Carleton University

95.495 Honors Project

A Self Organizing Social Insect Model for
Frequency Allocation in Cellular Telephone Networks

By: Michael Lawlor

Advisor: Dr. Tony White, Computer Science

August 9th 2002
Abstract

The cellular telephone networks of today are becoming larger and more complex on a daily basis. The systems in place to manage these networks have themselves become so complicated and intractable that no single human being can comprehend them. This paper finds a solution to the problem of frequency allocation in cellular networks that does not rely on a large centralized management system. Through the use of mobile agents whose behavior is modeled after such social insects as ants and wasps, a simpler, more distributed solution to this problem is proposed. In this solution, the agents have no knowledge of the global structure of the network. They follow a simple set rules which govern their actions in response to local stimulus. It is through these local interactions with each other and their environment that the overall organization of the entire system emerges. Thus, using swarms of these mobile agents, a network may become self-organized, adaptive, and may exhibit intelligent behavior that is far beyond the comprehension of the agents who populate it.
Acknowledgements

This project would not have been possible without the contributions of the following people:

Dr. Franz Oppacher who first introduced me to swarm intelligence, and through who’s lectures I developed an appreciation for its power and elegance.

Dr. Tony White who’s guidance, patience, and advice helped make this project both interesting and achievable.

The equations and algorithms used for many of the vital solutions in this project are taken from Chapter 3 of Swarm Intelligence by Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. See the References section for a full listing of this work.
# Table of Contents

## Chapter 1: Introduction
- 1.1 Objectives ................................ 6
- 1.2 Cellular Telephone Networks .......... 8
- 1.3 Swarm Intelligence ...................... 11

## Chapter 2: Project Components
- 2.1 A Simplified Cellular Network Simulation .. 13
- 2.2 Mobile Agents and the Cellular Network .... 16
- 2.3 Mobile Agent Division of Labor .......... 17
- 2.4 Adaptive Task Allocation ............... 19

## Chapter 3: Implementation
- 3.1 The Cellular Network ................... 21
- 3.2 The Swarm Intelligence .................. 23
- 3.3 The GUI .................................. 25

## Chapter 4: Experimentation
- 4.1 Experiment Overview and Justification ..... 26
- 4.2 Experimental Results ..................... 30
- 4.3 Observations .............................. 40

## Chapter 5: Conclusions
- 5.1 Explanation of Results .................. 43
- 5.2 Future Improvements ..................... 48
- 5.3 Summary .................................. 50
Table of Figures

Figure 1: Frequency Reuse .......................... 9
Figure 2: Call Instantiation Process ............... 10
Figure 3: Initial Channel Configuration .......... 15
Figure 4: Grid Directions ........................... 15
Figure 5: Example Grid Cell ....................... 26
Figure 6: Line Graph – 2 Calls Per Second ....... 31
Figure 7: Line Graph – 3 Calls Per Second ...... 32
Figure 8: Line Graph – 4 Calls Per Second ...... 33
Figure 9: Line Graph – 5 Calls Per Second ...... 34
Figure 10: Line Graph – 6 Calls Per Second .... 35
Figure 11: Line Graph – 7 Calls Per Second .... 36
Figure 12: Line Graph – 8 Calls Per Second .... 37
Figure 13: Line Graph – 9 Calls Per Second .... 38
Figure 14: Line Graph – 10 Calls Per Second .... 39

Table of Tables

Table 1: Statistics – 2 Calls Per Second ........... 31
Table 2: Statistics – 3 Calls Per Second .......... 32
Table 3: Statistics – 4 Calls Per Second .......... 33
Table 4: Statistics – 5 Calls Per Second .......... 34
Table 5: Statistics – 6 Calls Per Second .......... 35
Table 6: Statistics – 7 Calls Per Second .......... 36
Table 7: Statistics – 8 Calls Per Second .......... 37
Table 8: Statistics – 9 Calls Per Second .......... 38
Table 9: Statistics – 10 Calls Per Second ......... 39
Chapter 1: Introduction

1.1 Objectives

The cellular telephone networks of today have become a vastly complex and truly global web of heterogeneous services, environments, and implementations. The cellular telephone industry is the fastest growing segment of the communications industry in the world with an average of 150,000 new subscribers every day [Akyildiz, McNair, et al. 1999]. As a consequence, cellular telephone networks have become increasingly complicated and the task of managing them has become entangled by the sheer size and complexity of their programming. Furthermore, it is often necessary for human involvement to solve some of the issues that occur in load management in cellular telephone networks. As these centralized legacy network management systems become larger and more elaborate, the ability of human managers to cope with the problems they encounter diminishes. Thus, a more decentralized approach to these management issues should be considered – one in which the cellular network has the capacity to solve such problems as frequency allocation at its source, instead of consulting a central management program.
The object of this paper is to find a solution to the problem of frequency allocation in a cellular telephone network using swarm intelligence. The project will be considered a success if its solution produces similar or better results than that of the traditional centralized or fixed methods of frequency allocation. The results will be measured mainly by ease of traffic flow over the network, and the percentage of successfully connected calls.

In this project, swarm intelligence in the form of mobile agents will be employed. These agents are released into the cellular network where they will respond to demands and solve problems locally without the need for interference from central management. The inspiration for these agents comes from observations of the swarm behavior of social insects. Each agent follows a simple set of rules that govern its actions in response to environmental stimulus. The agents are completely ignorant of the overall workings of the system; however, when a well designed rule-set is used, their behavior causes a self-organization of the entire system to emerge. It is this emergent property [Johnson, 2001] that makes swarm intelligence a powerful tool for decentralization. The properties of the overall system are not explicitly defined. Instead, it is through the simple agent rule-sets
that the system becomes self organized and its inherent properties emerge.

The concepts of both cellular telephone networks and swarm intelligence are discussed in more detail in the following two sections.

1.2 Cellular Telephone Networks

Cellular telephones are actually extremely sophisticated radios. They use radio frequencies to transmit and receive both voice and digital information. A cellular system for a city-sized region is organized as a grid of cells. Each cell is roughly 10 to 20 square kilometers, and has its own base station consisting of a tower and the necessary radio transmitting equipment. This division into distinct cells allows for the reuse of frequencies over the cellular network. Two cells in the grid may use the same frequency simultaneously as long as they are not directly adjacent to one another. If the same frequency is used in two adjacent cells, the calls will interfere with each other and both calls may be lost.
Any loss of a cell-phone call is referred to as a *blocked call*. In the case where two calls in adjacent cells share the same frequency, it is said that the blocked call was due to frequency interference. A call may also be blocked upon generation if its home cell has no frequencies available to allocate to it. Finally, a call may be blocked while roaming when the cellular phone enters a new cell where no frequencies are available.

In addition to possibly thousands of cells, each city has one MTSO or Mobile Telephone Switching Office. This office is responsible for handling the connections of the cell-phones in its local network to the entire telephone network (both cellular and land-based), and managing the operations of the base stations. When a call is made in a cell of the network, the cellular phone first contacts its
local base station requesting a frequency. The base station then relays this request to the MTSO which responds with a frequency that it has determined to be suitable. The base station relays this response to the cellular phone which assigns itself to the indicated frequency and begins the call. All further communication between the cellular phone and its intended receiver occurs in a similar fashion: cellular phone to base station to MTSO to receiver’s network and back.

