A 21-cell hexagonal system: Uniform traffic distribution

The reason why Modified DCAA works quite well if we know the traffic distribution is that then, in any cell, we can look for channels in an order depending on the traffic distribution. This fixed nature of the algorithm tries to keep the system close to an optimal state. The dynamic nature of the algorithm allows the system to adjust to the randomness in call arrivals and the holding times. The overall performance can thus be expected to be good.

4.4 Conclusion

In this chapter, we compared several channel assignment algorithms, including the TDCAA, in terms of their performance and complexity. We also presented a heuristic algorithm, very similar to the TDCAA, which can be expected to give a good performance even for cellular systems with a small number of channels.

Chapter 5 Performance Limits for Cellular Systems with Adjacent Channel Constraints

5.1 Introduction

In a cellular system, in addition to the co-channel restrictions on the use of a frequency, we also have adjacent channel restrictions. These impose restrictions on the use of the nearby frequencies in neighbouring cells. This is because the filtering process is not ideal. The entire set of restrictions is specified by a compatibility matrix which we describe below. One important problem in the area of cellular communications is to develop tight bounds on the performance of channel assignment algorithms (CAAs). Earlier work [18] in this regard deals with cellular systems having only cochannel constraints. In this paper, we extend the results obtained in [18] to cellular systems with adjacent channel constraints. We will develop an upper bound (as a function of the number of channels) on the performance of a system with a given offered traffic per channel (perhaps different for different cells) and a given compatibility matrix. We will further show that as the number of channels becomes large, this bound approaches a limit and that there are channel assignment algorithms which achieve this limit. The bound, though computationally intensive, can be used as a benchmark against which the performance of channel assignment algorithms can be compared.

5.2 Model, Definitions and the Main Result

Let $I_N = \{1, 2, \ldots, N\}$ be the set of cells in the system. Let the offered traffic in cell i be r_i Erlangs per channel. Let $r = (r_1, r_2, \ldots, r_N)$. The compatibility matrix for the system is given by an $N \times N$ matrix $C = (c_{i,j})$ of non-negative integers such that

$c_{i,j}$ indicates the minimum separation required between the channels being used in cells i and j. Let c be the maximum value of the entries in C. For the rest of the paper, we shall assume that r and C are fixed. Let there be $n + (c - 1)$ channels in the system. Let the channels be numbered $1, 2, \ldots, n, n + 1, \ldots, n + (c - 1)$.

Example. Consider a system of three cells 1,2 and 3. Let the compatibility matrix be $\{\{3, 2, 1\}, \{2, 3, 2\}, \{1, 2, 3\}\}$. What this says is that the separation between channels in the same cell should be at least 3 and that the separation between channels being used in cells 1 and 2 should be at least 2 and so on. Here $c = 3$. We shall refer to this example throughout the paper.

Result. Let $T_A(n, r)$ be the carried traffic per channel when a particular CAA A is used. The algorithms we shall consider either accept a call or reject it. There is no call waiting and calls in progress cannot be dropped. We will show that there exists a function $T(n, r)$ such that $T_A(n, r) \leq T(n, r)$ for any algorithm A and show that as n tends to infinity, $T(n, r)$ approaches a limit $T(r)$ and there are channel assignment algorithms which achieve this limit.

Let $\Gamma = \{1, 2, \ldots, m\}$ be the set of all states a channel can be in, if we consider only co-channel constraints. Let $\Omega \subseteq \Gamma \times \Gamma \times \ldots \times \Gamma$ be the set of all states a set of c contiguous channels can be in. Let $|\Omega| = M$ and let the elements in Ω be represented by $1, 2, \ldots, M$. An element $\omega \in \Omega$ will be of the form $\omega = (\gamma_1, \gamma_2, \ldots, \gamma_c)$ where $\gamma_i \in \Gamma$ for $i = 1, 2 \ldots, c$. We should note that $\Omega = \Gamma \times \Gamma \times \ldots \times \Gamma$ iff we have only cochannel constraints, that is $c = 1$ or 0.

In the example above, let the state of a channel be denoted by a 3-tuple (a_1, a_2, a_3) where $a_i = 1$ if the channel is being used in cell i and is 0 otherwise, for $i = 1, 2, 3$. Then

$$\Gamma = \{(0, 0, 0), (1, 0, 0), (0, 1, 0), (0, 0, 1)\}.$$

We shall index these states by $0, 1, 2,$ and 3 respectively. Hence if we talk about a channel being in state 1, we shall mean that the channel is in state (1,0,0). Let $\omega \in \Omega$ be ((0,0,1),(0,0,0),(0,1,0)). We shall also indicate this by $\omega = (3, 0, 2)$, where 3 corresponds to the state (0,0,1), 0 corrresponds to the state (0,0,0) and 2 corresponds to the state (0,1,0). The set Ω is as follows:

$$\begin{aligned}\Omega = \{ \quad & (0,0,0),(0,0,1),(0,0,2),(0,0,3),(0,1,0),(0,1,3),(0,3,0),(0,3,1),\\ & (0,2,0),(2,0,0),(2,0,1),(2,0,3),(1,0,0),(1,0,2),(1,0,3),(3,0,0),\\ & (3,0,1),(3,0,2),(3,1,0),(1,3,0)\}.\end{aligned}$$

Thus we have $|\Omega| = M = 20$.

These are the only possible states a set of 3 contiguous channels can be in without violating the constraints imposed by the compatibility matrix.

Let I_n be the set of channels numbered $1, 2, \ldots, n$. Let us define the hyperstate of a channel $k \in I_n$ to be $\omega = (\gamma_1, \gamma_2, \ldots, \gamma_c) \in \Omega$ if channel $k + l - 1$ is in state γ_l for $1 \leq l \leq c$. Thus the hyperstate is well-defined for all the channels in the set I_n. Let Ω' be the set of states a set of $c - 1$ contiguous channels can be in. Consider $\omega' = \{\gamma_1', \gamma_2', \ldots, \gamma_{c-1}'\} \in \Omega'$. Let $\Omega_U(\omega')$ be a subset of Ω such that $\Omega_U(\omega') = \{\omega \mid \omega \in \Omega, \omega = (\gamma_1', \gamma_2', \ldots, \gamma_{c-1}', \gamma') \text{ with } \gamma' \in \Gamma\}$. Similarly, let $\Omega_L(\omega')$ be a subset of Ω such that $\Omega_L(\omega') = \{\omega \mid \omega \in \Omega, \omega = (\gamma', \gamma_1', \gamma_2', \ldots, \gamma_{c-1}') \text{ with } \gamma' \in \Gamma\}$.

For example, we say that channel 4 is in hyperstate $(2, 0, 3)$ if channel 4 is in state 2, channel 5 (=4+1) is in state 0 and channel 6 (=4+2) is in state 3. In the above example,

$$\Omega' = \{(0,0),(0,1),(0,2),(0,3),(1,0),(1,3),(2,0),(3,0),(3,1)\}.$$

Also,

$$\Omega_U((3,0)) = \{(3,0,0),(3,0,1),(3,0,2)\},$$

and

$$\Omega_L((3,0)) = \{(0,3,0),(1,3,0)\}.$$

When a channel is in a hyperstate $\omega = (\gamma_1, \gamma_2, \ldots, \gamma_c) \in \Omega$, let the carried traffic t_ω corresponding to it be the number of calls being carried by the channel. Let $B = (b_{i,\omega})$ be the incidence matrix, i.e., $b_{i,\omega}$ is 1 if a channel carries a call in cell i when it is in hyperstate $\omega = (\gamma_1, \gamma_2, \ldots, \gamma_c)$ and is 0 otherwise.

For example,

$$t_{(2,0,3)} = 1 \qquad \text{and} \qquad t_{(0,3,1)} = 0,$$

and

$$b_{1,(2,0,3)} = 0 \qquad \text{and} \qquad b_{2,(2,0,3)} = 1.$$

We should note that

$$t_\omega = \sum_{i \in I_N} b_{i,\omega}. \tag{5.1}$$

5.3 The function $T(n, r)$ as the Upper Bound

Let x_ω be the number of channels from the set I_n which are in hyperstate $\omega \in \Omega$. Let us denote the state of the system by an M-tuple $x = (x_1, x_2, \ldots, x_M)$. For all $\omega' \in \Omega'$, we have

$$\sum_{\omega \in \Omega_U(\omega')} x_\omega = \sum_{\omega \in \Omega_L(\omega')} x_\omega - \delta_{\omega'}, \tag{5.2}$$

where $\delta_{\omega'}$ is zero for all ω' or one of the $\delta_{\omega'}$'s is $+1$, one of them is -1 and the rest are 0's.

What this says is that the number of channels whose hyperstates are in $\Omega_U(\omega')$ is the same as the number of channels whose hyperstates are in $\Omega_L(\omega')$. This is true except perhaps for the ω' such that $\Omega_U(\omega')$ contains the hyperstate of channel 1 and for the ω' such that $\Omega_L(\omega')$ contains the hyperstate of channel n.

For example, let $n = 12$. Suppose that channels $1, 2, \ldots, 13, 14$ are in states $2, 0, 3, 1, 0, 0, 0, 2, 0, 3, 1, 0, 3, 0$ respectively. Then channel 1 is in hyperstate $(2, 0, 3)$, channel 2 is in hyperstate $(0, 3, 1)$, channel 3 is in hyperstate $(3, 1, 0)$ and so on. We should note that the number of channels whose hyperstates are in $\Omega_U((3, 1))$ is 2 (channels 3 and 10) and the number of channels whose hyperstates are in $\Omega_L((3, 1))$ is also 2 (channels 2 and 9). This is true for all $\omega' \in \Omega'$ except for the elements (2,0) and (3,0) of Ω'. This is because channel 1 is in hyperstate $(2, 0, 3) \in \Omega_U((2, 0))$ and channel 12 is in hyperstate $(0, 3, 0) \in \Omega_L((3, 0))$.

From (5.2) we get,

$$\sum_{\omega\in\Omega_U(\omega')} E(x_\omega) = \sum_{\omega\in\Omega_L(\omega')} E(x_\omega) - E(\delta_{\omega'}). \tag{5.3}$$

We should note that for all $\omega' \in \Omega'$,

$$|E(\delta_{\omega'})| \leq 1. \tag{5.4}$$

Since the carried traffic in any cell is less than or equal to the offered traffic in that cell, we have, for all $i \in I_N$,

$$\sum_{\omega} b_{i,\omega} E(x_\omega) \leq r_i(n + (c-1)). \tag{5.5}$$

Also,

$$T_A(n,r) \leq \sum_{\omega} t_\omega E(x_\omega)/(n+(c-1)) + k/(n+(c-1)), \tag{5.6}$$

where k is an upper bound on the traffic carried by the last $(c-1)$ channels.

Let us define the set S_n as follows:

$$\begin{aligned} S_n = \quad \{(s_1, s_2, \ldots, s_M) : \ & \sum_{\omega} b_{i,\omega} s_\omega \leq r_i(1+(c-1)/n) \quad \text{for all } i \in I_N, \\ & s_\omega \geq 0 \quad \text{for all } \omega \in \Omega, \\ & \sum_{\omega\in\Omega_U(\omega')} s_\omega = \sum_{\omega\in\Omega_L(\omega')} s_\omega - \delta_{\omega'}, \quad \text{for all } \omega' \in \Omega', \\ & -1/n \leq \delta_{\omega'} \leq 1/n, \quad \text{for all } \omega' \in \Omega', \\ & \sum_{\omega'} \delta_{\omega'} = 0, \sum_{\omega} s_\omega = 1\}. \end{aligned} \tag{5.7}$$

We should note from (5.3), (5.4) and (5.5) that

$$E(x) \in nS_n. \tag{5.8}$$

Let us define $T(n,r)$ as follows:

$$T(n,r) = \max(\sum_{\omega} t_\omega s_\omega : s = (s_1, s_2, \ldots, s_M) \in S_n). \tag{5.9}$$

Then it is clear from (5.6), (5.8) and (5.9) that

$$T_A(n,r) \leq T(n,r)(\frac{1}{1+(c-1)/n}) + k/(n+(c-1)). \tag{5.10}$$

Hence we have obtained an upper bound (as a function of the number of channels) on the carried traffic per channel for a cellular system with adjacent channel constraints.

5.4 The Function $T(r)$ as the Upper Bound Limit

Now we will show that this upper bound approaches a limit as n becomes large.

Consider the set S defined as follows :

$$\begin{aligned} S = \quad \{(s_1, s_2, \ldots, s_M) : \; & \sum_{\omega} b_{i,\omega} s_\omega \leq r_i \qquad \text{for all } i \in \; I_N, \\ & s_\omega \geq 0 \qquad \text{for all } \omega \in \Omega, \\ & \sum_{\omega \in \Omega_U(\omega')} s_\omega = \sum_{\omega \in \Omega_L(\omega')} s_\omega \qquad \text{for all } \omega' \in \Omega', \\ & \sum_{\omega} s_\omega = 1\}. \end{aligned} \tag{5.11}$$

Let us define $T(r)$ as follows:

$$T(r) = \max(\sum_{\omega} t_\omega s_\omega : s = (s_1, s_2, \ldots, s_M) \in S). \tag{5.12}$$

From (5.7) and (5.11), we have

$$\lim_{n \to \infty} S_n = S.$$

Now $T(n,r)$ and $T(r)$ are solutions to linear programs with the same objective function but over different spaces S_n and S. Also, as n tends to infinity, S_n tends to

S.

