Cross Layer Optimization:
System Design and Simulation Methodologies

By
Rahul Mahajan

Thesis submitted to the Faculty of
Virginia Polytechnic Institute and State University
In partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
In
Electrical Engineering

Dr. R. M. Buehrer, Chair
Dr. Luiz A. DaSilva
Dr. William Tranter

November 2003
Blacksburg, VA

Keywords: Cross-Layer Simulation Techniques, Mobile Ad-Hoc Networks with Smart Antennas, Medium Access Control with Directional Antennas, Power Control for Successive Interference Cancellation

Cross Layer Optimization:
System Design and Simulation Methodologies

Rahul Mahajan

Abstract

An important aspect of wireless networks is their dynamic behavior. The conventional protocol stack is inflexible as various protocol layers communicate in a strict manner. In such a case the layers are designed to operate under the worst conditions as opposed to adapting to changing conditions. This leads to inefficient use of spectrum and energy. Adaptation represents the ability of network protocols and applications to observe and respond to channel conditions.

Traditional simulation methodologies independently model the physical and higher layers. When multiple layer simulations are required, an abstraction of one layer is inserted into the other to provide the multiple layer simulation. However, recent advances in wireless communication technologies, such as adaptive modulation and adaptive antenna algorithms, demand a cross layer perspective to this problem in order to provide a sufficient level of fidelity. However, a full simulation of both layers often results in excessively burdensome simulation run-times. The benefits and possible parametric characterization issues arising due to the cross-layer integration of lower physical and higher network layers are investigated in this thesis. The primary objective of investigating cross-layer simulation techniques is to increase the fidelity of cross-layer network simulations while minimizing the simulation runtime penalties.

As a study of cross-layer system design a medium access control (MAC) scheme is studied for a MANET wherein the nodes are equipped with smart antennas. Traditional MAC protocols assume the use of omnidirectional antennas. Nodes with directional antennas are capable of transmitting in certain directions only and significantly reduce the chances of collision and increase the effective network capacity. MANETs using omni-directional antennas severely limit system performance as the entire space around a node up to its radio range is seen as a single logical channel. In this research a MAC protocol is studied that exploits space division multiple access at the physical layer. This is a strong example where physical and MAC design must be carried out simultaneously for adequate system performance.

Power control is a very important in the design of cellular CDMA systems which suffer from the near-far problem. Finally, the interaction between successive interference cancellation (SIC) receivers at the physical layer and power control, which is a layer 2 radio resource management issue, is studied. Traffic for future wireless networks is expected to be a mix of real-time traffic such as voice, multimedia teleconferencing, and games and data traffic such as web browsing, messaging, etc. All these applications will require very diverse quality of service guarantees. A power control algorithm is studied, which drives the average received powers to those required, based on the QoS requirements of the individual users for a cellular CDMA system using SIC receivers.

CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1 Introduction: The need for a cross-layer perspective

An important aspect of wireless networks is their dynamic behavior. The conventional protocol stack is inflexible as various protocol layers communicate in a strict manner. In such a case the layers are designed to operate under the worst conditions as opposed to adapting to changing conditions. This leads to inefficient use of spectrum and energy. Adaptation represents the ability of network protocols and applications to observe and respond to channel conditions.

This is especially important in mobile ad-hoc networks (MANETs). A MANET is characterized by no fixed infrastructure, it is formed spontaneously without any pre-planning. In addition to the inherently dynamic physical channel caused by shadowing, scattering, etc., the system must adapt to the dynamics arising due to the mobility of the nodes and constantly changing membership. A MANET must be able to operate in isolation and cannot rely on any infrastructure based services.

A MANET consists of nodes that communicate with each other using multi-hop routes. Numerous routing protocols address the problem of establishing and maintaining the routes in a dynamic network topology. However, most routing protocols are designed with less emphasis on the issues at lower layers like the variable link capacity at the physical layer and the fluctuating contention level at the MAC layer. By exploiting lower layer information through a cross-layer design concept, performance benefit may be obtained. Consider the cross-layer concept shown in Figure 1. At the physical layer, channel estimation is performed to obtain the instantaneous SNR of a link, which affects the data rate chosen, which in turn affects the transmission delay. The routing protocol then makes a routing decision based on the delay associated with each link. The routing decisions in turn affect the network load distribution and impact the lower layer parameters. Thus the performance of the layers is inter-related.

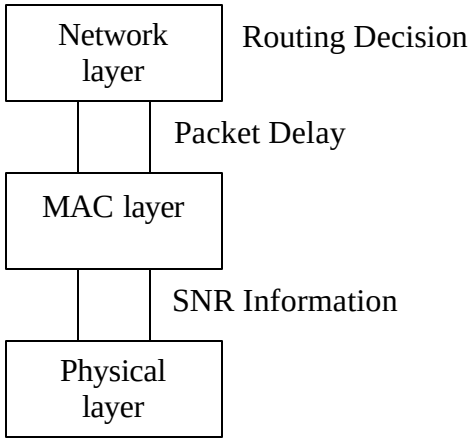

Figure 1 Cross-Layer Design Concept

Traditional simulation methodologies independently model the physical and higher layers. When multiple layer simulations are required, an abstraction of one layer is inserted into the other to provide the multiple layer simulation. However, as described, recent advances in wireless communication technologies, such as adaptive modulation and adaptive antenna algorithms, demand a cross-layer perspective to this problem in order to provide a sufficient level of fidelity. However, a full simulation of both layers often results in excessively burdensome simulation run-times. The benefits and possible parametric characterization issues arising due to the cross-layer integration of lower physical and higher network layers are investigated in chapter 2.

The primary objective of investigating cross-layer simulation techniques is to increase the fidelity of cross-layer network simulations while minimizing the simulation runtime penalties. Specifically, three different kinds of cross-layer simulation approaches are examined within this report:

- Baseline – Network Simulator Only
- Abstract Physical Layer Simulation
- Full Physical Layer Simulation

In the baseline approach, the network is simulated without consideration to the dynamic physical channel, merely using the facilities provided within a network simulation tool such as OPNET. In the abstract physical layer method, a simulation of the physical layer is performed, from which parameters are measured and inserted into the network simulation. In the full physical layer simulation, the complete, dynamic physical layer is implemented using a co-simulator for the physical layer. Two different co-simulation tools are utilized as part of this project: MATLAB and C.

These approaches are then compared for fidelity and for simulation run-time. As the network only implementation is expected to have the shortest run-time, it is used as the benchmark for comparing simulation run-times across the different approaches. For fidelity, the full physical layer simulation is used as the benchmark. A case study of cross-layer simulation to demonstrate the approaches developed chapter 2 is presented in chapter 3.

As a study of cross-layer system design, a medium access control (MAC) scheme is studied in chapter 4 for a MANET wherein the nodes are equipped with smart antennas. Traditional MAC protocols assume the use of omnidirectional antennas. Nodes equipped with directional antennas have the potential to significantly reduce the chances of collision and increase the effective network capacity. MANETs using omni-directional antennas severely limit system performance as the entire space around a node up to its radio range is seen as a single logical channel. In this research a MAC protocol is studied that exploits space division multiple access at the physical layer. This is a strong example where physical and MAC design must be carried out simultaneously for adequate system performance.

Power control is a very important in the design of cellular CDMA systems which suffer from the near-far problem. In chapter 5 the interaction between successive interference cancellation receivers at the physical layer and outer-loop power control, which is a layer 2 radio resource management issue, is studied.
Traffic for future wireless networks is expected to be a mix of real-time traffic such as voice, multimedia teleconferencing, games and data traffic such as web browsing, messaging, etc. All these applications will require very diverse quality of service guarantees. In this chapter, a power control algorithm is studied, which drives the average received powers to those required, based on the QoS requirements of the individual users for a cellular CDMA system using SIC receivers.

This thesis is organized as follows:

- A description of the simulation methodologies used for cross-layer systems and results obtained for cross-layer simulations is discussed in Chapter 2.
- A case study to demonstrate the cross-layer simulation approaches described in the preceding section is presented in Chapter 3.
- Chapter 4 discusses a MAC scheme for MANETs equipped with smart antennas as an example of cross-layer system design.
- A power control scheme for cellular CDMA systems using successive interference cancellation receivers with varying QoS requirements is studied in Chapter 5.
- Chapter 6 concludes this thesis with a summary of the key results, conclusions and future work possibilities.

2 Cross-layer Simulation Methodologies

2.1 Overview

Traditional simulation methodologies independently model the physical and higher layers. When multiple layer simulations are required, an abstraction of one layer is inserted into the other to provide the multiple layer simulation. However as described, recent advances in wireless communication technologies, such as adaptive modulation and adaptive antenna algorithms, demand a cross-layer perspective to this problem in order to provide a sufficient level of fidelity. Previous literature on cross-layer simulation methodologies is very sparse. Reference [15] focused on the effects of physical layer modeling on the performance of higher layer networking protocols and demonstrated the importance of physical layer modeling even if the higher later protocols do not directly interact with the physical layer. Future work suggested validation of physical layer modeling against wireless ad hoc networks and the impact of power control or smart antennas on the overall performance of a MANET.

In this chapter the inclusion of smart antenna algorithms into a network simulation implemented in OPNET is explored as an example of a cross-layer scenario. Smart antennas are playing an increasingly important role in wireless networks wherein SINR can be greatly increased in otherwise interference-limited networks. OPNET is the defacto standard for network simulations in the industry. Thus in addition to serving as a test case for the development of cross-layer simulation techniques, the integration of smart antennas into OPNET is also of long-term value for use as a simulation testbed.

However, OPNET is an event driven simulation tool. Since physical layer modeling is time-driven, OPNET by itself does not provide a convenient platform for investigation. Thus, techniques are explored for integrating traditional physical layer simulation, written in MATLAB or C, into OPNET in order to perform the necessary cross-layer simulations.

Simulation of the physical layer is traditionally one of the most computationally intensive operations in a simulation. The primary objective of this research is the investigation of methods to increase the fidelity of cross-layer network simulations while minimizing the simulation runtime penalties. Specifically, three different kinds of cross-layer simulation approaches are examined:

- Baseline Simulation (OPNET default)
- Abstracted Physical Layer
- Full Physical Layer Simulation (MATLAB and C).

2.1.1 Baseline Simulation

In the baseline simulations, performance is measured using the default OPNET physical layer processing. In order to approximate the effects of the physical layer for wireless networks, OPNET employs a 14-stage computational pipeline to process transmitted packets as shown in Figure 2. Each pipeline stage computes the value of a physical layer parameter(s) and adjusts an appropriate parameter in the packet for use in subsequent stages of the pipeline. A listing of each stage along with a brief description follows.

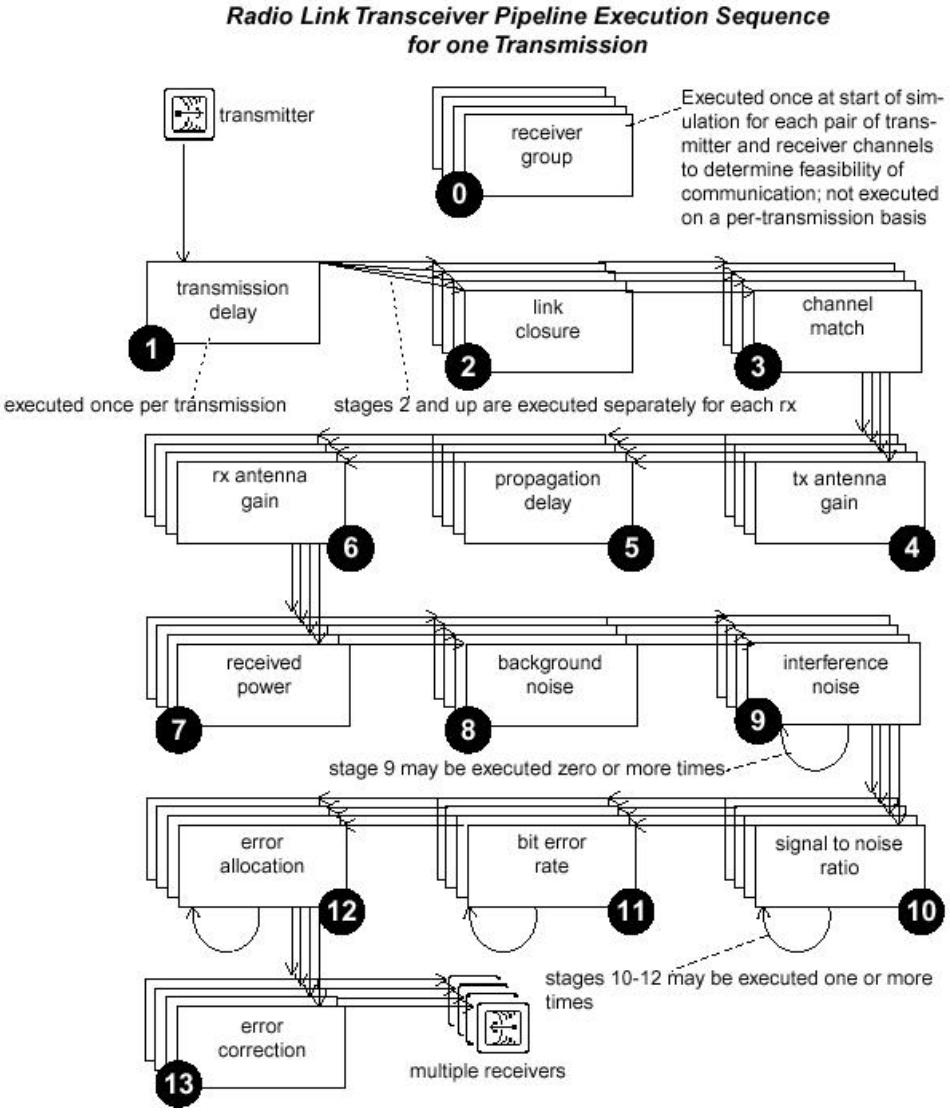

Figure 2 Radio Link Transceiver Pipeline Execution Sequence [1]

- *receiver group*: Evaluates connectivity between each radio transmitter channel and each radio receiver channel in the network. This stage determines the potential for communication between the transmitter channel and the receiver

channel. The purpose of this stage is to filter out ineligible receiver channels with respect to a particular transmitter channel in order to improve simulation performance.

- *transmission delay*: Computes the time required for the radio transmitter of interest to completely process and transmit the packet. The computed transmission delay corresponds to the simulated time interval separating the beginning of transmission of the first bit and the end of transmission of the last bit of the packet.

- *link closure*: Determines the connectivity between a radio transmitter and a radio receiver by invoking the closure model immediately after completion of the transmission delay model on a dynamic basis.

- *Channel match*: Determines the compatibility between a radio transmitter channel and a radio receiver. This model considers a transmitter channel and receiver channel to be compatible only if their common channel characteristics match. The channel attributes that are analyzed include frequency, bandwidth, data rate, spreading code and modulation.

- *tx antenna gain*: Determines the gain associated with a packet's transmitting antenna. This model computes gain based on the vector separating the transmitter and receiver and the pointing attributes of the transmitter's associated antenna.

- *propagation delay*: Computes the propagation delay of the received packet from the transmitter to the receiver. This model computes the delay based on the distance separating the transmitter and the receiver, and the propagation velocity of radio waves.

- *rx antenna gain*: Determines the gain associated with a packet's receiving antenna. The working of this particular pipeline stage is similar to tx antenna gain.

- *received power*: Computes the average power of the received signal. This provides a fundamental performance measure of the quality of the received signal.

- *background noise*: Used to determine noise sources other than explicitly modeled interferers. This model accounts for a constant ambient noise level, a constant source of background noise and a constant source of thermal noise at the receiver.

- *Interference noise*: Computes in-band interference due to other transmitters. This stage quantifies the mutual interference of concurrent transmissions.

- *signal to noise ratio*: Computes the signal-to-noise (interference included) ratio for the received packet. This stage depends on the computation of background noise and interference noise carried out during the previous stages.

- *bit error rate*: Computes the BER associated with a received packet based on previous pipeline stage inputs (SINR and modulation scheme).

- *error allocation*: Translates the given bit error rate into an actual set of bit errors for each valid packet which is received.
- *error correction*: Determines whether a packet can be accepted by the receiver based on the threshold (in terms of errors) of the receiver.

By using this pipeline, a link budget is formed for each packet that arrives at a node. The resulting power levels are used to calculate the SINR for each desired packet, which is then used to compute a BER based on the modulation type. This result is then used to assign bit errors to the received packet.

2.1.2 Abstracted Physical Layer

In the abstracted physical layer approach, a physical layer algorithm is implemented by one of the following methods:

- approximating physical layer behavior with the steady-state behavior of the physical layer; or
- implementing an equation that captures the desired aspect of the physical layer behavior, e.*g.*, BER over a Rayleigh channel.

To implement the abstracted behavior, the code of the related pipeline stage is modified. This is different from the baseline approach as physical layer simulations are actually run in MATLAB or C separately (not dynamically) and the steady state results inserted into OPNET.

2.1.3 Full Physical Layer Simulation

In this approach, the full physical layer is implemented by modifying the appropriate pipeline stage. This is implemented using two different approaches. In the first approach, OPNET passes the relevant node information and simulation parameters to MATLAB which processes this information to calculate the needed parameters. In the second approach these same steps are performed using C function calls from within the OPNET pipeline.

MATLAB is a commonly used development environment for physical layer simulations as it has a relatively shorter development cycle and excellent visualization tools. C is commonly used in physical layer simulations when simulation speed is critical. It has a relatively longer development cycle and no built-in visualization support, but as C simulations use compiled code whereas MATLAB simulations use interpreted code, C generally executes significantly faster than MATLAB. For this project, the MATLAB approach is expected to incur additional runtime penalties as OPNET and MATLAB maintain separate memory spaces and cycles must be used to transfer information between these memory spaces.

2.2 Adaptive Antenna Arrays

One of the main physical layer techniques examined in this thesis is adaptive antenna arrays. In this section the basics of adaptive antennas are described briefly.

2.2.1 Antenna Array Fundamentals

Depicted conceptually in Figure 3, an antenna array consists of N antenna elements. N complex baseband signals, $x_0 \ldots x_{N-1}$, are output from these elements, after down conversion and sampling, and weighted (multiplied) by a complex number $w_0 \ldots w_{N-1}$ to impart a magnitude and phase shift to each of the N signals. These weighted signals are then summed to produce the antenna array's output symbol y. The magnitude and phase of y will be dependent upon the magnitude and phases of both $\mathbf{x}$ and $\mathbf{w}$ where the bold scripting denotes a vector.

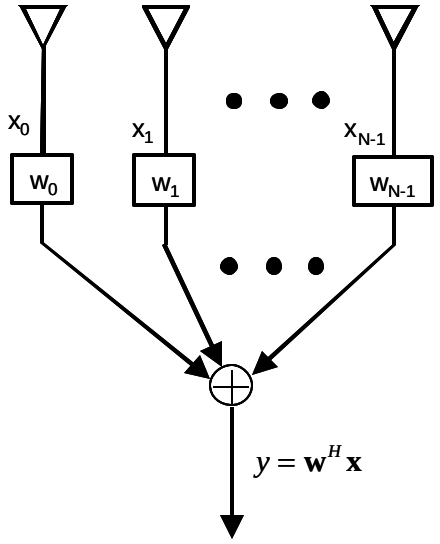

Figure 3 Conceptual Antenna Array

For closely spaced arrays, the component signals in $\mathbf{x}$ will be strongly correlated, only differing in the relative phases between each element. From an originating source, there will be some small propagation difference to each element that results in phase differences between the signals at each element. When the origination of a signal is significantly far away from the antenna array, the paths to each of the antenna elements are virtually parallel, and thus the propagation difference is only a function of the angle of arrival of the original signal to the antenna array. The exact relationship depends on the angular spread. In this work, zero angular spread is assumed.

Consider the two element array shown in Figure 4 where the wavefront of a signal is incident at some angle q and arrives at antenna element A_0 at time t_0 and at A_1 at time t_1. The relative phase difference between these two elements is then given by (1)

$$y_{1,0} = \frac{2p}{l} c\left(t_0 - t_1\right) \tag{1}$$

where c is the speed of light, l is the wavelength of the transmitted signal, and $y_{1,0}$ is the relative phase difference between element 1 and element 0. Through some simple trigonometry, it can be seen that the excess distance traveled, $c(t_0 - t_1)$, is also given by $d\sin(q)$ where d is the distance between elements 1 and 0. Thus the phase difference imparted to a signal as it arrives at each element can be found merely by knowing the distance between the elements, the signal's wavelength, and its AOA from (2).

$$y_{1,0} = \frac{2p}{l} d \sin(q) \tag{2}$$

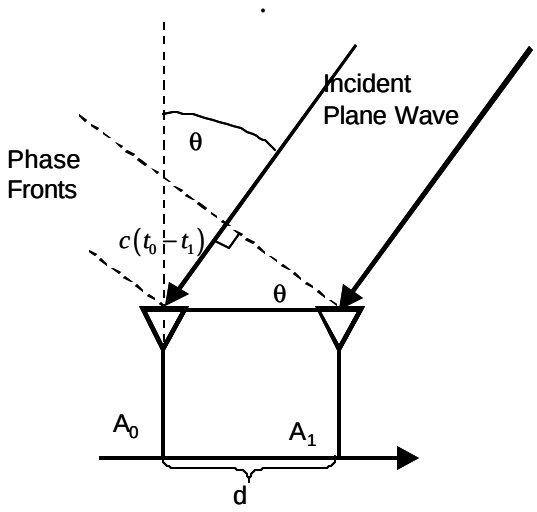

Figure 4 Illustration of Phase Difference for a Two Element Array

As the choice of where to measure the AOA is somewhat arbitrary, it is necessary to define a convention to ensure proper understanding. For all work performed in this thesis the AOA of a signal incident on the antenna array follows the convention shown in Figure 5. Note that AOA is measured as the clockwise deviation from the y-axis.

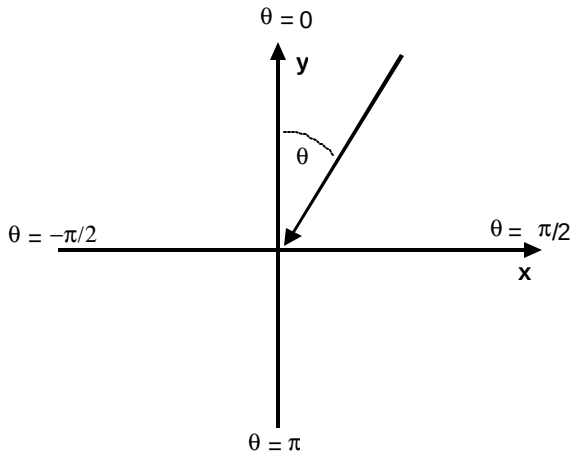

Figure 5 AOA Convention

Because of these phase differences, the choice of weights produces significantly different magnitudes dependent on the AOA. It is common to depict the magnitude of an antenna array versus the AOA in a plot known as a beam pattern. As shown in Figure 6, for the array depicted in Figure 4, two entirely different beam patterns have been created through the use of two different pairs of weights and assuming an omnidirectional antenna. In the figure on the left, the gain is two for AOAs of 0 or π and zero for AOAs of π/2 or -π/2. A more complex pattern is created in the figure on the right through the use of different weights. Notice that in both figures, there is symmetry about the horizontal axis. In fact, this symmetry will always be observed about the line of a linear array. Also, the narrowness of the beams is generally a function of the number of antenna elements, the antenna array geometry, and the spacing between elements. In general, as the number of elements is increased, the beams become narrower. Note that throughout this work, it is assumed that each element has an omni-directional pattern.

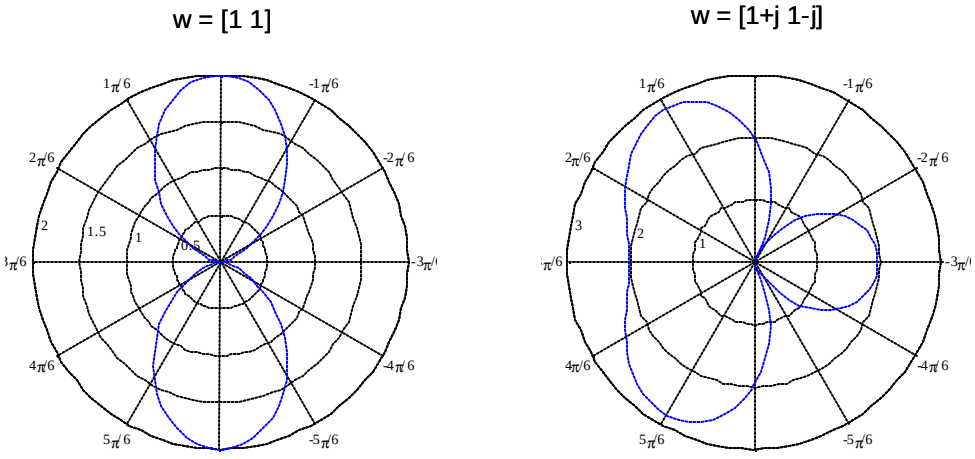

Figure 6 Example Beam Patterns for Two Element Array

The relative phase offset for a particular AOA between two arbitrarily located antenna elements can also be calculated using simple trigonometric identities. Consider the partial array shown below in Figure 7 where the only antenna elements of interest are A_k and A_{k+1} respectively located at (x_k, y_k) and (x_{k+1}, y_{k+1}) with a plane wave incident at an AOA q. Recall that the relative phase offset between the elements is given by (3).

$$y_{k,k+1} = \frac{2p}{l} c\left(t_{k+1} - t_k\right) \tag{3}$$

As can be seen in Figure 7, the relative displacement is given by (4)

$$\frac{t_{k+1} - t_k}{c} = d_{k,k+1} \sin\left(j_{k,k+1} + \frac{p}{2} - q\right) = d_{k,k+1} \cos\left(q - j_{k,k+1}\right) \tag{4}$$

where $j_{k,k+1}$ is the angle between A_k and A_{k+1} relative to the horizontal axis. Thus the phase offset between the elements is given by (5).

$$y_{k,k+1} = \frac{2p}{l} d_{k,k+1} \cos\left(q - j_{k,k+1}\right) \tag{5}$$

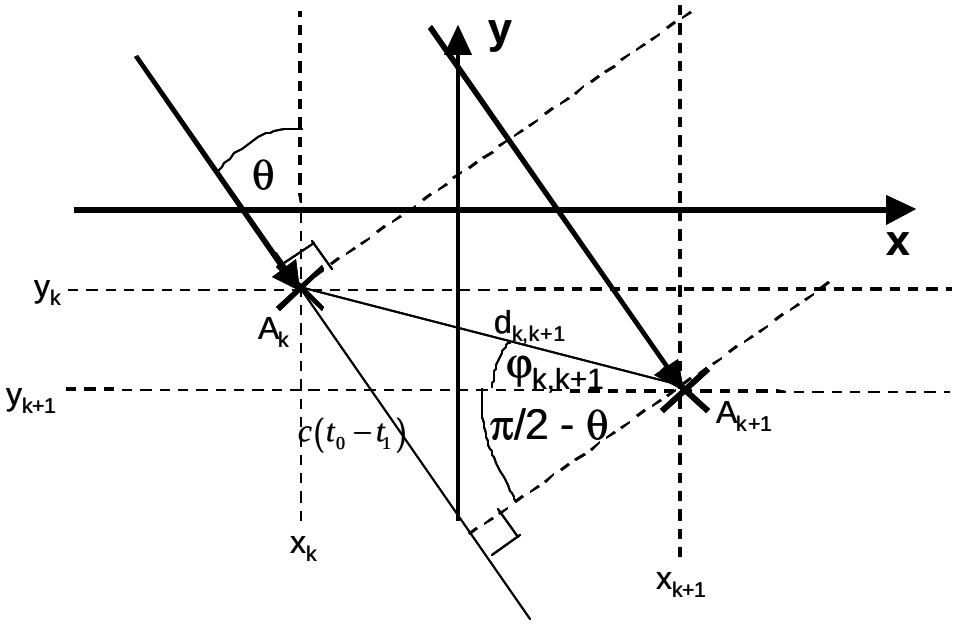

Figure 7 Illustration of Phase Delay for an Arbitrary Element Spacing

Using this method, the relative phase offsets for arbitrary planar antenna array geometries can be calculated by choosing one element to have a phase offset of 0 and then calculating the phase offset for the remaining elements with respect to this arbitrarily chosen element.

For the beam patterns shown in Figure 6, it can readily be seen that it would be desirable to have beam patterns where maxima lobes are pointed towards the desired node while nulls are pointed at undesired nodes. This results in an increased signal-to-interference-

plus-noise ratio (SINR), which, in turn, results in improved system performance. The challenge is then determining the optimal weights for maximizing SINR. Clearly not a straight forward process, this determination is complicated in a mobile environment where nodes may be moving or multipath characteristics may be changing. The typical solution to this problem is the use of adaptive antenna array algorithms that seek the optimal weights.

There are a wide variety of adaptive antenna array algorithms in the literature that operate in different manners. Some attempt to maximize SINR, others attempt to minimize mean squared error (MSE) of some desired sequence, some require additional information, others require no external information. Some of the considerations in the choice of an adaptive antenna array algorithm include:

- Available processing power
- Data rate that can be supported
- Convergence rate
- Number of antenna elements
- Available information

The basic structure for implementing an adaptive antenna array is shown in Figure 8. Notice that there are three basic steps in the adaptive antenna array operation:

- calculating the array output
- calculating an error vector
- updating the array weights.

The array output, *y*, is the complex dot product of the weight and receive vectors. Based on *y* and some side information, an error, *e*, is calculated. Numerous algorithms exist for calculating *e* including CMA (Constant Modulus Algorithm) which restores a constant envelope to the input signal and MMSE which exploits knowledge of a specific sequence of data to calculate *e*. Based on the previous weights, **x**, *y*, and *e* new values for **w** are calculated. Like the error calculation, several different algorithms exist for updating the values of the weights. Two of the most common include the Least Mean Squares (LMS) algorithm and the Recursive Least Squares (RLS) algorithm. In this thesis, only the LMS algorithm is implemented in the weight update algorithm.