An issue inherent in cellular networks is the concept of roaming. Since cellular phones are by nature mobile, they have a velocity associated with them. As a cellular phone moves toward the edge of its cell, its signal strength diminishes with respect to the base station. Once the signal strength falls below a specific threshold, the

Figure 2: Message Sequence Chart - The call instantiation process
base station begins a handoff procedure which associates the cellular phone with a new, closer base station. The new base station must contact the MTSO and request a suitable channel for the roaming call before the cellular phone can be handed over.

In the current scheme of the industry, the MTSO is responsible for handling all of the above mentioned issues. Frequency allocation for new calls as well as handoff calls is all taken care of by the MTSO. This method is extremely centralized, with the MTSO management software having to handle frequency allocation for the entire network. The mobile agents introduced in the following section hope to decentralize this process and thereby diminish the load on the MTSO.

1.3 Swarm Intelligence

Swarm intelligence is a field of artificial intelligence inspired by the behavior of such biological phenomena as swarms of social insects, flocks of birds, schools of fish, and aggregations of slime mold algae. In Swarm Intelligence, an animal society is observed – say a swarm of ants – performing a specific task. The behavior
of the swarm is then modeled, and using this model an artificial simulation is created. Often the simulation of a specific task is altered to fit some real-world application. For example the food foraging behavior of certain species of ants has been used to solve the problem of finding the shortest path between nodes in a network [Johnson, 2001].

The driving force behind swarm intelligence is the concept of emergence. Emergence occurs when an interconnected system of simple elements self-organizes and exhibits a form of intelligent, adaptive, higher level behavior [Johnson, 2001]. Using this principle, swarm intelligence is inherently distributed with a flexibility and robustness that is often absent in more centralized methodologies. It is these properties that have made swarm intelligence so appealing to researchers in such fields as combinatorial optimization, communications networks, and robotics. As computer systems become so large and complex that no one person can hope to comprehend them, swarm intelligence offers an alternative that avoids such massive centralized programming and control.

The key actors in swarm intelligence are the mobile agents. These agents are simple, autonomous, mobile metaphors for the individual organisms involved in such
biological systems as ant swarms or bird flocks. Mobile agents follow a very simple set of rules which govern their interactions with each other and their environment. They take cues from their immediate environment, making decisions and performing specific actions in response to these stimuli. No single agent is aware of the overall structure of the system. Instead, they mindlessly follow their instructions and interact with their surroundings, unaware of the complex swarm-level behavior that emerges as a result of the sum of their simple actions.

Details in regards to the specific function of the swarm intelligence employed in this project are found later on in this paper.

Chapter 2: Project Components

2.1 A Simplified Cellular Network Simulation

Since this project is utmost a demonstration of the applicability of swarm intelligence, it was not necessary to have a highly technical and complicated simulation of a cellular network. Therefore, a simplified version of a cellular network was implemented.
Technically, in a true cellular network each call requires three frequencies: one for outgoing voice, one for incoming voice, and a third data frequency for control communication. In the remainder of this paper, the term *Channel* will be used to refer to this three frequency combination which is required to establish a call. There are a total of 100 different Channels that the network may use.

In this simulation a ten-by-ten grid of hexagonal cells is used as the city or region controlled by a single MTSO. Each cell in the grid contains a vector of channels that are currently allocated to it. This vector contains both available, and in-use channels. Initially, each cell is allocated a group of ten channels. The initial configuration of these groups on the grid (Fig. 3) is such that no two adjacent cells have access to the same frequency. Each cell also contains a vector of all calls currently in progress within its boundaries. The amount of demand currently in a cell is simply the percentage of channels in its channel vector that are currently in use. As the number of calls increases, so does the demand. Conversely, as the number of free channels increases, the demand decreases.
When a call is generated in the network, its position on the grid is determined randomly. Once the position is set, the direction of the call is randomly chosen from the six possible directions (Fig. 4). Next, the holding time of the call is determined. This is the amount of time the call, once connected, will remain active in the network. The holding time of a call is also a random number between 20 and 3020 seconds. The average holding time of generated calls is 180 seconds (3 minutes).

When a roaming call is handed off from its current cell to a neighboring cell, the neighboring cell must allocate one of its channels to it. Once the handoff

---

**Figure 3:** The initial hexagonal grid channel configuration. Each letter corresponds to a specific group of ten channels. Only three rows of the grid are shown here, the remaining seven rows continue this pattern.

**Figure 4:** The six grid directions
procedure has begun, the call’s old cell removes the call from its calls vector and frees the call’s channel. If the roaming call is blocked by the neighboring cell, it is disconnected and the call is lost. This blocking will occur only if the neighboring cell has no available channel to allocate to the roaming call.

All other cellular telephone network protocols and technicalities are overlooked in this project as they have no bearing on frequency allocation and would not change the effectiveness (or ineffectiveness) of the mobile agents in this regard.

2.2 Mobile Agents and the Cellular Network

The Mobile Agents used in this project are able to move about freely within the cellular network from cell to cell. They have a threshold value associated with each cell in the network grid which will be discussed in the next section. Each mobile agent has the ability to allocate new channels to cells in the network where channels are in high demand. The agents are not able to simply “create” new channels out of thin air. Instead, an agent responding to a demand will arrive at the current
cell, and attempt to “steal” a channel from one of its neighbor cells. The agent determines the neighboring cell with the least demand and attempts to take a channel for its current cell. This channel must not be in use by any of the current cell’s neighbors, otherwise any call made on it will be blocked by interference. The agent is not allowed to steal a channel from a cell that has five or less channels allocated to it. This rule is in place to keep less busy cells from getting starved of channels by their busier neighbors.

2.3 Mobile Agent Division of Labor

Modeled after many species of social insects, this project employs mobile agents capable of division of labor. Using this strategy, each agent has a set of response thresholds which govern their responses to different stimuli. These stimuli are usually the demands for certain tasks to be performed in the system. As the demand for a task increases, its level of stimulation increases. If the intensity of a stimulus is at or above the threshold of a specific agent, then that agent will respond to the stimulus by performing the required task. If, on the other
hand the stimulus is below the agent’s threshold, the agent may choose to ignore the stimulus and continue whatever operation it was doing before it was stimulated.

In order to further facilitate the division of labor, the agents of this system are capable of a simple form of learning. Initially each agent is given the same mean threshold value for all stimuli. However, when an agent performs a given task, the agent “learns” that task. What this learning amounts to is a lowering of the agent’s response threshold to stimulation by this task. Therefore, in future instances when the same task is in demand, the agent will be quicker to react to this stimulus. Thus its can be said that the agent has specialized in this particular task. However, this specialization works both ways. The longer an agent goes without performing a specific task, the greater that agent’s threshold for that task becomes. It can therefore be said that an agent “forgets” a task that it hasn’t performed in a while.