Therefore

$$\lim_{n\to\infty} T(n,r) = T(r)$$

(This follows from [8], pp. 68-77.),

and therefore from (5.10),

$$\lim_{n\to\infty} T_A(n,r) \leq T(r).$$

Example. Consider once again the 3-cell system with compatibility matrix $\{\{3,2,1\},\{2,3,2\},\{1,2,3\}\}$. Suppose $\mathbf{r} = (1/3, 1/3, 1/3)r$ Erlangs per channel. Figure 5.1 shows the function $T(r)$. Compare it with $T(r)$ when the compatibility matrix is $\{\{1,1,0\},\{1,1,1\},\{0,1,1\}\}$, that is, with the $T(r)$ of a system having only cochannel constraints.

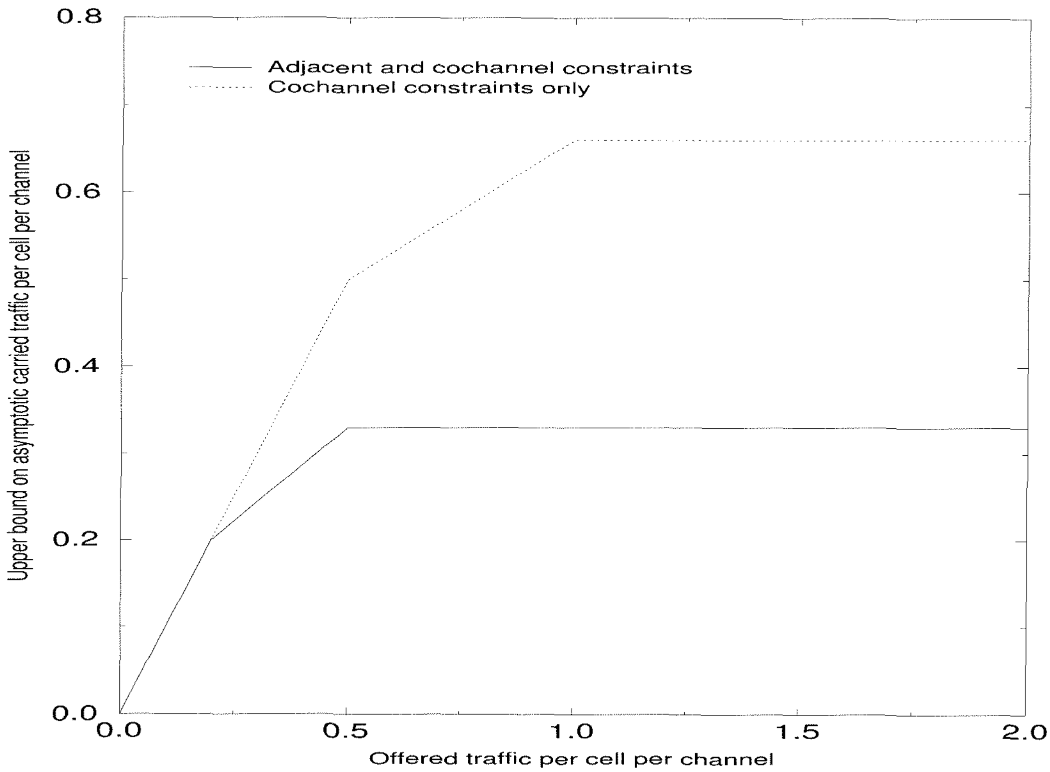


Figure 5.1: Asymptotic upper bound for the three-cell system with offered traffic $\mathbf{r} = (1/3, 1/3, 1/3)r$. The solid line indicates the upper bound when we have adjacent channel constraints and the dashed line gives the bound when we only have cochannel constraints.

In the next section we will show that the upper bound is achievable in the limit as the number of channels becomes large.

5.5 Asymptotic Tightness of the Upper Bound

Consider a directed graph $G = (X, E)$. Let there be a one to one correspondence between the elements of the vertex set X in G and the elements of the set Ω and we shall refer to them interchangeably. Let E consist of directed edges between the vertices of G. There is an edge from vertex i to vertex j if and only if there is an $\omega' \in \Omega'$ such that $i \in \Omega_L(\omega')$ and $j \in \Omega_U(\omega')$. Let $s = (s_1, s_2, \ldots, s_M) \in S$ give the maximum value $T(r)$ in (5.12). Let s_i be the value given to vertex i in G. Figure 5.2 shows the graph G for the 2-cell example we have been considering throughout this paper. It also shows the value s_i given to the vertex i. This $s \in S$ corresponds to $T(r)$ for $r = (1/6, 1/6, 1/6)$.

The graph G has the following properties: (1) For any $\omega' \in \Omega'$, there is a directed edge from every vertex in G which is in $\Omega_L(\omega')$ to every vertex in G which is in $\Omega_U(\omega')$. (2) The sum of the values given to those vertices in G which are in $\Omega_L(\omega')$ is equal to the sum of the values given to the vertices in G which are in $\Omega_U(\omega')$. This follows from the definition of the set S (5.11). Let us remove from G all the vertices having a value 0. This does not affect properties (1) and (2) described above. Figure 5.3 shows the graph G with all the vertices having a value 0 removed.

Consider the following operation of first picking a circuit Λ and then removing it from the graph. Pick up any vertex i. Since $s_i \neq 0.0$, there is at least one vertex j with a nonzero value which has an edge coming into it from i. If i and j are the same, we stop. Otherwise, there is at least one vertex k with a nonzero value which has an edge coming into it from vertex j. If k coincides with one of the vertices already encountered (in this case, vertices i and j), we stop. Else we continue in this manner. Finally, the process has to stop as the number of vertices is finite. The moment the process stops, we would have traced a circuit. Call it Λ. We should note that not all the vertices encountered during the process belong to Λ. Let y_Λ be the minimum value among the vertices in Λ. Subtract y_Λ from the values of all the vertices in Λ,

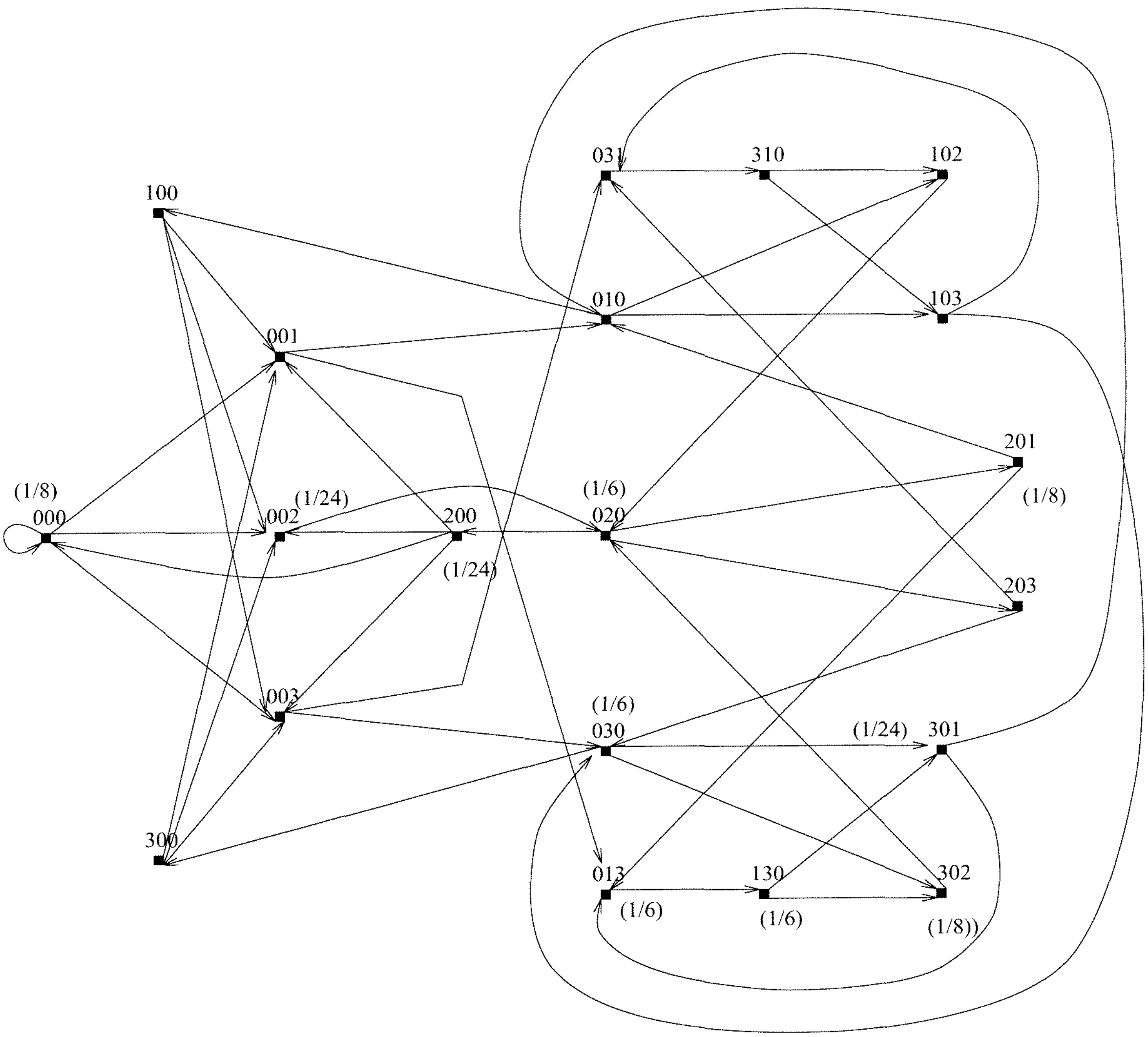


Figure 5.2: The directed, labeled graph G for the system with 3 cells. The label against a vertex indicates the hyperstate to which it corresponds and the figure in bracket gives the s_i value for $r = (1/6, 1/6, 1/6)$. An s_i value of zero is not indicated.

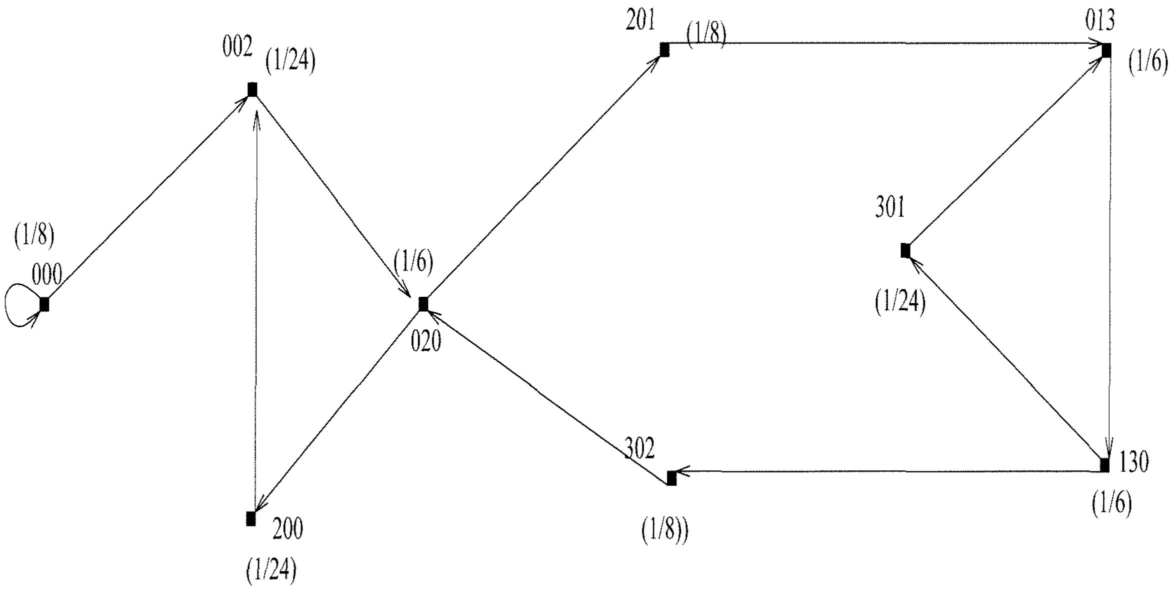


Figure 5.3: The graph G after all the vertices having a zero value have been removed.

remove all the vertices having a zero value (there is at least one such vertex) and also remove all the edges going into and out of these vertices. Let the new graph be G'. The new graph has at least one vertex less than the previous one. This new graph continues to have the properties (1) and (2) described above regarding the graph G. Hence we can continue this operation of picking and removing circuits from G' till we are left with an empty graph. The number of times we repeat this operation before we get an empty graph is finite as the graph G itself is a finite graph and contains a finite number of vertices. Hence the number of circuits we obtain is also finite, say M'. Figure 5.4 shows a circuit Λ and the graph G after the circuit has been removed from the graph.