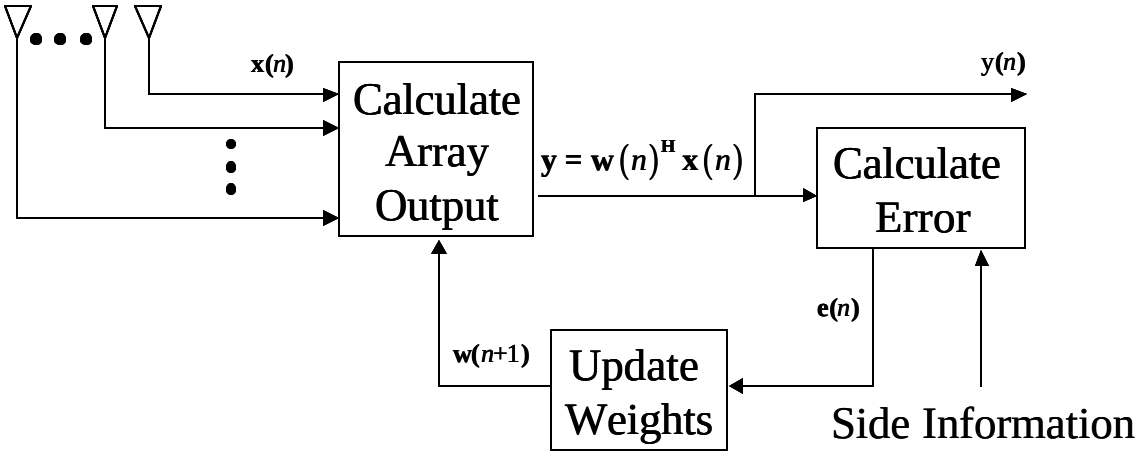

Figure 8 Basic Adaptive Antenna Array Operations

2.2.2 Adaptive Antenna Algorithm: CMA

In this system, the Constant Modulus Algorithm is one of the algorithms implemented for the error calculation. In CMA, the algorithm attempts to restore the envelope of the output signal *y*. In general, this can be expressed as solving for the weights that minimize the function given by (6)

$$J(\mathbf{w}_k) = E\left[\left(|y_k|^p - d^q\right)^q\right] \tag{6}$$

where d is the desired envelope level and p and q are integers. The most commonly implemented version of CMA uses $p = 2$ and $q = 1$ and is commonly written as CMA (2, 1).

The exact solution of the minimization of this function requires a matrix inversion which is generally extremely cycle intensive. Because of this fact, iterative approximations that converge to the minimum have been developed. The most common of these is the least mean squares (LMS) iterative solution which uses the gradient of steepest descent for its adaptation. For CMA (2, 1), it can be shown that this gradient is equal to (7)

$$\nabla J(\mathbf{w}_k) = \left(|y_k|^2 - 1\right) y_k^* \mathbf{x}_k \tag{7}$$

where * denotes the conjugate operation. Using the LMS algorithm for weight updating, the iterative solution for the weights takes on the form in (8)

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mu\left(|y_k|^2 - 1\right) y_k^* \mathbf{x}_k \tag{8}$$

where μ is the step size. Note that as per typical LMS implementations, larger values for μ result in faster convergence, but larger final error.

2.2.2.1 Comments on CMA

CMA is a particularly attractive algorithm for use in this simulation as it is a blind algorithm and thus obviates the need to create training sequences and perform timing recovery. However, it should be noted that CMA converges to the strongest constant modulus signal at the output of the array when the algorithm begins. Thus the algorithm may converge to an undesired signal. For instance, consider the two beam patterns illustrated in Figure 9. Both patterns began with the same initial weights, and use the same step size. However, in the figure on the left the node designated by the circle has the strongest transmit power, whereas on the right the node designated by the x has the strongest transmit power. It is seen that the 2 cases result in very different solutions.

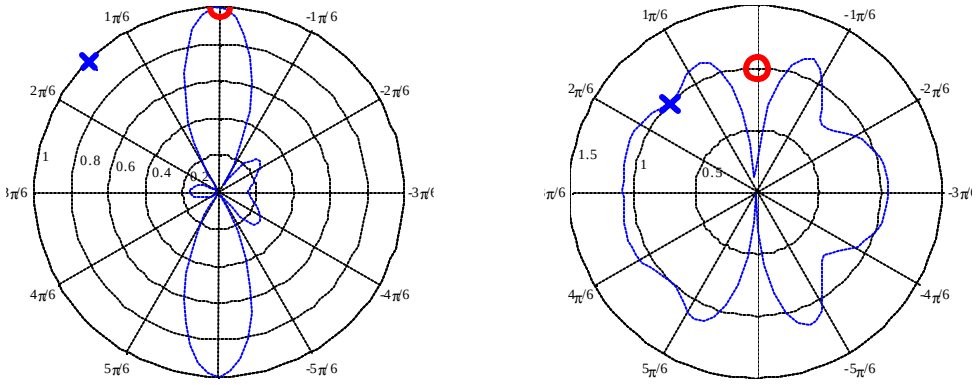

Figure 9 Effect of Received Power on Effect of CMA

For mobile environments, the rate of convergence compounds this problem. When a mobile node moves, the adaptation algorithm may not have sufficient time to converge to the steady state beam pattern. Thus the relative gain given to the desired signal and undesired signals may be less than optimal. This problem is exacerbated when an undesired node and a desired node are close to one another with respect to the difference in AOA that the antenna array can resolve. Because of the resulting fluctuations in gains, the algorithm can become unstable or lock onto a different node, radically altering SINR. These effects are made more pronounced depending upon the channel loss characteristics, the choice of step size, the relative power levels, antenna spacing and even the exact traffic patterns. Due to all of these effects, the performance of a network using an LMS implementation of a CMA algorithm is highly sensitive to changes in environment and initial conditions.

2.3 Modeled System

This section first provides an overview of the modeled system. This is followed by a sequence of sections detailing how each of the different cross-layer simulation methodologies is implemented, with particular attention given to the description of how interfaces between layers (co-simulations) are established and the procedural steps used to implement the physical layer within the co-simulation.

The system consists of three types of nodes: transmitters, receivers, and jammers. Each of these nodes may be fixed or mobile. The following gives a description of each of these nodes.

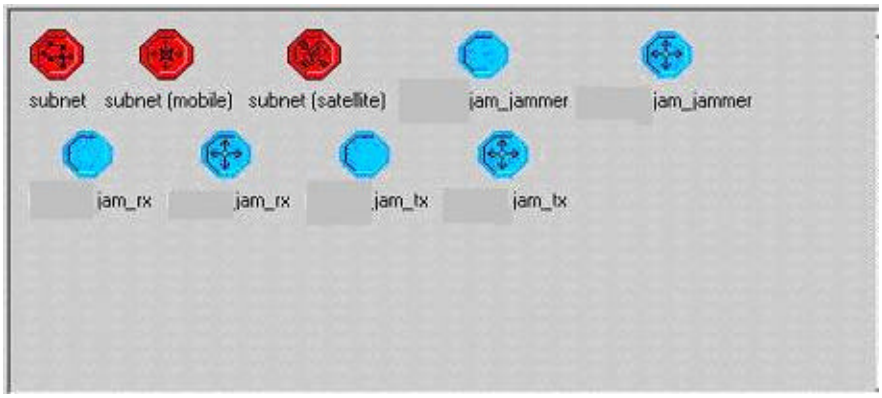

Figure 10 System Palette

2.3.1 Transmitter

The transmitter node model, shown in Figure 11, consists of a *simple source* module that generates packets and is generally treated as a desired node. The probability distribution functions of the packet size and packet inter-arrival time used within the simple source may be set as desired from those available within OPNET, and are set as constant in this case. After generation, packets move through a packet stream to the radio transmitter module that transmits the packets on a radio channel. To enhance simulation flexibility, the transmitter node's power attribute is promoted so it can be set at run-time by the user. Some of the specific parameters of the transmitter node are listed in Table 1.

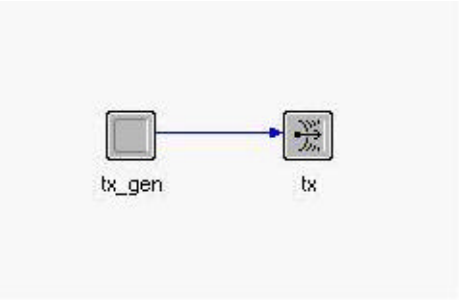

Figure 11 Transmitter Node Model

Table 1 Transmitter Attributes

Attribute	Type/ Value
Modulation	BPSK
Power	Promoted to be set by user at run time
Data Rate	100 Kbps
Bandwidth	100 KHz
Min. Frequency	1.8 GHz

2.3.2 Jammer

The network jammer node, whose node model is shown in Figure 12, introduces interference into the network. Like the transmitter node, it consists of a *simple source* packet generator module and a radio transmitter module. Its behavior is similar to the transmitter node, but its signal modulation is set to *jammod*. The jammer node's power attribute is also promoted so it can be set at run time by the user. Some of the specific parameters of the jammer node are listed in Table 2.

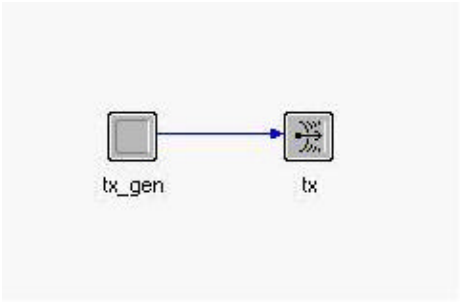

Figure 12 Jammer Node Model

Table 2 Jammer Attributes

Attribute	Type/ Value
Modulation	*jammod*
Power	Promoted to be set by user at run time
Data Rate	100 Kbps
Bandwidth	100 KHz
Min. Frequency	1.8 GHz

2.3.3 Receiver

The receiver node, whose node model is shown in Figure 13, consists of an antenna module, radio receiver module, and a sink processor module. The rx radio receiver contains the *ragain* pipeline which is changed in order to run each of the different scenarios as specified in Table 3. In order to run the abstracted physical layer, the *dra_ragain* pipeline stage must be selected and the *pattern* parameter in the rx_ant antenna module must be set to *beam_pattern_abstraction*. Table 4 lists additional node level attributes of the receiver node.

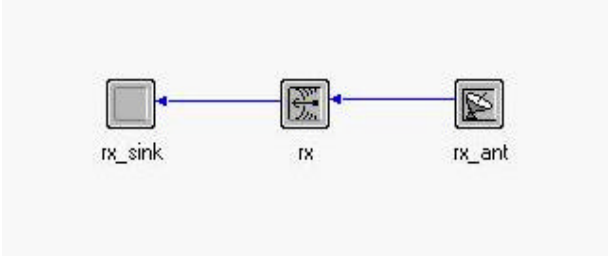

Figure 13 Receiver Node Model

Table 3 *Ragain* Pipeline Stages

Method	Pipeline Stage Name
Baseline	*dra_ragain*
Abstracted physical Layer	*dra_ragain*
Full physical Layer Simulation (MATLAB)	*adaptive_ antenna_ MATLAB*
Full physical Layer Simulation (C)	*adaptive_ antenna_ C*

Table 4 Receiver Attributes

Attribute	Type/Value
Modulation	BPSK
Noise Figure	1
Data Rate	100 Kbps
Bandwidth	100 KHz

2.4 Baseline Simulation

For the baseline simulation, no modifications were made on the standard OPNET pipeline

2.4.1 Interface

There is no interface as all operations are performed using OPNET KPs (Kernel Procedures).

2.4.2 Procedural Description

No procedural description is included for the baseline interface as no modifications are made to any pipeline stage.

2.5 Abstracted Physical Layer

In the abstracted physical layer approach, the physical layer is fully simulated for a particular transmitter/receiver/jammer scenario prior to running the full simulation. The steady-state weights from this simulation are then used to create an antenna pattern identical to the beam pattern produced from the physical layer simulation. Then for each packet input to the *ragain* pipeline, the gain indicated by the modified antenna pattern is used to set the OPC_TD_RA_RX_GAIN parameter in the packet. An example of this process is illustrated in Figure 14. By using the previously generated beam pattern, the *ragain* pipeline stage approximates the effects of the adaptive antenna array without incurring the additional cycles required to perform the full physical layer simulation.

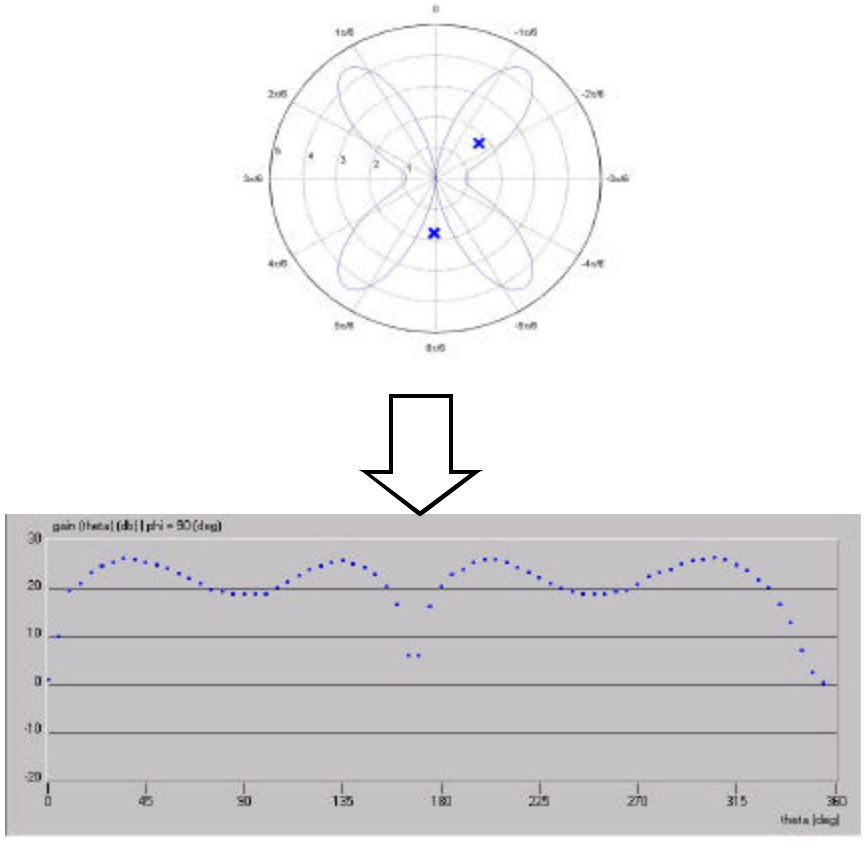

Figure 14 Translation of Array Beam Pattern to OPNET Antenna Pattern

2.5.1 Interface

There is no interface as all operations are performed using OPNET KPs.

2.5.2 Procedural Description

No procedural description is included for the abstracted interface as no modifications are made to any pipeline stage. However, alterations are made to the antenna beam pattern table used in the *ragain* stage to mimic the steady state beam pattern of the adaptive CMA algorithm.

2.6 Full Physical Layer Implementation (MATLAB)

2.6.1 Interface

The interface between OPNET and MATLAB is implemented using MATLAB engine routines. As part of the initialization routine of the adaptive antenna array pipeline, the MATLAB engine is initialized and stored in an Engine variable. This variable is then passed to the MATLAB antenna array pipeline function. This allows OPNET to maintain multiple simultaneous MATLAB engines which may be useful when distinct variable namespaces are required.

2.6.1.1 Operations Performed in OPNET

The following are the tasks performed in OPNET:

- Updating the positions of all relevant nodes
- Identifying the index of the receiving node (MATLAB is responsible for array index translation)
- Identifying the index of the transmitting node (MATLAB is responsible for array index translation)
- Determining the Array Geometry
- Setting λ
- Setting the number of bits, N_b, for use in the simulated packets
- Initializing MATLAB engine
- Storing antenna weights between calls to MATLAB
- Assigning the gain to the packet

2.6.1.2 Operations Performed in MATLAB

The following are the tasks performed in MATLAB:

- Calculating AOA for each transmitting node
- Calculating phase difference between each antenna element in the receiving node
- Generating transmitted signal at each antenna for each transmitted signal
- Performing channel operations
- Producing symbol estimates based on antenna weights
- Updating antenna weights according to adaptive antenna array algorithm (CMA)
- Determining the gain in the direction of the transmitting node

2.6.2 Procedural Description

Figure 15 shows the modified process flow for the *ragain* pipeline stage for the MATLAB implementation. Note that the purple boxes correspond to operations performed using OPNET KPs, black boxes correspond to operations performed using C routines, and blue boxes correspond to operations performed in MATLAB. Notice that C is only used to initialize parameters, to start the MATLAB engine, and to serve as an interface between OPNET and MATLAB. The use of the C routine to interface with MATLAB encapsulates the details of calling and using the MATLAB routines. To limit the possibility of memory leaks, all mxArrays created as part of the MATLAB interface are destroyed upon completion of that routine. Where possible, the vector processing

capabilities of MATLAB are exploited within these functions in an effort to improve the execution speed of this code. Further details of the signal generation process in MATLAB are shown in Figure 16.

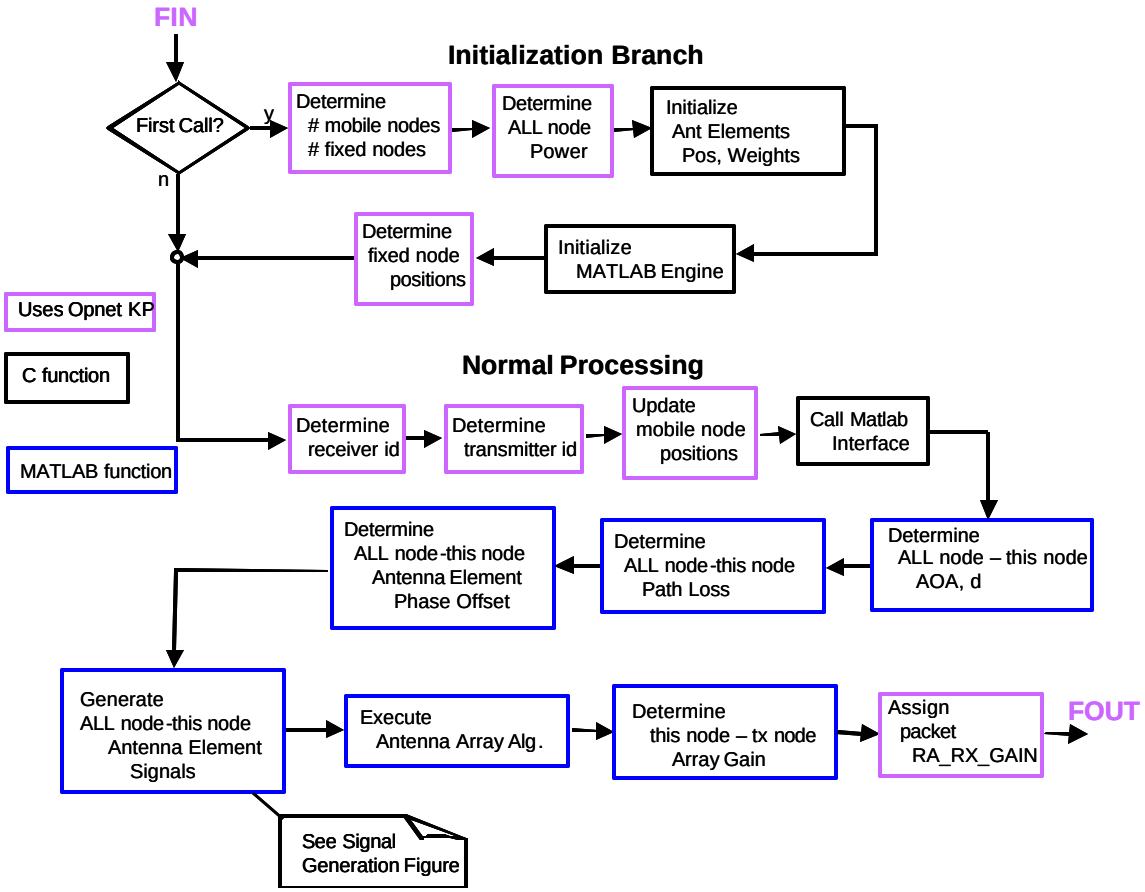

Figure 15 Modified Process Flow for *Ragain* for Physical Layer Simulation Using MATLAB

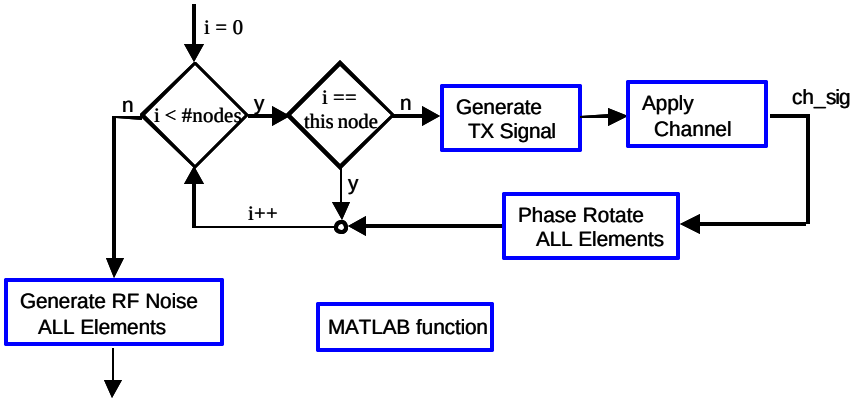

Figure 16 Signal Generation Algorithm for MATLAB

2.7 Full Physical Layer Implementation (C)

2.7.1 Interface

The interface between OPNET and C is implemented through a series of C function calls which are stored in header files and called sequentially within the pipeline.

2.7.1.1 Operations Performed in OPNET

The following are the tasks performed in OPNET:

- Updating the positions of all relevant nodes
- Identifying the index of the receiving node
- Identifying the index of the transmitting node
- Determining the Array Geometry
- Setting λ
- Setting the number of bits, N_b, for use in the simulated packets
- Storing antenna weights between calls to C
- Assigning the gain to the packet

2.7.1.2 Operations Performed in C

The following are the tasks performed in C:

- Calculating AOA for each transmitting node
- Calculating phase difference between each antenna element in the receiving node
- Generating transmitted signal at each antenna for each transmitted signal
- Performing channel operations
- Producing symbol estimates based on antenna weights
- Updating antenna weights according to adaptive antenna array algorithm (CMA)
- Determining the gain in the direction of the transmitting node

2.7.2 Procedural Description

Figure 17 shows the modified process flow for the *ragain* pipeline stage for the C implementation. Notice that during the initialization branch, all of the information that needs to be computed only once (such as the AOA and path loss between fixed nodes) is computed. In the normal operation branch, the AOA and path loss information for the mobile nodes are only calculated with respect to the currently receiving node. These simple optimizations help to reduce the number of cycles required to support the C implementation of the physical layer. Also notice that OPNET is used solely for retrieving and setting OPNET specific variables; all other processing is performed in C function calls. Further details of the signal generation process in C are shown in Figure 18.

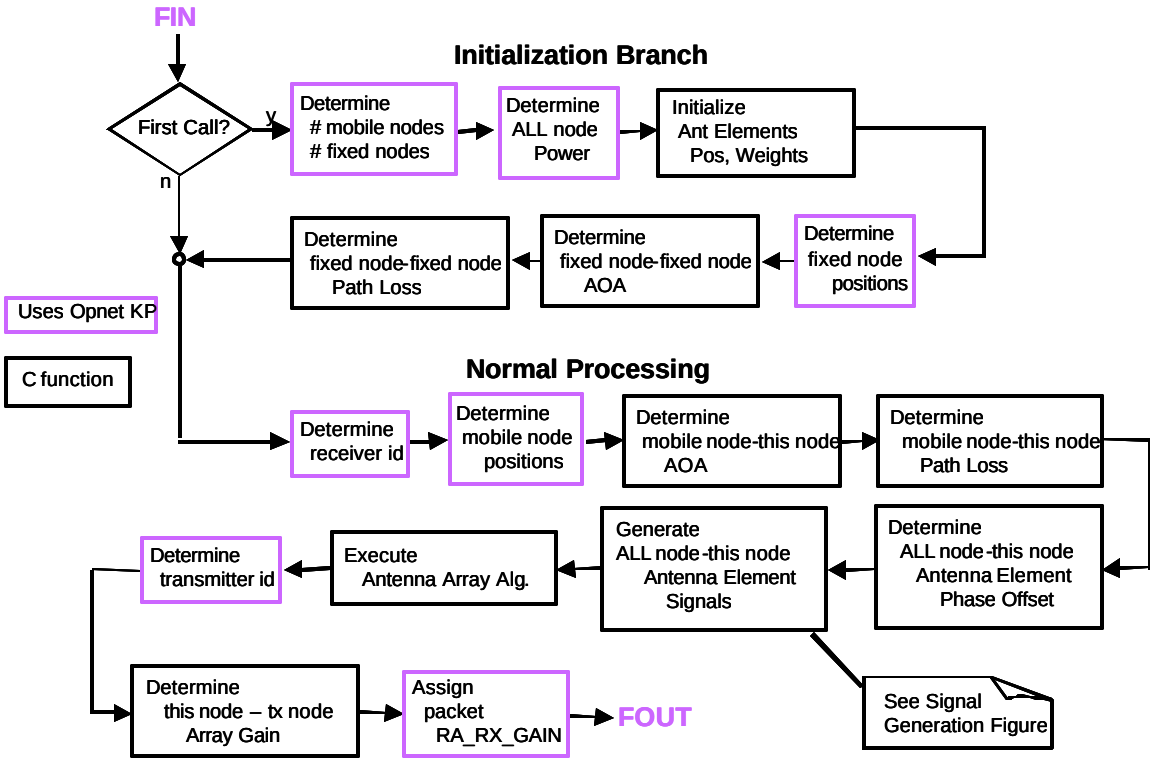

Figure 17 Modified Process Flow for *Ragain* for Physical Layer Simulation Using C

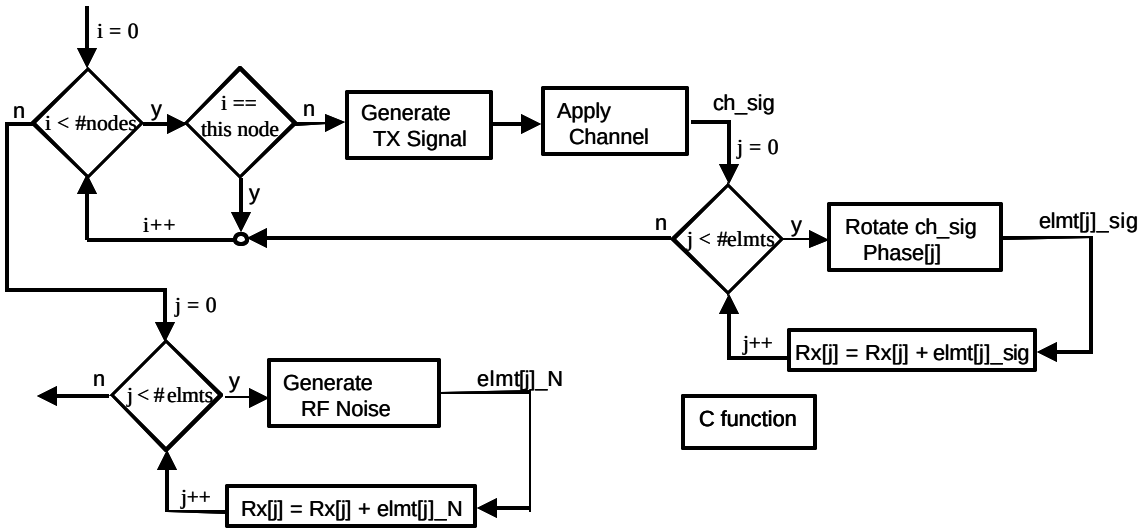

Figure 18 C Signal Generation Process Flow

2.8 Results

This section describes the specific parameters used in the simulation and presents the results of the simulations performed using the each of the different methodologies.

2.8.1 Scenario Description

The system consists of a fixed transmitter - receiver pair, with a mobile jammer moving in a trajectory around the receiver, as shown in Figure 19. The transmitter and jammer promoted power levels have been set to be 1 W and 0.25 W, respectively. The jammer follows changes to a new position every minute completing its path around the receiver in 10 minutes. Simulation time is 30 minutes during which the jammer completes three circles around the receiver moving at a rate of approximately 0.21 m/s. The rx node implements a 4 element linear array with a spacing of 0.25 m between elements and uses a step size of 0.05 in its CMA algorithm.

Figure 19 Network Scenario

2.8.2 Captured Results

This section lists the results captured for each of the simulation methods explored as part of this simulation. Specifically, bit error rate (BER), throughput in packets/sec presuming packet rejection when any errors occur in the packet, and SINR were measured for each method considered in this scenario. Note as the jammer and the transmitter actually generate packets asynchronously and independently, the BER and throughput are artificially improved due to transmit packets arriving when no jammer packet is present. This yields packets with no interference and thus exceedingly high

SINR. To provide a more realistic interpretation of the operation of the adaptive algorithms, SINR is plotted on a packet-by-packet basis instead of a time-average basis as shown in Figure 20. This yields 2 curves as seen which appear continuous but are actually discrete. The SINR plot is then zoomed in on to yield the SINR in the presence of the jammer as shown in the subsequently presented results. All X axes represent time with the Y axis representing the title of the plot.

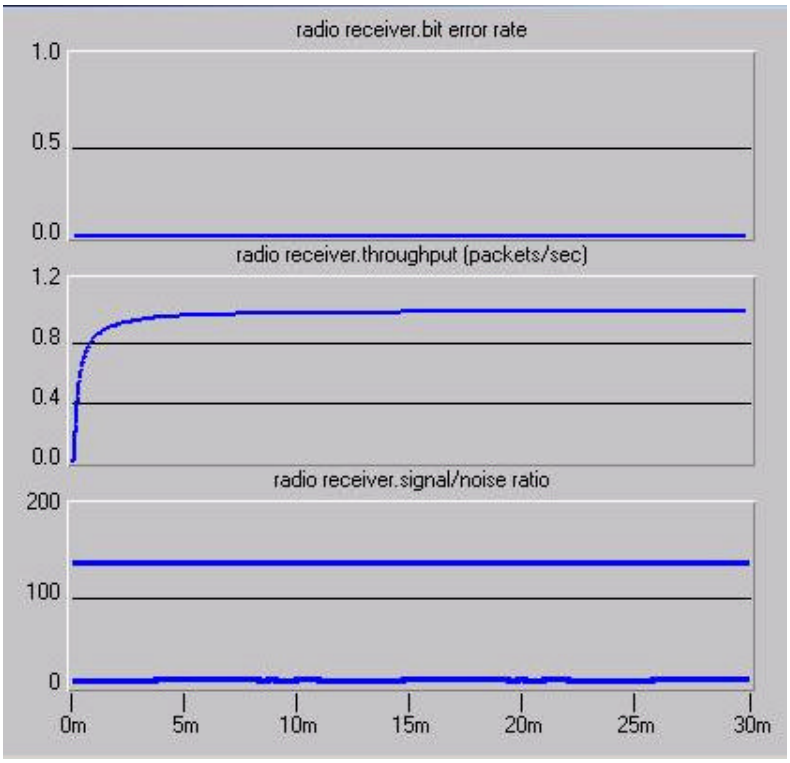

Figure 20 Example Initial Result

Figure 21 shows the results from the baseline method. The small variation in SINR is due to the jammer path not following a perfect circle around the receiver.