This concept of division of labor is used in tandem with an adaptive task allocation system to manage the agents in such a way that the global demand in the network is kept as low as possible. The specifics of this task allocation system are described in the following section.
2.4 Adaptive Task Allocation

As described in section 2.1, each cell in the cellular network grid has a demand associated with it. This demand is equal to the percentage of that cell’s channels that are currently in use. Using adaptive task allocation, once the demand of a cell has reached a certain percentage, this cell begins to stimulate agents in the system. This is accomplished through the system’s allocation algorithm which sweeps through the agents in the system, stimulating each available agent until one responds. The probability that an agent located in cell $i$ responds to a demand $S_j$ in cell $j$ is given by:

$$P_{i,j} = \frac{S_j^2}{S_j^2 + \alpha \theta_{a,j} + \beta d_{i,j}^2}$$  \hspace{1cm} \text{(Equation 1)}$$

$S_j$ is the amount demand or stimulus of cell $j$.
$\theta_{a,j}$ is the response threshold of agent $a$ to a demand from cell $j$.
$d_{i,j}$ is the distance between cell $i$ and cell $j$.
$\alpha$ and $\beta$ are two positive coefficients that weight the influences of the threshold and the distance on the probability.

Once agent $a$ has responded to the demand from cell $j$ it moves to that cell and attempts to allocate it a new channel. Every time agent $a$ allocates itself to cell $j$ for this purpose, it updates its response thresholds as follows:
\[ \theta_{a,j} = \theta_{a,j} - \xi_0 \] 
(Equation 2)

\[ \theta_{a,n(j)} = \theta_{a,n(j)} - \xi_1 \] 
for all \( n(j) \) 
(Equation 3)

\[ \theta_{a,k} = \theta_{a,k} + \varphi \] 
for all \( k \) not equal to \( j \), and not contained in \( n(j) \) 
(Equation 4)

\( n(j) \) is the set containing all of \( j \)'s neighbouring cells.
\( \xi_0 \) is the learning coefficient for cell \( j \).
\( \xi_1 \) is the learning coefficient for cell \( j \)'s neighbors.
\( \varphi \) is the forgetting coefficient for all other cells in the grid.

In words, Eq.(2) reduces the agent’s threshold for the cell it is currently working at. Eq.(3) reduces the agent’s response threshold for the neighbours of the cell it is currently working on. Eq.(4) is responsible for updating the agent’s threshold for all other cells in the grid.

Since these cells are neither the agent’s current cell, nor are they its neighbours, the agent’s response threshold for them increases.

Instead of specializing in different tasks, the agents specialize in the regions of the grid that they frequent the most. In other words, as an agent spends more time working in a specific area, the agent becomes “better” at working in that area and is more likely to respond to its demands. Conversely, if an agent doesn’t spend any time in an area, it becomes “worse” at working in that area. The agent will therefore be less likely to respond to demands from this less familiar area.
This adaptive form of task allocation is highly flexible and robust, allowing the system to adjust to unforeseen fluctuations in demand as they occur.

Chapter 3: Implementation

This project was implemented entirely in JAVA. Included in the implementation: the cellular network simulation, the mobile agent software, and a GUI for visualization and experimentation.

3.1 The Cellular Network

The simulated cellular network is comprised of four classes: Channels (Channel.java), Calls (Call.java), Cells (Cell.java), and the Network Control class (named Antwork.java as it also handles the mobile agents).

Channel

Channel is a simple class consisting of a channel identification number, and a boolean flag indicating whether the channel is available or currently in use.
Call

The Call class keeps track of the speed, direction, holding time, current cell position, and current channel for a call. It also contains a Boolean flag indicating if the call is progress.

Cell

The Cell class is more complex than the above two classes. Each cell keeps track of its position on the grid, its on-screen coordinates for rendering on the GUI, a Vector of the Channels allocated to it, a Vector of the Calls currently in progress in it, its demand, and a boolean flag indicating whether there is currently an agent assigned to it.

The Cell class contains functions responsible for connecting, handing off, and disconnecting calls. It also has functions for adding and removing Channels that agents have allocated to and from it respectively.

Antwork

With respect to the Cellular Network, the Antwork contains the actual 10x10 grid of Cells. The Antwork also contains a number of statistical variables to keep track
of: time, total calls, in progress calls, total blocked calls, the call generation rate (user defined), the number of handoffs, and the number of "stolen" or allocated Channels.

The Antwork is responsible for running the entire cellular network over time. This includes initializing the cells in the grid with their initial channel configurations, maintaining the neighbour relations between cells, handling the handoffs encountered in the network, and generating the calls with their required values (eg. speed, direction, holding time, etc.).

3.2 The Swarm Intelligence

The swarm intelligence portion of the application is made up of the mobile agents (Ant.java) which will be referred to as Ants for the remainder of this paper. The behaviour of the Ants in terms of the cellular network is handled by the Network Control (Antwork.java).

Ant

Each Ant keeps track of its current position in the grid, its response threshold for every cell in the grid,
its learning and forgetting coefficients, and two boolean flags to indicate whether the ant is working, or in transit to a demanding cell.

The Ant class also contains the functions responsible for specialization (learning and forgetting cells), stimulation (deciding whether to respond to stimulus from a cell), and the allocation of frequencies to cells in demand.

**Antwork**

With respect to the swarm intelligence portion of the application, the Antwork keeps a Vector of all the Ants in the network.

The Antwork contains functions responsible for moving idle Ants randomly around the network, and moving stimulated Ants towards their intended destinations. The Antwork also contains the cell allocation algorithm which finds cells with high demand and sweeps through the Ants until one is stimulated enough to respond to this demand.

In this way, the Antwork is responsible for the operations of both the Cellular Network and the Mobile Agents.
3.3 The GUI

The GUI for this application is comprised of two classes: the Network Frame (NetworkFrame.java), and the Network Panel (NetworkPanel.java).

Network Frame

The Network Frame is the container for the entire GUI, and handles the user input, the visualization of the network, and the runtime statistics of the system.

It has two sliders for user defined variables: Number of Ants, and Call Generation Rate. It contains the action buttons for starting, pausing, stopping, and stepping through the simulation. Furthermore, it contains both the Panel depicting the cellular network, and fields containing its runtime statistics.

Network Panel

The Network Panel is a canvas located within the Network Frame upon which the visual representation of the network is displayed.

This visualization includes the outlines of each hexagonal cell in the grid. Within each such hexagon, the number of calls, the total number of channels, and the
demand of the cell are displayed. As the demand in the cell reaches 65%, the demand is coloured yellow. When the demand reaches 90% and above, the demand is coloured red. The Ants are also displayed in their current locations as coloured dots. Idle ants are coloured yellow, ants in transit to high-demand cells are coloured orange, and working ants are coloured red.

Figure 5: An example grid cell. There are currently 4 calls and a total of 10 channels in this cell. The demand is currently 25%. The yellow dot is an ant.

Chapter 4: Experimentation

4.1 Experiment Overview and Justification

Experimentation was accomplished through numerous runs of the application, utilizing the full range of user
defined values for call generation and ant number. For each combination of call generation rate and ant number, the system was run for a total of 1200 ticks. Each tick is considered a second in the lifetime of the network, so each run simulates twenty minutes of network operation. This twenty minute period is a sufficient test of the network’s capabilities because it allows ample time for call generation reach a point of levelling off in the number of in-progress calls. The system then runs for many more cycles after this plateau has been reached. Over these 1200 seconds, all functionality of the system including call handoff, connection, disconnection, and frequency allocation are tested numerous times.