Let

$$Y_\Lambda = \lfloor n y_\Lambda \rfloor. \tag{5.13}$$

We have for all $\omega \in \Omega$,

$$s_\omega = \sum_{\Lambda \ni \omega} y_\Lambda. \tag{5.14}$$

Given a circuit Λ containing vertices $\lambda_1, \lambda_2, \ldots, \lambda_{k_\Lambda}$ (where there is an edge from

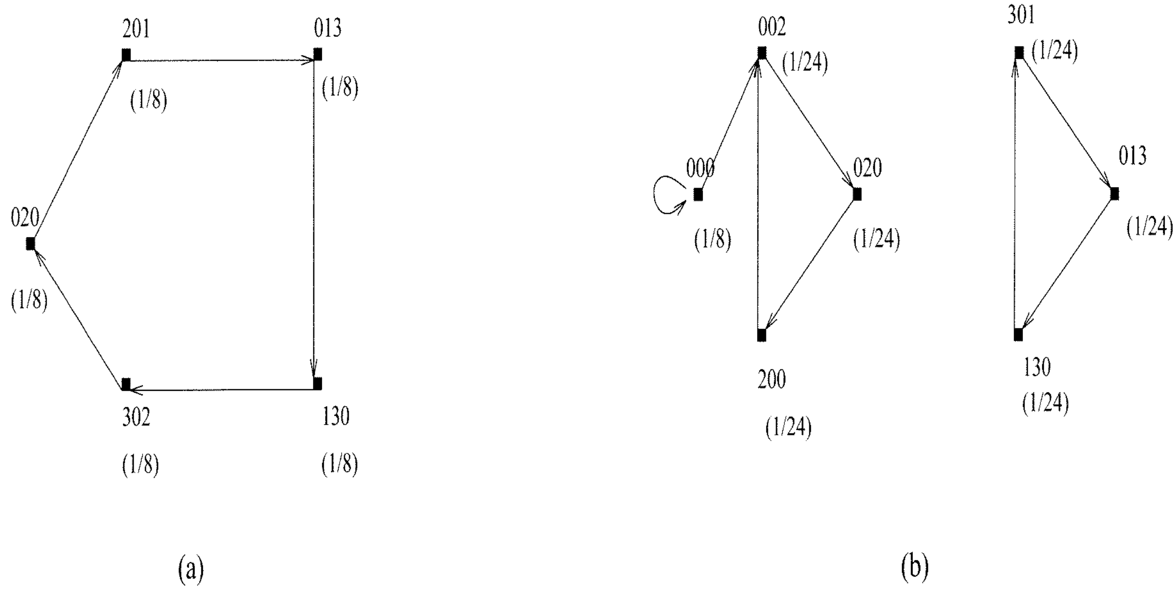
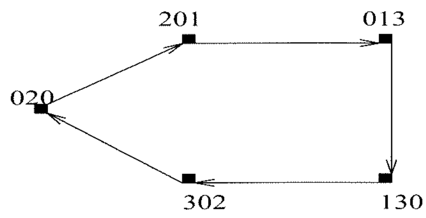


Figure 5.4: (a) A circuit Λ with $y_\Lambda = 1/8$ and (b) the graph G after the circuit Λ has been removed.

λ_i to λ_{i+1}) and an integer value Y_Λ associated with Λ, we will now describe how to assign states to a set of $k_\Lambda \times Y_\Lambda + (c-1)$ contiguous channels such that from this set, we will have Y_Λ channels in hyperstate corresponding to λ_i for $i = 1, 2, \ldots, k_\Lambda$. Suppose $\lambda_1 = (\gamma_1, \gamma_2, \ldots, \gamma_c)$. Then let $\lambda_2 = (\gamma_2, \gamma_3, \ldots, \gamma_c, \gamma_{1,2})$. In general, let $\gamma_{i,j}$ be in Γ such that if we append a channel which has been assigned the state $\gamma_{i,j}$ to the last $(c-1)$ channels of the vertex λ_i, we get the hyperstate corresponding to vertex j. Number the channels $1, 2, \ldots, k_\Lambda \times Y_\Lambda, \ldots, k_\Lambda \times Y_\Lambda + (c-1)$. Let channel number 1 be assigned the state γ_1, channel 2 be assigned the state γ_2, and so on so that finally we have channel 1 in hyperstate λ_1. Now if we assign the state $\gamma_{1,2}$ to channel $(c+1)$, then we will have channel 2 in hyperstate λ_2. Now to have channel 3 in hyperstate λ_3, we assign the state $\gamma_{2,3}$ to channel $(c+2)$. In this way we can continue till finally we have Y_Λ channels in hyperstates $\lambda_1, \lambda_2, \ldots, \lambda_{k_\Lambda}$. We would have used exactly $k_\Lambda \times Y_\Lambda + (c-1)$ channels. Figure 5.5 shows a circuit Λ with $k_\Lambda = 5, Y_\Lambda = 2$ and the assignment of states to a set of $5 \times 2 + (3-1) = 12$ contiguous channels such that for each hyperstate in Λ, we have $Y_\Lambda = 2$ channels in that hyperstate.



(a)

Channel Number	Channel State	Hyperstate
1	0	020
2	2	201
3	0	013
4	1	130
5	3	302
6	0	020
7	2	201
8	0	013
9	1	130
10	3	302
11	0	
12	2	

(b)

Figure 5.5: (a) A circuit Λ with $Y_\Lambda = 2$ and (b) the assignment of states to 12 channels corresponding to this circuit.

Now suppose that $\gamma_0 \in \Gamma$ denotes the all-zero state. Then append $(c-1)$ channels which have been assigned the state γ_0 to the set of $k_\Lambda \times Y_\Lambda + (c-1)$ channels corresponding to the circuit Λ. Then we will have a set of $k_\Lambda \times Y_\Lambda + 2(c-1)$ channels corresponding to the circuit Λ. Since γ_0 is the all-zero state, we can concatenate the channels corresponding to the various circuits without violating the adjacent channel and cochannel reuse constraints. Thus we will have a set of $n' = \sum_\Lambda (k_\Lambda \times Y_\Lambda + 2(c-1))$ channels. This set will contain at least $\sum_{\Lambda \ni \omega} Y_\Lambda$ channels which would have been assigned the hyperstate ω.

We should note the following:

$$\sum_{\Lambda} k_\Lambda y_\Lambda = 1.$$

Therefore we have,

$$\begin{aligned} n &= \sum_{\Lambda} k_\Lambda y_\Lambda n \\ &\geq \sum_{\Lambda} k_\Lambda \lfloor y_\Lambda n \rfloor \\ &= \sum_{\Lambda} k_\Lambda Y_\Lambda \\ &= n' - \sum_{\Lambda} 2(c-1), \end{aligned}$$

and

$$\begin{aligned} n &= \sum_{\Lambda} k_\Lambda y_\Lambda n \\ &\leq \sum_{\Lambda} k_\Lambda (\lfloor y_\Lambda n \rfloor + 1) \\ &= \sum_{\Lambda} k_\Lambda (Y_\Lambda + 1) \\ &= n' - \sum_{\Lambda} 2(c-1) + \sum_{\Lambda} k_\Lambda. \end{aligned}$$

Therefore,

$$n' - \sum_{\Lambda}(2(c-1)) \leq n \leq n' - \sum_{\Lambda}(2(c-1)) + \sum_{\Lambda} k_\Lambda.$$

Let

$$n'' = n' - \sum_{\Lambda}(2(c-1)) + \sum_{\Lambda} k_\Lambda.$$

Then

$$n'' - \sum_{\Lambda} k_\Lambda \leq n \leq n''. \tag{5.15}$$

Hence

$$\lim_{n\to\infty} \frac{n}{n''} = 1. \tag{5.16}$$

Fixed Channel Assignment Algorithm: Suppose we have a system with n''

channels with a certain offered traffic per channel $r = (r_1, r_2, \ldots, r_N)$. The channels are assigned states as described above. Suppose that the channels which are assigned the hyperstate ω are allocated to cell i if and only if $b_{i,\omega} = 1$. The algorithm is as follows: When a call comes to cell i, if there is a channel which has been allocated to the cell and is not being used by a call in the cell, then the call is given one such channel. Otherwise the call is blocked.

Suppose we consider only the channels which were actually assigned the hyperstate $\omega \in \Omega$ and ignore the channels which happen to be in hyperstate ω because of appending channels in state γ_0 to the channels corresponding to a circuit or because of concatenating channels corresponding to various circuits. Let X_i be the the number of channels allocated to cell i using this scheme. Then we can say that

$$X_i = \sum_{\omega} \sum_{\Lambda \ni \omega} b_{i,w} Y_\Lambda. \tag{5.17}$$

Therefore,

$$\begin{aligned} \lim_{n'' \to \infty} \frac{X_i}{n''} &= \lim_{n \to \infty} \frac{X_i}{n} && (\text{ from (5.16)}) \\ &= \lim_{n \to \infty} \sum_{\omega} \sum_{\Lambda \ni \omega} b_{i,w} Y_\Lambda / n && (\text{ from (5.17)}) \\ &= \sum_{\omega} \sum_{\Lambda \ni \omega} b_{i,w} y_\Lambda && (\text{ from (5.13)}). \end{aligned} \tag{5.18}$$

Also, we have

$$\begin{aligned} X_i &= \sum_{\omega} \sum_{\Lambda \ni \omega} b_{i,w} Y_\Lambda \\ &\leq \sum_{\omega} \sum_{\Lambda \ni \omega} b_{i,w} y_\Lambda n \\ &= \sum_{\omega} b_{i,w} (\sum_{\Lambda \ni \omega} y_\Lambda) n \\ &= \sum_{\omega} b_{i,w} s_\omega n && (\text{ from (5.14)}) \\ &\leq r_i n && (\text{ from (5.11)}) \\ &\leq r_i n'' && (\text{ from (5.15)}). \end{aligned}$$

What this says is that the offered traffic in cell i is greater than the number of channels which it has been allocated under the Fixed channel Assignment Algorithm (FCAA).

Let q_i be the carried traffic in cell i. Then from the Asymptotic Traffic Property satisfied by independent Poisson arrivals [18] we have,

$$\lim_{n''\to\infty} \frac{q_i}{X_i} = 1.$$

So

$$\lim_{n''\to\infty} \frac{q_i/n''}{X_i/n''} = 1.$$

Since from (5.18),

$$\lim_{n''\to\infty} \frac{X_i}{n''} = \sum_{\omega} \sum_{\Lambda\ni\omega} b_{i,w} y_\Lambda,$$

we have

$$\lim_{n''\to\infty} \frac{q_i}{n''} = \sum_{\omega} \sum_{\Lambda\ni\omega} b_{i,w} y_\Lambda. \tag{5.19}$$

Let $T'(n'', r)$ be the total carried traffic per channel when the number of channels is n'' and the FCAA described above is used. Then we have,

$$\begin{aligned}
\lim_{n''\to\infty} T'(n'', r) &= \lim_{n''\to\infty} \sum_{i\in I_N} \frac{q_i}{n''} \\
&= \sum_{i\in I_N} \sum_{\omega} \sum_{\Lambda\ni\omega} b_{i,w} y_\Lambda \qquad \text{(from (5.19))} \\
&= \sum_{\omega} \Big(\sum_{i\in I_N} b_{i,\omega} \Big) \Big(\sum_{\Lambda\ni\omega} y_\Lambda \Big) \\
&= \sum_{\omega} t_\omega s_\omega \qquad \text{(from (5.1) and (5.14))} \\
&= T(r).
\end{aligned}$$

This shows that as the number of channels tends to infinity, there are channel assignment algorithms which achieve the upper bound given by $T(r)$ and hence the bound is asymptotically sharp.

5.6 Conclusion

In this chapter, we developed an upper bound on the performance of cellular systems with adjacent channel constraints. The bound obtained is tight when the number of channels is large. The bound is computationally intensive, but can be used as a benchmark against which the performance of algorithms can be compared.

Chapter 6 Interference and Availability

6.1 Introduction

We have assumed so far that we are given the set of allowable states a system can be in, and have tried to design and analyze "good" channel assignment algorithms based on this assumption. One main problem in cellular radio is to determine the set of allowable states. In this chapter, we will discuss some ways of determining the set of allowable states for a system. Also, in any given state, users experience interference from one another. We will introduce "availability" as a measure of how good the call quality for a particular user is in the presence of other users and hence develop another measure for comparing the performance of channel assignment algorithms. In microcellular systems, determining the set of allowable states becomes a very complex task because of the large system size. Also, there is no single channel model which fits all situations. The set of allowable states based on a certain channel model may not be very suitable. We might be underutilizing the capacity of the system or the service quality might be bad. One possible way to solve the problem is to have algorithms which choose channels based on the instantaneous measurement, by the base stations and the mobiles, of the interference power in the channels. This implies the use of algorithms which are "measurement based" [3] rather than "prediction based", as the algorithms discussed so far have been. In this chapter, we also present some heuristic measurement-based algorithms and give some simulation results.

6.2 Interference and Availability

Consider a single-channel cellular system. Let $I_N = \{0, 1, 2, \ldots, N-1\}$ denote the set of cells in the system and let $\Omega = \{1, 2, \ldots, 2^N\}$ be the set of all possible states that a single channel can be in. Consider a state j of the system in which we have users in cells $i_1, i_2, \ldots, i_k$. Let us focus on the user in cell i_1. In any urban cellular system, the signal power received by a given user in this cell will undergo rapid fading caused by the variations in the channel as well as shadowing caused by obstacles in the signal path from the transmitter to the receiver. At the same time, this user will experience interference from users in cells $i_2, i_3, \ldots, i_k$. This appears as noise to the receiver. Thus the signal to interference ratio (S/I) for the user will be a random variable. We assume that there is a threshold such that if $S/I > z$, the call quality is acceptable to the user, whereas if $S/I < z$, the call quality is not acceptable. We then define the availability for a call-in-progress as the probability that $S/I > z$. We denote the availability for a call in cell i when the system is in state j by $p_{i,j}$.