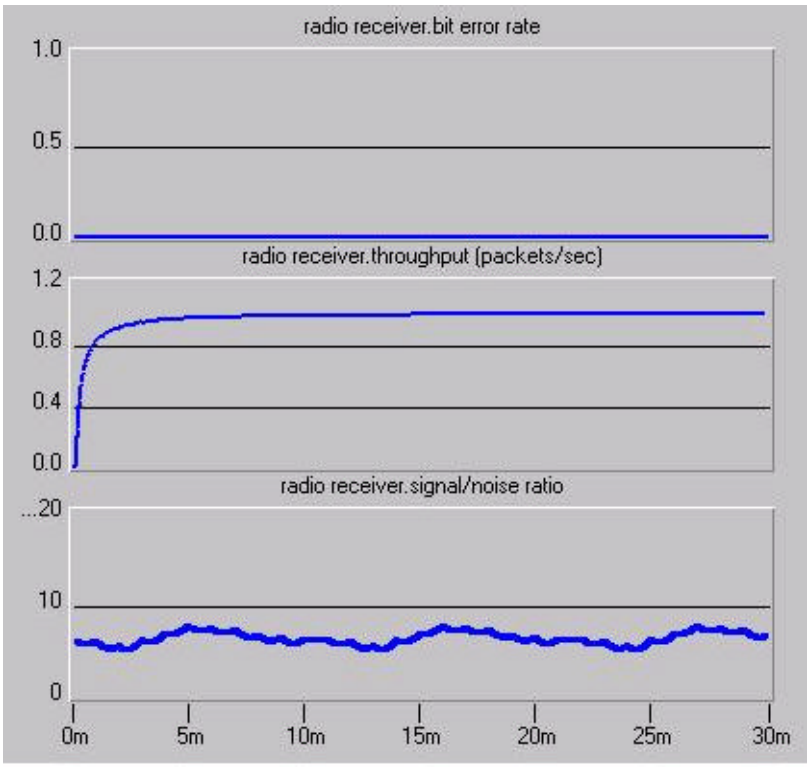

Figure 21 System Statistics for Baseline Simulation

To perform the abstracted physical layer, the physical layer was simulated separately in MATLAB assuming fixed nodes. These results were captured and used to approximate the effects of the CMA algorithm by modifying the antenna gain pattern in OPNET as shown in Figure 22.

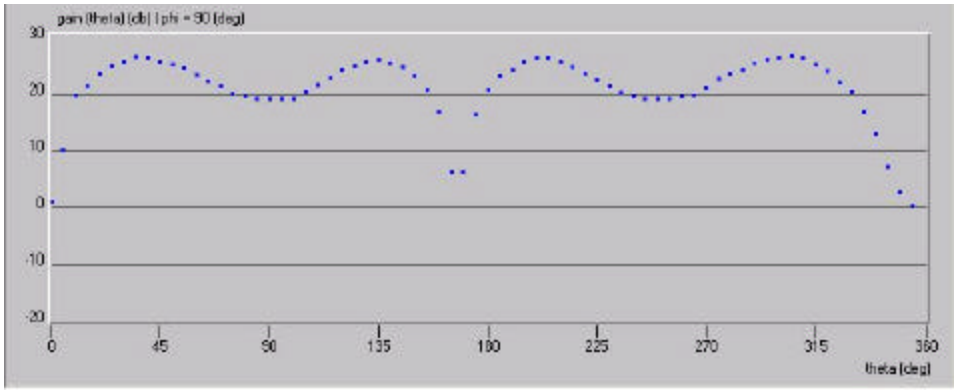

Figure 22 Fixed Antenna Pattern Defined in OPNET

This was then simulated presuming both a stationary jammer as well as for the mobile jammer. For the stationary scenario the system achieved zero BER with an SINR of 20 dB, shown in Figure 23, in accordance to the MATLAB simulation.

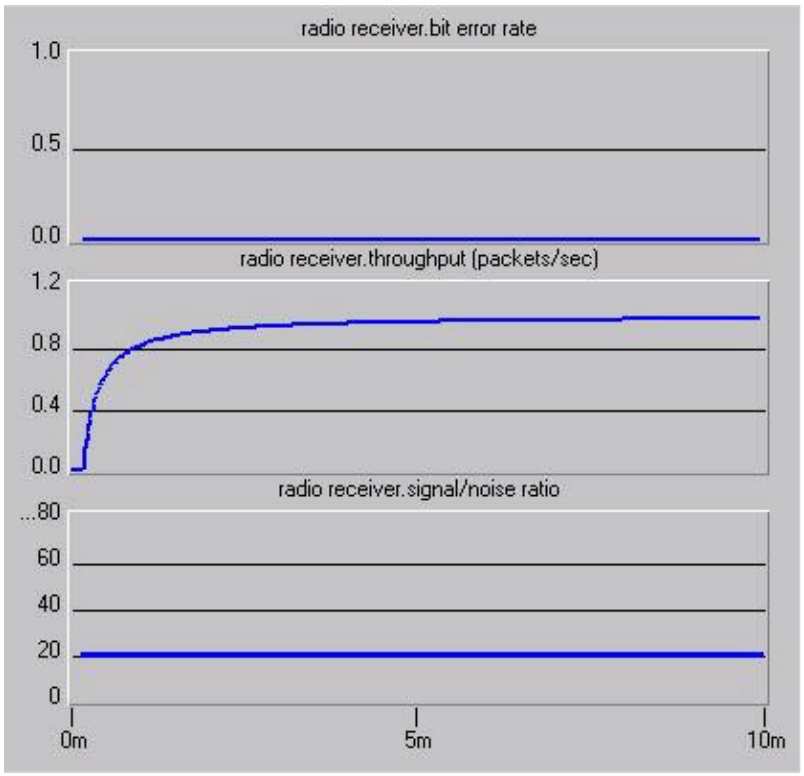

Figure 23 System Statistics with Abstracted Physical Layer, Stationary Jammer

However as shown in Figure 24, with a mobile jammer and the abstracted physical layer, the SINR remains nearly constant at 20 dB, except for small variations due to variation in the distance of the jammer to the receiver, the lone exceptions being when the jammer exactly lined up with the transmitter.

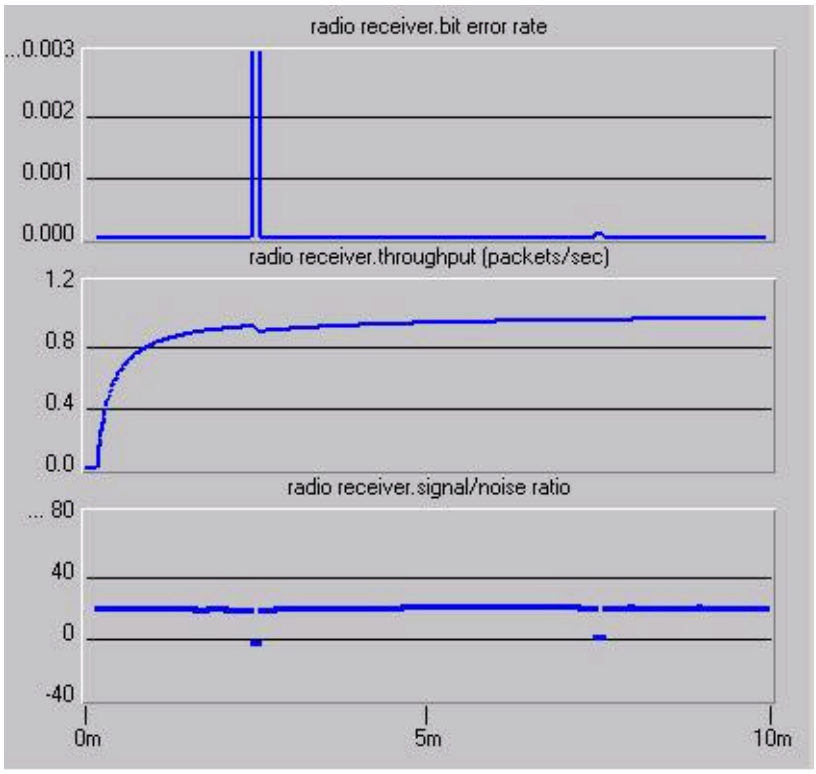

Figure 24 System Statistics with Abstracted Physical Layer, Mobile Jammer

Figure 25 illustrates the results captured from the full physical layer simulation using MATLAB. As expected the average SINR (plotted in dB) is better than for the omni-directional case modeled under the baseline method. However, the SINR full implementation is not nearly as smooth as seen in the abstracted physical layer simulation as there are numerous peaks when near perfect nulling is achieved or relatively low SINR when the algorithm either has insufficient time to adapt or antenna geometries prevent optimal adaptation.

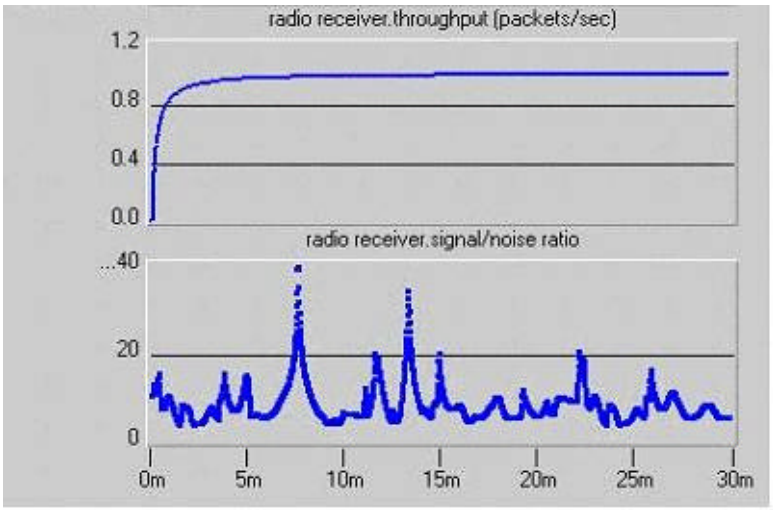

Figure 25 System Statistics with Physical Layer Co-Simulated in MATLAB

Figure 26 summarizes the results seen from a particular simulation in C. Note again that the SINR (plotted in dB) is relatively choppy as seen in MATLAB. Due to different traffic generation, in this result, there are actually short periods where the algorithm appears to track the jammer rather than the transmitter.

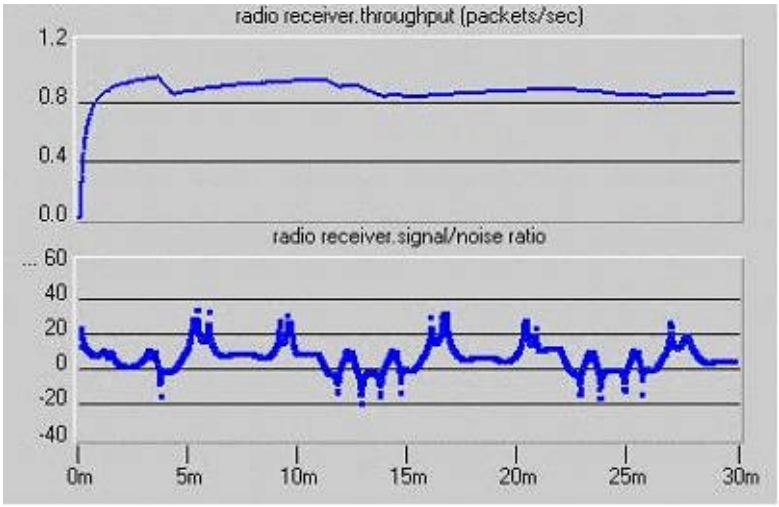

Figure 26 System Statistics for Physical Layer Co-Simulation in C

2.8.3 Simulation Time Benchmark

This section compares each simulation against the simulation time benchmark. Note that the rate at which events are processed is dependent upon the platform on which the simulation is run. For the results presented in this section, Table 5 lists the specifications of the machine used to perform the simulations. The relative performance between methods is expected to hold regardless of platform.

Table 5 Simulation Platform Specifications

Parameter	Value
Processor	Intel Pentium II 400 MHz
RAM	256 MB
Operating System	Windows 2000 Professional
Virtual Memory	384 MB

Table 6 lists the results for the simulation speed benchmark. Treating the baseline simulation as the benchmark, it is seen that both the abstracted physical layer and the full physical layer implemented in C both achieve a significant fraction of the speed of the baseline method. However, the MATLAB simulation performs poorly in terms of simulation speed when compared to the other methods.

Table 6 Simulation Speed Benchmark

Method	Simulation Speed (Events/Sec)	Benchmark	Performance
Baseline	8471	100% (Benchmark)	Perfect
Abstracted physical Layer	6943	81.96%	Excellent
Full physical Layer Simulation (MATLAB)	83	0.98%	Poor
Full physical Layer Simulation (C)	4424	55.23%	Good

2.8.4 Fidelity Benchmark

Treating the full physical layer simulation methods as the fidelity benchmark, it can be seen from the results that the baseline method fails to capture in any way the operation of the CMA algorithm. The abstracted physical layer successfully reflects the average SINR of 20dB, but it does not do an adequate job approximating the positional sensitivity of the CMA algorithm.

Table 7 Simulation Fidelity Benchmark

Method	Fidelity
Baseline	Unacceptable
Abstracted physical Layer	Poor
Full physical Layer Simulation (MATLAB)	Perfect
Full physical Layer Simulation (C)	Perfect

2.9 Summary and Conclusions

This phase of the research has demonstrated the feasibility of incorporating a realistic physical layer into OPNET network simulations, specifically a physical layer that includes the CMA(2,1) adaptive algorithm. The physical layer is included with a minimum amount of intrusion into normal OPNET processing through the modification of a single pipeline stage, *ragain*. It has been demonstrated that this can be performed with a high degree of fidelity through the use of MATLAB or C with a relatively small penalty in run time for implementations using C. Neither the baseline approach nor the abstracted physical layer adequately modeled the sensitive nature of the CMA algorithm in a mobile environment. However, it is expected that the abstracted physical layer would yield sufficient fidelity for a purely stationary wireless environment. The benchmark results for this study are summarized in Table 8.

Table 8 Qualitative Benchmarks Results Matrix

Simulation Approach	Simulation Time	Fidelity
Baseline (Isotropic)	Perfect	Unacceptable
Abstracted physical Layer	Excellent	Poor
Full physical Layer (MATLAB)	Poor	Perfect
Full physical Layer (C)	Good	Perfect

2.10 Recommendations for Cross-layer Simulations Using OPNET

Based on this work, the following recommendations are made for cross layer simulations:

- When modeling a system with stationary operating parameters, an abstraction of the physical layer is suggested as it will provide reasonable fidelity and an excellent simulation time.
- When modeling a system with non-stationary operating parameters (such as seen when adaptive algorithms are modeled), it is necessary to implement a full physical layer in order to achieve an acceptable level of fidelity. While MATLAB may be more attractive for development and small simulations, implementation in C results in significantly reduced simulation run-times.
- When implementing adaptive antenna algorithms, the modified algorithms should be called from within the *ragain* or *tagain* to produce a minimal impact on the operation of the remaining stages.

2.11 Future Directions

It is anticipated that further reductions in simulation run-time can be made through the use of hardware-in-the-loop simulations to implement computationally intense portions of the physical layer. This will necessitate the development of middleware to ensure proper message passing between software and hardware and to encapsulate the hardware devices.

3 Case Study: Cross-Layer Simulation

This section uses the techniques developed in the preceding chapter and describes a *Stop and Wait* acknowledgement (ACK) scheme for a MANET with convolutionally encoded and decoded data, as a case study of cross-layer simulation. Techniques to incorporate the convolutional encoding and decoding in the network simulator (OPNET) are presented and compared.

3.1 System Description

The system consists of a transmitter and a receiver communicating over a radio link established in OPNET with a jammer providing interference to the system. The system being modeled is an instance of an ad-hoc network with a transmitter communicating to a receiver over the interference caused by jammers as described in Section 2.3.

3.2 The ACK scheme

A stop and wait ACK scheme is incorporated in the transmitter and receiver. In the system under consideration, the transmitter does not transmit data at a rate faster than what the receiver can accept. Hence, flow control is not required and the purpose of the ACK scheme is reliable transmission as packets may get corrupted by the interference caused by the jammers.

The transmitter on sending a packet waits for an *acknowledgement packet* (ACK) to be received from the receiver before sending the next data packet. If an ACK is pending for the last packet sent, the current packet is queued. On receiving an ACK, the transmitter first transmits the queued packets before those arriving from the network layer on a first-in first-out basis.

3.3 Physical Layer Description

At the physical layer, an exponential path loss, with path loss coefficient n = 2, is assumed, and AWGN is generated to simulate the noise introduced in each receiver chain. Additionally convolutional decoding is incorporated at the receiver. Simulation results for static and fast fading are compared. The following gives a detailed description of the underlying operations being modeled in the physical layer simulation.

3.3.1 Modulation

Each node in the modeled system uses BPSK modulation generated on a symbol by symbol basis implemented at baseband using a complex envelope representation where a symbol is given by

$$b(t) = \sqrt{P_i}\, b_k\, e^{j j_k}\ \ p(t) \tag{9}$$

where b_k is the k^{th} baseband symbol, j_k is the phase of symbol k, and P_i is the transmit power of node i, $p(t)$ is the square pulse of T second duration. For BPSK, the desired

envelope level is generally chosen to be identically one which dictates that d = 1 for the CMA algorithm.

3.3.2 Channel

Additive white Gaussian noise introduced at the receiver. The path loss between two points is given by (10)

$$PL = 10n\log_{10}\left(\frac{d}{d_0}\right) + \left(P_0\right) \tag{10}$$

where n = 2 is the path loss exponent, d_0 =10 m is the reference distance, and $\left(P_0\right)$ = 1 dB is the path loss at the reference distance.

The noise introduced due to the random motion of each by each receiver chain is an AWGN source and independent samples are generated for each receiver chain.

3.3.2.1 Generation of AWGN and Rayleigh coefficients

Additive White Gaussian Noise (AWGN) samples and the Rayleigh coefficients to model fading are generated using the built-in OPNET kernel function *op_dist_uniform*() and the Box-Mueller algorithm. The Box-Mueller algorithm is described as follows.

It is known that given two independent zero-mean Gaussian random variables, x and y, with identical standard deviations, $\sigma_x = \sigma_y = 2^{-1/2}$, a Rayleigh distributed random variable, r, with standard deviation $\sigma_r = 1$ is given by (11).

$$r = \sqrt{x^2 + y^2} \tag{11}$$

Additionally, it is known that a uniform random variable,q, distributed over the range [0, 2π] can be generated by (12).

$$q = \tan^{-1}\left(\frac{y}{x}\right) \tag{12}$$

Thus with a simple polar transformation, two zero-mean independent Gaussian random variables can be created by evaluating (13) and (14)

$$x = r\cos q \tag{13}$$

$$y = r\sin q \tag{14}$$

where r is a Rayleigh distributed random variable and q is a random variable uniformly distributed over the range [0, 2π].

To implement (13) and (14), both r and q must be generated. Fortunately, this can be accomplished with simple transformations of two independent uniform random variables, u_1 and u_2 distributed over the range [0, 1]. These transformations are given by (15) and (16).

$$r = \sqrt{-2\ln(1-u_1)} \tag{15}$$

$$q = 2p\,u_2 \tag{16}$$

Then two Gaussian random variables, x and y, are created.

It is straightforward to modify the variables x or y to create Gaussian random variables with arbitrary means and standard deviations by defining the Gaussian random variable x_1 through the transformation in (17).

$$x_1 = s\,x + m \tag{17}$$

Since x is zero mean and has $\sigma_x = 1$, x_1 will have standard deviation σ and mean μ.
By using this function with a mean of 0 and the appropriately chosen standard deviation, AWGN samples can be generated.

3.3.3 Convolutional Coding

The implemented physical layer also supports convolutional encoding and decoding. Convolutional encoding is a special kind of forward error correction (FEC) that injects redundancy into the data stream by introducing sequential dependencies. This is most commonly implemented using a convolutional encoder. Figure 27 shows a systematic rate 1/3 encoder. For every input bit three output bits are produced, one of which is an information bit while the others are parity bits. Note that although there is an explicit information bit and parity bits, this is not necessary for proper operation of a convolutional encoder. In many encoders, information and parity are effectively shared over the output bits. Because there are two memory elements in this encoder, the length of sequential dependencies of this code is two previous symbols and is said to have a constraint length of 2.

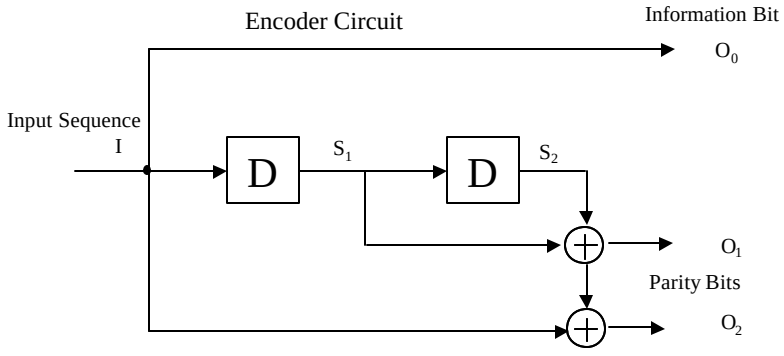

Figure 27 Rate 1/3 Convolutional Encoder

The generation of the output bits for an encoder is characterized by a set of encoder polynomials. For this encoder, the set of polynomials is given by (18).

$$g_0(D) = 1$$
$$g_1(D) = D + D^2 \qquad (18)$$
$$g_2(D) = 1 + D^2$$

The behavior of the encoder can also be characterized by a state diagram. In this scheme, the state is determined by the values currently held in memory, state transitions are determined by the input bit, and the output is a function of the state and the inputs. The state diagram associated with the encoder shown in Figure 27 is shown in Figure 28. Note that since there are two memory elements in the encoder and a binary alphabet is assumed, there are $2^2 = 4$ states in the state diagram.

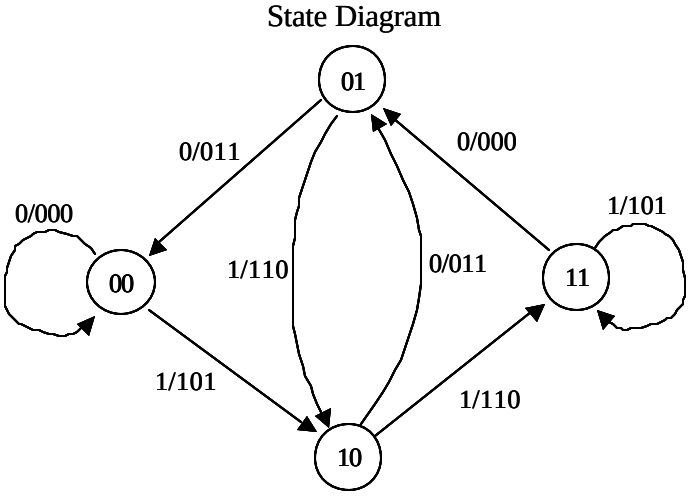

Figure 28 Example State Diagram

For faster computation, this information can be collected into a pair of tables, a state output table and a state transition table. The state output table tabulates the output bits for all possible inputs and states. The state transition table tabulates the next state given the input bit and the current state. Figure 29 shows the state tables for the encoder shown in Figure 27.

State Output Table

I	S_1	S_2	O_0	O_1	O_2
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	0
1	1	1	1	0	1

State Transition Table

I	S_1	S_2	S_1	S_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

Figure 29 Example State Tables for a Rate 1/3 Code given in Figure 27

3.3.3.1 Decoding Convolutional Codes

Because of the sequential dependencies of the convolutional code, it can be shown that the optimal performance is given by a Maximum Likelihood Sequence Estimator (MLSE). This is most commonly implemented using a Viterbi decoder which is considered for physical layer implementation. Example performance of a Viterbi decoder is shown in Figure 30, for r=1/2, r=1/3 and hard or soft decoding. Note that for common operating SNRs, convolutionally encoded systems outperform uncoded systems. However, for low SNRs, convolutionally encoded systems underperform uncoded systems.

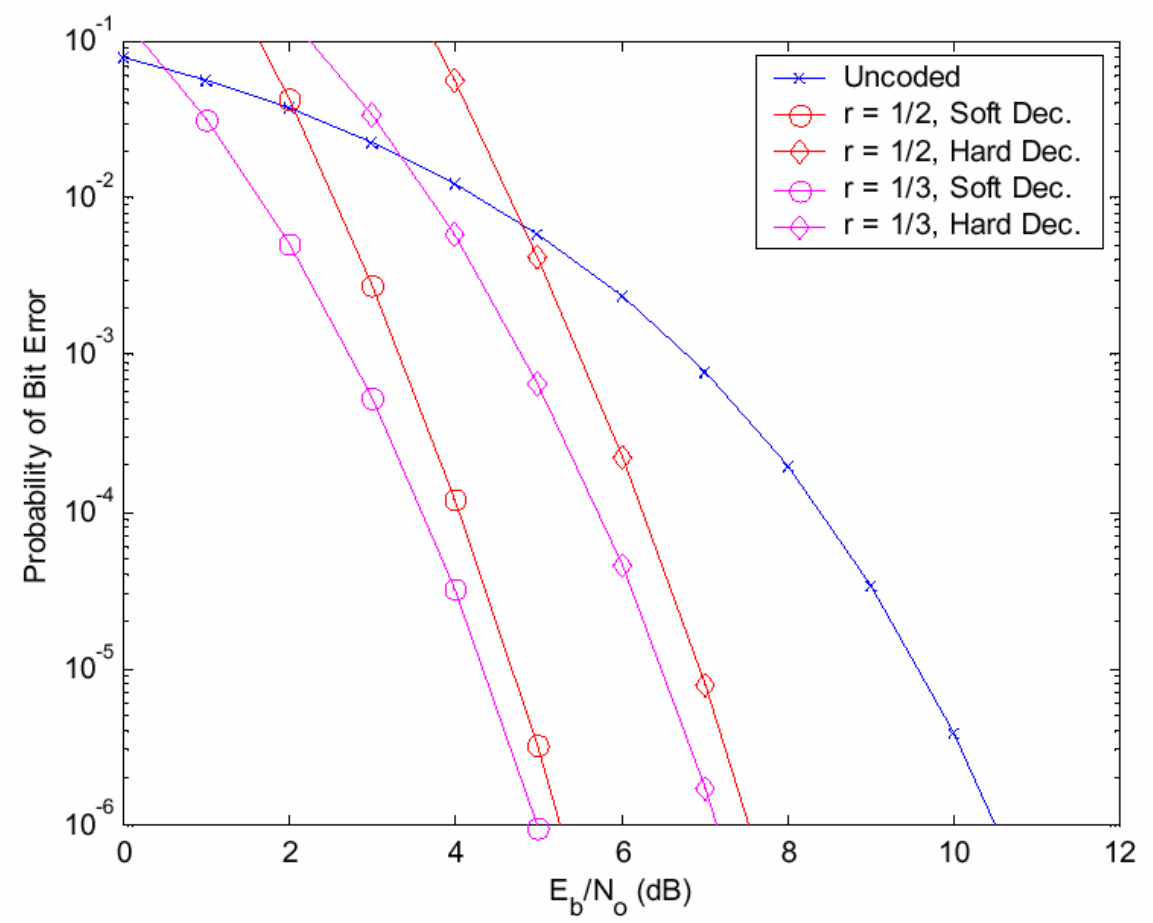

Figure 30 Viterbi Decoder Performance

A Viterbi decoder estimates which sequence of states is the most likely transmitted sequence of states by iteratively applying a metric between the sequence of received words and the words that would be produced for a sequence of state transitions. For hard decision decoding, this metric can be calculated from the Hamming distance between the received word and the potentially encoded word. The metrics from each state are calculated as (19)

$$m_i = H\left(O\left(s_j, s_k\right), r\right) \tag{19}$$

where $O(s_j,s_k)$ is the output word produced by a transition from state j to state k, and $H(a,b)$ is the Hamming distance between words a and b. The precalculation and storage of output state tables and state transition tables, like those shown in Figure 29 facilitate this operation.

As it would be impractical to compute all possible sequences, the Viterbi decoder only maintains the most likely sequences, also known as the surviving sequences, for each state. A sequence coming into a state survives if its metric is lower than the other sequences coming into the state. This survival process can be readily accomplished by maintaining a vector of sequences with a length equal to the number of states in the system and a vector of metrics associated with each sequence.

Side information can also be used to improve the performance of the Viterbi algorithm. For instance, if it is known that the initial state must be the relaxed or all 0s state, then initial sequence estimation must only consider transitions from the 0 state. Since handling this as a special case can be cumbersome, it is customary to effectively accomplish this by initializing the metrics for all sequences that do not begin in the 0 state with very large values.

3.3.3.2 Viterbi Algorithm Implementation

The Viterbi algorithm can be implemented as follows:

1. Initialize state transition table, state output table.
2. Initialize metrics vector with 0 in state 0, and a large number in the other states.
3. Initialize the sequence vector with zeros.
4. Read in a received word, *r*.
5. Create a temporary copy of the metric vector and the sequence vector.
6. Compute transition metrics, m_1 and m_2 for the two possible transitions for state 0.
7. Temporarily add this to the metrics from the originating states to form two other 8. metrics $m_{1,\text{temp}}$ and $m_{2,\text{temp}}$.
8. Compare $m_{1,\text{temp}}$ and $m_{2,\text{temp}}$,
   if $m_{1,\text{temp}}$ is smaller
   set state 0's element in the temporary copy of the metric vector to $m_{1,\text{temp}}$
   set state 0's element in the temporary copy of the sequence vector to the sequence held in the element in the sequence vector associated with originating state appended with the bit that would cause the transition.
   else
   do the same as for $m_{1,\text{temp}}$ but using the other possible originating state
9. Repeat steps 6-8 for all states.
10. Copy the temporary copy of the metric vector and the sequence vector to the primary copy.
11. Repeat steps 4-10 for all elements in the received stream.

3.3.3.3 Convolutional Coding Implementation

For this simulation, the polynomials specified for 802.11a were used. These are repeated for completeness in (20). Note that this is a rate 1/2 encoder with six memory elements.

$$\begin{aligned} g_0(D) &= 1 + D^2 + D^3 + D^5 + D^6 \\ g_1(D) &= 1 + D + D^2 + D^3 + D^6 \end{aligned} \tag{20}$$

This use of 6 memory elements means that 64 states are maintained in the associated tables. As the total length of the encoded sequence is undefined, a maximum trellis depth of 32 is used. Outputs are produced before the trellis is forced back to state 0 by the tail bits. Note that this is slightly less than the typical run-time decoder depth of 5.8K or 35. This tradeoff was made so that the survivor sequence estimators could be implemented with an array of 64 integers.