The results of these experiments are organized in a series of tables. Each individual table contains all results for a particular call generation rate. The call generation rates are taken from the range of [2-10] calls per second. Each table includes an initial run with no ants in the system, followed by nine more runs with increasing numbers of ants. The ten runs of the system use the following sequence of ant numbers: 0, 1, 5, 10, 15, 20, 25, 30, 35, 40.

The initial run without any ants is intended to test the ability of the network to handle call generation and
migration when the channels assigned to each cell are fixed. In this case, each cell has ten channels to allocate for calls, and there is no way for a cell to either gain or lose channels to neighbouring cells.

The second run with one ant is merely included as a boundary case. Since the swarm intelligence solution to frequency allocation is intended for a swarm (ie. more than one ant), this run should not produce very good nor very useful results. The case with one ant will most likely produce results similar or worse than that of the initial zero-ant run.

The most meaningful results will be reaped from the 5-40 ant runs. As the number of ants increases, the more cell demands will be satisfied. Furthermore, with the ant’s abilities to specialize in specific regions of the grid, ants will begin to localize themselves to these sections. This local specialty of the ants will decrease the amount of time required for an ant to respond to a demand, and the amount of time it takes for the ant to arrive at the demanding cell.

It should be made clear that the call generation rate value set by the user is not a fixed value for calls generated per second. Instead, this value is a boundary condition for the random amount of calls that may be
generated per second. The actual amount of calls generated per second will be a uniformly distributed random number between zero and the call generation rate value inputted by the user. For example, if the user enters a rate of four calls per second, then every second a random number of calls will be generated between zero and four inclusively.

This randomness of call generation raises some issues with the experimental results. Firstly, the total number of calls generated over the 1200 second period will be different in each run. Also, the number of in-progress calls at any given second will be different for each run. This decision was made to make the network appear more organic, and to create a variety of load situations for the network to deal with. As a result, exact values attained from the experimental runs should not be compared directly. Instead, the overall patterns or trends in these values are the important results. Such results as the percentage of total calls that were blocked, the percentage of blocked calls that were caused by handoffs, and the number of channels “stolen” per ant are the true indicators of the effectiveness of each run. It is through the overall statistical patterns produced by these experiments that any conclusions as to the success or failure of the mobile agent solution will be evaluated.
### 4.2 Experimental Results

In order to fit each table on the page, the headings for each column are abbreviated. Refer to the legend for full column headings. The columns containing more important or remarkable statistics are coloured in blue.

The line graphs visualize how the number of ants in the system affects the percentage of blocked calls. The markers along the lines represent the number of ants used to attain the blocked call percentage at that point. Thus, the first marker is the initial ant case with zero ants, and each consecutive marker thereafter corresponds to a member of the sequence \(\{1, 5, 10, 15, 20, 25, 30, 35, 40\}\).

<table>
<thead>
<tr>
<th>Heading Legend:</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Number of Ants</td>
</tr>
<tr>
<td>C/S</td>
<td>Calls per Second</td>
</tr>
<tr>
<td>TC</td>
<td>Total Calls</td>
</tr>
<tr>
<td>IP</td>
<td>Calls In Progress at end of run</td>
</tr>
<tr>
<td>BC</td>
<td>Total Blocked Calls</td>
</tr>
<tr>
<td>TH</td>
<td>Total Handoffs</td>
</tr>
<tr>
<td>BH</td>
<td>Blocked Handoffs</td>
</tr>
<tr>
<td>SC</td>
<td>Total Number of Stolen Channels</td>
</tr>
<tr>
<td>%B</td>
<td>Percentage of Total Calls that were Blocked</td>
</tr>
<tr>
<td>HB%</td>
<td>Percentage of Blocked calls that occurred during a Handoff</td>
</tr>
<tr>
<td>S/A</td>
<td>Number of Stolen Channels Per Ant</td>
</tr>
</tbody>
</table>
Table 1: Two Calls Per Second – Complete Statistics

<table>
<thead>
<tr>
<th>A</th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>1084</td>
<td>269</td>
<td>0</td>
<td>287</td>
<td>0</td>
<td>0</td>
<td>0.00%</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1030</td>
<td>277</td>
<td>0</td>
<td>295</td>
<td>0</td>
<td>41</td>
<td>0.00%</td>
<td>n/a</td>
<td>41</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1076</td>
<td>292</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>33</td>
<td>0.00%</td>
<td>n/a</td>
<td>6.6</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1034</td>
<td>273</td>
<td>0</td>
<td>276</td>
<td>0</td>
<td>20</td>
<td>0.00%</td>
<td>n/a</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>1065</td>
<td>293</td>
<td>0</td>
<td>276</td>
<td>0</td>
<td>18</td>
<td>0.00%</td>
<td>n/a</td>
<td>1.2</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>1048</td>
<td>284</td>
<td>0</td>
<td>298</td>
<td>0</td>
<td>36</td>
<td>0.00%</td>
<td>n/a</td>
<td>1.8</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>1052</td>
<td>260</td>
<td>0</td>
<td>286</td>
<td>0</td>
<td>33</td>
<td>0.00%</td>
<td>n/a</td>
<td>1.32</td>
</tr>
<tr>
<td>30</td>
<td>2</td>
<td>1051</td>
<td>279</td>
<td>0</td>
<td>285</td>
<td>0</td>
<td>28</td>
<td>0.00%</td>
<td>n/a</td>
<td>0.9333333333</td>
</tr>
<tr>
<td>35</td>
<td>2</td>
<td>1036</td>
<td>266</td>
<td>0</td>
<td>284</td>
<td>0</td>
<td>30</td>
<td>0.00%</td>
<td>n/a</td>
<td>0.857142857</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
<td>1055</td>
<td>274</td>
<td>0</td>
<td>291</td>
<td>0</td>
<td>38</td>
<td>0.00%</td>
<td>n/a</td>
<td>0.95</td>
</tr>
</tbody>
</table>

Two Calls Per Second: Percentage of Blocked Calls

Figure 6
### Table 2: Three Calls Per Second – Complete Statistics

<table>
<thead>
<tr>
<th>A</th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>1478</td>
<td>417</td>
<td>3</td>
<td>419</td>
<td>1</td>
<td>0</td>
<td>0.20%</td>
<td>33.33%</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1466</td>
<td>353</td>
<td>0</td>
<td>420</td>
<td>0</td>
<td>128</td>
<td>0.00%</td>
<td>n/a</td>
<td>128</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1441</td>
<td>346</td>
<td>0</td>
<td>381</td>
<td>0</td>
<td>136</td>
<td>0.00%</td>
<td>n/a</td>
<td>27.2</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>1505</td>
<td>381</td>
<td>1</td>
<td>391</td>
<td>0</td>
<td>204</td>
<td>0.07%</td>
<td>0.00%</td>
<td>20.4</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>1447</td>
<td>410</td>
<td>0</td>
<td>359</td>
<td>0</td>
<td>172</td>
<td>0.00%</td>
<td>n/a</td>
<td>11.46666667</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>1558</td>
<td>414</td>
<td>1</td>
<td>392</td>
<td>0</td>
<td>230</td>
<td>0.06%</td>
<td>0.00%</td>
<td>11.5</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>1382</td>
<td>363</td>
<td>0</td>
<td>359</td>
<td>0</td>
<td>93</td>
<td>0.00%</td>
<td>na</td>
<td>3.72</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1380</td>
<td>371</td>
<td>0</td>
<td>334</td>
<td>0</td>
<td>123</td>
<td>0.00%</td>
<td>n/a</td>
<td>4.1</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
<td>1406</td>
<td>367</td>
<td>0</td>
<td>426</td>
<td>0</td>
<td>174</td>
<td>0.00%</td>
<td>n/a</td>
<td>4.971428571</td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>1485</td>
<td>398</td>
<td>0</td>
<td>393</td>
<td>0</td>
<td>172</td>
<td>0.00%</td>
<td>n/a</td>
<td>4.3</td>
</tr>
</tbody>
</table>