One problem is how to calculate the $p_{i,j}$'s. These numbers depend on many factors: the frequency being used, the weather conditions, the height of the antenna, the geography of the region etc. However, for a given cellular system operating in a certain frequency range, the $p_{i,j}$'s can be approximately obtained using the following formula due to Linnartz [15]:

$$p_{i,j} = \frac{1}{\sqrt{\pi}} \sum_{l=1}^{m} w_l \prod_{s=1}^{n} P_l(s). \tag{6.1}$$

We can say that $p_{i,j}$ consists of m terms and that $P_l(s)$ may be regarded as the contribution of the s^{th} interferer to the l^{th} term in $p_{i,j}$. Here the w_l's are constants associated with the m-point Hermite integration [1].

In order to calculate $P_l(s)$ in (6.1), the user is placed at the farthest position from the base station, and the interferer is placed at the center of the interfering cell. This is not exactly the worst case situation, but is in between the worst case and the average.

Thus the following term $Q_l(s)$ [15]:

$$Q_l(s) = \frac{1}{\sqrt{\pi}} \sum_{k=1}^{m} w_k \frac{r_s^{\beta} \exp\{x_l \sigma \sqrt{2}\}}{r_s^{\beta} \exp\{x_l \sigma \sqrt{2}\} + z r_0^{\beta} \exp\{x_k \sigma \sqrt{2}\}}, \tag{6.2}$$

may be used for $P_l(s)$ in (6.1).

In (6.1) and (6.2), the w_k's and the x_l's are constants associated with m-point Hermite integration [1]; σ is the logarithmic standard deviation of the shadowing; β is the signal attenuation constant, which is usually a number between 3 and 4; n is the number of interferers for user i; the r_s's indicate the distance of the interferer s from the center of the cell i; r_0 is the distance of the user from the corresponding base station; $(n+1)$ is the total number of cells in which the channel is being used; and z is the S/I threshold. See Figure 6.1.

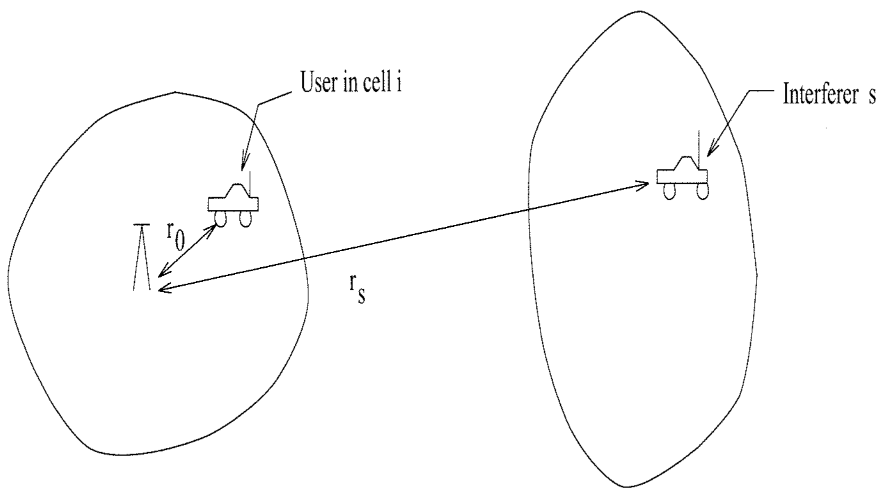


Figure 6.1: Linnartz formula: equation 6.2

In our numerical studies of the formula (6.2), we have found that using $m = 3$ is sufficient to get a very good approximation for the $p_{i,j}$'s. For example, Table 6.1 gives the values of the $p_{i,j}$'s for a linear array of three cells with $z = 10$db, $\beta = 4$, and $\sigma = 1.38$. These were calculated from (6.2) using $m = 20$. The figures in the brackets indicate the values obtained using $m = 3$, which as can be seen are very close to those with $m = 20$.

For example, if we want only those states in which the availability to each user is 90 percent or more to be allowable, then the set of allowable states is $\{(000), (100), (010), (001)\}$. On the other hand, if we say that a state is allowable if each user has an availability of 85 percent or more, then the set of allowable states is $\{(000), (100), (010), (001), (101)\}$. Once we have the set of $p_{i,j}$'s, we can get the set of allowable states

States	000	001	010	100	110	101	011	111
Cell 1	-	-	-	1.0	0.5715 (0.5740)	0.8911 (0.8907)	-	0.6041 (0.607)
Cell 2	-	-	1.0	-	0.5715 (0.5740)	-	0.5715 (0.5740)	0.371 (0.371)
Cell 3	-	1.0	-	-	-	0.8911 (0.8907)	0.5715 (0.5740)	0.6041 (0.607)

Table 6.1: $p_{i,j}$'s for a linear array of three cells with z = 10db, $\beta = 4$, and $\sigma = 1.38$. The figures in the bracket indicate the $p_{i,j}$'s obtained using $m = 3$.

depending on the call quality we want to guarantee the users.

A different way of getting the set of allowable states

In order to calculate the interference experienced by a user from other cochannel users, normally the user is placed at the worst possible spot: as far as possible from the base station with which he is communicating, and the interferers are placed at the centers of the interfering cells. One of the possible reasons for this might have been that the amount of calculation needed to get the interference between users in different cells was enormous and so it was difficult to take any kind of an average. However, the Linnartz formula enables us to average the interference over the entire area of the cells in which the users are, with very little computation.

Consider two non-overlapping cells of unit area with their base stations distance d apart. Let

$$Q'_l(d) = \int_{(R,R')} \frac{1}{\sqrt{\pi}} \sum_m w_m \frac{((x')^2 + (y')^2 + d^2 + 2dx')^{\beta/2}}{(((x')^2 + (y')^2 + d^2 + 2dx')^{\beta/2} + k(l,m)(x^2 + y^2)^{\beta/2})} dx\, dy\, dx'\, dy'. \tag{6.3}$$

Here R and R' indicate the regions of the two cells, (x, y) is the position of the user in one cell with respect to its center and (x', y') is the position of the user in the other cell with respect to the center of that cell (See Figure 6.2).

Suppose a channel is in state j and we have users in cells i and s. Let the distance between their base stations be $d(s)$. Then to get the contribution of the interferer in cell s to the l^{th} term in $p_{i,j}$, if we use $Q'_l(d(s))$ given by (6.3) for $P_l(s)$ in (6.1),

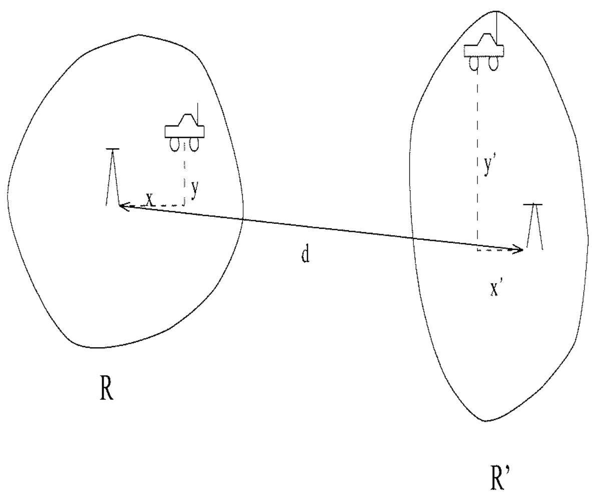


Figure 6.2: Averaging method: equation 6.3

States	000	001	010	100	110	101	011	111
Cell 1	-	-	-	1.0	0.6703	0.9253	-	0.6351
Cell 2	-	-	1.0	-	0.6703	-	0.6703	0.4958
Cell 3	-	1.0	-	-	-	0.9253	0.6703	0.6351

Table 6.2: $p_{i,j}$ values for the 3-cell example, obtained using the averaging method.

instead of using $Q_l(s)$ given by (6.2), we can expect the carried traffic to increase. The reason is that we are averaging the availability over the cells in which the users and interferers are, instead of taking something between the average and the worst case.

Table 6.2 gives the values of the $p_{i,j}$'s obtained by averaging the availability over the cells in which the users and interferers are. As is clear, even if we have an availability criterion of 90 percent, the set of allowable states is more.

Figure 6.4 shows the upper bound on the carried traffic [18] for a 19-cell hexagonal system shown in Figure 6.3 for the case when the set of allowable states is obtained by using the averaging method and when it is obtained using the "worst-case" method.

The parameters for calculating interference are z = 10db, $\beta = 4$, and $\sigma = 1.38$. The availability threshold is 90 percent. We have found via simulation that, if the averaging method is used to find the set of allowable states, then the number of users who experience an availability less than the threshold is quite low. Thus the averaging method is seen to increase the capacity without affecting the call quality.

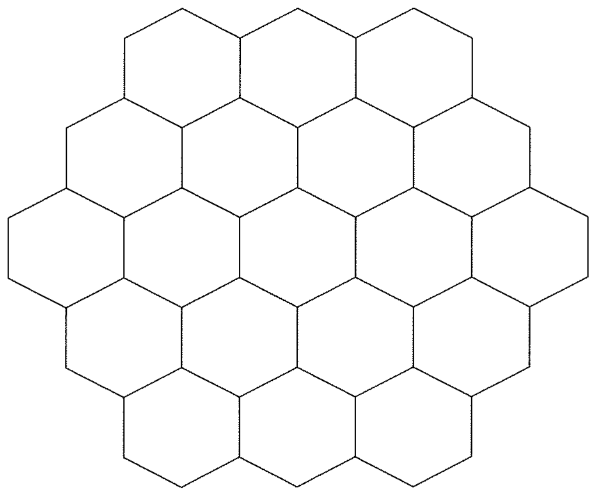


Figure 6.3: A 19-cell hexagonal system

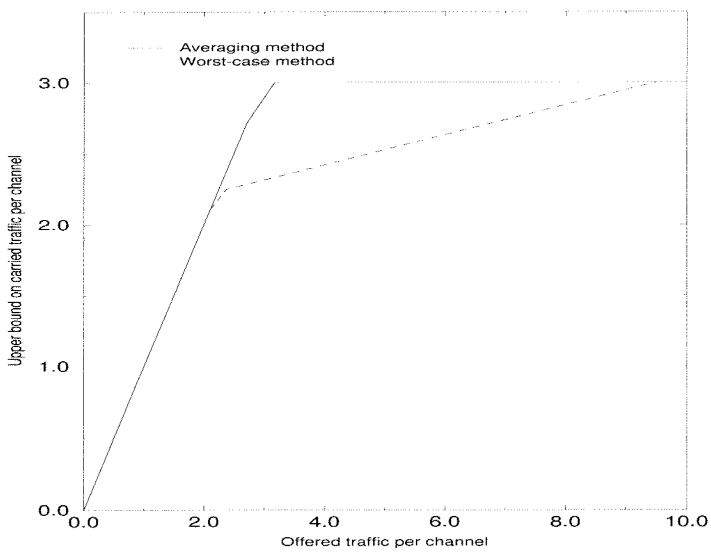


Figure 6.4: Upper bound on the carried traffic for the 19-cell system for the averaging method and for the worst-case method

Above Threshold Carried Traffic

Let us define the weight w_j of a state j by

$$w_j = \sum_{i:a_{i,j}=1} p_{i,j}. \tag{6.4}$$

Here $a_{i,j}$ is the incidence matrix. $a_{i,j}$ is 1 if there is a call in cell i when the system is in state j and is 0 otherwise. The weight w_j is a measure of the "above threshold carried traffic" (ATCT) for state j.

We now define the ATCT for a given algorithm, say algorithm A, as

$$t_c(A) = \sum_j \pi_j w_j, \tag{6.5}$$

where π_j is the probability that the system is in state j.

In this new setting, our goal is to find algorithms which maximize the ATCT, rather than simply the carried traffic.

Assuming that the matrix of $p_{i,j}$'s is known, we have been able to show (a sketch of the proof is given in Appendix E) that for a given cellular system with a given offered traffic, there is an upper bound on the ATCT as defined above and that there are algorithms which achieve this upper bound asymptotically. The proof is similar to that in [18]. Our goal is to look for algorithms which are "practical" and which give a "good performance" even when the number of channels is not very large.

Simulation: We conducted simulation for the following two algorithms to get an idea of how the call quality and performance are affected when we use different methods (mentioned above) for calculating the $p_{i,j}$'s. We use the Above Threshold Carried Traffic (ATCT) as our performance measure.

(1) Algorithm 1: Here the interference between various cells is calculated by averaging it over the area of the cells as mentioned above (6.3). When a call comes to a cell, the following is done for each channel: Assume that the user has been given that channel. Then the availability for all the users using that channel is calculated. If the availability is above the minimum for all such users, the change in the above threshold carried traffic is calculated assuming that the new call has been assigned

this channel. If the availability falls below the minimum for any user, the channel is assumed unavailable. From all the channels which pass the availability criterion, the channel which results in the maximum increase in the ATCT is selected.

(2) Algorithm 2: Similar to algorithm 1 except that instead of averaging over the areas of the cells in which the users are, when we calculate the interference experienced by any user, we assume that the user is at the farthest point from the center of cell in which he is and the interferers are at the centers of the interfering cells.