An optimized rate 1/2 encoder and hard decision decoder are implemented. Rates 2/3 and 3/4 codes are implemented by puncturing the output stream appropriately and appropriately inserting erasure events in the received stream. The process of puncturing and inserting erasures is implemented according to the specifications of 802.11a. All operations are performed in C.

For the purposes of this comparison, only the rate 1/2 encoder is considered.

3.3.3.3.1 Baseline Simulation

For the baseline simulation, no modifications from the standard OPNET pipeline were made. This represents an uncoded system.

3.3.3.3.2 Abstracted Physical Layer

In the abstracted physical layer approach, a union bound for the performance of the
For a rate 1/2 convolutional code with a constraint length of 7, the probability of bit error can be approximated as (21)

$$P_e \le \frac{1}{2}\left[36D^{10} + 211D^{12} + 1404D^{14} + 11633D^{16}\right] \tag{21}$$

where D is given by

$$D = \sqrt{4e\left(1-e\right)} \tag{22}$$

where e is the uncoded probability of bit error.

As this abstraction is a function of the raw uncoded BER, the decoding abstraction conceptually should be implemented in the *ecc* stage. Based on the ber value in the packet field *OPC_TDA_RA_BER* is evaluated. This result is then used to calculate (21). This forms the decoded BER that is the used to determine whether or not the packet is accepted.

3.3.3.3.3 Full Physical Layer Simulation

In this approach, the full encoding and decoding algorithms are implemented by modifying the *ecc* stage. While MATLAB provides an attractive development environment, Section 2.2 demonstrated that co-simulation with C provided identical fidelity to the MATLAB implementation with a significant reduction in simulation time. Thus this full physical layer simulation only considers a C co-simulation.

Operations Performed in OPNET
The following are the tasks performed in OPNET (within the modified *ecc* stage):

- Determining packet size
- Fetching the packet's modulation and SNR
- Determining error threshold
- Assigning packet acceptance/rejection
- Releasing the receiver channel

Note that the error threshold should generally be identically 0 for this stage as the *ecc* stage is commonly used to model the number of errors which could be corrected. As the full error correction scheme is implemented in this pipeline, any residual errors would represent uncorrectable errors. The release of the receiver channel is required for any *ecc* pipeline stage code and is not specific to this implementation.

Operations Performed in C

The following are the tasks performed in C:

- Initializing the encoder
- Generating random data (including space allocation)
- Encoding the data
- Assigning bit errors by flipping bits in the encoded bit stream
- Decoding the data
- Counting errors in the decoded data

Procedural Description

Figure 31 shows the modified process flow for the modified *ecc* pipeline stage for the C implementation. In the diagram, purple boxes correspond to operations performed using OPNET KP's and black boxes correspond to operations performed using C routines. Notice that during the initialization branch, all of the information that needs to be only computed once (such as state output tables and state transition tables) is computed. In the normal operation branch, a random baseband data signal is generated according to the packet length. This is a time consuming process and should be avoided when possible to directly assign bits within the packet. Also notice that OPNET is used solely for retrieving and setting OPNET specific variables; all other processing is performed in C function calls.

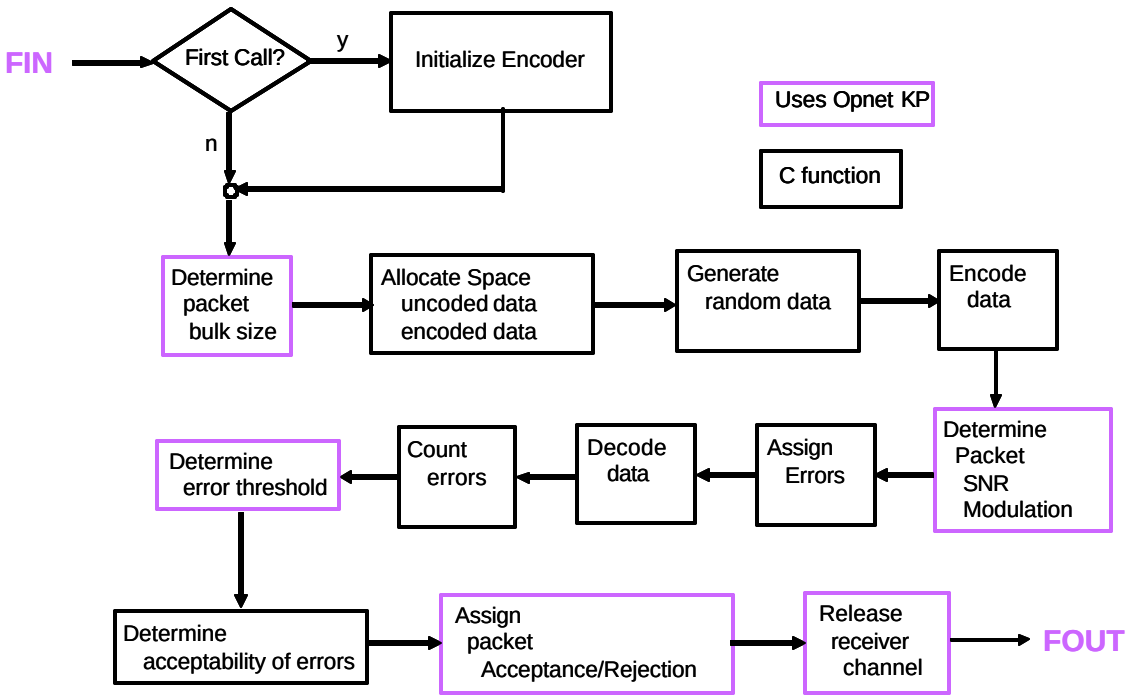

Figure 31 Modified *ecc* Process Flow for Convolutional Encoding

3.4 System Implementation

This section provides an overview of the implemented system. Particular attention is paid to the implementation of the link layer protocol. For details on how interfaces between layers (co-simulations) are established and the procedural steps used to implement the physical layer in the co-simulation please refer to sections 3.3.3.2 and 3.3.3.3.

3.4.1 System Overview

As shown in Figure 32, the system consists of three types of nodes: transmitters, receivers, and jammers. Each of these nodes may be fixed or mobile. The following gives a description of each of these nodes.

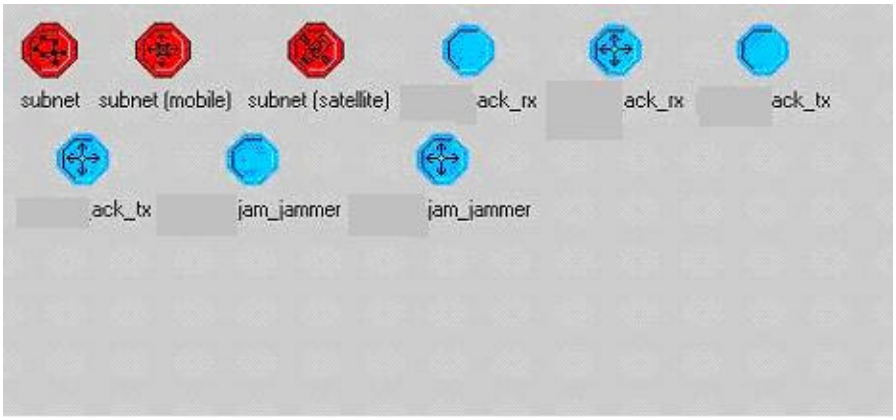

Figure 32 System Palette

3.4.1.1 Transmitter

The transmitter node model shown in Figure 33 consists of the following:

- A source module (*src*) which uses OPNET's *simple source* process model to generate packets. The probability distribution functions of the packet size and packet inter-arrival time used within the simple source may be set as desired from those available within OPNET.
- A queue module (*ack_tx_proc*), the procedural behavior of this queue, custom written to implement the *stop and wait* ACK link layer protocol is detailed in a later section.
- A radio transmitter module (*tx_tx*) which provides an interface to transmit the packets on a radio channel.
- A radio receiver module (*tx_rx*) which provides an interface to receive ACKS from the radio channel.
- A sink module (*sink*) which sinks received ACK packets.

Packet flow:
After generation, packets move through a packet stream to the *ack_tx_proc* module before being sent to the radio transmitter. Acknowledgements move from the radio receiver to the *ack_tx_proc* module and are sent to the sink.

To avoid interference between data packets and control packets, the transmitter (*tx_tx*) and receiver (*tx_rx*) modules occupy different frequency bands.

To enhance simulation flexibility, the transmitter node's power attribute is promoted so it can be set at run-time by the user.

Some of the specific parameters of the transmitter node are listed in Table 9.

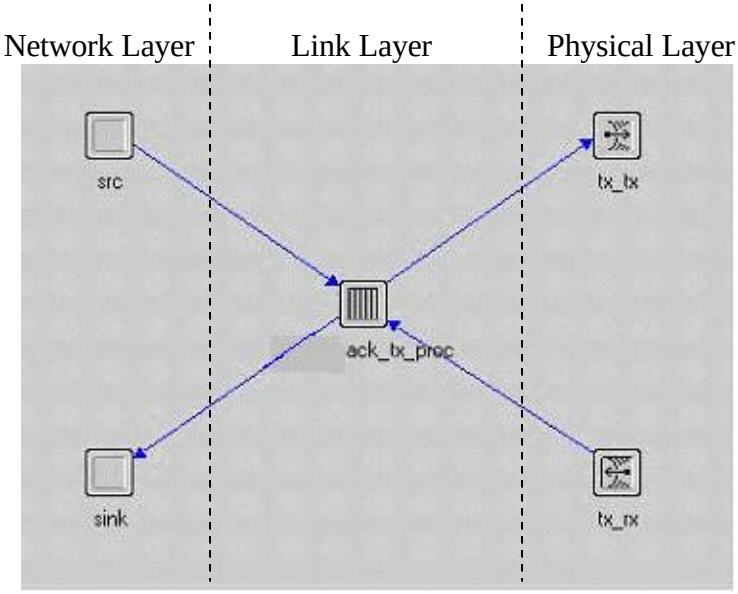

Figure 33 Transmitter Node Model

Table 9 Transmitter Attributes

Attribute	Type/ Value
Modulation	BPSK
Power	Promoted to be set by user at run time
Data Rate	100 Kbps
Bandwidth	100 KHz
Min. Frequency (*tx_tx*)	1.8 GHz
Min. Frequency (*tx_rx*)	2.2 GHz

Note: The transmitter and receiver modules in the nodes use different frequency bands in order to provide different logical channels for the exchange of control and data packets.

3.4.1.2 Receiver

The receiver node model shown in Figure 34 consists of the following:

- A radio receiver module (*rx_rx*) which provides an interface to receive data packets sent by the transmitter.
- A radio transmitter module (*rx_tx*) which provides an interface to send acknowledgements to the transmitter.
- A queue module (*ack_rx_proc*), the procedural behavior of this queue, custom written to implement the *stop and wait* ACK link layer protocol is detailed in a later section.
- A sink module (*sink*) which sinks data packets received.

Table 10 lists additional node level attributes of the receiver node.

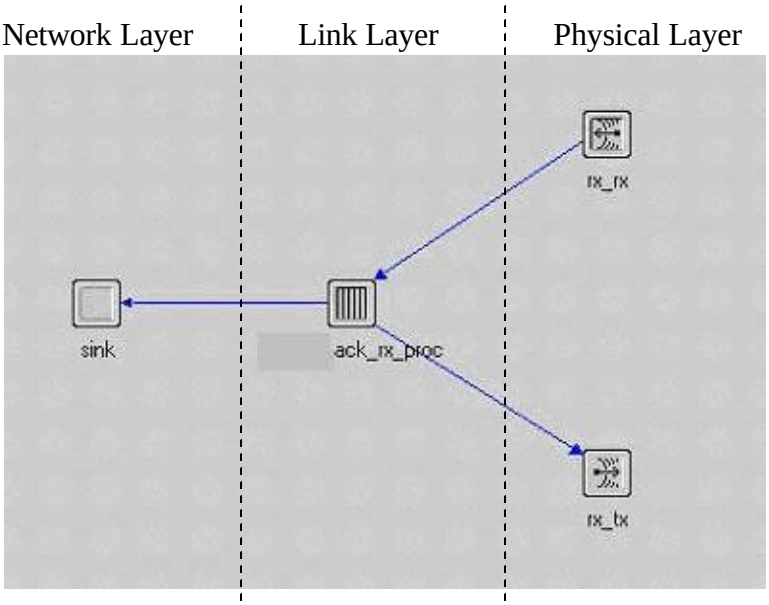

Figure 34 Receiver Node Model

Table 10 Receiver Attributes

Attribute	Type/ Value
Modulation	BPSK
Data Rate	100 Kbps
Bandwidth	100 KHz
Min. Frequency (*rx_rx*)	1.8 GHz
Min. Frequency (*rx_tx*)	2.2 GHz

3.4.1.3 Jammer

The jammer node, shown in Figure 35, introduces interference into the network. Like the transmitter node, it consists of the following:

- A source module (*src*) which generates packets.
- A radio transmitter module (*jam_tx*) which provides an interface to transmit the packets on a radio channel. Its behavior is similar to the transmitter node, but its signal modulation is set to *jammod*.

The jammer node's power attribute is also promoted so it can be set at run time by the user.
Some of the specific parameters of the jammer node are listed in Table 11.

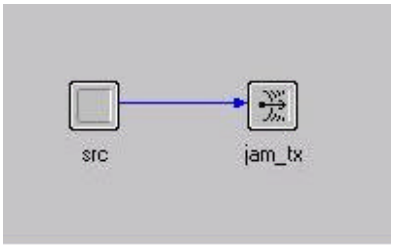

Figure 35 Jammer Node Model

Table 11 Jammer Attributes

Attribute	Type/ Value
Modulation	*jammod**
Power	Promoted to be set by user at run time
Data Rate	100 Kbps
Bandwidth	100 KHz
Min. Frequency (*rx_rx*)	1.8 GHz

***Note :** The Radio version of OPNET provides for the modeling of link level effects in radio networks by computing average power levels for received signals. This computation is a link budget that by default includes propagation losses, antenna gains, and transmitter power level.
Several procedures of the Radio Link Pipeline are devoted to modeling the interaction of concurrently transmitted signals. This allows models to be constructed that take into account self-interference and jamming. A distinction is made between signals that carry properly formatted and receivable information and those which are treated simply as interference. The *jammod* modulation type is not a specific waveform but a means to recognize a signal as interference by OPNET. This categorization is performed by the Channel Match Model of the pipeline. Interference signals undergo the same processing with regard to computation of their received power level; however they are not considered for eventual forwarding to the higher layers of the receiver node.

3.4.2 Process Models in OPNET

In OPNET, a network is made up of individual nodes, and a node is made of modules. The hierarchy of packet flows through the different modules in a node effectively models the different layers in a networking system. The *process model* defines a module's behavior. Certain modules are limited in the types of behavior they can represent. Generators are simple traffic sources provided by OPNET for convenience and various transmitters and receivers represent interfaces to links defined in the network domain. Two types of modules, *queues* and *processors*, support fully user-specified behavioral modeling through the building of process models.

OPNET thus provides a convenient platform for building higher layers tailored to exact user-defined behavioral specifications or protocol standards through the building of process models.

A process model is a finite state machine (FSM). It represents a module's logic and behavior. An FSM consists of any number of states that a module may be in and the necessary criteria for changing states. OPNET simulations are made up of events. Process models respond to events (such as packet arrivals, etc.) and may schedule new ones. In an OPNET simulation, when an event occurs that affects a module, the simulation kernel passes control to the module's process model via an interrupt. The process model executes related user-defined code and returns control to the simulation kernel.

Hence, the functionality provided by a layer in a network is easily modeled by a module with a process model that emulates the sequence of packet operations in that layer.

3.4.3 Process Model for the Stop and Wait Protocol

3.4.3.1 The Transmitter Process Model

As shown in Figure 36 the FSM for the transmitter process model consists of the following states:

- *init*: The first time a process model is invoked, it begins in the initialization state. This state initializes the variables used in the process and registers the statistics to be collected.

- *idle*: By default the process is always in the idle state.

- *transmit*: A *READY* condition which is an interrupt triggered by a packet arriving from the generator causes the process to transition from the *idle* to the *transmit* state. The packet which caused this interrupt is transmitted if no ACK is pending from the receiver. If an ACK has yet to be received, then the packet is queued. A timer is started for a transmitted packet and an ACK pending is recorded. The statistic for the number of packets generated is recorded in this state.

- *rcv*: Acknowledgements arriving from the receiver node trigger an interrupt, i.e. the *ACK_ARRIVAL* condition, to pass control to this state. The ACK received is recorded, and the timer is reset. If there are any packets, queued, the first packet in the queue is transmitted, a timer is started for the transmitted packet and an ACK pending is recorded.

- *resend*: A *TIMEOUT* occurs if the timer for a transmitted packet expires with no ACK having been received. The packet is resent and the timer started again. An ACK pending is recorded for the transmitted packet. The statistic for the number of packets resent is recorded in this state.

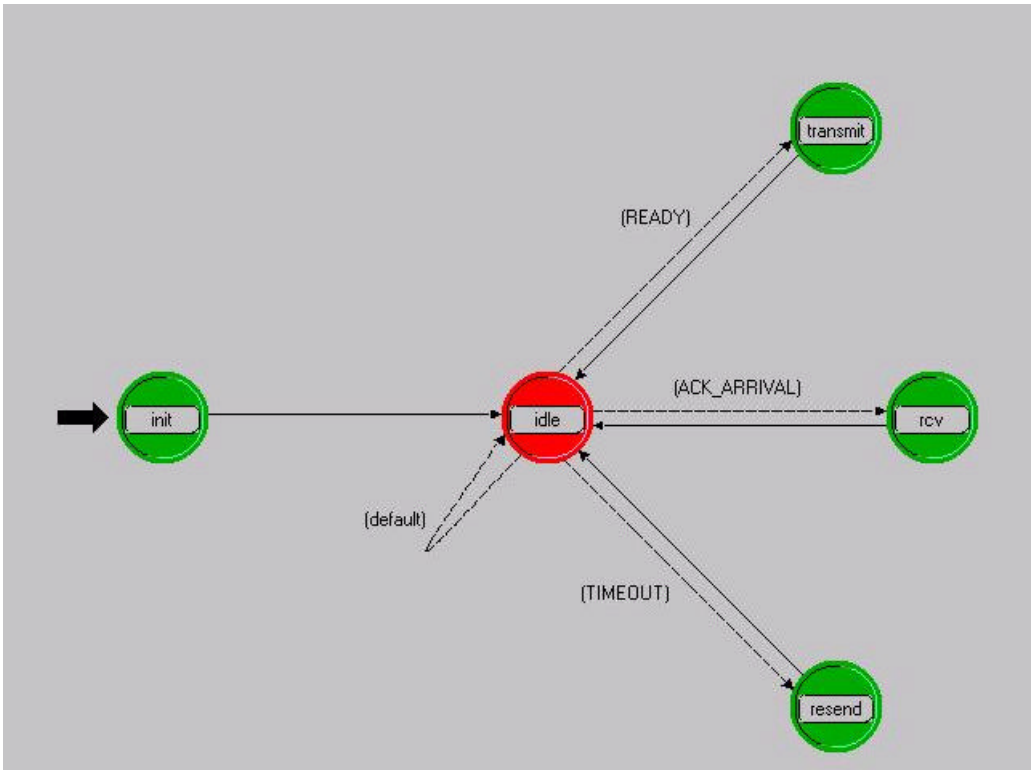

Figure 36 Transmitter Process Model

Note: Solid lines represent unconditional transitions between states while dashed lines represent conditional transitions between states. For example, if the process is in the *transmit* state, the process moves to the *idle* state on completion of the code sequence in the *transmit* state. However, if the process is in the *idle* state, it will wait for the *ACK_ARRIVAL* condition to be satisfied before it moves to the *rcv* state.

3.4.3.2 The Receiver Process Model

As shown in Figure 37 the FSM for the receiver process model consists of the following states:

- *init* and *idle* as described in the previous section.
- *rcv*: A packet arriving from the transmitter triggers an interrupt, the *FRAME_ARRIVAL* condition, to pass control to the *rcv* state. The packet is accepted, an ACK is sent to the transmitter for the packet received, and that packet is sent to the sink.

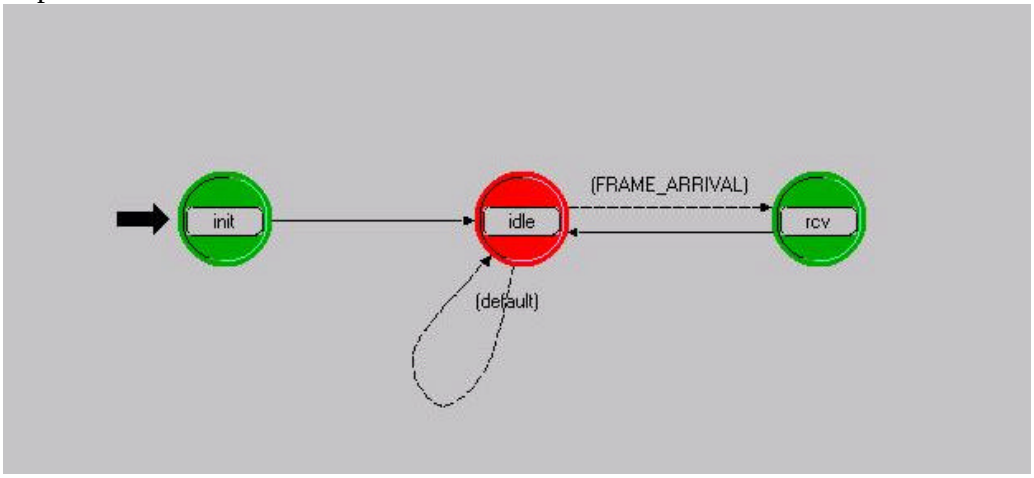

Figure 37 Receiver Process Model

3.5 Results

The system consists of a fixed transmitter–receiver pair, with a jammer as shown in Figure 38.

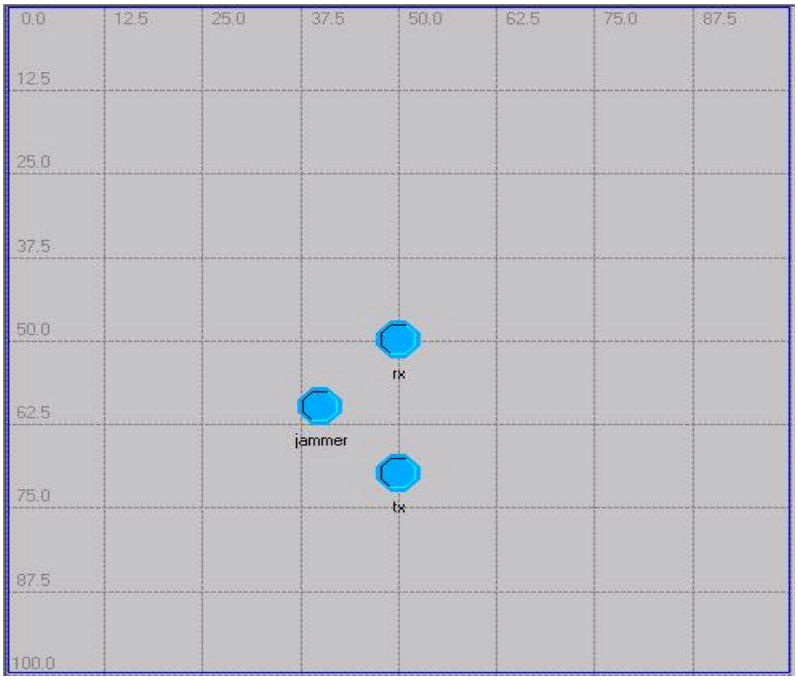

Figure 38 Network Scenario 1

The jammer power was set equal to the transmitter power of 1W. Convolutional encoding and decoding was implemented in the system through the modification of the error correction pipeline stage in the radio receiver module of the receiver through running the Viterbi decoder in C.

Figure 39 and Figure 40 show the packets generated by the transmitter and those received at the receiver without and with convolutional coding implemented respectively. An increase in the number of packets sent in seen without coding, as resent packets are included in this statistic. There is significant improvement in channel throughput. In the case with channel coding, all packets sent are received, whereas without any channel coding, the jammer degrades system performance and only 45% of the packets sent a received.

Figure 41 and Figure 42 show the number of packets resent and the queue size at the radio transmitter module without and with channel coding respectively. The spikes observed in the radio transmitter queue size without channel coding are due to queuing of incoming packets when packets are being resent. These spikes correspond to the times when a packet is being resent. With channel coding the number of resends drops to zero from 33.

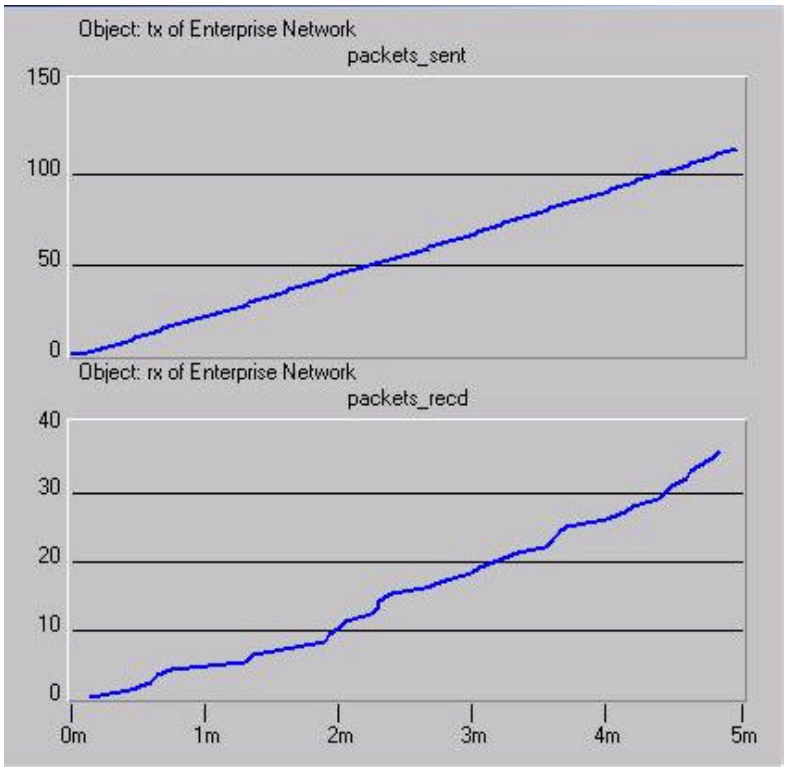

Figure 39 Packets Sent and Received without Channel Coding

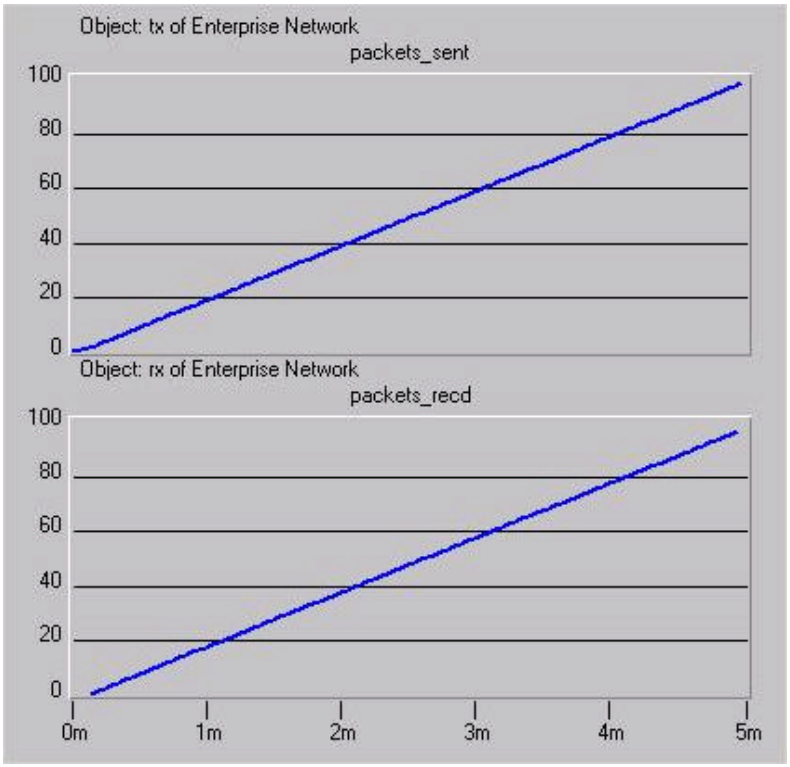

Figure 40 Packets Sent and Received with Channel Coding

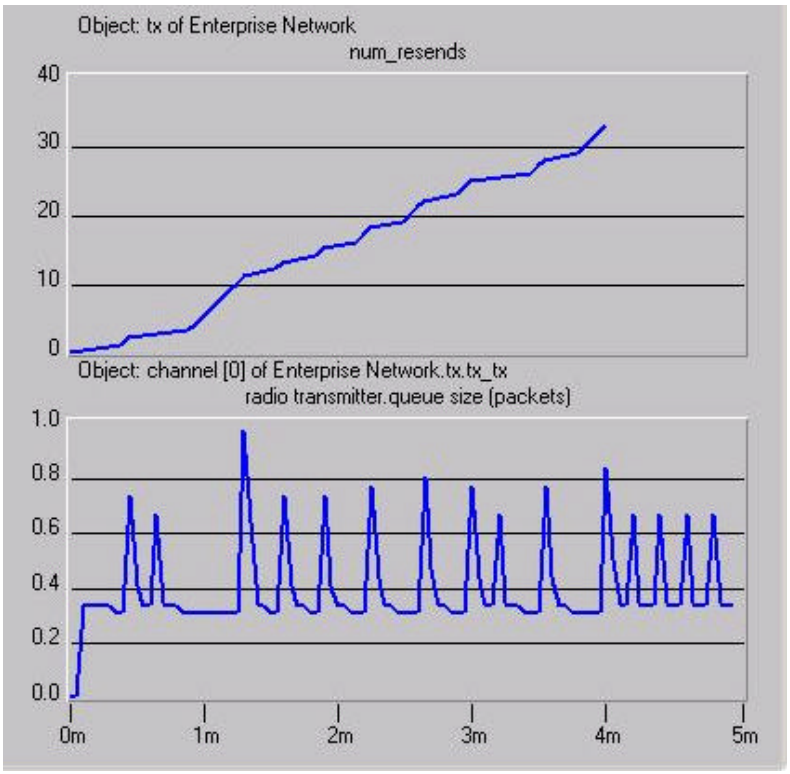

Figure 41 Number of Resent Packets and Queue Size at Radio Transmitter without Channel Coding

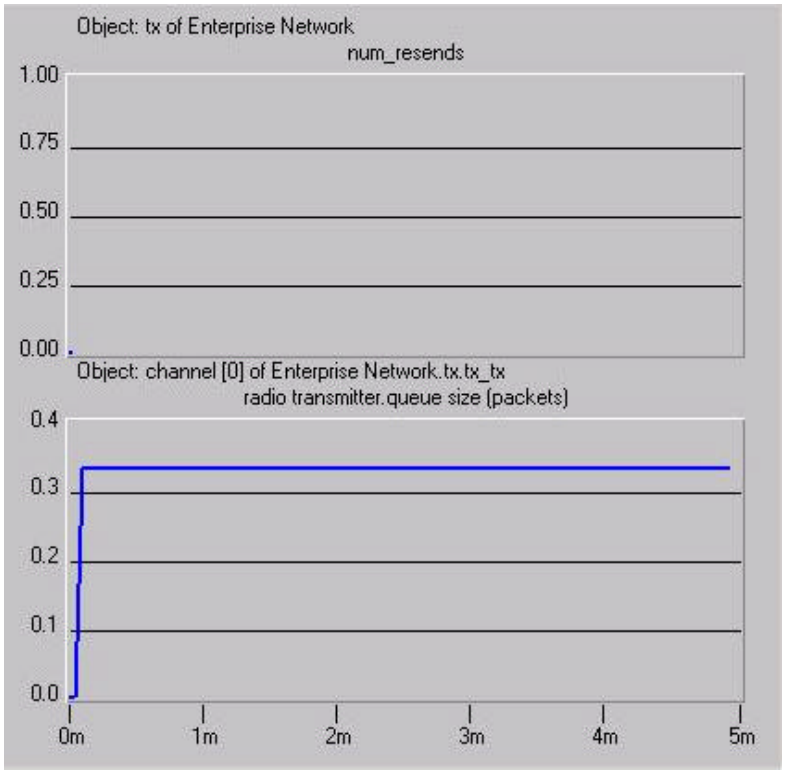

Figure 42 Number of Packets Resent and Queue Size at Radio Transmitter with Channel Coding

Table 12 summarizes the key results on the performance of the ACK scheme without and with coding implemented at the receiver.