**Figure 7**

Three Calls Per Second: Percentage of Blocked Calls
Table 3: Four Calls Per Second – Complete Statistics

<table>
<thead>
<tr>
<th>A</th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>1792</td>
<td>474</td>
<td>10</td>
<td>483</td>
<td>2</td>
<td>0</td>
<td>0.56%</td>
<td>20.00%</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1764</td>
<td>448</td>
<td>7</td>
<td>452</td>
<td>0</td>
<td>183</td>
<td>0.40%</td>
<td>0.00%</td>
<td>183</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1836</td>
<td>505</td>
<td>6</td>
<td>502</td>
<td>3</td>
<td>447</td>
<td>0.33%</td>
<td>50.00%</td>
<td>89.4</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>1830</td>
<td>456</td>
<td>5</td>
<td>457</td>
<td>1</td>
<td>392</td>
<td>0.27%</td>
<td>20.00%</td>
<td>39.2</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>1795</td>
<td>460</td>
<td>5</td>
<td>425</td>
<td>0</td>
<td>430</td>
<td>0.28%</td>
<td>0.00%</td>
<td>28.66666667</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>1812</td>
<td>473</td>
<td>5</td>
<td>441</td>
<td>2</td>
<td>407</td>
<td>0.28%</td>
<td>40.00%</td>
<td>20.35</td>
</tr>
<tr>
<td>25</td>
<td>4</td>
<td>1776</td>
<td>501</td>
<td>1</td>
<td>443</td>
<td>0</td>
<td>369</td>
<td>0.06%</td>
<td>0.00%</td>
<td>14.76</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>1752</td>
<td>456</td>
<td>2</td>
<td>423</td>
<td>2</td>
<td>354</td>
<td>0.11%</td>
<td>100.00%</td>
<td>11.8</td>
</tr>
<tr>
<td>35</td>
<td>4</td>
<td>1836</td>
<td>480</td>
<td>5</td>
<td>453</td>
<td>2</td>
<td>458</td>
<td>0.27%</td>
<td>40.00%</td>
<td>13.08571429</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>1810</td>
<td>458</td>
<td>7</td>
<td>430</td>
<td>2</td>
<td>418</td>
<td>0.39%</td>
<td>28.57%</td>
<td>10.45</td>
</tr>
</tbody>
</table>

Figure 8: Four Calls Per Second: Percentage of Blocked Calls.
Table 4: Five Calls Per Second – Complete Statistics

<table>
<thead>
<tr>
<th>A</th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>2125</td>
<td>516</td>
<td>50</td>
<td>535</td>
<td>12</td>
<td>0</td>
<td>2.35%</td>
<td>24.00%</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>2216</td>
<td>546</td>
<td>59</td>
<td>520</td>
<td>11</td>
<td>244</td>
<td>2.66%</td>
<td>18.64%</td>
<td>244</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2053</td>
<td>521</td>
<td>20</td>
<td>543</td>
<td>7</td>
<td>598</td>
<td>0.97%</td>
<td>35.00%</td>
<td>119.6</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>2137</td>
<td>545</td>
<td>23</td>
<td>541</td>
<td>5</td>
<td>874</td>
<td>1.08%</td>
<td>21.74%</td>
<td>87.4</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>2136</td>
<td>572</td>
<td>11</td>
<td>582</td>
<td>6</td>
<td>893</td>
<td>0.51%</td>
<td>54.55%</td>
<td>59.53333333</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>2192</td>
<td>558</td>
<td>12</td>
<td>533</td>
<td>1</td>
<td>1036</td>
<td>0.55%</td>
<td>8.33%</td>
<td>51.8</td>
</tr>
<tr>
<td>25</td>
<td>5</td>
<td>2049</td>
<td>561</td>
<td>12</td>
<td>520</td>
<td>3</td>
<td>791</td>
<td>0.59%</td>
<td>25.00%</td>
<td>31.64</td>
</tr>
<tr>
<td>30</td>
<td>5</td>
<td>2130</td>
<td>522</td>
<td>33</td>
<td>568</td>
<td>10</td>
<td>969</td>
<td>1.55%</td>
<td>30.30%</td>
<td>32.3</td>
</tr>
<tr>
<td>35</td>
<td>5</td>
<td>2105</td>
<td>548</td>
<td>16</td>
<td>508</td>
<td>7</td>
<td>843</td>
<td>0.76%</td>
<td>43.75%</td>
<td>24.08571429</td>
</tr>
<tr>
<td>40</td>
<td>5</td>
<td>2176</td>
<td>545</td>
<td>19</td>
<td>555</td>
<td>8</td>
<td>1014</td>
<td>0.87%</td>
<td>42.11%</td>
<td>25.35</td>
</tr>
</tbody>
</table>

Figure 9
### Table 5: Six Calls Per Second: Complete Statistics

<table>
<thead>
<tr>
<th>A</th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
<td>2401</td>
<td>577</td>
<td>95</td>
<td>641</td>
<td>17</td>
<td>0</td>
<td>3.96%</td>
<td>17.89%</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>2452</td>
<td>599</td>
<td>76</td>
<td>577</td>
<td>20</td>
<td>235</td>
<td>3.10%</td>
<td>26.32%</td>
<td>235</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>2466</td>
<td>594</td>
<td>114</td>
<td>663</td>
<td>25</td>
<td>951</td>
<td>4.62%</td>
<td>21.93%</td>
<td>190.2</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>2393</td>
<td>601</td>
<td>57</td>
<td>624</td>
<td>20</td>
<td>1157</td>
<td>2.38%</td>
<td>35.09%</td>
<td>115.7</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>2466</td>
<td>608</td>
<td>79</td>
<td>601</td>
<td>19</td>
<td>1914</td>
<td>3.20%</td>
<td>24.05%</td>
<td>127.6</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>2480</td>
<td>620</td>
<td>56</td>
<td>621</td>
<td>12</td>
<td>2036</td>
<td>2.26%</td>
<td>21.43%</td>
<td>101.8</td>
</tr>
<tr>
<td>25</td>
<td>6</td>
<td>2459</td>
<td>614</td>
<td>53</td>
<td>681</td>
<td>15</td>
<td>2403</td>
<td>2.16%</td>
<td>28.30%</td>
<td>96.12</td>
</tr>
<tr>
<td>30</td>
<td>6</td>
<td>2425</td>
<td>621</td>
<td>32</td>
<td>676</td>
<td>9</td>
<td>2400</td>
<td>1.32%</td>
<td>28.13%</td>
<td>80</td>
</tr>
<tr>
<td>35</td>
<td>6</td>
<td>2469</td>
<td>628</td>
<td>37</td>
<td>611</td>
<td>13</td>
<td>2541</td>
<td>1.50%</td>
<td>35.14%</td>
<td>72.6</td>
</tr>
<tr>
<td>40</td>
<td>6</td>
<td>2353</td>
<td>645</td>
<td>25</td>
<td>646</td>
<td>7</td>
<td>2738</td>
<td>1.06%</td>
<td>28.00%</td>
<td>68.45</td>
</tr>
</tbody>
</table>