We conducted simulation for the 19-cell hexagonal system shown in Figure 6.3 with uniform traffic distribution and 100 channels. We used $z = 10$ db, $\beta = 4.0$, and a minimum availability requirement of 90 percent. Figure 6.5 shows the simulation results. As expected, Algorithm 1 performs better than Algorithm 2. Also, although it is not shown in the figure, simulation shows that the fraction of users who experience an availability of less than 90 percent is very small. For the purpose of comparison, the figure shows the upper bound on the ATCT for the 19-cell system for two cases, when the $p_{i,j}$'s are calculated using the averaging method and when they are calculated using the "worst-case" method.

6.3 Measurement Based Algorithms

As mentioned in the introduction, the algorithms discussed so far are prediction based. The set of allowable states is determined based on some model of the channel which may not be suitable for different situations and which may require updating as the channel conditions change. Hence we need to look at algorithms which are based on instantaneous measurement, rather than on prediction. In this section, we look at two such algorithms and compare their performance with the prediction based algorithms mentioned above, through simulation.

(1) Algorithm 3: Here the actual location of the user is taken into account to calculate the interference. When a call comes to a cell, the following is done for all the channels: Assume that the user has been given that channel. Then the availability for all the users using the channel is calculated. If the availability is above the

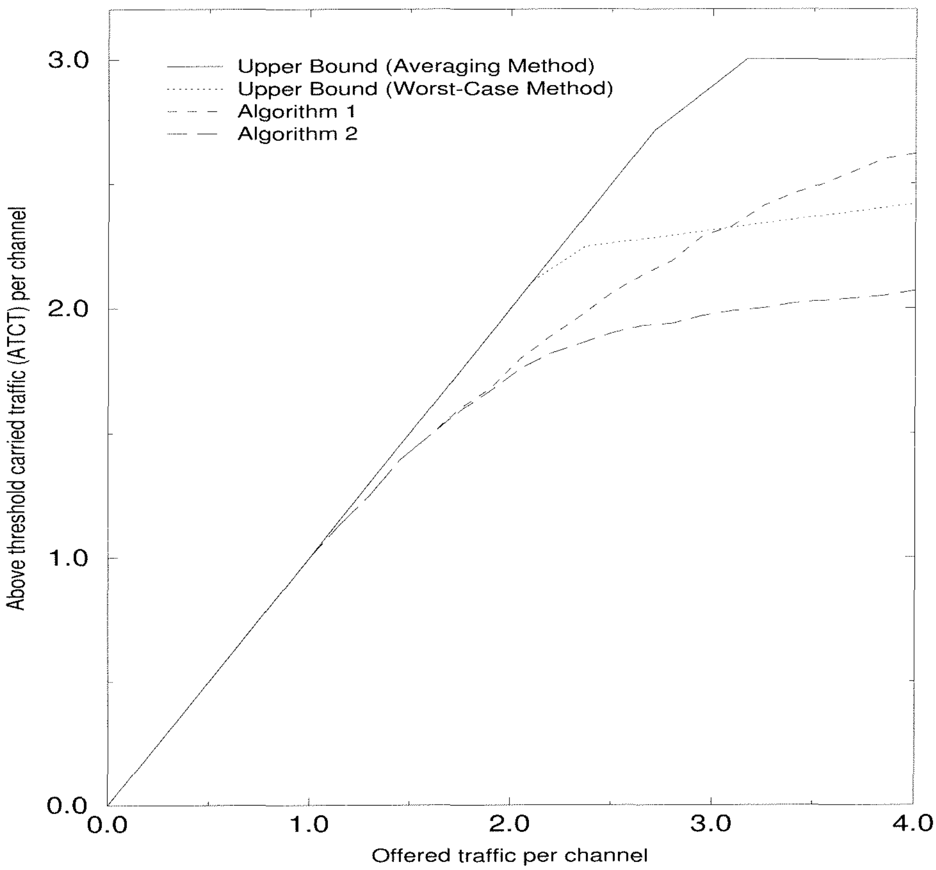


Figure 6.5: The performance of Algorithms 1 and 2 for a 19-cell hexagonal system with uniform traffic distribution and 100 channels

minimum for all the users using the channel, the change in the above threshold carried traffic is calculated assuming that the new call has been assigned this channel. If the availability falls below the minimum for any user, the channel is assumed unavailable. From all the channels which pass the availability criterion, the channel which results in the maximum increase in the ATCT is selected.

(2)Algorithm 4: Similar to Algorithm 3, except that only the availability for the new user is calculated. A channel is assumed unavailable if this availability is below a minimum. From among the channels which remain, the channel which provides the maximum availability for the new user is chosen.

Algorithm 3 requires a centralized decision making process, which may not be practical for microcellular systems. Algorithm 4 is more practical as the decision as to whether a channel is available in a particular cell is made locally. The simulation for both these algorithms assume that the base stations can determine the power from the interferers in the various channels exactly, as if they knew the exact locations of the users of the channels and the Linnartz model [15] is perfect.

We conducted simulation for these two algorithms for the 19-cell hexagonal system mentioned above, assuming the same values for the various parameters. Our results indicated that the fraction of users who experience availability less than 90 percent, when Algorithm 4 is used, is very small. The results for all the four algorithms are shown in Figure 6.6. The contribution to ATCT from all the users is taken into account. Algorithm 3 seems to perform better than Algorithm 4. Both these measurement based algorithms perform better than the two prediction based algorithms 1 and 2.

6.4 Conclusion

In this chapter, we discussed some methods of determining the set of allowable states. We also discussed some heuristic "measurement-based" algorithms. These algorithms are particularly useful for microcellular systems which have a large number of cells, because of the decentralized nature of the algorithms and because such algorithms

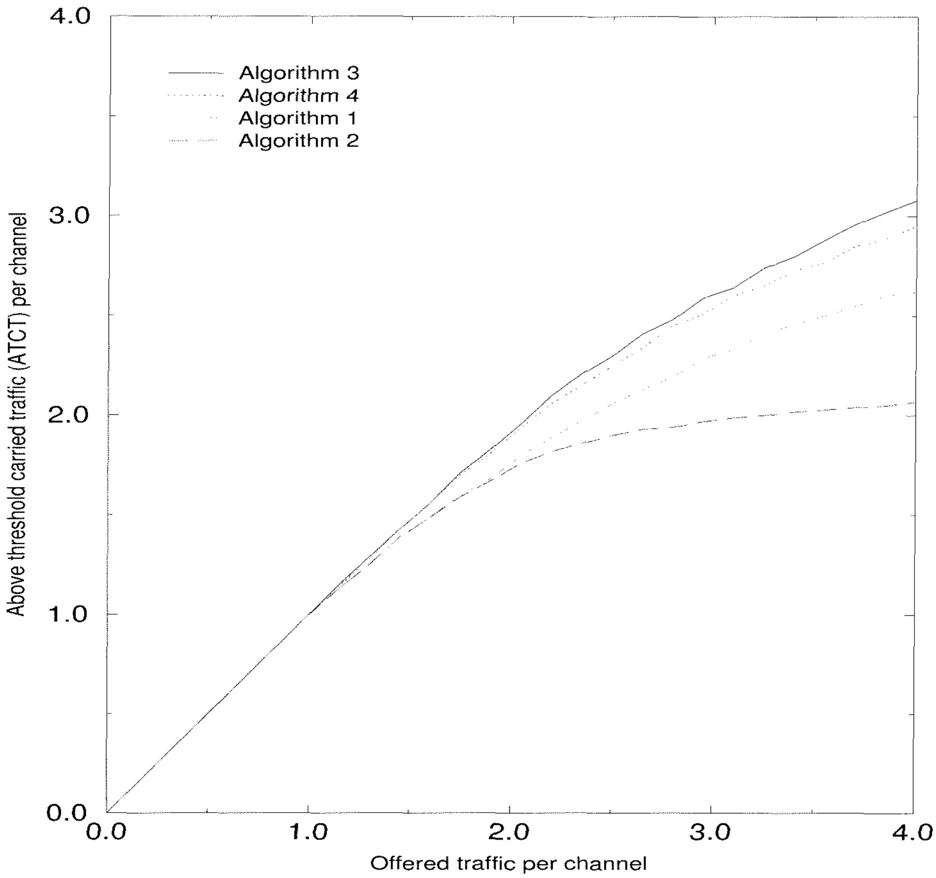


Figure 6.6: The performance of Algorithms 1,2,3 and 4 for the 19-cell hexagonal system with uniform traffic distribution and 100 channels

don't require any frequency planning. Also such algorithms are adaptive in the sense that they maintain acceptable call quality even when channel conditions change. More work needs to be done in this respect.

Conclusions and Future Work

The work in this thesis on channel assignment algorithms applies to channelized cellular systems, for example, systems using Frequency Division Multiple Access (FDMA) or Time Division Multiple Access (TDMA). It does not apply to systems using Code Division Multiple Access (CDMA). It is still an open question as to which one is "better", and a lot of work needs to be done before one can come close to answering the question.

In this thesis, we studied and compared the performance of various channel assignment algorithms (CAAs). One of the ways to increase the capacity of cellular systems is through the use of microcells and our aim in this thesis was to develop decentralized, low-complexity CAAs which are suitable for microcellular systems. We analyzed the performance of a very simple CAA which we call the Timid DCAA, in the limiting case of a large number of channels. We showed, under a plausible mathematical hypothesis, that the algorithm is asymptotically optimal, where "asymptotically" refers to a system with a large number of channels. This is very surprising as there are algorithms of very high complexity which provably do not have the property of asymptotic optimality. However, the Timid DCAA appears to give a good performance only when the number of channels is large. We developed an algorithm, called the Modified DCAA, which retains the simplicity of the Timid DCAA, but unlike the Timid DCAA, can be expected to give a good performance even when the number of channels is small. Such an algorithm is extremely suitable for microcellular systems.

The cellular system model we considered does not include some important features which we describe below. One of the most important features we excluded in our model is user motion. Also, power control methods are being proposed to increase the capacity and improve the quality of cellular systems. Although we have not talked about power control in the thesis, the results regarding the Timid DCAA can be extended to a system using power control in the following manner. We can assume that the area served by a base station is divided up into a very large number of smaller cells, such that when power control methods are used, the power transmitted to and from anywhere in a small cell can be assumed to be constant. Then the set of allowable states can be determined, depending upon the propagation model, the power control algorithm used, and the call quality which we want to provide. Since the cellular system model in Chapter 3 allows the set of allowable states to be arbitrary for a system, all the results regarding the Timid DCAA apply. However the Modified DCAA needs to be changed further to incorporate power control, along with user motion, and its performance relative to other algorithms under these new conditions need to be studied. Also, measurement-based algorithms of the type described in Chapter 6 are also being proposed. These become particularly important for systems where it becomes extremely difficult to model the propagation channels which vary a lot in space and time. Also, cellular systems might be offering different grades of service to customers of different types. This will greatly influence the power control algorithms, as well as the channel allocation algorithms. We think that the major thrust now should be to develop measurement-based algorithms which incorporate power control, take user motion into account, and which offer different grades of service to customers of different types, and to study the performance and complexity of such algorithms.

Appendix A Proof of Some Lemmas Related to TDCAA

Lemma A.1 *There exists an n_0 such that for all $n > n_0$ and all $i \in \Gamma$, $P_{b_i} < P_{th}$.*

Proof: From (3.8) and (3.13), we have

$$\begin{aligned} P_i' &= \sum_{l=0}^{n-1} \frac{1}{1+l} \nu_i'(l) \\ &= \frac{1}{n^{\delta_i}}. \end{aligned}$$

Also,

$$\Gamma = \{i : i \in I_N, \delta_i > \delta_{\text{th}}\}.$$

We will only consider the case when $i \in \Gamma$, i.e.,

$$P_i' < \frac{1}{n^{\delta_{\text{th}}}}.$$

Let ϵ be any positive number less than δ_{th}. Let γ_i be the probability that $|S_i'| < \lfloor n^\epsilon \rfloor$. Now

$$\begin{aligned} P_i' &= \sum_{l=0}^{n-1} \frac{1}{1+l} \nu_i'(l) \\ &\geq \sum_{l=0}^{\lfloor n^\epsilon \rfloor - 1} \frac{1}{1+l} \nu_i'(l) \\ &\geq \frac{1}{\lfloor n^\epsilon \rfloor} \sum_{l=0}^{\lfloor n^\epsilon \rfloor - 1} \nu_i'(l) \\ &= \frac{1}{\lfloor n^\epsilon \rfloor} \gamma_i'. \end{aligned}$$

Therefore,

$$\begin{aligned}
\gamma_i' &\leq \lfloor n^{\epsilon} \rfloor P_i' \\
&\leq \frac{n^{\epsilon}}{n^{\delta_i}} \\
&= \frac{1}{n^{\delta_i - \epsilon}} \\
&< \frac{1}{n^{\delta_{\text{th}} - \epsilon}}, \qquad \text{since} \qquad \delta_i > \delta_{th}.
\end{aligned}$$

P_{b_i} is the probability that a call arriving in cell i gets blocked. Also S_i' is the set of channels in S' which can accept a call in cell i. We can say that

$$\begin{aligned}
P_{b_i} &\leq \Pr(|S_i'| = 0) \\
&\leq \Pr(|S_i'| < \lfloor n^{\epsilon} \rfloor) \\
&= \gamma_i \\
&< \frac{1}{n^{\delta_{\text{th}} - \epsilon}}.
\end{aligned}$$

Hence we can say that there exists an n_0 such that for all $n > n_0$ and all $i \in \Gamma$,

$$P_{b_i} < P_{\text{th}}.$$

∎

Lemma A.2 Let Λ be a set defined as follows:

$$\Lambda = \{j : j \in \Omega, (w_{\max} - w_j) > \epsilon_1\}. \tag{A.1}$$

There exists an n_1 such that for all $n > n_1$ and for all $j \in \Lambda$,

$$\pi_j' < \epsilon_2.$$

Here ϵ_1 and ϵ_2 are arbitrary positive numbers.