Table 12 Summary of Results: Scenario 2

	WITHOUT CHANNEL CODING	WITH CHANNEL CODING
Number of Resent Packets	33	0
System Throughput	45%	100%

On increasing the number of jammers in the system to 2, the power of each being set to 1 W as shown in Figure 43 it is seen that with no coding, the throughput of the system drops to zero, due to the high level of interference. With convolutional coding, the throughput is not 100%, as was the case with a single jammer, and the number of resent packets rises to 5 from 0. This is shown in Figure 44 and Figure 45.

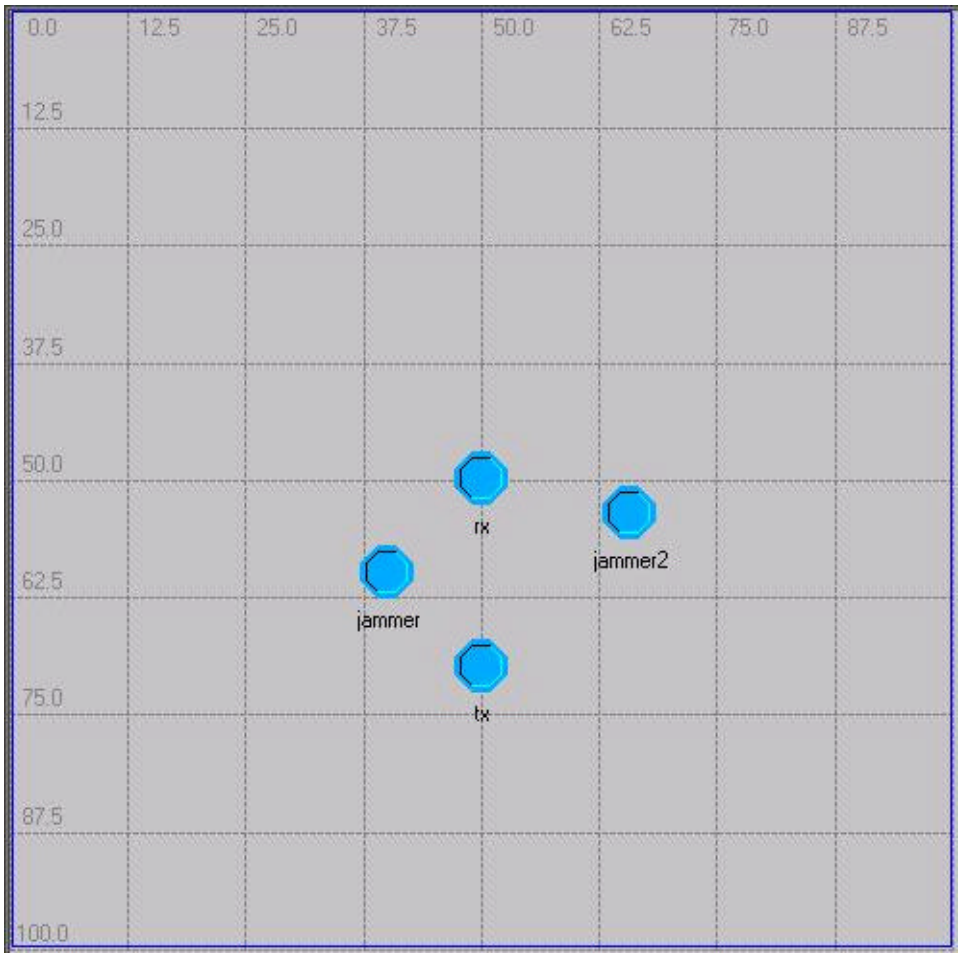

Figure 43 Network Scenario 2

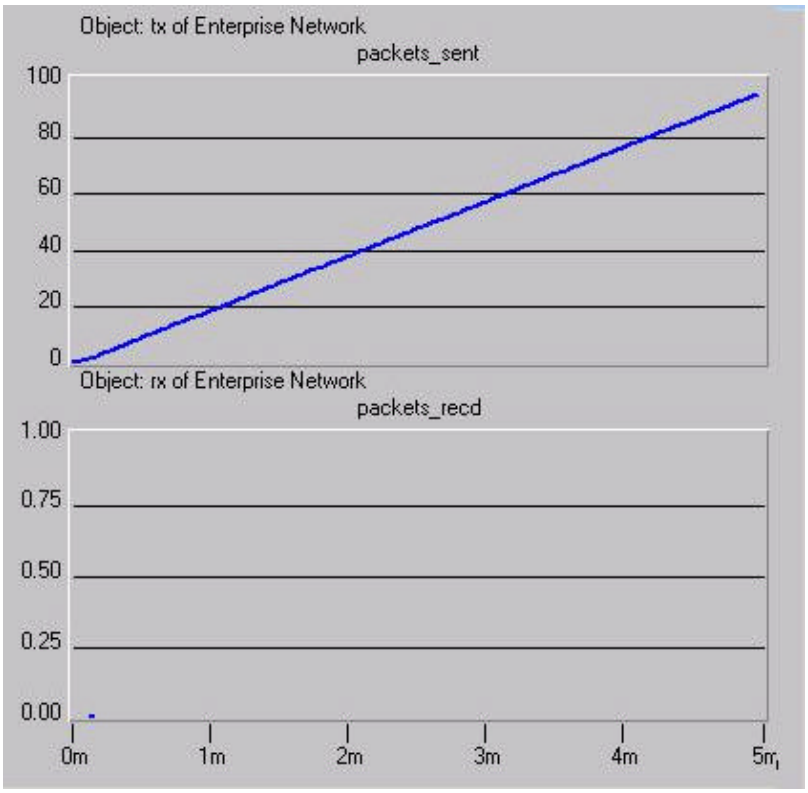

Figure 44 Packets Sent and Received without Channel Coding and 2 Jammers

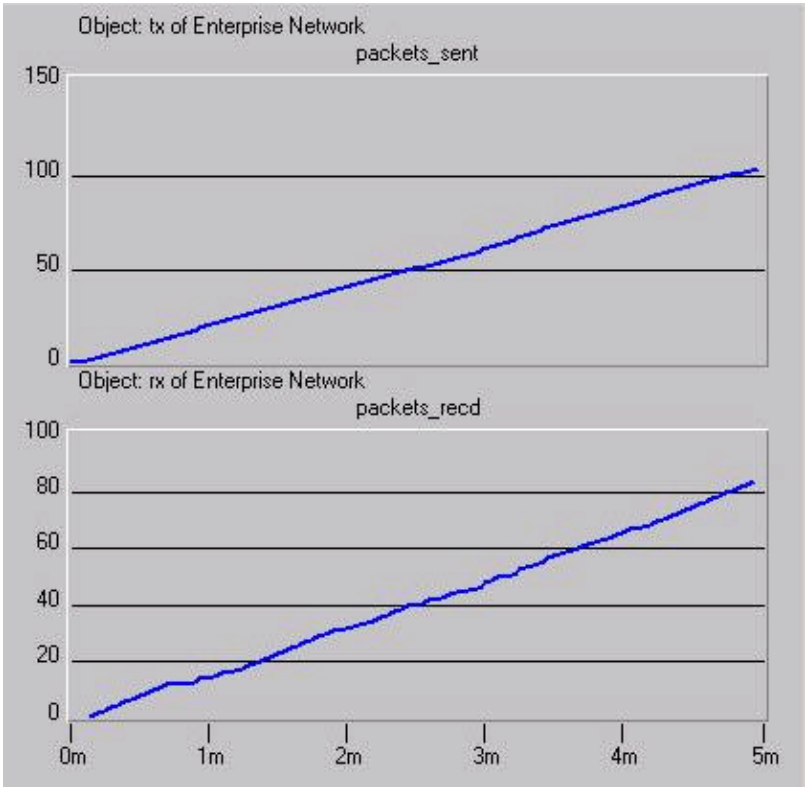

Figure 45 Packets Sent and Received with Channel Coding and 2 Jammers

Table 13 Summary of Results: Scenario 2

	WITHOUT CHANNEL CODING	WITH CHANNEL CODING
Number of Resent Packets	90	5
System Throughput	0%	82%

The scenarios described in Figure 38 and Figure 43 were simulated using both the abstracted physical layer and full physical layer simulations. The results obtained were identical in both cases. However the full physical layer simulation ran at a speed of 630 events/second while the abstracted physical layer simulation ran at a speed of 2397 events per second. Hence, a 280.5 % increase in speed is seen with no loss in fidelity in simulating the abstracted physical layer.

However, an important aspect to note is that the abstracted physical layer simulation implicitly assumes a static channel over the duration of a packet, as the simulation granularity is not as fine as the bit level. This is a reasonable assumption for an AWGN channel as seen from the results so far wherein both the abstracted physical layer and the full physical layer simulations yield identical results.

To compare the 2 simulation approaches further, Rayleigh fading is introduced into the channel model for the network in Scenario 1.

In the abstracted physical layer simulation, the SINR transmission data attribute in each transmitted packet is multiplied with the Rayleigh fading coefficient. Based on this SINR value, OPNET provides a BER value through a table look up. This is the uncoded BER through which the coded BER is calculated as in section 3.3.3.3.2.

In the full physical layer simulation, each transmitted bit is multiplied by the fading coefficient and fed to the Viterbi decoder.

As expected the system performs worse in the abstracted physical layer simulation, as this corresponds to static fading, in which entire packets may be wiped out by burst of errors. With fast fading, the errors are interleaved and the Viterbi decoder has a better chance of correction leading to better system performance. Hence the abstracted physical layer provides a pessimistic view of the system, while completely simulating the physical layer is a more accurate representation of a real system.

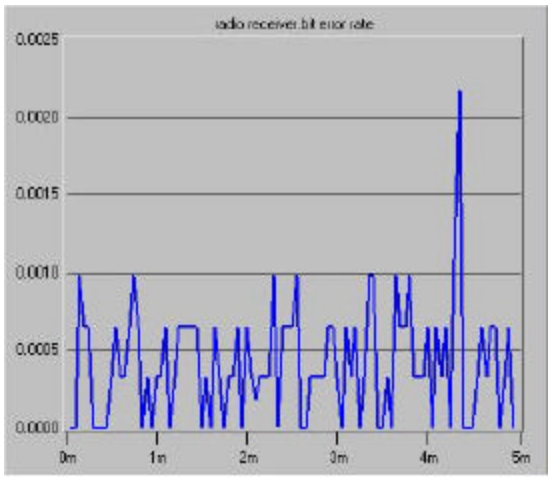

Figure 46 Receiver BER with Fast Fading: Full Physical Layer Simulation

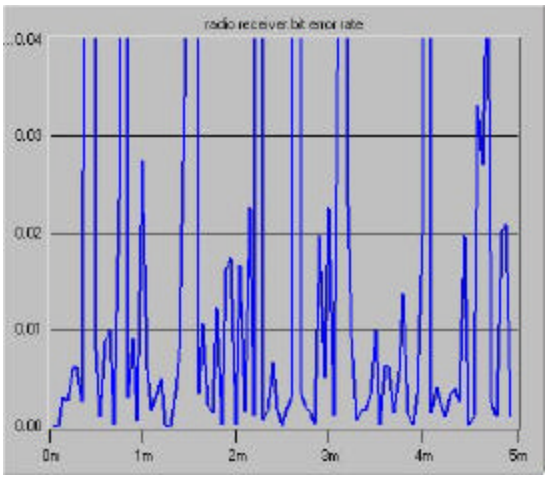

Figure 47 Receiver BER with Static Fading: Abstracted Physical Layer Simulation

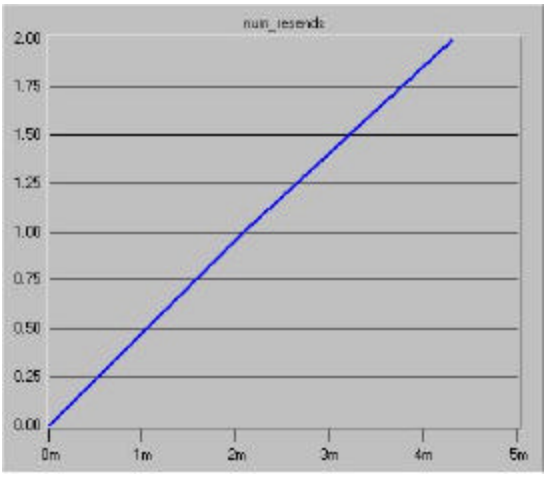

Figure 48 Number of Resent Packets with Fast Fading: Full Physical Layer Simulation

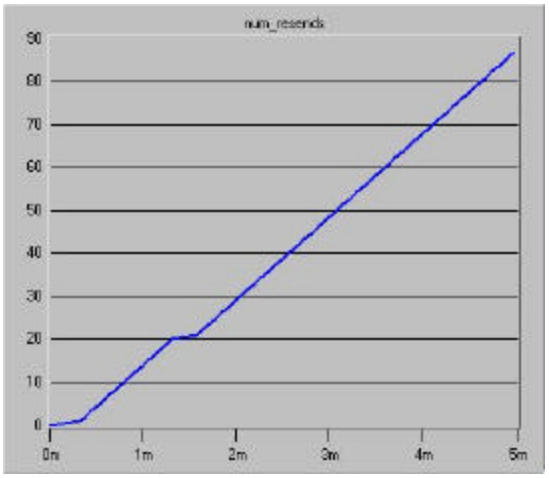

Figure 49 Number of Resent Packets with Static Fading: Abstracted Physical Layer Simulation

3.6 Conclusions

The fundamental communication mechanism in OPNET is packet flow. This is suitable for network-level simulations where higher layer communications are based on packets. However at the physical level communication occurs through symbol streams. OPNET abstracts the transmission of a stream of symbols and the physical layer processes via a set of transmission data attributes attached to each packet, such as BER, SNR, and delay. These attributes are set and modified through a series of pipeline stages as detailed in Section 2.1.1.

While the approach used by OPNET is much better in terms of simulation run-time, it does not provide a sufficient level of simulation fidelity for complex physical layer processes. For example, OPNET calculates the BER of packets by doing a table look-up based on the modulation scheme and SINR. This value is then stored in the packet's BER transmission data attribute. *While fast, this method implicitly assumes a static channel over the duration of a packet*. This ignores an important category of MANET operation.

For systems in which physical layer dynamics play an important role in overall system performance, such as MANETs with fading channels, the physical layer must be completely simulated. This is done by carrying out simulations in C and/or MATLAB which are carried out in the pipeline stages. OPNET allows the pipeline stages to be modified or user–defined. But a complete calculation of these parameters increases simulation run-time considerably. Hence, in cross-layer simulation a trade-off must be made between simulation fidelity and run-time.

Specifically for convolutionally encoded data, for a system where the channel may be assumed to be static over the duration of a packet, simulation should be performed with an abstracted representation of the performance of the encoder, such as the union bound used in Section 3.3.3.3.2. This provides both excellent fidelity and excellent run-time. However, for a dynamic channel (significant changes in SNR over the length of a packet), the encoder needs to be fully implemented.

4 A Medium Access Control Scheme for MANETs Equipped with Smart Antennas

4.1 Introduction

The primary function of the MAC (Medium Access Control) layer is to mediate access to the communication medium, which the physical layer provides an interface to. The MAC layer uses services provided by the physical layer such as carrier sensing, transmission and reception of MAC frames, etc. In turn, physical layer parameters such as SINR are affected by the performance of the MAC layer. An inherent coupling between the physical and MAC layers is seen to exist. Hence, the MAC layer is a good example of where this coupling may be exploited through a cross-layer design approach to improve overall system performance. This section describes such a medium access control scheme developed for a MANET equipped with smart antennas at each node, as an example of cross-layer system design.

4.1.1 Medium Access Control

Before transmitting frames, a station must first gain access to the medium, which is a radio channel that stations share. The MAC layer manages communications between nodes by coordinating access to the shared channel and utilizing protocols that enhance communications over the shared medium. The MAC layer uses the physical layer to perform functions such as carrier sensing, transmission and reception of MAC frames in order to perform its basic function of controlling access to the shared communication medium.

The most common MAC protocol for local area wireless networks is the CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Under CSMA/CA, a node that wants to send a data packet will first wait for the channel to become available and then transmit a short RTS (Request To Send) packet. The potential receiver, assuming it perceives an idle channel will immediately respond with a CTS (Clear To Send) packet that authorizes the initiating node to transmit and also informs neighboring hidden nodes (i.e. nodes that are outside the communication range of the transmitter and might harm the ongoing transmission) to refrain from transmitting during the duration of the transmission.
Nodes that overhear the RTS packet will refrain from transmitting until the CTS packet can be received, and those overhearing the CTS packet will refrain from transmitting until the whole data packet is sent. In this way the hidden node and exposed node problems are alleviated.

Since CSMA/CA requires that the potential transmitter must receive a control packet, it is necessary for not only the nodes around the transmitter but also those around the receiver to remain silent leading to reduced spatial reuse.

The 802.11 standard specifies a common medium access control layer that supports the operation of 802.11-based wireless LANs. The standard defines two forms of medium access, the distributed coordination function (DCF) and the point coordination function

(PCF). The DCF is mandatory and based on the CSMA/CA (carrier sense multiple access with collision avoidance) protocol. CSMA/CD (collision detection) is not used because a station is unable to listen to the channel (physical carrier sensing) while transmitting. With DCF, 802.11 stations contend for access and attempt to send frames when there is no other station transmitting. If another station is sending a frame, stations are polite and wait until the channel is free.
As a condition to accessing the medium, the MAC layer checks the value of its network allocation vector (NAV), which is a counter resident at each station that represents the amount of time that the previous frame needs to send its frame. The NAV must be zero before a station can attempt to send a frame. This is known as virtual carrier sensing. Prior to transmitting a frame, a station calculates the amount of time necessary to send the frame based on the frame's length and data rate. The station places a value representing this time in the duration field in the header of the frame. When stations receive the frame, they examine this duration field value and use it as the basis for setting their corresponding NAVs. This process reserves the medium for the sending station.
An important aspect of the DCF is a random back off timer that a station uses if it detects a busy medium. If the channel is in use, the station must wait a *random* period of time before attempting to access the medium again. This ensures that multiple stations wanting to send data do not transmit at the same time. The random delay causes stations to wait different periods of time and avoids all of them sensing the medium at exactly the same time, finding the channel idle, transmitting, and colliding with each other. The back off timer significantly reduces the number of collisions and corresponding retransmissions, especially when the number of active users increases.
For supporting time-bounded delivery of data frames, the 802.11 standard defines the optional point coordination function (PCF) where the access point grants access to an individual station to the medium by polling the station during the contention free period. Stations cannot transmit frames unless the access point polls them first. The period of time for PCF-based data traffic (if enabled) occurs alternately between contention (DCF) periods. The access point polls stations according to a polling list, then switches to a contention period when stations use DCF. The optional request-to send and clear-to-send (RTS/CTS) function allows the access point to control use of the medium for stations activating RTS/CTS.

Although the 802.11 standard does specify an infrastructure through the use of access points, the DCF form of the medium access applies to any arbitrary topology of nodes without the use of access points, and hence is relevant to the discussion of medium access control for MANETs.

In 802.11 if a node is aware of an ongoing transmission it will not participate in a transfer of data itself. Directional MAC schemes apply a similar logic, but on a per beamwidth basis for the antenna arrays. Hence, capacity may be increased by a factor proportional to the angular width of the antenna beams.

4.1.2 Medium Access Control with Directional Antennas

The CSMA protocol when applied to ad hoc networks causes the hidden terminal and exposed terminal problems. CSMA/CA, medium access control with collision avoidance (MACA) relieves these problems by using the RTS and CTS frames.

These traditional MAC protocols assume the use of omnidirectional antennas. Since CSMA/CA requires that the potential transmitter must receive a control packet, it is necessary for not only the nodes around the transmitter but also those around the receiver to remain silent leading to reduced spatial reuse. Nodes with directional antennas are capable of transmitting and/or receiving selectively in certain directions and significantly reduce the chances of collision and increase the effective network capacity.

MANETs using omni-directional antennas severely limit system performance as the entire space around a node up to its radio range is seen as a single logical channel. This limitation is magnified in multi-hop routing mechanisms as a node may lie on many active routes and need considerably more bandwidth. These nodes become bottlenecks for the route on which they lie and the network capacity is surprisingly low due to poor spatial reuse.

In one of the first efforts to adapt the IEEE 802.11 to directional antennas a directional RTS frame is used [11]. It narrows the area in which an unintended receiver can overhear the RTS frame and thus significantly reduces the exposed terminal problem. Also, by recording the directions from which the CTS frames are recently overheard and then blocking communication in those directions, a node is allowed to transmit in directions that will not collide with other data transmissions, in addition to relieving the hidden terminal problem. However it relied on extra location tracking support. Reference [11] applies directional antenna to the IEEE 802.11 MAC DCF specification by sending the RTS and ACK frames directionally and succeeds in achieving better network throughput. However, a directional RTS frame requires knowledge of the direction of the intended receiver which is known as the location tracking problem.

To avoid this problem [8] utilized omnidirectional RTS/CTS frames in order to identify the directions of communicating nodes before sending directional data. However it did not have the benefit of reserving the channel directionally. A detailed comparison of omnidirectional and directional MAC protocols for ad hoc networks is detailed in [14].

4.2 A Smart Antenna based MAC scheme

A fundamental problem in designing a cross-layer MAC scheme with smart antennas for a MANET is to identify the directions of the nodes in the MANET, or the location tracking problem. One solution is to equip every node with GPS support and have nodes exchange location information periodically [11]. It is noted that direction finding information must be broadcast and some part of the control exchange must use omnidirectional packets.

In the scheme considered in this research, the MMSE (Minimum Mean Square Error) adaptive antenna algorithm is used to track the positions of the nodes. The MMSE algorithm achieves adaptive beamforming through the use of *training sequences* unique to each node

Each receiver sends out an omnidirectional *Ready To Receive* (RTR) packet. The transmission of these packets is timed to avoid collision. This is done by generating a uniformly distributed random number between 1 and N, N being the number of receiver nodes in the system. A receiver node waits this random amount of time before transmitting an RTR. These packets contain the receiver MAC address and a training sequence for that receiver.

Transmitters would need to maintain a table of receiver address-training sequence pairs in order to know which sequence to look for when it has data intended for a particular receiver.

On receiving an RTR packet from its intended receiver, a transmitter node, uses the training sequence obtained from the RTR packet to form a beam towards that receiver. The transmitter then sends directional data to the receiver such that it does not interfere with other ongoing transmissions if the angular separation of the nodes is greater than the beam width.

This is illustrated in Figure 50.

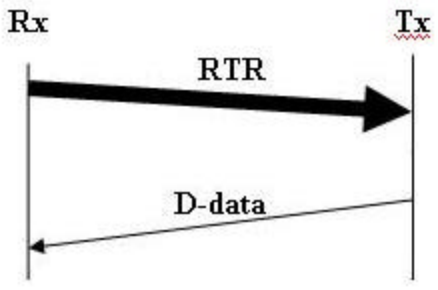

Figure 50 Timing Diagram for the Considered MAC Scheme using Smart Antennas

Hence, point to point data-transfer between different transmitter-receiver pairs takes place simultaneously without mutual interference.

To summarize, in this scheme, carrier sensing is not used, instead omni-directional RTR packets are transmitted (in a receiver initiated approach). This scheme achieves collision avoidance without using carrier sensing through the broadcasting of RTR packets and then employing spatial filtering. The transmission of the RTR packets is timed to avoid collision. Spatial multiple access is achieved by transmitting a training sequence in the RTR packet. A transmitting node would use this training sequence to form a directional beam towards an intended receiver using the MMSE beamforming algorithm. The receiver address is also contained in the RTR packet, so that a transmitter which hears that RTR would know whether it wants to transmit to that receiver or not. Transmission and reception are time division duplexed.

If 2 transmitters want to transmit to the same receiver they both transmit directional Ready to Send (D-RTS) packets to that receiver. The receiver would receive these 2 D-RTS packets and send a directional Clear to Send (D-CTS) packet to each of the transmitting nodes. The D-RTS packets would contain the training sequences for the transmitter nodes. Using receiver beamforming the training sequence of a transmitter node enables the beamforming weights at the receiver to converge and form a main lobe in the desired direction. Simultaneous beams in separate directions are formed by applying different steering vectors, one for each beam to the signals at all antenna elements, using programmable DSPs. This is detailed in [2]. The benefit of simultaneous reception/transmission of D-RTS/D-CTS packets is seen at bottleneck nodes in a network using multi-hop message transfer [2]. This scheme is illustrated in Figure 51.

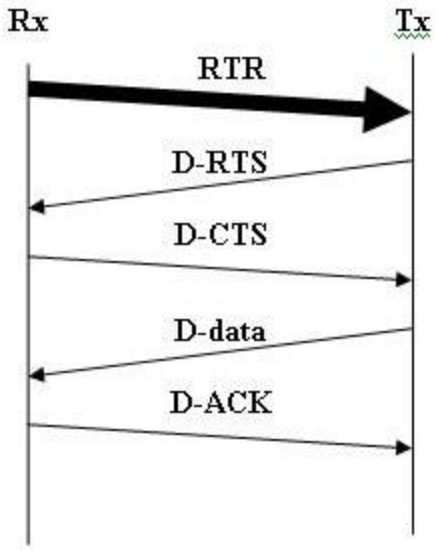

Figure 51 Timing Diagram for a MAC Scheme using Smart Antennas

However, in the system under consideration, simultaneous multi-hop message transfer is not being modeled. Point to point communication is assumed between transmitter receiver pairs. Hence, communication between nodes is assumed to take place on a one to one basis and the forwarding/ routing of packets between nodes is ignored. Hence, modeling the system with pairs of transmitters and receiver communicating is a reasonable simplification as this would be the case in a real system. Hence for the purposes of this study, it is assumed that the transmitter-receiver pairs are fixed and that only one transmitter would want to communicate with a receiver. D-RTS/D-CTS is ignored and communication between fixed pairs of nodes is compared for a MAC scheme using directional and omni-directional antennas.

4.3 Physical Layer Description

At the Physical layer, the MMSE smart antenna algorithm is implemented on each receiver, an exponential path loss is assumed, and AWGN is generated to simulate the noise introduced in each receiver chain. The following gives a detailed description of the underlying operations being modeled in the Physical layer simulation.

4.3.1 Modulation

Each node in the modeled system uses BPSK modulation generated on a symbol by symbol basis implemented at baseband using a complex envelope representation where a symbol is given by

$$b(t) = \sqrt{P_i}\, b_k\, e^{j j_k}\ \ p(t) \tag{23}$$

where b_k is the original baseband symbol k, j_k is the phase of symbol k, and P_i is the transmit power of node *I*, *p(t)* is the square pulse of T second duration.

4.3.2 Channel

Additive white Gaussian noise introduced at the receiver. The path loss between two points is given by (10)

$$PL = 10n\log_{10}\left(\frac{d}{d_0}\right) + (P_0) \tag{24}$$

where n is the path loss exponent, d_0 is reference distance, and (P_0) is the path loss at the reference distance.

4.3.3 Adaptive Antenna Algorithm: MMSE

In this system, the Minimum Mean Square Error (MMSE) algorithm is also implemented for the error calculation. In MMSE, the algorithm attempts to make the received signal match a desired signal. In general, this can be expressed as solving for the weights that minimize the function given by (25)

$$J(\mathbf{w}_k) = E[y_k - d_k] \tag{25}$$

where d_k is the desired symbol and $y_k = \mathrm{w}^{\dagger} r_k$. Since knowledge of d_k before reception implies that no information transfer is occurring, exact knowledge of d_k is typically only known for a small percentage of a total transmitted sequence; the sequence of known symbols is referred to as a pilot or training sequence. When not operating with a pilot sequence, the MMSE algorithm operates with symbol estimates based on y_k which for BPSK is given by

$$d_k = \mathrm{sgn}(y_k) \tag{26}$$

This is known as decision directed approach. The exact solution of the minimization function again requires a matrix inversion which is generally extremely cycle intensive. Because of this fact, LMS is used again. For MMSE the iterative LMS solution for the weights takes on the form in (27)

$$\mathbf{w}_{k+1} = \mathbf{w}_k - m\left(y_k - d_k\right)y_k^*\mathbf{x}_k \tag{27}$$

Where m is the step size. Note that larger values for m result in faster convergence, but larger final error.