### Six Calls Per Second: Percentage of Blocked Calls

![Graph showing the percentage of blocked calls against the number of ants (0-40).](image)

**Figure 10**
Table 6: Seven Calls Per Second: Complete Statistics

<table>
<thead>
<tr>
<th>A</th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7</td>
<td>2687</td>
<td>641</td>
<td>164</td>
<td>659</td>
<td>36</td>
<td>0</td>
<td>6.10%</td>
<td>21.95%</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td>2635</td>
<td>629</td>
<td>137</td>
<td>635</td>
<td>27</td>
<td>278</td>
<td>5.20%</td>
<td>19.71%</td>
<td>278</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2729</td>
<td>600</td>
<td>203</td>
<td>713</td>
<td>63</td>
<td>1082</td>
<td>7.44%</td>
<td>31.03%</td>
<td>216.4</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>2786</td>
<td>625</td>
<td>169</td>
<td>633</td>
<td>47</td>
<td>1720</td>
<td>6.07%</td>
<td>27.81%</td>
<td>172</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>2756</td>
<td>583</td>
<td>142</td>
<td>700</td>
<td>43</td>
<td>2596</td>
<td>5.15%</td>
<td>30.28%</td>
<td>173.0666667</td>
</tr>
<tr>
<td>20</td>
<td>7</td>
<td>2708</td>
<td>651</td>
<td>108</td>
<td>689</td>
<td>26</td>
<td>3133</td>
<td>3.99%</td>
<td>24.07%</td>
<td>156.65</td>
</tr>
<tr>
<td>25</td>
<td>7</td>
<td>2648</td>
<td>645</td>
<td>63</td>
<td>696</td>
<td>11</td>
<td>3241</td>
<td>2.38%</td>
<td>17.46%</td>
<td>129.64</td>
</tr>
<tr>
<td>30</td>
<td>7</td>
<td>2632</td>
<td>651</td>
<td>50</td>
<td>668</td>
<td>14</td>
<td>3126</td>
<td>1.90%</td>
<td>28.00%</td>
<td>104.2</td>
</tr>
<tr>
<td>35</td>
<td>7</td>
<td>2671</td>
<td>686</td>
<td>70</td>
<td>663</td>
<td>15</td>
<td>4229</td>
<td>2.62%</td>
<td>21.43%</td>
<td>120.8285714</td>
</tr>
<tr>
<td>40</td>
<td>7</td>
<td>2659</td>
<td>675</td>
<td>105</td>
<td>674</td>
<td>26</td>
<td>4981</td>
<td>3.95%</td>
<td>24.76%</td>
<td>124.525</td>
</tr>
</tbody>
</table>

Seven Calls Per Second: Percentage of Blocked Calls

![Graph showing the percentage of blocked calls for different numbers of ants]

Figure 11
### Table 7: Eight Calls Per Minute: Complete Statistics

<table>
<thead>
<tr>
<th>A</th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>2929</td>
<td>683</td>
<td>207</td>
<td>728</td>
<td>61</td>
<td>0</td>
<td>7.07%</td>
<td>29.47%</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>2904</td>
<td>645</td>
<td>288</td>
<td>731</td>
<td>73</td>
<td>310</td>
<td>9.92%</td>
<td>25.35%</td>
<td>310</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>2873</td>
<td>678</td>
<td>244</td>
<td>697</td>
<td>59</td>
<td>1973</td>
<td>8.49%</td>
<td>24.18%</td>
<td>394.6</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>2839</td>
<td>639</td>
<td>219</td>
<td>696</td>
<td>57</td>
<td>1993</td>
<td>7.71%</td>
<td>26.03%</td>
<td>199.3</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>2936</td>
<td>661</td>
<td>201</td>
<td>764</td>
<td>61</td>
<td>2920</td>
<td>6.85%</td>
<td>30.35%</td>
<td>194.6666667</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>2920</td>
<td>666</td>
<td>213</td>
<td>752</td>
<td>64</td>
<td>3823</td>
<td>7.29%</td>
<td>30.05%</td>
<td>191.15</td>
</tr>
<tr>
<td>25</td>
<td>8</td>
<td>3000</td>
<td>674</td>
<td>244</td>
<td>788</td>
<td>59</td>
<td>5011</td>
<td>8.13%</td>
<td>24.18%</td>
<td>200.44</td>
</tr>
<tr>
<td>30</td>
<td>8</td>
<td>2922</td>
<td>676</td>
<td>280</td>
<td>713</td>
<td>73</td>
<td>5112</td>
<td>9.58%</td>
<td>26.07%</td>
<td>170.4</td>
</tr>
<tr>
<td>35</td>
<td>8</td>
<td>3021</td>
<td>693</td>
<td>237</td>
<td>789</td>
<td>61</td>
<td>7064</td>
<td>7.85%</td>
<td>25.74%</td>
<td>201.8285714</td>
</tr>
<tr>
<td>40</td>
<td>8</td>
<td>2869</td>
<td>701</td>
<td>133</td>
<td>734</td>
<td>36</td>
<td>6803</td>
<td>4.64%</td>
<td>27.07%</td>
<td>170.075</td>
</tr>
</tbody>
</table>

#### Eight Calls Per Second: Percentage of Blocked Calls

![Eight Calls Per Second: Percentage of Blocked Calls](image)

**Figure 12**
### Table 8: Nine Calls Per Second: Complete Statistics

<table>
<thead>
<tr>
<th></th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9</td>
<td>3108</td>
<td>689</td>
<td>279</td>
<td>755</td>
<td>70</td>
<td>0</td>
<td>8.98%</td>
<td>25.09%</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>3097</td>
<td>636</td>
<td>323</td>
<td>733</td>
<td>88</td>
<td>282</td>
<td>10.43%</td>
<td>27.24%</td>
<td>282</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>3052</td>
<td>687</td>
<td>280</td>
<td>698</td>
<td>62</td>
<td>1174</td>
<td>9.17%</td>
<td>22.14%</td>
<td>234.8</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>3211</td>
<td>673</td>
<td>419</td>
<td>815</td>
<td>102</td>
<td>2370</td>
<td>13.05%</td>
<td>24.34%</td>
<td>237</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>3189</td>
<td>677</td>
<td>432</td>
<td>730</td>
<td>118</td>
<td>3221</td>
<td>13.55%</td>
<td>27.31%</td>
<td>214.7333333</td>
</tr>
<tr>
<td>20</td>
<td>9</td>
<td>3111</td>
<td>686</td>
<td>290</td>
<td>763</td>
<td>93</td>
<td>4038</td>
<td>9.32%</td>
<td>32.07%</td>
<td>201.9</td>
</tr>
<tr>
<td>25</td>
<td>9</td>
<td>3260</td>
<td>693</td>
<td>477</td>
<td>764</td>
<td>118</td>
<td>5990</td>
<td>14.63%</td>
<td>24.74%</td>
<td>239.6</td>
</tr>
<tr>
<td>30</td>
<td>9</td>
<td>3116</td>
<td>698</td>
<td>282</td>
<td>716</td>
<td>69</td>
<td>5663</td>
<td>9.05%</td>
<td>24.47%</td>
<td>188.7666667</td>
</tr>
<tr>
<td>35</td>
<td>9</td>
<td>3178</td>
<td>696</td>
<td>290</td>
<td>810</td>
<td>78</td>
<td>7472</td>
<td>9.13%</td>
<td>26.90%</td>
<td>213.4857143</td>
</tr>
<tr>
<td>40</td>
<td>9</td>
<td>3176</td>
<td>708</td>
<td>254</td>
<td>781</td>
<td>59</td>
<td>8060</td>
<td>8.00%</td>
<td>23.23%</td>
<td>201.5</td>
</tr>
</tbody>
</table>