Proof: From (3.18), we have for $j \in \Omega$,

$$\pi'_j = v_j n^{w_j} / \left(\sum_{l \in \Omega} v_l n^{w_l} \right).$$

Let $v_{\min} = \min(v_j : j \in \Omega)$ and $v_{\max} = \max(v_j : j \in \Omega)$. Then we can write

$$\begin{aligned} \pi'_j &\leq (v_{\max} n^{w_j}) / (v_{\min} n^{w_{\max}}) \\ &= (v_{\max}/v_{\min})\, n^{w_j - w_{\max}}. \end{aligned}$$

Therefore for $j \in \Lambda$, we have

$$\pi'_j \leq \frac{v_{\max}}{v_{\min}} n^{-\epsilon_1}.$$

It is clear from the above expression that there exists an n_1 such that

$$\frac{v_{\max}}{v_{\min}} n_1^{-\epsilon_1} < \epsilon_2.$$

Hence, there exists an n_1 such that for all $n > n_1$ and for all $j \in \Lambda$,

$$\pi'_j < \epsilon_2.$$

∎

Lemma A.3 $\mathbf{J}^T[\mathbf{r} - \mathbf{AP}] \leq \mathbf{f}^T[\mathbf{r} - \mathbf{AP}'] + \delta_{\text{th}} r + P_{\text{th}} r - \mathbf{f}^T \mathbf{A} \zeta$.

Proof: $(r_i - (\mathbf{AP})_i)$ gives the blocked traffic in cell i when the TDCAA is used. Since P_{b_i} is the blocking probability in cell i, we can also write for all $i \in I_N$,

$$r_i - (\mathbf{AP})_i = r_i P_{b_i}. \tag{A.2}$$

Now,

$$\begin{aligned} \Delta &= \sum_{i=1}^{N} (r_i - (\mathbf{AP})_i) \qquad \text{(from(3.20))} \\ &= \mathbf{J}^T[\mathbf{r} - \mathbf{AP}] \end{aligned} \tag{A.3}$$

$$
\begin{aligned}
&= (\mathbf{f}^T + \delta^T)[\mathbf{r} - \mathbf{AP}] && \text{(from(3.16))} \\
&= \mathbf{f}^T[\mathbf{r} - \mathbf{AP}] + \delta^T[\mathbf{r} - \mathbf{AP}] \\
&= \mathbf{f}^T[\mathbf{r} - \mathbf{AP}'] - \mathbf{f}^T\mathbf{A}\zeta + \delta^T[\mathbf{r} - \mathbf{AP}] && \text{(from(3.11)).} \qquad \text{(A.4)}
\end{aligned}
$$

Also,

$$
\begin{aligned}
\delta^T[\mathbf{r} - \mathbf{AP}] &= \sum_{i \in I_N} \delta_i(r_i - (\mathbf{AP})_i) \\
&= \sum_{i \in I} \delta_i r_i P_{b_i} \qquad \text{(from (A.2))} \\
&= \sum_{i \in \Gamma} \delta_i r_i P_{b_i} + \sum_{i \in \Gamma'} \delta_i r_i P_{b_i}. \qquad \text{(A.5)}
\end{aligned}
$$

Now, $P_{b_i} \leq 1$ and $\delta_i \leq 1$ for all $i \in I_N$.

Also for $i \in \Gamma$, from Lemma A.1 in Appendix A, we know that

$$P_{b_i} < P_{\text{th}},$$

and for $i \in \Gamma'$, we have by definition,

$$\delta_i \leq \delta_{\text{th}}.$$

Therefore from (A.5), we have

$$
\begin{aligned}
\delta^T[\mathbf{r} - \mathbf{AP}] &= \sum_{i \in \Gamma} \delta_i r_i P_{b_i} + \sum_{i \in \Gamma'} \delta_i r_i P_{b_i} \\
&\leq P_{\text{th}} \sum_{i \in \Gamma} \delta_i r_i + \delta_{\text{th}} \sum_{i \in \Gamma'} r_i P_{b_i} \\
&\leq P_{\text{th}} \sum_{i \in \Gamma} r_i + \delta_{\text{th}} \sum_{i \in \Gamma'} r_i \\
&\leq P_{\text{th}} \sum_{i \in I_N} r_i + \delta_{\text{th}} \sum_{i \in I_N} r_i \\
&\leq P_{\text{th}} r + \delta_{\text{th}} r,
\end{aligned}
$$

where $r = \sum_{i \in I_N} r_i$ is the total offered traffic per channel.

Hence, from (A.3) and (A.4), we have

$$\mathbf{J}^T[\mathbf{r}-\mathbf{AP}] \leq \mathbf{f}^T[\mathbf{r}-\mathbf{AP}'] + \delta_{\text{th}} r + P_{\text{th}} r - \mathbf{f}^T\mathbf{A}\zeta.$$

∎

Lemma A.4 *For any probability distribution* $\mathbf{Q} = (q_1, q_2, \ldots, q_m)$ *on* Ω*,*

$$\mathbf{f}^T\mathbf{AQ} \leq w_{\max}.$$

Proof: From (3.19), we know that

$$\mathbf{w}^T = \mathbf{f}^T\mathbf{A}.$$

Also,

$$w_{\max} = \max_{j\in\Omega}\{w_j\}.$$

Therefore,

$$\begin{aligned}
\mathbf{f}^T\mathbf{AQ} &= \mathbf{w}^T\mathbf{Q} \\
&= \sum_{j\in\Omega} w_j q_j \\
&\leq \sum_{j\in\Omega} w_{\max} q_j \\
&= w_{\max} \sum_{j\in\Omega} q_j \\
&= w_{\max}.
\end{aligned}$$

∎

Lemma A.5 *For the distribution* $\mathbf{P}'$ *on* Ω *obtained for the system* X'_n*,*

$$\mathbf{f}^T\mathbf{AP}' \geq (w_{\max} - \epsilon_1)(1 - m\epsilon_2).$$

Proof: From (A.1), we have

$$\Lambda = \{j : j \in \Omega, (w_{\max} - w_j) > \epsilon_1\},$$

and from (3.19), we have

$$\begin{aligned} \mathbf{f}^T\mathbf{A}\mathbf{P}' &= \mathbf{w}^T\mathbf{P}' \\ &= \sum_{j\in\Omega} w_j\pi'_j \\ &= \sum_{j\in\Lambda} w_j\pi'_j + \sum_{j\in\Omega\backslash\Lambda} w_j\pi'_j. \end{aligned} \tag{A.6}$$

Let η denote the probability of channel β being in a state in Λ and let η' denote the probability of channel β being in a state in $(\Omega\backslash\Lambda)$. Also we have proved in Lemma A.2 in Appendix A, that for $j \in \Lambda$,

$$\pi'_j < \epsilon_2,$$

and by definition, for $j \in \Omega\backslash\Lambda$,

$$w_j \geq w_{\max} - \epsilon_1. \tag{A.7}$$

Therefore,

$$\begin{aligned} \eta &= \sum_{j\in\Lambda} \pi'_j \\ &\leq \sum_{j\in\Lambda} \epsilon_2 \\ &\leq \sum_{j\in\Omega} \epsilon_2 \\ &= m\epsilon_2. \end{aligned}$$

Since

$$\eta + \eta' = 1,$$

we have

$$\eta' \geq 1 - m\epsilon_2.$$

Therefore, from (A.6), we have

$$\begin{aligned}
\mathbf{f}^T\mathbf{A}\mathbf{P}' &= \sum_{j\in\Lambda} w_j\pi_j' + \sum_{j\in\Omega\backslash\Lambda} w_j\pi_j' \\
&\geq \sum_{j\in\Omega\backslash\Lambda} w_j\pi_j' \\
&\geq (w_{\max} - \epsilon_1) \sum_{j\in\Omega\backslash\Lambda} \pi_j' \qquad\qquad \text{(from (A.7))} \\
&\geq (w_{\max} - \epsilon_1)(1 - m\epsilon_2). \qquad\qquad \blacksquare
\end{aligned}$$

Appendix B $x(t)$ - An "Almost Markov" Process

Consider the cellular system S described in section 3.2, Chapter 3. $x(t)$ denotes the single-channel system containing channel β. We will now show that the single channel β does not describe a Markov process. Consider two states j and k such that $a_k = a_j + e_i$ for some $i \in I_N$. Here e_i is a vector of length N with a 1 in the i^{th} position and a 0 everywhere else. Then the transition rate from state j to state k given the present state of the system is not independent of the past states of the system. It depends on the state of S' which is not independent of the past states of the system $x(t)$ given the present state of the system $x(t)$.

To be precise, let A be the event that channel β is in a state $a_j \in \Omega_i$ and let B be some information regarding the past states of the channel. Let $\lambda'_{A,B}$ be the call arrival rate in cell i to channel β when both A and B are given and λ'_A be the arrival rate in cell i to channel β when only A is given. Then we have

$$\begin{aligned}\lambda'_{A,B} &= \text{Call arrival rate in cell } i \times \Pr(\text{ A new call in cell } i \text{ is given} \\ &\quad \text{to channel } \beta \mid A, B) \\ &= \lambda_i n \sum_{l=0}^{n-1} \frac{1}{1+l} \Pr(|S'_i| = l \mid A, B), \end{aligned} \tag{B.1}$$

and

$$\begin{aligned}\lambda'_A &= \text{Call arrival rate in cell } i \times \Pr(\text{ A new call in cell } i \text{ is given to channel } \beta \mid A) \\ &= \lambda_i n \sum_{l=0}^{n-1} \frac{1}{1+l} \Pr(|S'_i| = l \mid A). \end{aligned} \tag{B.2}$$

Since, in general, B gives us some information about the state of the system S', we can say that

$$\Pr(|S'_i| = l | A, B) \neq \Pr(|S'_i| = l | A).$$

Hence,

$$\lambda'_{A,B} \neq \lambda'_A.$$

But for the process described by channel β to be a Markov process, the call arrival rate in cell i to channel β should be independent of the past states of the channel given its present state, i.e., it should be independent of B given A. That is, we should have

$$\lambda'_{A,B} = \lambda'_A.$$

This is not true in general and hence the process $x(t)$ described by the single channel β is not a Markov process.

Example:

Consider the 3-cell system of Figure 1.1. Suppose the system has 2 channels. Suppose the call arrival rate in each cell is 1 per second and the call departure rate is 1 per second. We denote the state of the system by a two-tuple (i,j) where i denotes the state of channel 1 and j denotes the state of channel 2 . Let $P_{i,j}$ be the probability of the system being in a state (i,j). Let A be the event that channel 1 is in state 1 and B be the event that channel 1 has been in state 1 for a "long" time. Let λ'_A be the call arrival rate in cell 1 for channel 1 when A is given and let $\lambda'_{A,B}$ be the call arrival rate in cell 1 for channel 1 when both A and B are given. Then from (B.1) and (B.2), we have

$$\lambda'_A = \sum_{l=0}^{1} \frac{1}{1+l} \Pr(|S'_1| = l|A), \tag{B.3}$$

and

$$\lambda'_{A,B} = \sum_{l=0}^{1} \frac{1}{1+l} \Pr(|S'_1| = l|A,B). \tag{B.4}$$

By solving for the $P_{i,j}$'s, we get

$$\begin{aligned} P_{1,1} &= 166/3717, \\ P_{1,2} &= 178/3717, \\ P_{1,3} &= 178/3717, \end{aligned}$$

$$
\begin{aligned}
P_{1,4} &= 166/3717, \\
P_{1,5} &= 102/3717.
\end{aligned}
$$

Let π_1 be the probability that channel 1 is in state 1. Then we have

$$
\begin{aligned}
\pi_1 &= \sum_{j=1}^{5} P_{1,j} \\
&= 790/3717.
\end{aligned}
$$

When a channel is in states 2,4 or 5, it cannot accept a call in cell 1. Therefore,

$$
\begin{aligned}
\Pr(|S_1'| = 0|A) &= \frac{P_{1,2} + P_{1,4} + P_{1,5}}{\pi_0} \\
&= 223/395,
\end{aligned}
$$

and

$$
\begin{aligned}
\Pr(|S_1'| = 1|A) &= \frac{P_{1,1} + P_{1,3}}{\pi_0} \\
&= 172/395.
\end{aligned}
$$

Therefore, from (B.3), we get

$$
\begin{aligned}
\lambda_A' &= 1 \times \Pr(|S_1'| = 0|A) + \frac{1}{2} \times \Pr(|S_1'| = 1|A) \\
&= 223/395 + 1/2 \times 172/395 \\
&= 309/395. \qquad \text{(B.5)}
\end{aligned}
$$

Now suppose that it is given that channel 1 has been in state 1 for a "long" time. Given that channel 1 is in state 1, channel 2 describes a Markov process whose state diagram is given in Figure B.1. We should note that this state diagram is obtained from the state diagram for the entire system of 2 channels by keeping only those states which have channel 1 in state 1 and keeping only the transitions occurring between these states. The transition rates remain unchanged. By "long" time, we mean time large enough for the channel 2 to have attained equilibrium given that channel 1 has been in state 1. Then the probabilities of the system being in the states $(1,1), (1,2), (1,3), (1,4)$, and $(1,5)$ are given by the probabilities of the Markov process whose state diagram is shown in Figure B.1.