4.3.3.1 Comments on MMSE

While MMSE requires side knowledge of the desired signal and results in a lower data rate, it performs significantly better than the CMA when the desired signal weaker than the interferer. The use of the training sequence, to which the jammer's sequence is presumably uncorrelated, ensures convergence to the proper signal. Reconsidering the system in Figure 9, Figure 52 shows the resulting beam patterns. Note that now the pattern converges to the desired signal, located at the red 'o' in both cases even though on the right the desired signal is weaker than the interferer. Also note that when the interferer is stronger (on the right), a much sharper null is steered towards the interferer.

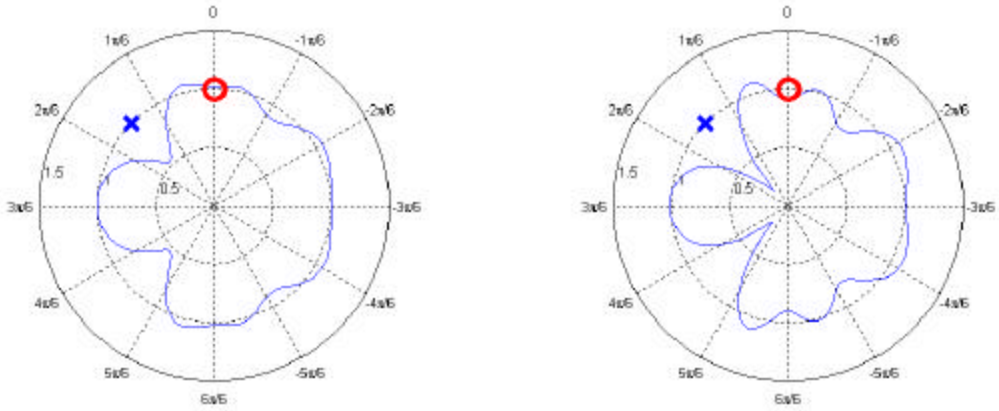

Figure 52 Effect of Received Power on Effect of MMSE for 4 Antenna Elements

4.4 System Implementation

4.4.1 System Overview

As shown in Figure 53, the system consists of 2 types of nodes: transmitters and receivers. Each of these nodes may be fixed or mobile. The following gives a description of each of these nodes.

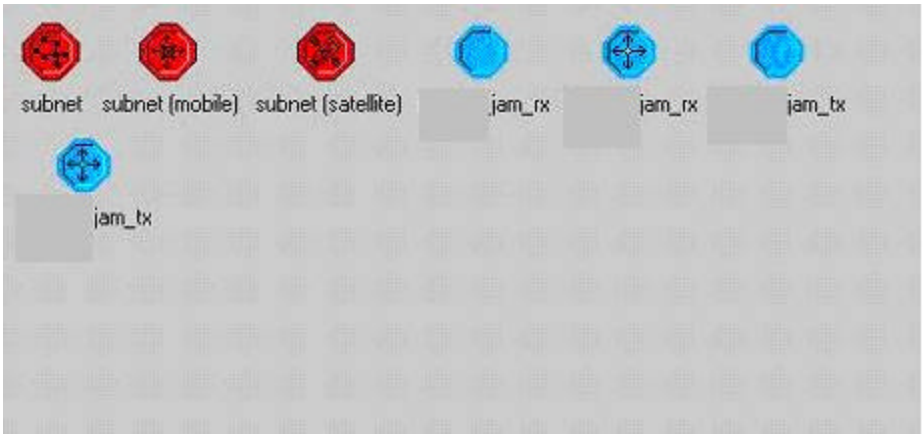

Figure 53 System Palette

4.4.1.1 Transmitter

The transmitter node model shown in Figure 54 consists of the following:

- A source module (*src*) which uses OPNET's *simple source* process model to generate packets. The probability distribution functions of the packet size and packet inter-arrival time used within the simple source may be set as desired from those available within OPNET.
- A queue module (*MAC A Tx*), the procedural behavior of this queue, custom written to implement the MAC link layer protocol is detailed in a later section.
- A radio transmitter module (*tx_tx*) which provides an interface to transmit packets on the radio channel.
- A radio receiver module (*tx_rx*) which provides an interface to receive RTR packets from the receiver.

Packet flow:
After generation, packets move through a packet stream to the *MAC A Tx* link layer module before being sent to the radio transmitter. Control packets received move from the radio receiver to the *MAC A Tx* module.

To enhance simulation flexibility, the transmitter node's power attribute is promoted so it can be set at run-time by the user.

Some of the specific parameters of the transmitter node are listed in Table 14.

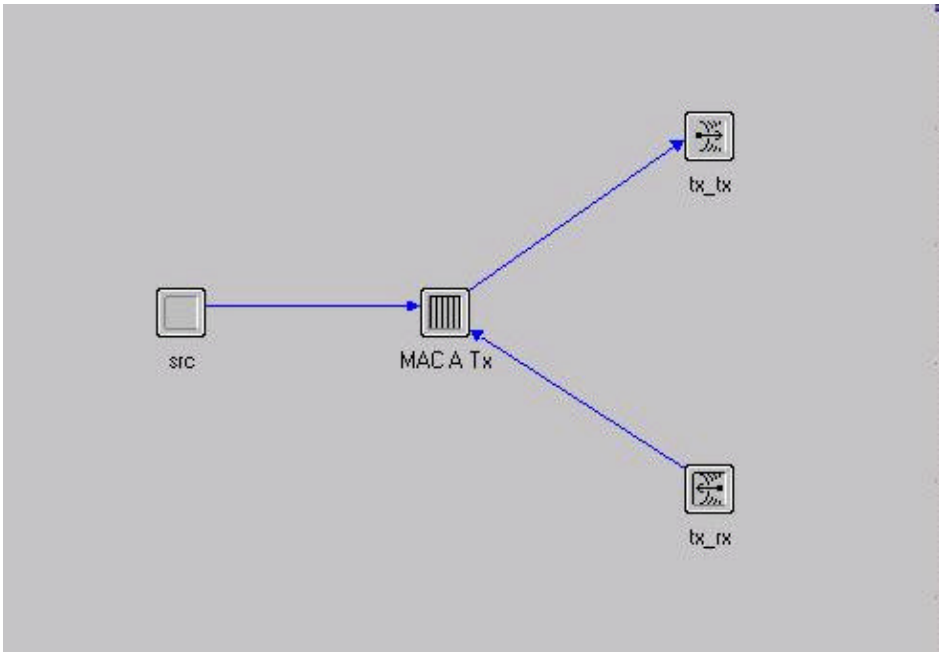

Figure 54 Transmitter Node Model

Table 14 Transmitter Attributes

Attribute	Type/ Value
Modulation	BPSK
Power	Promoted to be set by user at run time
Data Rate	100 Kbps
Bandwidth	100 KHz
Min. Frequency	1.8 GHz

4.4.1.2 Receiver

The receiver node model shown in Figure 55 consists of the following:

- A source module (*src*) which generates the control packets.
- A radio receiver module (*rx_rx*) which provides an interface to receive data packets sent by the transmitter.
- A radio transmitter module (*rx_tx*) which provides an interface to send RTR control to the transmitter.
- A queue module (*MAC A Rx*), the procedural behavior of this queue, custom written to implement the MAC link layer protocol is detailed in a later section.
- A sink module (*sink*) which sinks data packets received.

Packet flow:

After generation, the packets move through a packet stream to the *MAC A Rx* link layer module before being sent to the radio transmitter. Data packets received move from the radio receiver to the *MAC A Rx* module from where they are sent to the sink.

Table 15 lists additional node level attributes of the receiver node.

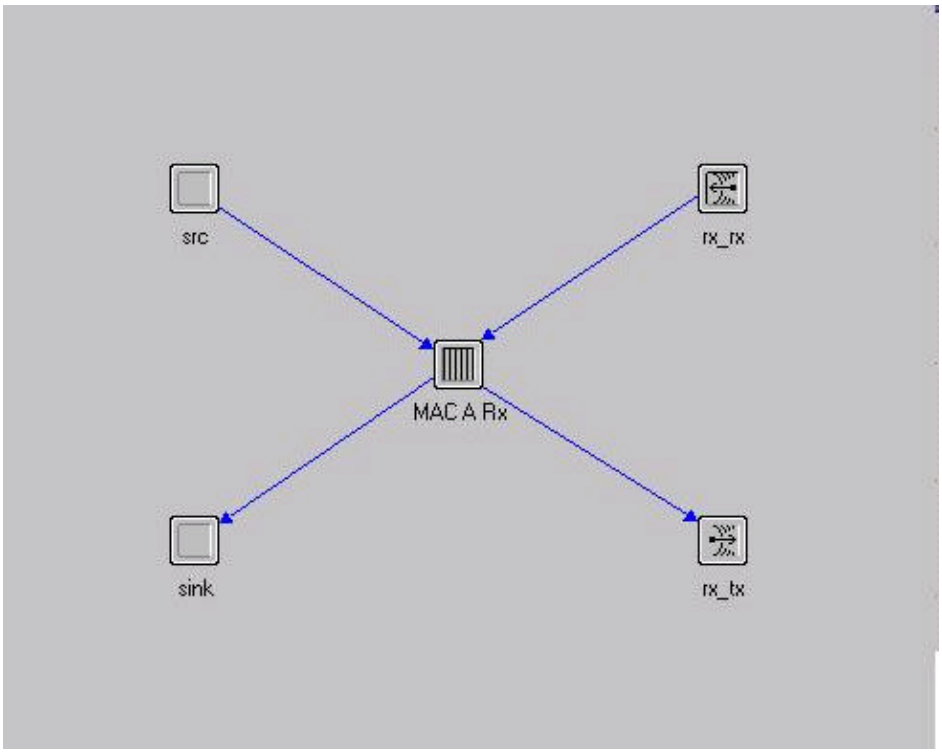

Figure 55 Receiver Node Model

Table 15 Receiver Attributes

Attribute	Type/ Value
Modulation	BPSK
Data Rate	100 Kbps
Bandwidth	100 KHz
Min. Frequency	1.8 GHz

4.4.2 Process Model for the Medium Access Control Scheme

4.4.2.1 Transmitter Process Model

As shown in Figure 56 the FSM for the transmitter process model consists of the following states:

- *INIT*: The first time a process model is invoked, it begins in the initialization state. This state initializes the variables used in the process and registers the statistics to be collected.

- *IDLE*: By default the process is always in the idle state.

- *RCV_RTR*: An *RTR_ARRIVAL* condition which is an interrupt triggered by an RTR packet arriving from the receiver causes the process to transition from the *IDLE* to the *RCR_RTR* state. If the source address of the packet is the same as the intended receiver for that transmitter the RTR is accepted and the training sequence of that receiver recorded. On execution of all the ode in this state, control returns to the *IDLE* state.

- *SEND_D_DATA*: If an RTR packet has been received from the intended receiver for a particular transmitter and a packet arrives from the network layer, the *DATA_ARRIVAL* condition is satisfied. The process on transitioning to the SEND_D_DATA state stamps the training sequence in a packet field which is then sent to the radio transmitter module. In the transmitter antenna gain calculation pipeline stage of the radio transmitter module the training sequence is extracted from the packet and the packet is transmitted over a directional beam to the intended receiver. For the procedural description of the implementation of the adaptive antenna algorithm refer to section 2.7.

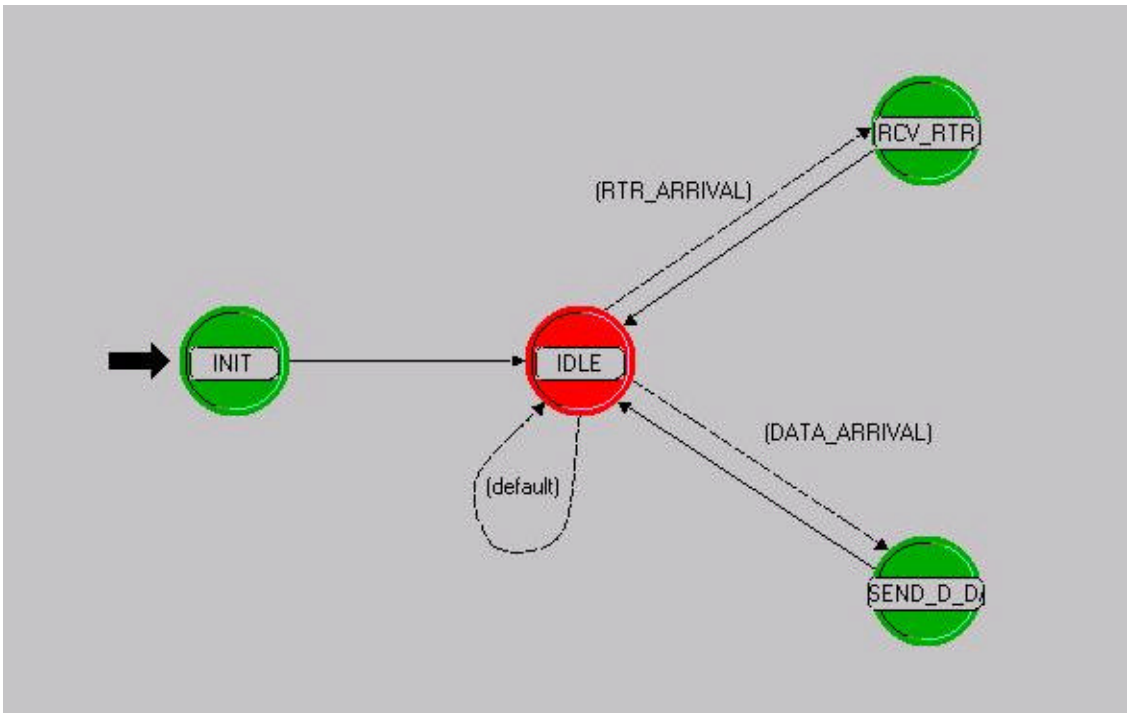

Figure 56 Transmitter Process Model

4.4.2.2 Receiver Process Model

As shown in Figure 57, the FSM for the receiver process model consists of the following states:

- *INIT*: The first time a process model is invoked, it begins in the initialization state. This state initializes the variables used in the process and registers the statistics to be collected.

- *IDLE*: By default the process is always in the idle state.

- *RTR*: A packet arriving from the source triggers the *READY_TO_RCV* condition and causes the process to transition from the *IDLE* to the R*TR* state. The training sequence for that receiver address is stamped onto the packet which is then broadcasted in the network.

- *RCVD_DATA*: Packets arriving from the transmitter trigger the DATA_ARRIVAL condition. These packets are then accepted and sent to the sink.

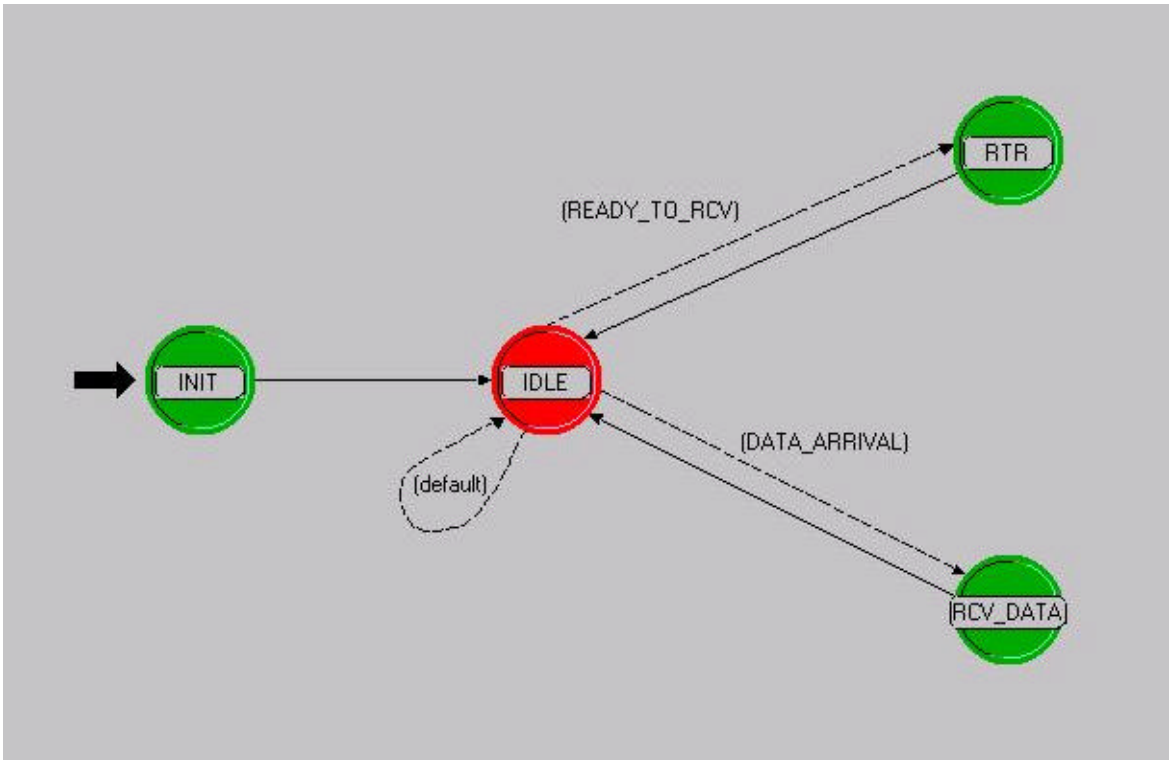

Figure 57 Receiver Process Model

4.5 Results

4.5.1 Preliminary Results: Comparison of Omnidirectional and Directional Antennas

The system consists of a 2 transmitter–receiver pairs, Tx1-Rx1 and Tx2-Rx2, as shown in Figure 58, the system was simulated for 3 minutes. The power of transmitter, Tx1, was set to 1 W while the power of transmitter Tx2, was set to 2 W.

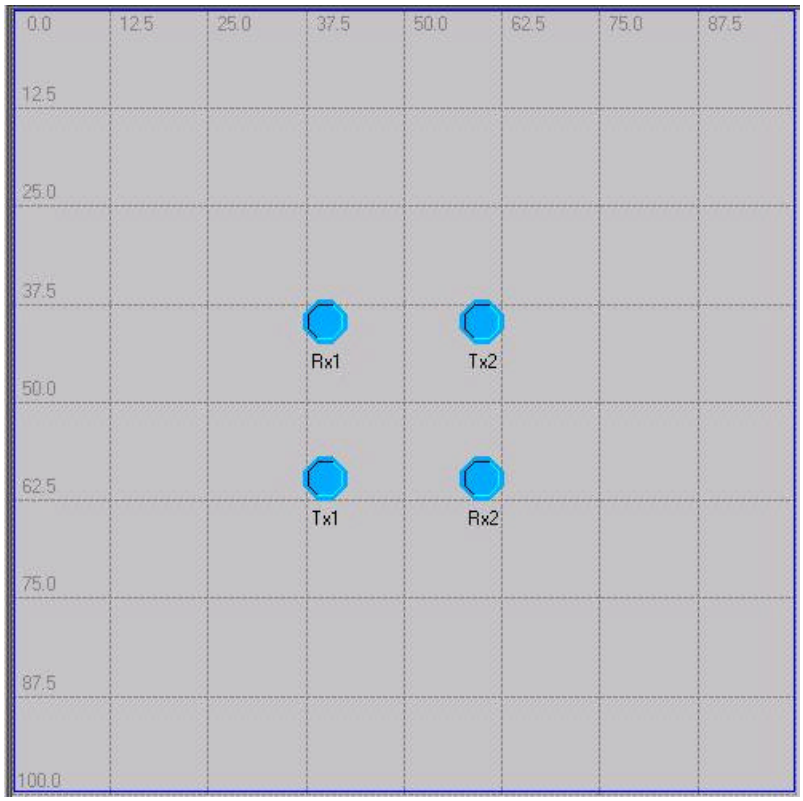

Figure 58 Network Scenario 1

Statistics looked at were the receiver BER, and the transmitter and receiver throughout for node Tx1 and Rx1.

Figure 59 and Figure 60 show the BER at receiver 1 with the MAC protocol using omnidirectional antennas and a directional antenna array respectively. With the MAC protocol using adaptive beamforming, both transmitter send data over beams directed at their intended receiver. These transmissions are spatially separated and do no cause interference to each other, and we observe a BER of zero. With omnidirectional antenna patterns, the transmitter packets collide and we observe a BER as in Figure 59.

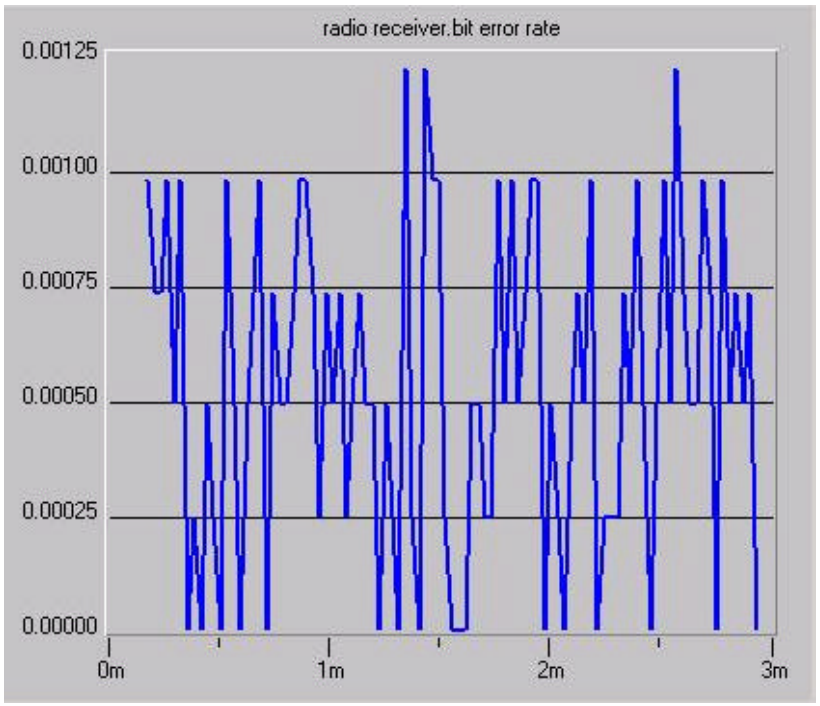

Figure 59 Receiver BER: MAC with Omnidirectional Antennas

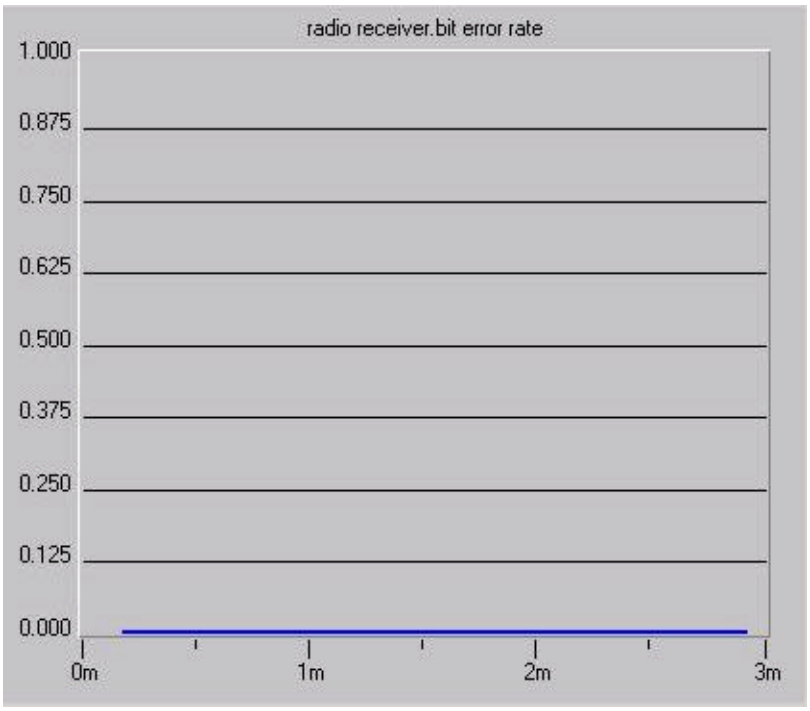

Figure 60 Receiver BER: MAC with Adaptive Beamforming

On running the same scenario with a 802.11 CAMA/CA based MAC wherein the nodes are quipped with omnidirectional antennas an average receiver BER as shown in Figure 61 is obtained.

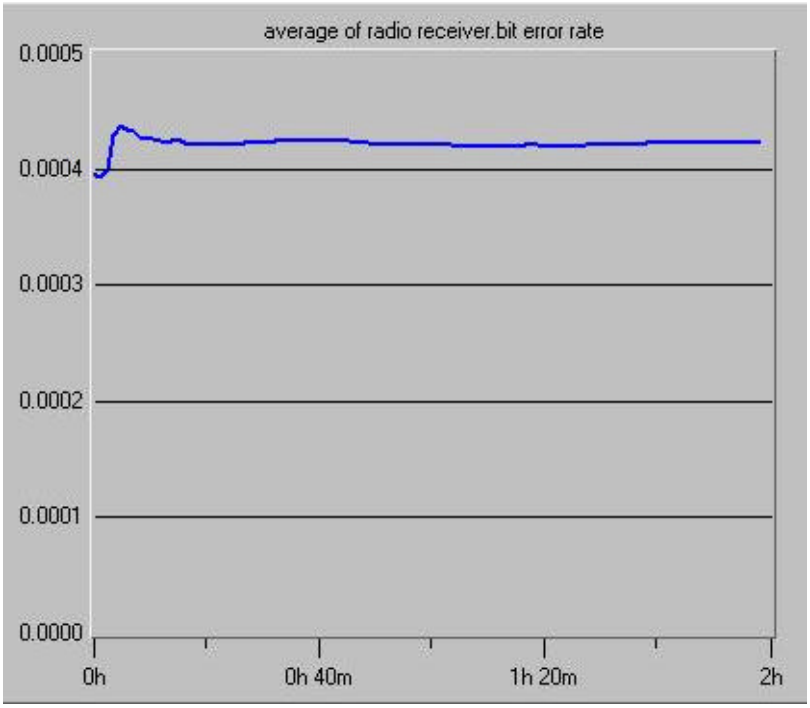

Figure 61 Receiver Average BER: 802.11 based MAC with Omnidirectional Antennas

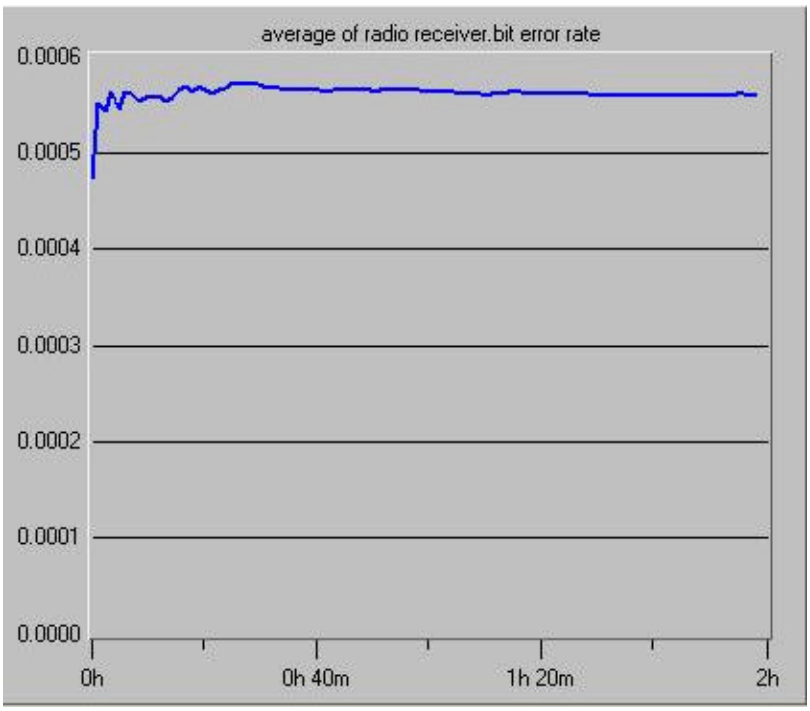

Figure 62 Receiver Average BER: Smart Antenna based MAC with Omnidirectional Antennas

As can be seen, the 802.11 based MAC scheme with omnidirectional antennas performs better than the MAC scheme studied in this research *with omnidirectional antennas* as can be seen from the receiver BERs in Figure 61 and Figure 62. This is as expected as the MAC scheme studied and implemented in meant for use with smart antennas. This scheme when used in a system along with adaptive beamforming performs better than the 802.11 MAC scheme based on CSMA/CA as seen from Figure 60.

Figure 63 and Figure 64 show the transmitter and receiver throughputs with the MAC protocol using omnidirectional antennas and a directional antennas respectively. The transmitter sends packets at a constant rate of 1 packet per second. With the omnidirectional antennas, transmissions from the 2 transmitter collide, these packets get corrupted and are not accepted by the receiver. The system throughput degrades as shown in Figure 63. With the MAC protocol using smart antennas, the 2 transmissions are spatially separated. Hence packets from these 2 simultaneous transmissions do not collide and the receiver throughput is the same as the transmitter throughput.