**Nine Calls Per Second: Percentage of Blocked Calls**

![Graph](image)

**Figure 13**
Table 9: Ten Calls Per Second: Complete Statistics

<table>
<thead>
<tr>
<th>A</th>
<th>C/S</th>
<th>TC</th>
<th>IP</th>
<th>BC</th>
<th>TH</th>
<th>BH</th>
<th>SC</th>
<th>%B</th>
<th>HB%</th>
<th>S/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>3392</td>
<td>694</td>
<td>413</td>
<td>796</td>
<td>65</td>
<td>0</td>
<td>12.18%</td>
<td>15.74%</td>
<td>n/a</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>3584</td>
<td>756</td>
<td>532</td>
<td>851</td>
<td>145</td>
<td>206</td>
<td>14.84%</td>
<td>27.26%</td>
<td>206</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>3325</td>
<td>681</td>
<td>480</td>
<td>752</td>
<td>114</td>
<td>1228</td>
<td>14.44%</td>
<td>23.75%</td>
<td>245.6</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>3379</td>
<td>711</td>
<td>554</td>
<td>743</td>
<td>128</td>
<td>2433</td>
<td>16.40%</td>
<td>23.10%</td>
<td>243.3</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>3507</td>
<td>690</td>
<td>501</td>
<td>773</td>
<td>118</td>
<td>3513</td>
<td>14.29%</td>
<td>23.55%</td>
<td>234.2</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>3381</td>
<td>692</td>
<td>405</td>
<td>843</td>
<td>110</td>
<td>4623</td>
<td>11.98%</td>
<td>27.16%</td>
<td>231.15</td>
</tr>
<tr>
<td>25</td>
<td>10</td>
<td>3450</td>
<td>709</td>
<td>454</td>
<td>730</td>
<td>96</td>
<td>6307</td>
<td>13.16%</td>
<td>21.15%</td>
<td>252.28</td>
</tr>
<tr>
<td>30</td>
<td>10</td>
<td>3517</td>
<td>716</td>
<td>439</td>
<td>787</td>
<td>92</td>
<td>7619</td>
<td>12.48%</td>
<td>20.96%</td>
<td>253.9666667</td>
</tr>
<tr>
<td>35</td>
<td>10</td>
<td>3409</td>
<td>750</td>
<td>311</td>
<td>766</td>
<td>67</td>
<td>7286</td>
<td>9.12%</td>
<td>21.54%</td>
<td>208.1714286</td>
</tr>
<tr>
<td>40</td>
<td>10</td>
<td>3423</td>
<td>749</td>
<td>426</td>
<td>880</td>
<td>116</td>
<td>11669</td>
<td>12.45%</td>
<td>27.23%</td>
<td>291.725</td>
</tr>
</tbody>
</table>

Ten Calls Per Second: Percentage of Blocked Calls

Figure 14
4.3 Observations

In light of the data gathered in these experiments, some observations may be made in regards to the effectiveness of the mobile agents in reducing call blockage.

In the set of results produced using a call generation rate of 2 calls per second, no noticeable difference was observed between the fixed channel case, and any of the cases with ants involved. In none of these cases was there a single blocked call. It can therefore be deduced that with such a low call generation rate, the method of fixed channel configurations is effective enough to produce 100% probability that calls will be connected. In the cases where ants were used, they spent most of their time idle. The number of channels stolen by the ants ranged from 18 to 41 with an average of 28 stolen channels. Over 1200 seconds, this amount of stolen channels had little effect on the overall configuration of the network.

Similarly, in the 3 calls per second set of tests, the fixed channel configuration produced 3 blockages while the most any run using ants produced was 1 blockage. These values are mostly insignificant, and the difference between them is almost negligible. In terms of blocked call
percentages, both the fixed system, and the ant system produced desirable results.

The most ideal conditions for the ants to improve network traffic flow were observed in the range of 4 to 7 calls per second. In most of these cases, an initial increase in blocked calls was observed for the interval between 1 and 10 ants. However; after this initial hiccup, the addition of more ants noticeably decreased the blocked call percentage to levels well below that of the fixed channel system. As more ants were added to these systems, the percentage of blocked calls tended to hit a minimum value and then increase slightly as the number of ants exceeded a specific point. This trend gives the line graphs a slight U-shape with the leftmost endpoint higher than the rightmost one. The largest reduction of blocked call percentage occurred with a call generation rate of 4 calls per second and an ant number of 25. In this case, the percentage of blocked calls is reduced from 0.56% to 0.06%. In other words, the use of 25 ants reduced the percentage of blocked calls by almost 90% in comparison to its corresponding fixed channel configuration.

In the two sets of runs for call generation rates 8 and 9, a less discernable pattern emerges. For all three sets, the use of ants has an unpredictable effect on the
percentage of blocked calls experienced. In some cases, a certain number of ants was seen to slightly increase the percentage of blocked calls, while in other cases it reduced it or had no effect at all. It can be stated that for these cases, the use of ants produced roughly the same percentage of blocked calls as when ants were not used.

Finally, with a call generation rate of 10, the blocked call percentages were the same or worse than those generated by the fixed configuration method. Although the use of ants creates at most 4% more blocked calls, it must be stated that the fixed configuration method produces consistently better results.

Another pattern observed in these experiments was that as the number of ants increases, the number of stolen channels increases as well. There seems to be no limit to the amount of channel shuffling that will occur as more ants are added to the system. As the total amount of calls increases over the different call generation rates, this number of stolen channels gets larger and larger. In fact, with a call generation rate of 10, an ant number of 40, and a call total of 3423, the number of stolen channels hits 11669. This is a remarkably large amount of channel shuffling.
Chapter 5: Conclusions

5.1 Explanation of Results

From the results listed above it may be stated that a distributed, mobile agent solution to the problem of frequency allocation in a cellular network is feasible. In networks where the amount of calls generated is within certain bounds, the addition of mobile agents to the system creates a remarkable improvement in network traffic flow.

For a cases where the number of calls generated is very low (in the case of this experiment, a call generation rate of 1 to 3 would suffice), the network performs with 100% reliability with either the fixed channel configuration method, or with mobile agents. This reliability stems from the fact that with such low call generation rates, there are much fewer calls in progress in the network at any given time. Thus, the overall amount of demand in the network remains low, and there is little problem allocating channels to new or handed-off calls.