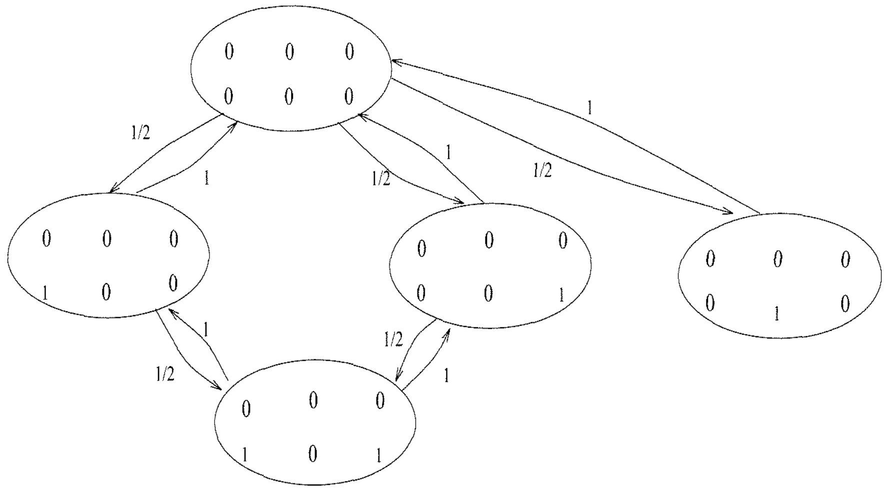


Figure B.1: Markov process described by channel 2 given that channel 1 is in state 1

Let us denote by $P'_{1,j}$ the probability of the system being in state $(1,j)$ given that channel 1 has been in state 1 for a "long" time. Then by solving for the state probabilities of the Markov process whose state diagram is in Figure B.1, we get

$$\begin{aligned} P'_{1,1} &= 4/11, \\ P'_{1,2} &= 2/11, \\ P'_{1,3} &= 2/11, \\ P'_{1,4} &= 2/11, \\ P'_{1,5} &= 1/11. \end{aligned}$$

Therefore, we have

$$\begin{aligned} \Pr(|S'_1| = 0|A,B) &= P'_{1,2} + P'_{1,4} + P'_{1,5} \\ &= 5/11, \end{aligned}$$

and

$$\begin{aligned}\Pr(|S_1'| = 1|A, B) &= P_{1,1}' + P_{1,3}' \\ &= 6/11.\end{aligned}$$

Hence, from (B.4), we get

$$\begin{aligned}\lambda_{A,B}' &= 1 \times \Pr(|S_1'| = 0|A, B) + \frac{1}{2} \times \Pr(|S_1'| = 1|A, B) \\ &= 5/11 + 1/2 \times 6/11 \\ &= 8/11. \end{aligned} \tag{B.6}$$

As is clear from (B.5) and (B.6),

$$\lambda_{A,B}' \neq \lambda_A'.$$

Hence the call arrival rate in cell 1 for channel 1 when it is in state 0 is a function of the time the channel has spent in state 0. Hence the resulting process is not Markov.

Claim: Although the process $x(t)$ is not Markov, its equilibrium distribution is the same as that of a Markov process which has the same state space as $x(t)$ and for which the transition rates between two states are given by the transition rates between the corresponding states of the process $x(t)$ conditioned only on the present state of the system.

Proof: For the single-channel system that we are considering, let a_j and a_k be the vectors corresponding to the states j and k. Let e_i be a $N \times 1$ vector with a 1 in the i^{th} position and a 0 everywhere else. Let $\lambda_{j,k}(t)$ be the transition rate from state j to state k at time t given that the system is in state j at time t. Then we have

$$\lambda_{j,k}(t) = \begin{cases} \lambda_i n P_{i,j}'(t), & \text{if } a_k = a_j + e_i \text{ for some } i \in I_N, \\ \mu & \text{if } a_j = a_k + e_i \text{ for some } i \in I_N, \\ 0 & \text{otherwise.} \end{cases}$$

Here, $P'_{i,j}(t)$ is the probability that a call in cell i will be given to channel β given that channel β is in a state $j \in \Omega_i$ at time t. We should note that

$$\frac{1}{n} \leq P'_{i,j}(t) \leq 1.$$

Let us define $\nu'_{i,j}(l,t)$ as follows:

$$\nu'_{i,j}(l,t) = \Pr(|S'_i| = l \text{ at time } t \mid \text{channel } \beta \text{ is in a state } j \in \Omega_i \text{ at time } t).$$

Then we can write

$$P'_{i,j}(t) = \sum_{l=0}^{n-1} \frac{1}{1+l} \nu'_{i,j}(l,t).$$

Since the entire system describes a Markov process, we know that

$$\lim_{t \to \infty} \nu'_{i,j}(l,t) = \nu'_{i,j}(l).$$

Here, $\nu'_{i,j}(l)$ is the equilibrium probability that $|S'_i| = l$ given that channel β is in a state $j \in \Omega$.

Therefore,

$$\begin{aligned} \lim_{t \to \infty} P'_{i,j}(t) &= \sum_{l=0}^{n-1} \frac{1}{1+l} \nu'_{i,j}(l) \\ &= P'_{i,j}. \end{aligned}$$

Let

$$\lambda_{j,k} = \begin{cases} \lambda_i n P'_{i,j}, & \text{if } a_k = a_j + e_i \text{ for some } i \in I_N, \\ \mu & \text{if } a_j = a_k + e_i \text{ for some } i \in I_N, \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\lim_{t \to \infty} \lambda_{j,k}(t) = \lambda_{j,k}.$$

We should note that for $\lambda_{j,k}(t) \neq 0$,

$$\lim_{t \to \infty} \frac{\lambda_{j,k}(t)}{\lambda_{j,k}} = 1.$$

Also since the entire system describes a Markov process, we can say that the single channel-system has a steady state distribution, i.e.,

$$\lim_{t \to \infty} \Pr(\text{ channel } \beta \text{ is in state } j \text{ at time } t) = \pi_j.$$

Also we should note that for $\lambda_{j,k}(t) \neq 0$, we have

$$0 < \min\{\mu, \lambda_i : i \in I_N\} \leq \lambda_{j,k}(t) \leq \max\{\mu, \lambda_i n : i \in I_N\}.$$

Since the single-channel system satisfies all the properties which should be satisfied by a process for Theorem 2.2 in chapter 2 to be applicable, our claim follows.

Appendix C Reversible Processes

A stochastic process $X(t)$ is said to be reversible if $(X(t_1), X(t_2), \ldots, X(t_n))$ has the same distribution as $(X(t-t_1), X(t-t_2), \ldots, X(t-t_n))$ for all $t_1, t_2, \ldots, t_n, t$ [12]. Let S be the set of states which a stationary Markov chain can be in and let $p(j,k)$ be the transition rate from state j to state k for $j, k \in S$. Then the stationary Markov chain is reversible if and only if there exists a collection of positive numbers $\pi_j, j \in S$, summing to unity that satisfy the following detailed balanced conditions for all $j, k \in S$:

$$\pi_j p(j,k) = \pi_k p(k,j)$$

When there exists such a collection , it is the equilibrium distribution of the process.

Kolmogorov's Criteria for Reversibility

Kolmogorov's criteria allows us to establish the reversibility of a process directly from the transition rates.

A stationary Markov chain is reversible if and only if its transition probabilities satisfy

$$p(j_1, j_2)p(j_2, j_3) \ldots p(j_{n-1}, j_n)p(j_n, j_1) = p(j_1, j_n)p(j_n, j_{n-1}) \ldots p(j_2, j_1)$$

for any finite sequence of states $j_1, j_2, \ldots, j_n \in S$.

Cellular System and Reversibility

Consider a single-channel cellular system with N cells and m states. Let $I_N = (1, 2, \ldots, N)$ denote the set of cells and $\Omega = (1, 2, \ldots, m)$ denote the set of states of the system. Let the departure rate of a call be μ in all the cells. Let Ω_i be the set of states in which a channel can accept a call in cell i. Suppose the rate at which calls come in cell i is independent of which particular state of $j \in \Omega_i$ the channel is in. Let us denote it by λ_i. Also assume that the resulting Markov chain is reflexive, that is, if the transition rate from state i to state j is non-zero , then the transition rate from state j to state i is also non-zero. In that case we can verify that the transition rates satisfy Kolmogorov's criteria for reversibility and hence the process is reversible.

Let $\mathbf{A}$ be the $N \times m$ incidence matrix such that

$$a_{i,j} = \begin{cases} 1, & \text{if there is a call in progress in cell } i \text{ when the system is in state } j; \\ 0, & \text{otherwise.} \end{cases}$$

Then it can be verified that

$$\pi_j = \prod_{i:a_{i,j}=1} r_i \pi_1.$$

Here, $r_i = \frac{\lambda_i}{\mu}$ and π_1 is the probability of the all-zero state.

Since,

$$\sum_{j \in \Omega} \pi_j = 1,$$

we can write

$$\pi_j = \frac{\prod_{i:a_{i,j}=1} r_i}{1 + \sum_{j \in \Omega \setminus \{1\}} \prod_{i:a_{i,j}=1} r_i}.$$

One of the advantages of having a reversible process is that the equilibrium probability of the states of the system can be written down in terms of the transition rates in a simple manner which is very useful for analysis. The state probabilities are said to have a product form solution.

Appendix D An Interesting Theorem

In this section, we will state and prove an interesting theorem regarding probability distributions which maximize a given function. This came up while proving some of the results in Chapter 3.

Theorem D.1 Consider a set $I_N = \{1, 2 \ldots, N\}$. Let $0 \le c_i \le 1$ be a number associated with each $i \in I_N$. Let S be a collection of subsets of I_N with the property that if $S_j \in S$, then all subsets of S_j are also elements of S. Let us denote $S = \{1, 2, \ldots, m\}$. Let $\mathbf{A} = (a_{i,j})$ be an $N \times m$ matrix defined as follows:

$$a_{i,j} = \begin{cases} 1, & \text{if } i \in S_j; \\ 0, & \text{otherwise.} \end{cases} \tag{D.1}$$

Let

$$\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} \qquad \text{and} \qquad \mathbf{J} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \tag{D.2}$$

Consider a probability distribution $\mathbf{P} = (p_1, p_2, \ldots, p_m)$ on S satisfying

$$\mathbf{AP}^T \le \mathbf{c}. \tag{D.3}$$

Let χ be the set of all $\mathbf{P}$'s which satisfy (D.3).

Let $\mathbf{t} = (t_1, t_2, \ldots, t_N)^T = \mathbf{AP}^T$. Let

$$T_{\mathbf{P}} = \mathbf{J}^T \mathbf{AP}^T,$$

and

$$T_{\max} = \max_{P \in \chi} \{T_{\mathbf{P}}\}.$$

Define

$$f_i = \begin{cases} 1, & \text{if } t_i < c_i; \\ x_i, \quad (0 \le x_i \le 1) & \text{if } t_i = c_i. \end{cases} \tag{D.4}$$

Let $w = (w_1, w_2, \ldots, w_m)$ be defined by

$$w_j = \sum_{i: i \in S_j} f_i, \tag{D.5}$$

and

$$w_{\max} = \max\{w_j\}. \tag{D.6}$$

For a given $\mathbf{P} \in \chi$*, if there exist* x_i*'s such that for all* j*'s,*

$$w_j < w_{\max} \Rightarrow p_j = 0 \tag{D.7}$$

then

$$T_{\mathbf{P}} = T_{\max}$$

Proof: Let $\mathbf{P} = (p_1, p_2, \ldots, p_m) \in \chi$ be a probability distribution such that there exists a collection of x_i's such that $w_j < w_{\max} \Rightarrow p_j = 0$. Let $\mathbf{f}$ be a vector defined from the corresponding f_i's:

$$\mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{pmatrix}.$$

Let $\mathbf{P}' = (p'_1, p'_2, \ldots, p'_m) \in \chi$ be any other probability distribution. Then for all i, we have from (D.3), $(\mathbf{AP}^T)_i \le c_i$ and $(\mathbf{AP'}^T)_i \le c_i$.