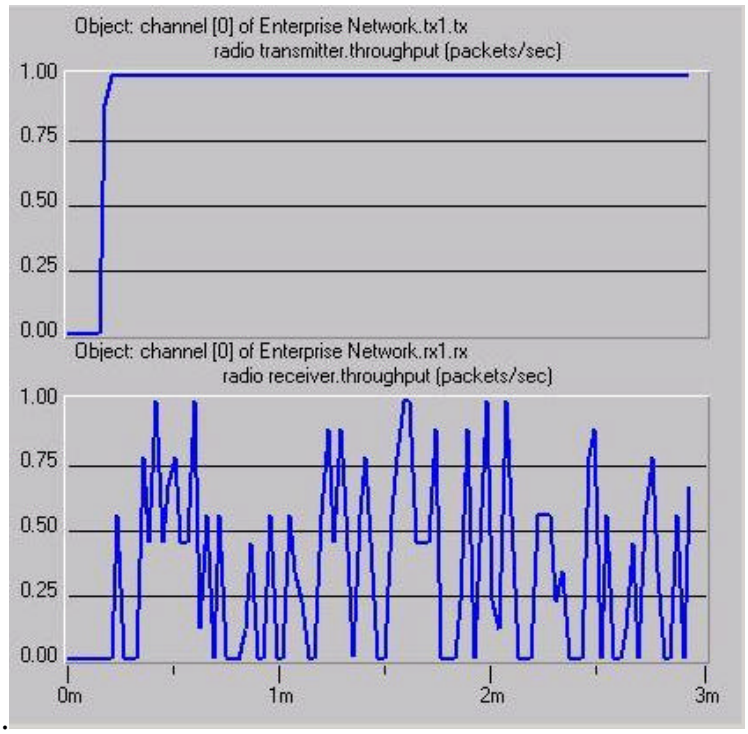

Figure 63 Transmitter and Receiver Throughput: Smart Antenna MAC with Omnidirectional Antennas

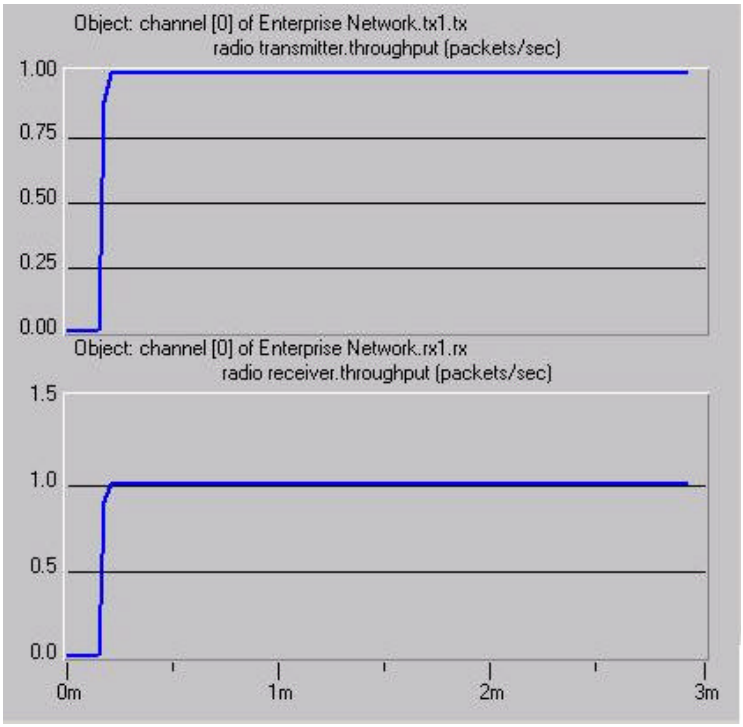

Figure 64 Transmitter and Receiver Throughput: MAC with Adaptive Transmitter Beamforming

4.5.2 Effect of Training Sequence Length on Performance

The results shown above assume all the bits generated for a particular receiver are used as the training sequence by the MMSE beam forming algorithm. Figure 65 shows the receiver BER and throughput when 5% of the bits (5 bits in a 100 bit packet) are absolutely known and the rest of the training sequence uses decision decoded bits. The MMSE algorithm now takes a longer time to converge as seen from the peaks in the BER curve. This emphasizes the importance of dynamic simulations.

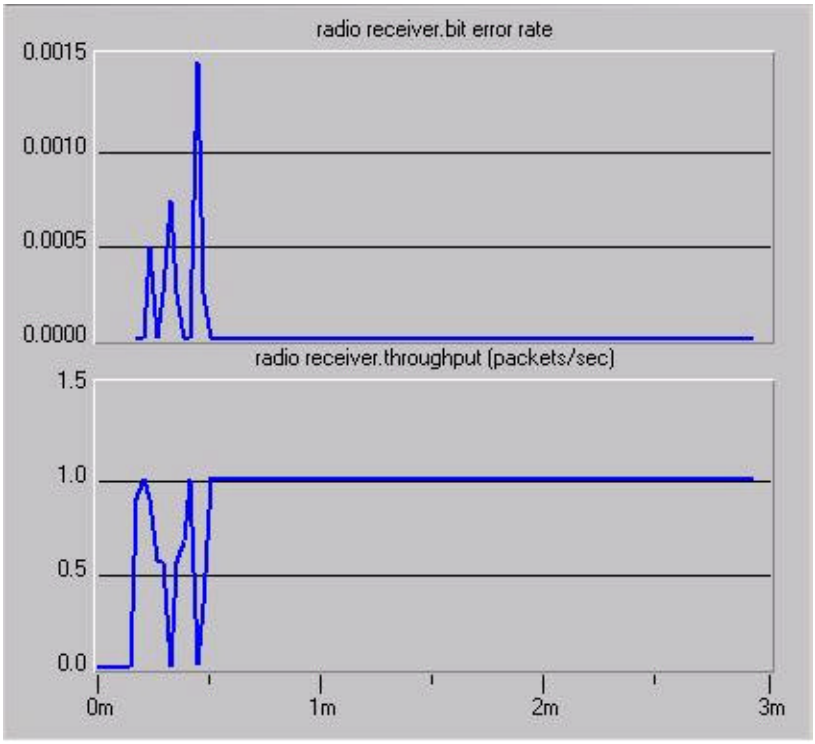

Figure 65 Receiver BER and Throughput with Decision Directed Decoding

4.5.3 Transmitter Beamforming Vs. Transmitter and Receiver Beamforming

The MAC scheme uses either transmitter beamforming or receiver beamforming. The use of both would increase processing overhead while not contributing to throughput improvement except in specific scenarios, where performance would be restricted due to the antenna geometry as shown in Figure 66.

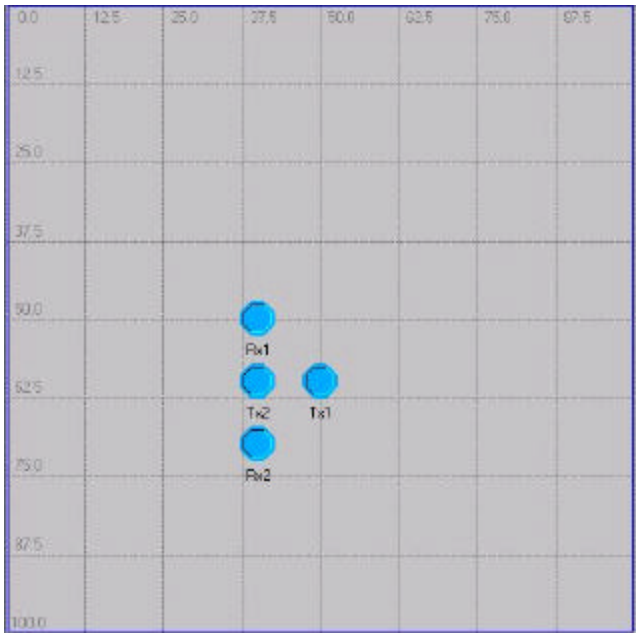

Figure 66 Network Scenario 2

The scenario shown in Figure 66 is simulated to compare the effect of only transmitter beamforming as opposed to transmitter and receiver beamforming. In this scenario, Tx2 while, beaming towards Rx2, will also have the same gain in the direction of Rx1, due to the vertical symmetry of the antenna array. Hence, Rx1 will see interference (Figure 67). This would be cancelled (Figure 68) if Rx1 also employed beamforming in which case it would steer a null towards Tx2 while beaming towards Tx1. This interference cancellation comes at the expense of the exchange of higher processing required at the receiver.

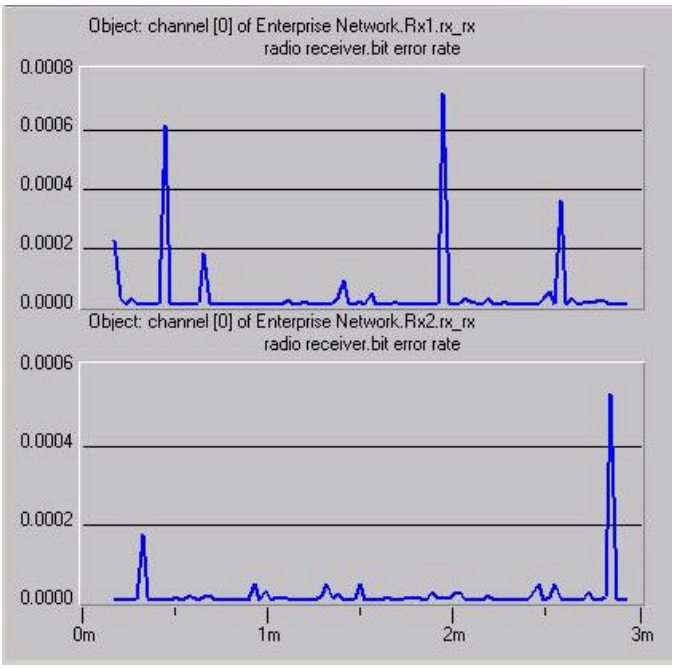

Figure 67 Receiver BERs with only Transmitter Beamforming

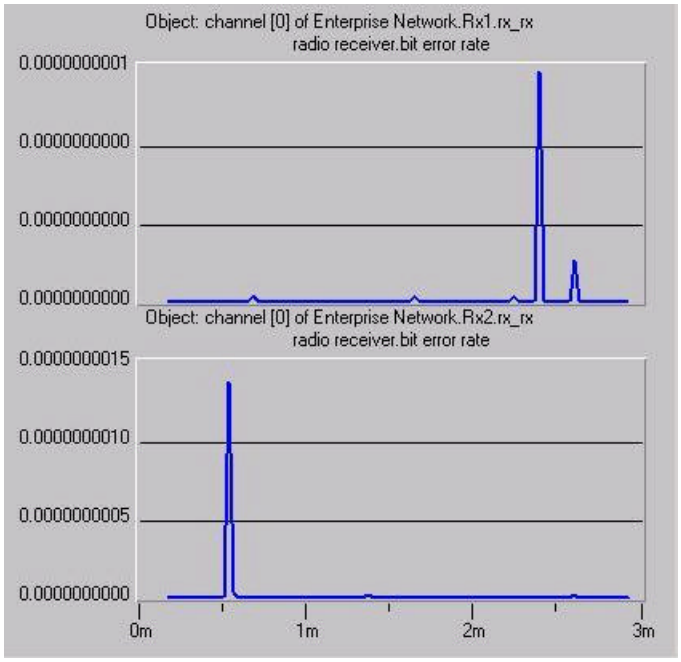

Figure 68 Receiver BERs with Transmitter and Receiver Beamforming

However, in an actual system, where nodes are moving about in a random manner it is felt that such a scenario where nodes would be exactly symmetrical such that only transmitter beamforming would affect system performance due to the symmetry of the antenna array would not occur very often. To that effect the scenario shown in Figure 69 where nodes rx1 and tx1 follow random trajectories is simulated.

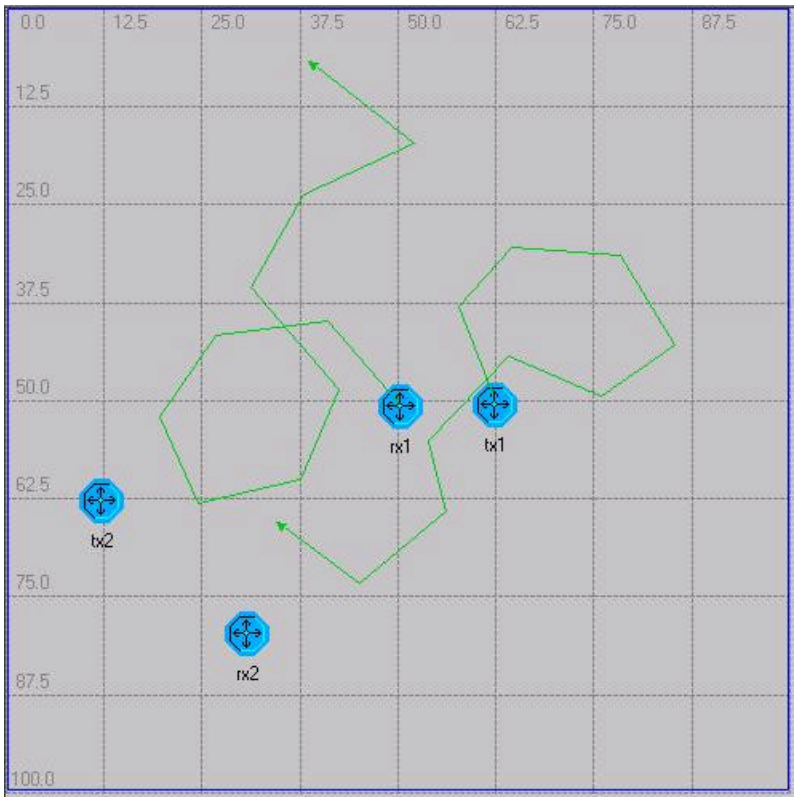

Figure 69 Network Scenario 3

From Figure 70 and Figure 71 it is seen that using transmitter beamforming only does not hinder system performance substantially. While transmitter and receiver beamforming achieves marginally better performance, it contributes to increase in processing power required.

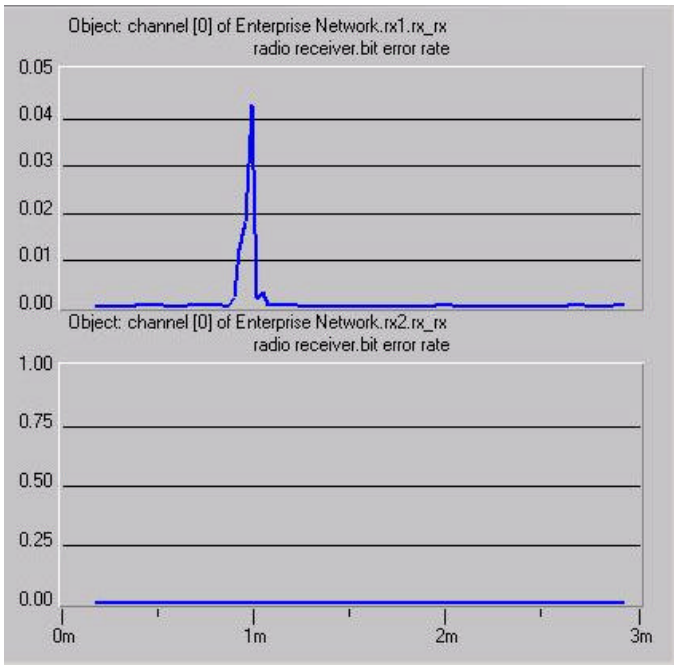

Figure 70 Receiver BERs with only Transmitter Beamforming

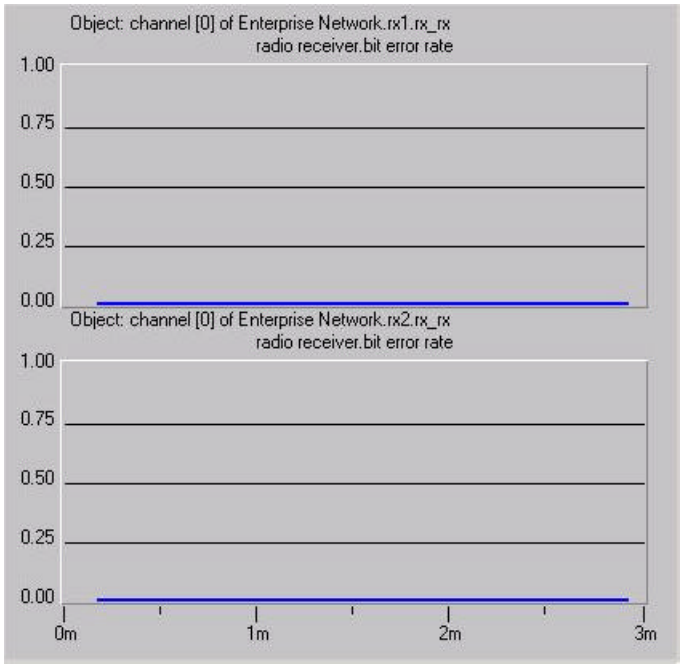

Figure 71 Receiver BERs with Transmitter and Receiver Beamforming

4.5.4 Network Capacity as a Measure of Performance

To fully appreciate the effect of spatial filtering on medium access control, system capacity vs. throughput is examined. The number of nodes in the system is increased from 2 to 38 as shown in Figure 72 in steps of 2 and the corresponding throughput is plotted for the omnidirectional MAC and the smart antenna based MAC scheme. It is observed that when using smart antennas the throughout stays constant at about 100% up to 18 nodes. When using omnidirectional antennas the throughout drops sharply when the number of nodes in the system is increased from 6 to 8. With 10 nodes in the system the throughout with the MAC using omnidirectional antennas is only 1.98% while the throughput with the smart antenna based MAC is 100%. Increasing the number of nodes further drops the throughput to zero in the omnidirectional case while for the smart antenna based MAC there is a negligible drop to 99.8 %.

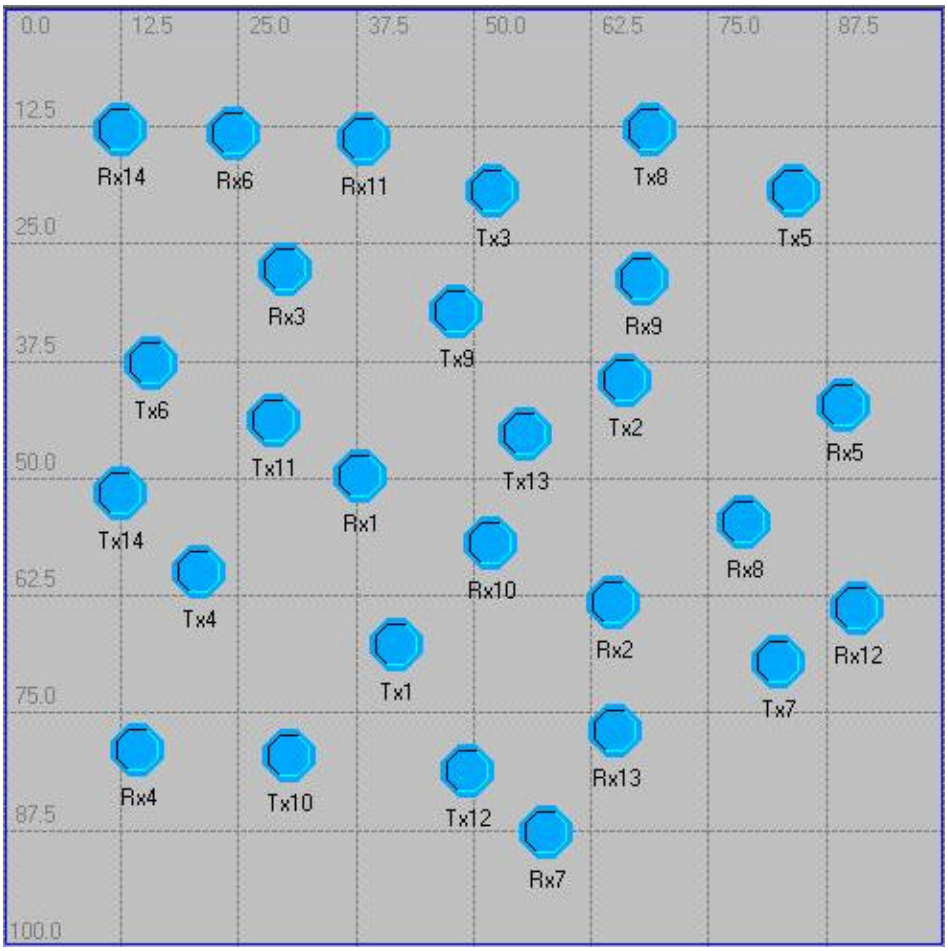

Figure 72 Network Scenario 4

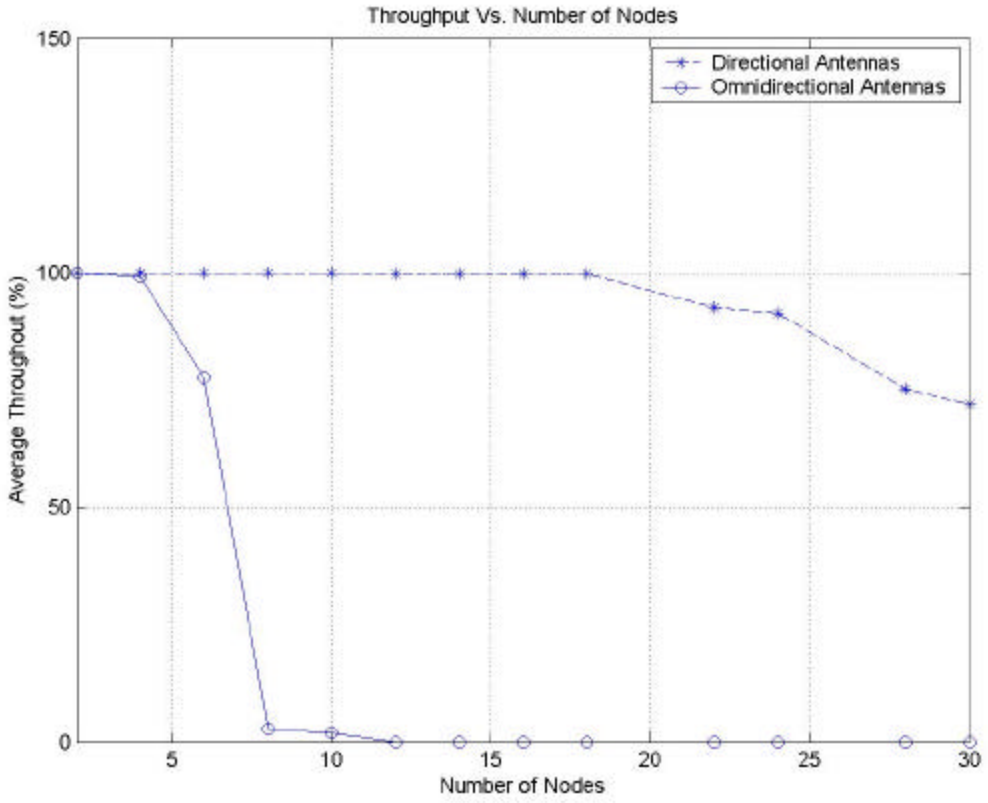

Figure 73 Effect of System Capacity on Throughput

Transmitter only beamforming is compared with transmitter and receiver beamforming in Figure 74. It is seen that adding receiver beamforming helps throughput as the number of nodes increases.

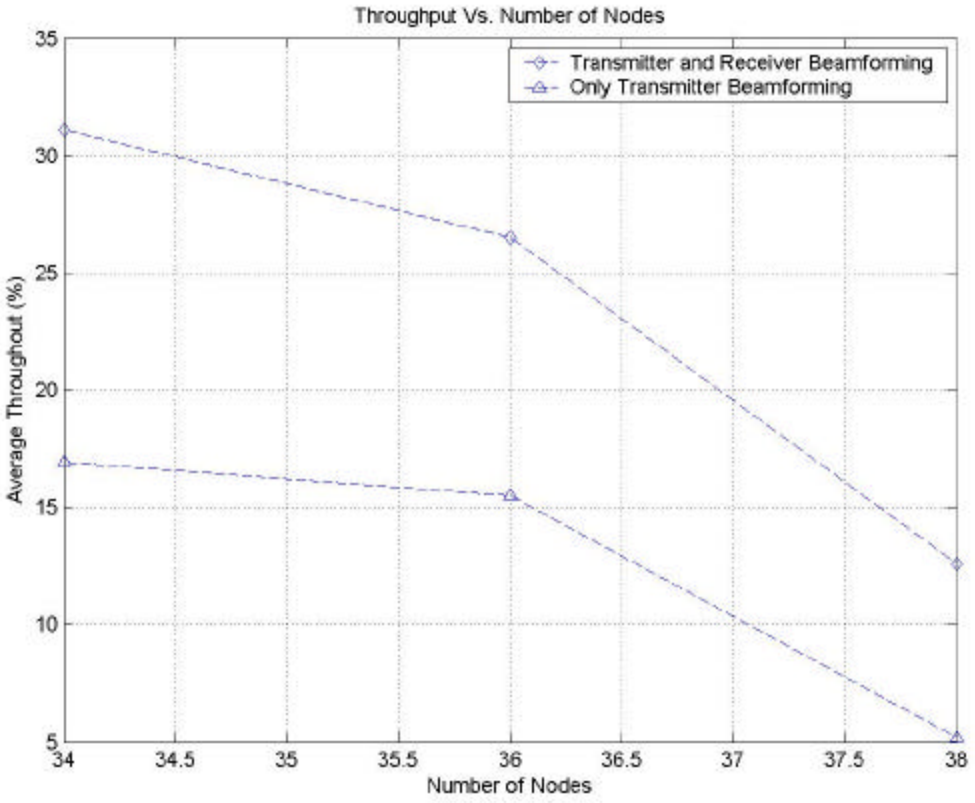

Figure 74 Effect of System Capacity on Throughput

4.6 Conclusions

The 802.11 based MAC scheme with omnidirectional antennas performs better than the MAC scheme studied in this research *with omnidirectional antennas* as can be seen from the receiver BERs in Figure 61 and Figure 62. This is as expected as the MAC scheme studied and implemented in meant for use with smart antennas. This scheme when used in a system along with adaptive beamforming performs better than the 802.11 MAC scheme based on CSMA/CA as seen from Figure 60. This shows the importance of cross-layer system design.

Using transmitter beamforming only does not hinder system performance substantially. While transmitter and receiver beamforming achieve marginally better performance, it contributes to increase in the processing power required.

The true benefit of this MAC scheme using smart antennas and the performance benefit of cross-layer design is seen from the increase in network capacity. It is seen that the performance of the smart antenna based MAC degrades gracefully with increase in the number of nodes in the system and not sharply as in the case of the MAC using omnidirectional antennas. For a throughput of 80% an increase of approximately 21 nodes in system capacity is observed.

5 A Power Control Scheme with Successive Interference Cancellation to satisfy Varying QoS Requirements for CDMA systems

5.1 Introduction

Wireless mobile ad hoc networks are collections of nodes communicating over a wireless channel. Since wireless power decays exponentially with distance and in the presence of obstructions, each node can only communicate directly with some nodes that lie in its vicinity. On the other hand, the traffic requirements of the nodes are taken to be arbitrary, and hence it is necessary that nodes cooperate to forward packets to their final destination. In [3] it is shown that the gains from allowing power control appear modest due to the multi-hop nature of MANETs. In multi-hop MANETs it is important that a node transmits to a distant node so that the packet moves closer to its destination, however, power control favors nodes that are close, since now the transmitter can use less power and create less interference and still be able to transmit its packet to its destination.

However, power control is a very important in the design of cellular CDMA systems which suffer from the near-far problem. Hence, in this chapter the role successive interference cancellation receivers at the physical layer play on power control which is a layer 2 radio resource management issue is studied. With the increasing need to differentiate classes of service among users, the deployment of applications that lead to mixed traffic QoS (Quality of Service) is a very important issue in wireless systems. In this section, a power control algorithm is studied, which drives the average received powers to those required, based on the QoS requirements of the individual users for a cellular CDMA system using SIC receivers.

5.1.1 Successive Interference Cancellation

Multi-user detection has been proposed as a means of improving the capacity of code-division multiple-access (CDMA) systems relative to the conventional matched filter receiver, which treats multiple-access interference as Gaussian noise. SIC (Successive Interference Cancellation) is a form of multi-user detection in which signals are detected in the order of perceived reliability. A specific signal is detected, regenerated and cancelled from the total received signal, which mitigates the interference seen by the remaining signals. Subsequently detected users will thereby experience reduced interference. Initially the strongest signal will be perceived as the most reliable, which may be detected in the presence of more interference, hence it will be detected first and cancelled from the composite received signal. Also, the strongest signal would contribute the most to the interference, and removing the effect of this signal would have the greatest benefit. In this manner individual signals are detected and subtracted successively from the composite received signal.

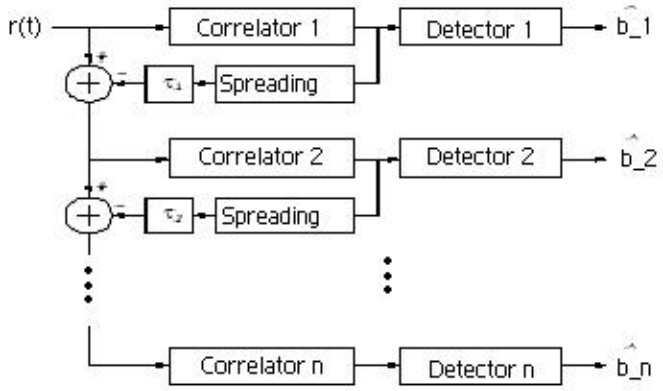

Figure 75 Block diagram of Successive Interference Cancellation

In order to implement ideal SIC detection, the receiver would need to have knowledge of all the spreading waveforms in the system. Typically, SIC finds its practical relevance to the uplink where the subsets of cancelled users are restricted to the same cell.

Clearly the first signal to be detected receives no benefit from this scheme as compared to conventional matched filter detection, while the signals lower in the cancellation order benefit more in that they see lesser and lesser interference. Hence when the received powers are roughly equal, the user detected first will obtain performance comparable to the conventional receiver while the last user detected may have a performance that approaches the single user bound. By varying the received powers one can obtain equal performance for all the users. In [5] , it is shown that a geometric distribution of received powers will provide equal bit-error rate (BER) for all users.

Hence, we see that one difficulty in the implementation of SIC is that for equal received signal strengths, the BER performance will vary significantly. Power control in current CDMA systems is designed to guarantee equal received power for all signals using a combination of inner-loop and outer-loop power control. To obtain equal BER for all signals with SIC, the received power profile must vary geometrically according to the cancellation order. Creating power control tables which have unequal thresholds to maintain such received power levels using inner-loop power control could be problematic. In this letter we show that FER-based outer-loop power control will provide sufficient control for SIC without using different power control tables and without causing degradation in SIC performance. Further, using outer-loop control to guarantee performance with SIC simply requires a coarse knowledge of the power limits of the mobile. That is, we must use some knowledge of the relative shadow fading and path loss experienced by each mobile to determine cancellation order. This information is available from the output of the receiver's acquisition and searcher circuits. We show that when the power limits are known incorrectly, it affects only the performance of the signal in question but does not cause catastrophic system failure.

5.1.2 Power Control in CDMA

Power control is an extremely important part of current CDMA systems. If multi-user detection is to play a part in commercial CDMA, its interaction with power control must be well understood. Compared to the massive amount of work done on multi-user detection, to date relatively little work has been done on the interaction between power control and multi-user detection [19] [20] [21] [22] [23] [24]. The combination of power control with the linear MMSE receiver was considered in [19] [20] [21] [22] [23] while [24] considered power control with the linear decorrelator. Further, the impact of power control with SIC on system interference levels was analyzed in [18] although the method of power control was not discussed. All of the approaches considered SINR-based measurements at the base station which were used to determine an optimal power setting for each mobile. This information was then communicated back to the mobile for power adjustment. Essentially, these methods can be considered "outer-loop" techniques since they are aimed at adjusting long-term power to provide an optimal received power distribution between users. This is as opposed to "inner-loop" control which makes fine adjustments to counter Rayleigh fading.