The most ideal conditions for the success of the mobile agents occurred when the call generation rates ranged from 4 to 7. In these cases, the increased generation rate means more calls in progress at any given
time. This in turn gives rise to an increase in the demand experienced in the network. It is in these cases that the fixed channel configuration method falls short. Using this method, each cell has a choice of ten channels to use, and no ability to borrow extra channels from its neighbours. As the number of calls increases in the network, the more cells begin to use up their ten available channels. When this occurs, any calls that attempt to be generated in these cells will be blocked. Furthermore, any calls that are handed off into these full cells will be blocked as well.

It is in these cases (4 to 7 calls per second) that the mobile agents have their greatest impact. Although the call generation rate is higher than in the initial three cases, it does not completely overload the network. This means that not every cell in the network is in demand at the same time. Thus, the ants are able to move about throughout the network and can improve the ability high demand cells have for connecting calls.

In the cases where call generation rates range from 8 to 10 calls per second, the effectiveness of the mobile agents decreases. These cases may be considered instances where the network is overloaded. At any given moment, there are more cells in demand in the network than there
are ants to deal with these demands. Thus, the first cells allowed to stimulate agents tend to hold on to them for most of the run cycle. This occurs because the demand of cells in the network is always increasing. Thus once an ant reaches a cell and allocates a channel to it, the ant specializes in that cell. Since the number of calls in this cell is almost always increasing, the demand of this cell rebounds instantaneously, and then ant must again allocate another channel to it. The ant specializes further in this cell, and the cycle continues, each time the ant’s response threshold to this cell diminishes until it basically becomes the property of this cell. In these high generation rate cases, the cells earliest in the ant sweeping cycle retain the most ants, thereby starving those cells that occur later on in the cycle of ants. This leads to a large number of calls being blocked in these later cells which have lost the ability to steal channels from their neighbours.

The number of ants in each system had a noticeable effect on their respective blocked call percentages. The case with one ant made little difference to the percentage of blocked calls in the network. This is because one ant is not enough to create a global pattern or organization in the network. This lone ant responded to demands from
particular cells, and specialized in these localities. By mid-way through the cycle, the ant became so specialized in a specific locality of the grid, that it completely ignored demands from any other area. Thus, one portion of the grid used the ant to shuffle channels around, while the rest of the network remained with a fixed channel configuration.

The addition of five ants to the network almost always resulted in slightly worse blocked call percentages than that of the fixed configuration method. This occurred again because five ants are not enough to create a global self organization in a network grid that contains 100 cells. Similar to the case with one ant, these five ants became over-specialized in particular areas and ignored the demands of other areas. Thus, cells that had no ants in their immediate vicinity were unable to steal channels and suffered a larger amount of blocked calls.

As the number of ants ranged from 10 to 40, a global pattern of self-organization began to emerge in the network. Ants still specialized in specific regions of the grid, but the larger amount of ants meant that these regions could be more evenly distributed throughout it. For each call generation rate used, there was a corresponding number of ants that seemed to work the best. As this call generation rate increased, so did this
corresponding number of ants. This is intuitive because

with more calls comes more demand, and more demand means

more work required from the ants. It follows that more

work can be accomplished by the ants if there are more ants

in the network. This is not to say that the more ants

added to the network, the more effective they were. In

fact, too many ants in the network created an increase in

blocked call percentages. This occurred because too many

channels were needlessly shuffled causing complications

with channel interference and cells being stripped of too

many channels.

Overall these experiments proved that there does exist

a simpler, distributed solution to frequency allocation in

cellular networks. The results were not all positive, but

in the more realistic cases with call generation rates

ranging from low to moderately high, the addition of mobile

agents to the network effectively improved traffic flow,

and reduced the percentage of calls being blocked.
5.2 Future Improvements

Of course there are many improvements that could have been made to this project which would have made it more realistic. Many decisions made in the creation and implementation of this project were based on simplicity and time restrictions.

Utmost among these is the simulation of the cellular network. The simulation used for this project is incredibly simplified, and only reflects a fraction of the behaviour exhibited by a normal cellular telephone network. In future work, more realistic methods for call generation should be used. The rate of call generation should increase during peak hours of the day, and decrease during slower hours. Furthermore, the experiments should be run for a simulated week’s worth of network activity. Due to time constraints, this was not possible for this project.

Secondly, the specialization of ants occurs quite rapidly in scenarios where the ants are in high demand. This specialization may occur too fast and hinder the system’s ability to evenly distribute the ants to all demanding cells. This becomes more of an issue as the call generation rate increases. Here, more work is done by each ant and more specialization occurs. Thus, a region in high
demand early on in the lifetime of the network may require
many ants and may therefore over-specialize a
disproportionate amount of ants in that region. These
over-specialized ants have such low response thresholds for
this region, that they respond to any stimulation from here
and ignore larger demands from other cells. In these high
demand cases, ants become trapped in the regions they first
specialize in and cannot effectively respond to later
demands from other parts of the network.

Finally, the sweep algorithm used to allocate ants to
cells in high demand should be randomized. Currently, this
algorithm sweeps through the cells in sequential order,
finds a cell in high demand, and then sweeps the ants in
sequential order until an ant is stimulated and responds.
In cases where a large percentage of the cells are in high
demand at a particular time, this algorithm unfairly
allocates ants to those cells that happen to be swept
through first. When many cells are in such high demand,
al all of the ants are allocated to cells in the earlier
portion of the sweep cycle, leaving those cells that occur
later in the cycle to starve. Since this occurs every
sweep cycle, the ants become over-specialized in the cells
early in the cycle and their ability to manage channel
allocation for the entire network is lost.
5.3 Summary

Based on the experimental results, and their subsequent analysis, the solution proposed at the outset of this project may be considered a success. In the worst test case results, the mobile agent method performed just slightly worse than the fixed configuration method. In the majority of cases where more reasonable values were used for call generation rates, there was always a threshold number of mobile agents that, when added to the network, improved its ability to handle channel allocation. In some cases the improvement was quite large, while not so in others. However, where improvement was detected, the percentage of blocked calls was reduced to a fraction of its original level.

The results reached in this project will hopefully open the door to further research in distributed, swarm intelligence alternatives in communications network management. Above all else, this project has helped prove that the use of mobile agents with division of labour and adaptive task allocation can be applied to situations once reserved for more complex centralized management systems. A solution similar to the one demonstrated here may one day be considered by a cellular provider or even implemented in
an actual cellular network. As these cellular networks become larger and more complex, the need for them to adapt and self-organize will only increase. Using inspiration taken from the observed behaviour of insect societies – systems finely tuned by millions of years of evolution – a simpler, more organic solution can be applied to cellular networks which have themselves become complex organisms.
References


Appendix A

Files included on disk:

Ant.java
Ant.class
Antwork.java
Antwork.class
Call.java
Call.class
Cell.java
Cell.class
Channel.java
Channel.class
NetworkPanel.java
NetworkPanel.class
NetworkFrame.java
NetworkFrame.class

The main() is found in NetworkFrame.java. To run the application simply type “java NetworkFrame” from the command line. You must have a JAVA compiler installed for this to work.