We would like to show that

$$\mathbf{J}^T\mathbf{AP}^T \ge \mathbf{J}^T\mathbf{AP'}^T.$$

Now from (D.2), (D.3) and (D.4), we have

$$\mathbf{J}^T[\mathbf{c} - \mathbf{A}\mathbf{P}^T] = \mathbf{f}^T[\mathbf{c} - \mathbf{A}\mathbf{P}^T]. \tag{D.8}$$

Also, since $\mathbf{c} \geq \mathbf{A}\mathbf{P}'^T$, we have

$$\mathbf{J}^T[\mathbf{c} - \mathbf{A}\mathbf{P}'^T] \geq \mathbf{f}^T[\mathbf{c} - \mathbf{A}\mathbf{P}'^T]. \tag{D.9}$$

Therefore,

$$\begin{aligned} T_{\mathbf{P}} - T_{\mathbf{P}'} &= \mathbf{J}^T\mathbf{A}\mathbf{P}^T - \mathbf{J}^T\mathbf{A}\mathbf{P}'^T \\ &= -\mathbf{J}^T[\mathbf{c} - \mathbf{A}\mathbf{P}^T] + \mathbf{J}^T[\mathbf{c} - \mathbf{A}\mathbf{P}'^T] \\ &\geq -\mathbf{f}^T[\mathbf{c} - \mathbf{A}\mathbf{P}^T] + \mathbf{f}^T[\mathbf{c} - \mathbf{A}\mathbf{P}'^T] \qquad \text{(from (D.8) and (D.9))} \\ &= \mathbf{f}^T\mathbf{A}\mathbf{P}^T - \mathbf{f}^T\mathbf{A}\mathbf{P}'^T. \end{aligned}$$

Now, since $\mathbf{P}$ satisfies (D.7), we have

$$\mathbf{f}^T\mathbf{A}\mathbf{P}^T = w_{\max},$$

and

$$\mathbf{f}^T\mathbf{A}\mathbf{P}'^T \leq w_{\max}.$$

Hence,

$$T_{\mathbf{P}} \geq T_{\mathbf{P}'}.$$

∎

Appendix E Some Results Regarding the Above Threshold Carried Traffic

E.1 Upper Bound on the Above Threshold Carried Traffic and its Achievability

Consider a single-channel cellular system Y_1 with N cells. Let the corresponding system with n channels be denoted by Y_n. A single channel in the system can be in $M = 2^N$ states. Let us denote the set of states by $I_M = \{1, 2, \ldots, M\}$. Let us define a $N \times M$ matrix $A = (a_{ij})$ as

$$a_{ij} = \begin{cases} 1, & \text{if a user in cell } i \text{ is using the channel when the channel is in state } j; \\ 0, & \text{otherwise.} \end{cases}$$

For any state $j \in I_M$, let x_{ij} denote the probability that when the system is in state j, the signal to interference ratio for user in cell i is above a certain threshold. Let $X = (x_{ij})$ be an $N \times M$ matrix. Let J_N be an all-one vector of length N . Let us define a vector $b = (b_1, b_2, \ldots, b_M)$ as

$$b = J_N X.$$

We can look upon b_j as the revenue that the system gets when a channel is in state j.

Let us denote the offered traffic per channel in each cell by $\mathbf{r} = (r_1, r_2, \ldots, r_N)$. Let $s = (s_1, s_2, \ldots, s_M)$ be a probability distribution on the set of states $I_M = \{1, 2, \ldots, M\}$. Let $S(\mathbf{r})$ be the set of all probabiltiy distributions on I_M such that $s \in S(\mathbf{r})$ if and only if

$$sA^T \leq \mathbf{r}.$$

Let us define $T(r)$ as follows:

$$T(\mathbf{r}) = \max\{sb^T : s \in S(\mathbf{r})\}. \tag{E.1}$$

Let S_n denote the set of all feasible states for the system Y_n. By a feasible state, we mean a M-vector $m = (m_1, m_2, \ldots, m_M)$ such that m_j indicates the number of channels in state j and $\sum_j m_j = n$. We have

$$E(mA^T) \leq \mathbf{r}n. \tag{E.2}$$

Therefore,

$$E(m) \in nS(\mathbf{r}). \tag{E.3}$$

(E.2) is the constraint that the carried traffic in any cell is less than or equal to the offered traffic in that cell.

Let us denote by $T_n(\mathbf{r})$ the maximum possible revenue for the system Y_n when the offered traffic per channel is $\mathbf{r}$. We will show that for $n = 1, 2\ldots$,

$$\frac{1}{n}T_n(\mathbf{r}) \leq T(\mathbf{r}). \tag{E.4}$$

We will also show that

$$\lim_{n\to\infty} \frac{1}{n}T_n(\mathbf{r}) = T(\mathbf{r}). \tag{E.5}$$

This is the main result. What the above results say is that the function $T(\mathbf{r})$ is the normalised upper bound on the revenue the system can get for a given offered traffic distribution r and that this upper bound is asymptotically tight.

Theorem E.1 *The function $T(\mathbf{r})$ is non-decreasing, continuous and convex.*

Proof: Since the set over which the maximum occurs is increasing in r, it follows that $T(\mathbf{r})$ is nondecreasing. To prove that $T(\mathbf{r})$ is convex, let r_1 and r_2 be arbitrary demand vectors, and let p and q be positive real numbers such that $p + q = 1$. Let $s_1 \in S(r_1)$ achieve $T(r_1)$, i.e., $s_1b^T = \max\{sb^T : sA^T \leq r_1\}$ and similarly let $s_2 \in S(r_2)$ achieve

$T(r_2)$, i.e., $s_2 b^T = \max\{sb^T : sA^T \leq r_2\}$. Let $s = ps_1 + qs_2$. Then $s \in S(pr_1 + qr_2)$, and

$$sb^T = (ps_1 + qs_2)b^T = ps_1 b^T + qs_2 b^T = pT(r_1) + qT(r_2).$$

Therefore,

$$T(pr_1 + qr_2) \geq pT(r_1) + qT(r_2).$$

∎

Theorem E.2 *Let $T_A(n\mathbf{r})$ be the expected value of the function mb^T, when a particular algorithm A is used and when the offered traffic is $n\mathbf{r}$. Then,*

$$\frac{1}{n}T_A(n\mathbf{r}) \leq T(\mathbf{r})$$

Proof: We have by definition, and by linearity of the expectation operator,

$$T_A(n\mathbf{r}) = E(mb^T) = E(m)b^T.$$

Therefore, from (E.3) and (E.2), we have

$$\begin{aligned} T_A(n\mathbf{r}) = E(m)b^T &\leq \max\{sb^T : s \in nS(\mathbf{r})\} \\ &= n \max\{sb^T : s \in S(\mathbf{r})\} \\ &= nT(\mathbf{r}). \end{aligned}$$

Since this is true for all algorithms A, (E.4) follows.

E.2 Asymptotic Optimality of $T(\mathbf{r})$

Theorem E.3

$$\lim_{n\to\infty} \frac{1}{n}T_n(\mathbf{r}) = T(\mathbf{r}).$$

Proof: To prove this, we will prove the following inequality which together with (4) will complete the proof :

$$\lim_{n\to\infty} \frac{1}{n} T_n(\mathbf{r}) \geq T(\mathbf{r}).$$

Consider the following algorithm A. Choose $s = (s^1, s^2, \ldots, s^M) \in S(\mathbf{r})$ arbitrarily and select a sequence $s_n = (s_n^1, s_n^2, \ldots, s_n^M) \in S_n(\mathbf{r})$ such that

$$\lim_{n\to\infty} \frac{s_n}{n} = s.$$

This can be done by choosing for example s_n such that $\lfloor ns^j \rfloor \leq s_n^j \leq \lfloor ns^j \rfloor + 1$.

Consider a fixed channel assignment scheme in which s_n^j channels are "reserved" for state j or are of service type j. Let $\tau(s_n^j)$ be the average number of channels from these allocated s_n^j channels which are in state j. Suppose the call arrival requests are processed as follows: When a call comes in cell i, it is sent to the channels reserved for state j with probability s^j/r_i, if $a_{i,j} = 1$. (This can be done for all the states and for all the cells simultaneously because $\sum_i s^j a_{i,j} \leq r_i$.) If one of those channels can accept the call, the call is accepted. Otherwise the call is blocked.

Let us denote the revenue function for this algorithm by $T_n^\star(\mathbf{r})$. Then we have

$$T_n^\star(\mathbf{r}) = E(mb^T) = \sum_{j=1}^{M} E(m_j) b_j \geq \sum_{j=1}^{M} \tau(s_n^j) b_j. \tag{E.6}$$

We will prove in the next section that under the operation of the above algorithm A,

$$\lim_{n\to\infty} \frac{\tau(s_n^j)}{ns^j} = 1.$$

If we divide (E.6) by n and take the limit as $n \to \infty$, then we get

$$\begin{aligned} \lim_{n\to\infty} \frac{1}{n} T_n^\star(n\mathbf{r}) &\geq \lim_{n\to\infty} \sum_{j=1}^{M} \frac{\tau(s_n^j)}{n} b_j \\ &= \sum_{j=1}^{n} s^j b_j \end{aligned}$$

$$= \quad sb^T.$$

But since this holds for every $s \in S(\mathbf{r})$, we choose the $s = s(\mathbf{r})$ which maximizes it, i.e., the one for which $s(\mathbf{r})b^T = T(\mathbf{r})$. For this particular choice of s, we have

$$\lim_{n\to\infty} \frac{1}{n} T_n^\star(n\mathbf{r}) = T(\mathbf{r}).$$

This completes the proof.

Theorem E.4 *When algorithm A is used,*

$$\lim_{n\to\infty} \frac{\tau(s_n^j)}{ns^j} = 1.$$

Proof: Suppose $s^j n$ channels are reserved for state j. We will consider those values of n for which $s^j n$ is an integer to avoid complications and we will assume that $s_n^j = s^j n$. With a slight modification, the proof can be made valid for all n. Then if $a_{i,j} = 1$, then the traffic in cell i which goes to this set of channels is $nr_i \times \frac{s^j}{r_i}$, i.e., the traffic in cell i for this set of channels is 1 Erlang per channel. So we have reduced our problem to the following problem:

We have a set of n channels and there are N cells. Each channel can carry a call in any cell, irrespective of other cells which are using it. The offered traffic per channel in each cell is 1 Erlang per channel. Let $x(n)$ be the number of channels which carry a call in all the N cells at a given time. We have to prove that,

$$\lim_{n\to\infty} \frac{E(x(n))}{n} = 1.$$

This follows directly from the Asymptotic Traffic Property of Poisson arrivals proved in [18]. ∎

Bibliography

[1] M. Abromowitz and I. A. Stengun, *Handbook of Mathematical Functions*, Dover Pu. Inc., NY, 1965.

[2] X-R Cao and J. C-I Chuang, " A Set-Theory Framework and Algorithms to the Channel Assignment Problems," in preparation.

[3] J. C-I Chuang," Performance Issues and Algorithms for Dynamic Channel Assignment,"*IEEE J. on Sel. Areas in Com.*, vol. 11, no. 6, August 1993, pp. 955-963.

[4] L. J. Cimini, Jr. and G. J. Foschini, "Distributed Dynamic Algorithms for Dynamic Channel Allocation in Microcellular Systems," Proc. 1992 Vehicular Technology Conference (VTC 1992), pp. 641-644.

[5] J. L. Doob, *Stochastic Processes.* New York: Wiley and Sons, 1953.

[6] D. Everitt and D. Manfield, "Performance Analysis of Cellular Mobile Radio Communication Systems with Dynamic Channel Assignment," *IEEE J. Sel. Areas Commn.*, vol. 7, pp. 1172-1180, Oct. 1989.

[7] W. Feller, *An Introduction to Probability Theory and its Applications*, Third Edition, John Wiley & Sons, Inc., New York, 1968.

[8] J. Franklin, *Methods of Mathematical Economics*, Springer-Verlag New York Inc., 1980.

[9] A. Gamst, "Some Lower Bounds for a Class of Frequency Assignment Problems," *IEEE Trans. Veh. Tech.*, vol. VT-35, pp. 8-14, Feb. 1986.

[10] William K. Hale,"Frequency Assignment: Theory and Applicaions, "*Proceedings of the IEEE*, vol. 68, No. 12, December 1980.

[11] Wireless Personal Communications: Part 1,*IEEE Journal on Sel. Areas in Com.*, August 1993.

[12] F. P. Kelly, *Reversibility and Stochastic Networks*, John Wiley & Sons Ltd., NY, 1979.

[13] F. P. Kelly, "Blocking Probabilities in Large Circuit-Switched Networks," *Adv. in Appl. Prob.* , vol. 18,no. 3, pp. 473-505, 1986

[14] F. P. Kelly,"Stochastic Models of Computer Communication Systems," *J. Royal Stat. Soc. B*, Vol. 7, no. 3, 1985, pp. 379-395.

[15] Linnartz, Jean-Paul M. G. "Exact Analysis of the Outage Probability in Multiple-User Mobile Radio," *IEEE Trans. on Comm.,* vol. 40, no. 1 (Jan. 1992), pp. 20-23

[16] V. H. MacDonald, " The Cellular Concept," *The Bell System Tech. J.*, vol. 58, pp. 15-41, Jan. 1979.

[17] R. J. McEliece, E. Rodemich and S. K. Deora,"Why Channel Assignment Algorithms are Asymptotically Smart,"*Proc. 31st Annual Allerton Conference on Communication, Control, and Computing*, 1993, pp. 207-216.

[18] R. J. McEliece and K. Sivarajan, "Performance Limits for Channelized Cellular Telephone Systems," *IEEE Trans. Inform. Theory*, Jan. 1994, pp. 21-34.

[19] R. J. McEliece and K. Sivarajan, "Heavy Traffic Performance of a Class of Channel Assignment Algorithms," Proc. Third International Symposium on Personal,Indoor and Mobile Radio Communications (PIMRC '92). Boston : October 19-21,1992, pp. 550-555.

[20] R. J. McEliece and K. Sivarajan, "Maximizing Marginal Revenue In Generalized Blocking Service Networks," Proc. 1992 Allerton Conference on Communication, Control and Computing, pp. 455-464.

[21] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison-Wesley Publishing Co., 1987.

[22] K. N. Sivarajan, *Spectrum Efficient Frequency Assignment for Cellular Radio*, Doctoral Dissertation, California Institute of Technology, 1990.

[23] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchum," Channel Assignment in Cellular Radio," Proc. IEEE VTC '89, pp. 846-850, 1989.