Reverse link power control is accomplished by transmitting power control bits on the forward link to the mobile unit, which it uses to increase or decrease its transmit power. The base station estimates the received E_b/I_o and compares this estimate with an internally saved threshold known as the E_b/I_o set-point. If the estimate is less than the threshold the mobile is requested to transmit more energy and if the estimate is more, the mobile is requested to transmit less energy. This type of E_b/I_o -based power control is often called *fast power control* since it operates at a rate of up to 800 Hz and is intended to combat short-term multipath fading. This is the *inner-loop* portion of the E_b/I_o-based power control. What happens with the E_b/I_o set-point is known as the *outer-loop.*

5.1.3 FER-based Outer-Loop Power Control

In FER-based outer-loop power control, after every frame the base station determines whether it made an error in decoding or not (via a cyclic-redundancy check or some other method). If the frame is in error, the base station raises its set-point by some value. If the frame is not in error, the base station lowers its set-point by some (usually smaller value). In this manner a target FER may be maintained. Over time, this raising and lowering of the set-point is intended to drive the *average* received powers to those required to obtain a desired FER.

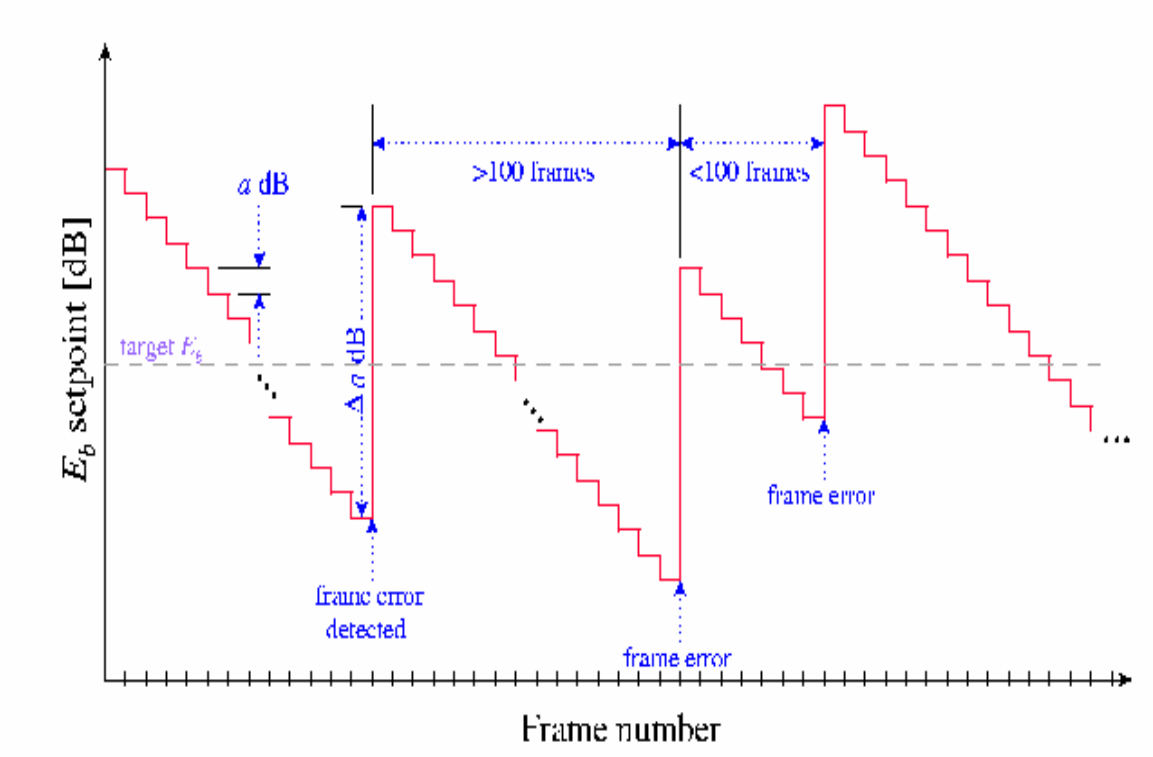

Figure 76 FER-based Outer-Loop Power Control [16]

5.2 System Model and Analysis

We consider a CDMA system with K=10 active transmitters communicating with a common base station in a single cell. The system is synchronous with random phases and random spreading codes. The received signal at the base station can be expressed as

$$r(t) = \sum_{k=1}^{K} \sqrt{P_k(t)} b_k(t - t_k) a_k(t - t_k) e^{jf_k(t)} + n(t) \tag{28}$$

where P_k is the received power from transmitter k, $b_k(t)$ and $a_k(t)$ are the data and spreading waveforms of the k^{th} signal respectively, t_k and f_k are the relative delays and phases of each signal with respect to an arbitrary reference and $n(t)$ represents thermal noise with variance s_n^2. $\sqrt{P_k(t)} e^{jf_k(t)}$ represents the complex distortion caused by the wireless channel. We assume no attenuation i.e. $\left|e^{jf_k(t)}\right|$=1, with f_k uniformly distributed between 0 and $2p$.

An SIC receiver detects signals in succession by canceling each signal from the aggregate received signal after detection. We represent the signal used to detect the k^{th} data signal as

$$r_k(t) = r(t) - \sum_{i=1}^{k-1} \hat{s}_i(t - t_i) \tag{29}$$

where $s_i(t) = \sqrt{P_i(t)} a_i(t) e^{jf_i(t)}$ and $\hat{s}_i(t)$ are the received signal and the estimate of the received signal from mobile *i.* We estimate the received signal as

$$\hat{s}_k(t) = \sum_{m=-\infty}^{\infty} z_{k,m} P_T(t - mT) a_k(t) \tag{30}$$

where $P_T(t)$ is a unit pulse defined over $[0, T)$, $z_{k,m}$ is the projection of the received signal onto the spreading code of signal k after cancellation of signal *k-1* during the m^{th} symbol interval, *i.e.*,

$$z_{k,m} = \frac{1}{T} \int_{(m-1)T + t_k}^{mT + t_k} r_k(t) a_k^*(t - t_k) dt \tag{31}$$

T is the symbol duration and * represents the complex conjugate. Re-writing $z_{k,m}$ as,

$$z_{k,m} = \frac{1}{T} \left[\int_{(m-1)T + t_k}^{mT + t_k} \left(\sum s_i(t - \tau_i) \right) a_k^*(t - \tau_k) \, dt - \right. \tag{32}$$

$$\left. \int_{(m-1)T + t_k}^{mT + t_k} \left(\sum_{i=1}^{k-1} \hat{s}_i(t - \tau_i) \right) a_k^*(t - \tau_k) \, dt \right] + N_{k,m} \tag{33}$$

$$= \sqrt{P_k} b_{k,m} e^{jf_k} + \sum_{i=1}^{k-1} \tilde{I}_{k,i,m} + \sum_{i=k+1}^{K} I_{k,i,m} + N_{k,m} \tag{34}$$

where

$$N_{k,m} = \int_{(m-1)T+t_k}^{mT+t_k} n(t)a_k^*(t-t_k)dt \tag{35}$$

and $I_{k\,i,m}$ is the cross-correlation between signal k and signal I during k's m^{th} symbol interval, and $\tilde{I}_{k,i,m}$ is the residual cross-correlation between spreading codes after cancellation. If a Gaussian approximation is made on the decision statistic as is common in CDMA system analysis, we can define the SINR for signal k as

$$\Gamma_k = \frac{\mathrm{E}\left\{\mathrm{Re}[z_{k,m}]\middle|b_{k,m}\right\}}{\mathrm{var}\left\{\mathrm{Re}[z_{k,m}]\middle|b_{k,m}\right\}} \tag{36}$$

Further, it is shown in [2] that the given receiver can provide equal SINR (*i.e.* $\Gamma_k = \Gamma \forall k$) and thus equal FER if the flowing power profile is used

$$p_{opt} = \left(\frac{1}{\Gamma}\mathbf{I} - \frac{2}{N}\mathbf{X}\right)^{-1} bs^2 \text{ (for synchronous random phases)} \tag{37}$$

where $p = [p_1, p_2, \ldots, p_K]^T$, $\mathbf{I}$ is a K $\times$ K identity matrix, b is a K $\times$ 1 vector with $(1+2/N)^{i-1}$ as the i^{th} element, $\mathbf{X}$ is a K $\times$ 1 matrix with x_i^T as the i^{th} row and x_k is a K $\times$ 1 vector defined as

$$x_{k,m} = \begin{cases} 0 & m = 1 \\ (1+2/N)^{k-1} - (1+2/N)^{k-m} & m \le k \\ (1+2/N)^{k-1} & m > k \end{cases}$$

This power profile is shown in Figure 77 . We can see that each signal requires a different received power to obtain equal performance (BER) based on cancellation order.

Now, when we consider the effect of each user having a different SINR requirement (different target FER) specified by the QoS class of that user, the power profile required is

$$p_{opt} = \left(\mathbf{Q}^{-1} - \frac{2}{N}\mathbf{X}\right)^{-1} bs^2 \tag{38}$$

where $\mathbf{Q}$ is a diagonal matrix with the diagonal elements, q_{ii} = the SINR required by the i^{th} user.

5.3 Results

Simulations were run using the above SIC receiver design along with FER-based outer-loop power control as described above. It was assumed that the system was synchronous with random phases and random spreading codes. The outer-loop used a step size of 0.2 dB per frame error. The frame size considered was 50 bits. These simulations focused on a spreading gain of N= 64 with K=10, 20, 30 and 40 users in the system.

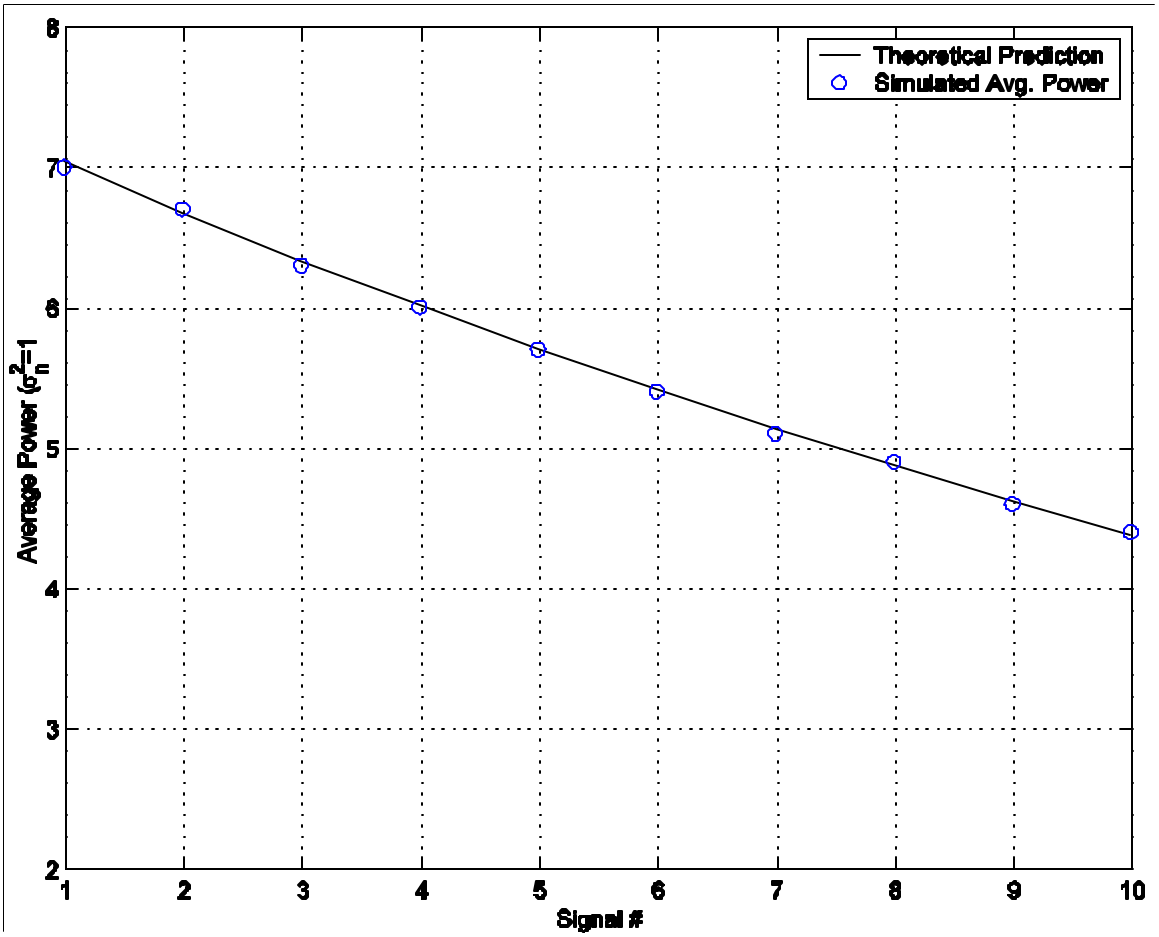

Figure 77 Simulated and Theoretical Power Profile for 10 Users

Figure 77 presents the average received power for each of the 10 signals in the system when no power limits are placed on the mobiles. The predicted normalized power is also plotted. Note that signals are ordered according to decreasing received power which is also the cancellation order. All signals obtained the target 12% frame error rate which corresponds to a BER of 0.02%. Thus, the outer-loop was stable and the prediction provided a good estimate of the required receive power. The outer-loop power control did indeed drive the mobile powers to the optimal power profile to achieve the target FER. The main assumption here is that all mobiles have sufficient power for the cancellation slot they are assigned. However, this requires intelligent assignment of cancellation order. If the first signal in the cancellation order corresponds to the signal furthest from the base station, not only will it result in more out-of-cell interference, but it

is possible that the signal would have insufficient power to achieve its target. A coarse knowledge is available from open-loop power control at call set-up as mentioned.

To determine the effect that such a limit (*i.e.* incorrect cancellation order) would have on the receiver performance, simulations in which some mobiles have a power limit were run. In the first example shown in Figure 78, signal 4 is limited to a normalized received power (noise variance =1) of 4.4. As we can see this prevents signal 4 from achieving its target FER since the power limit is below that necessary for the assigned cancellation slot. If that signal were assigned to the first cancellation slot, it would have achieved the target FER. Of further interest is the effect that the power-limited signal has on those signals which are cancelled before and after it. Signals cancelled before the power-limited signal *(# 1-3*) benefit in that they see less interference power (due to the lower received power of signal 4) and thus can transmit less power and still achieve the target FER. Signals which are cancelled after the power-limited signal are unaffected by the increased error rate and the lower received power. The effect of higher error rate (which increases interference) is offset by the lower received power (which decreases interference).

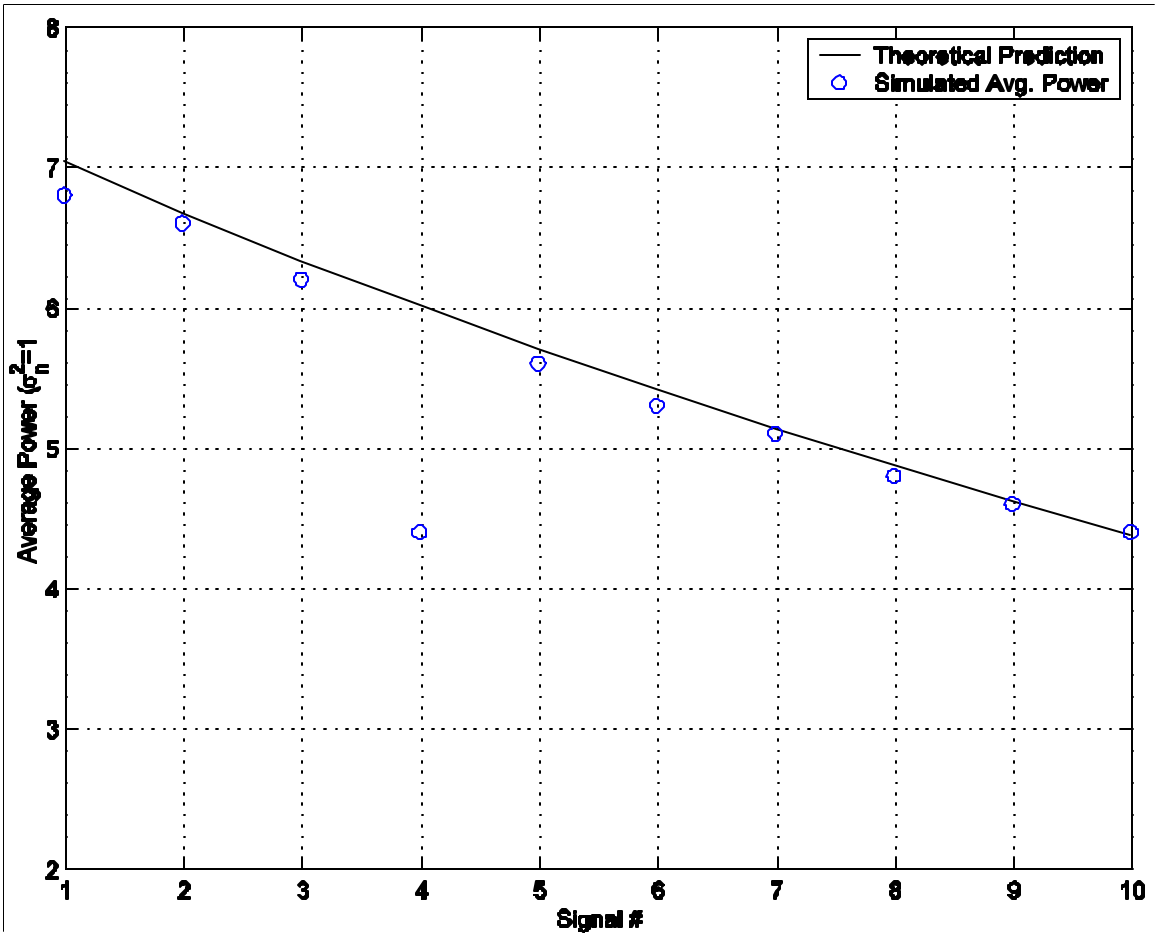

Figure 78 Simulated and Theoretical Power Profile for 10 Users with Signal 4 Power Limited

In Figure 79 signals 9 and 10 are limited to a normalized value of 3. These signals cannot reach the FER target but the others signals are not affected. Since these are the last two signals cancelled, the limits have the effect of reducing the necessary transmit power required for the signals cancelled before them.

A more serious potential problem is a limit on the first two signals cancelled as shown in Figure 80. Here it is anticipated that since the first two signals will not achieve their target FER, they will not be cancelled properly and will thus introduce more interference to those signals cancelled afterwards due to error propagation. However, since they are power limited, their effect is both increased (due to inaccurate cancellation) and decreased (due to reduced power). The net effect is that the signals cancelled afterward require essentially the same receive power as without the limits. The receiver with SIC and power control is thus fairly robust. The main problem then with inaccurate ordering is (1) the performance of the mobile in question may be degraded (i.e. it may not achieve target) and (2) the system creates larger out-of-cell interference than with proper ordering.

Figure 81 and Figure 82 plot the average received power and the achieved frame error rate for the power limit respectively. This essentially represents an ordering which is the opposite of the optimal ordering scheme since the more limited signals are cancelled first and the less limited signals are cancelled last. Again, we see that signals which are power-limited cannot reach their target FER, but they do not have an adverse effect on the other signals in the system. Again, this suggests that the SIC receiver with outer-loop power control is fairly robust. The base station need not specify separate fine-tuned threshold tables for each cancellation situation in order for SIC to be effective. Instead, the base station simply needs to have a rough idea of the relative shadow fading and path loss (as a gauge to the relative power limits of the mobiles) in order to make an ordering assignment. This can be gauged by using energy estimates obtained by the searcher and acquisition circuitry. Further, FER is estimated at the base station. If a particular signal is not meeting its FER target, it can be moved down in the cancellation order to improve its performance based on its power limit. This order control can be done at a much slower rate to combat long term fading changes. System software can detect high FER values and instruct the base station to move the signal down in the cancellation order.

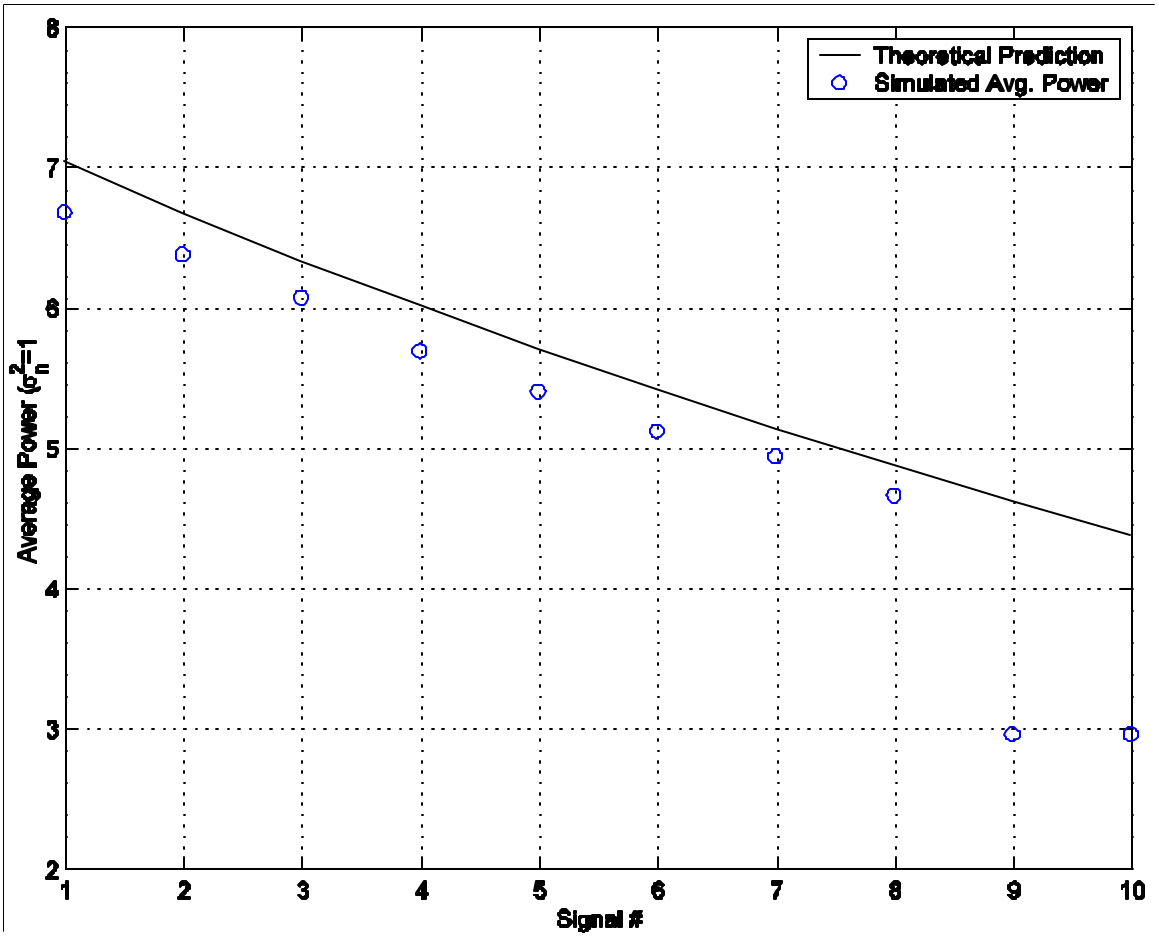

Figure 79 Simulated and Theoretical Power Profile for 10 Users with Signals 9 and 10 Power Limited

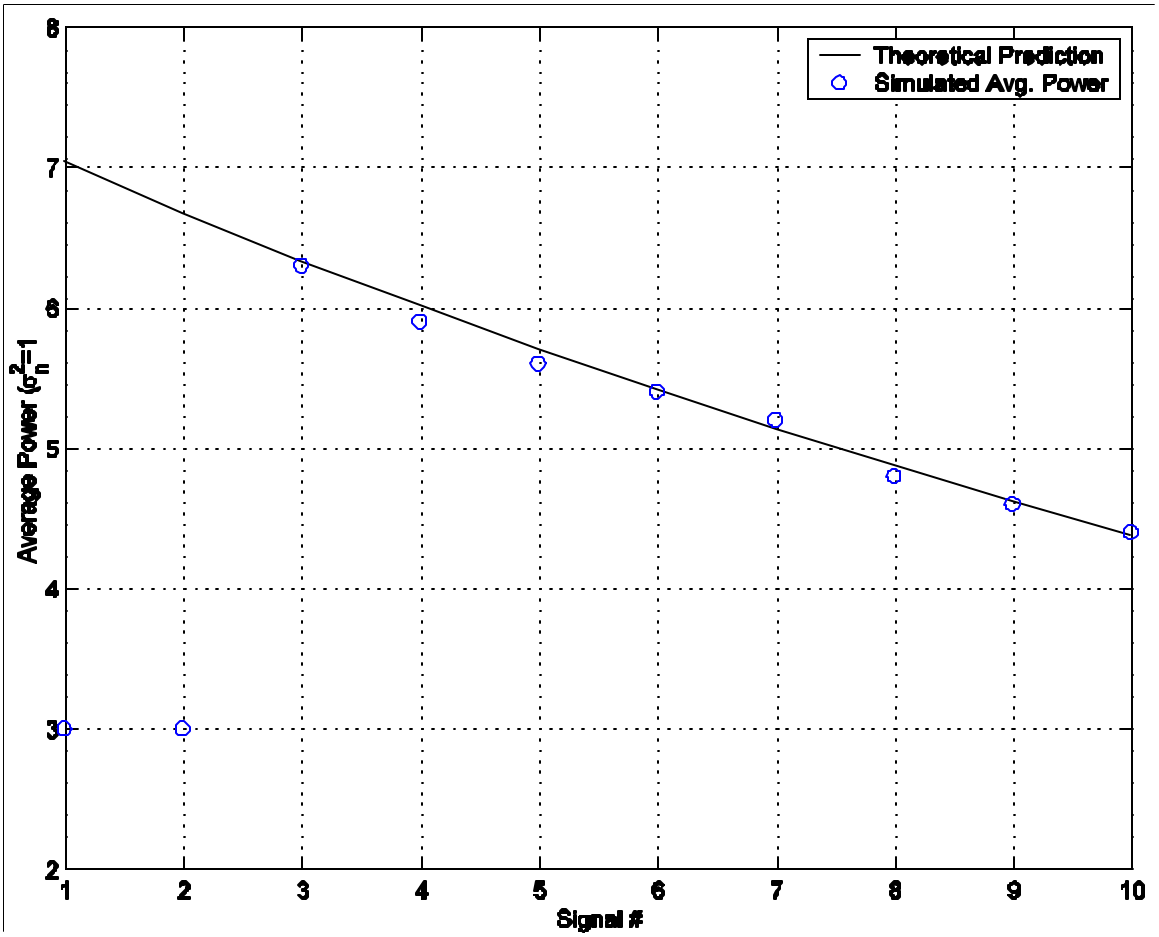

Figure 80 Simulated and Theoretical Power Profile for 10 Users with Signals 1 and 2 Power Limited

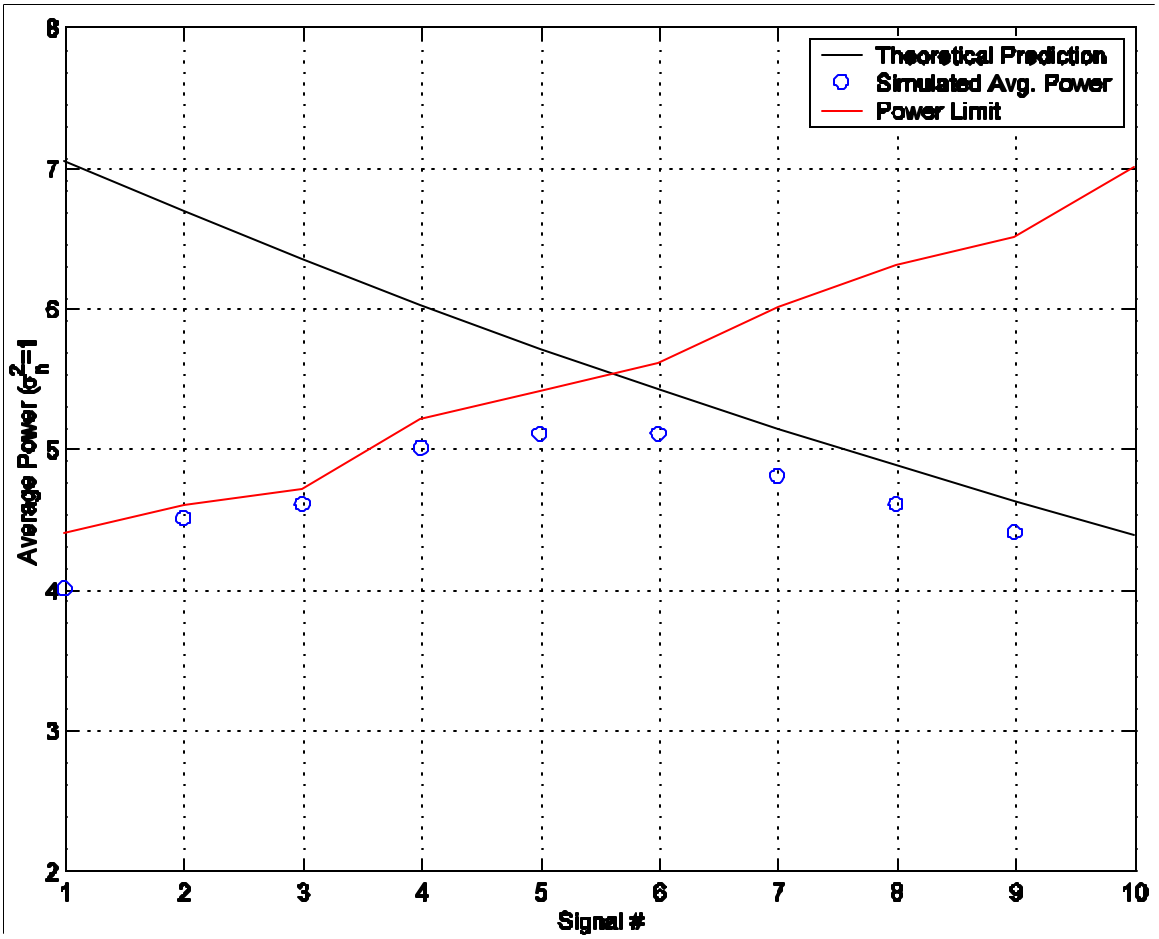

Figure 81 Simulated and Theoretical Power Profile for 10 Users with Reverse Ordering with respect to Power Limits

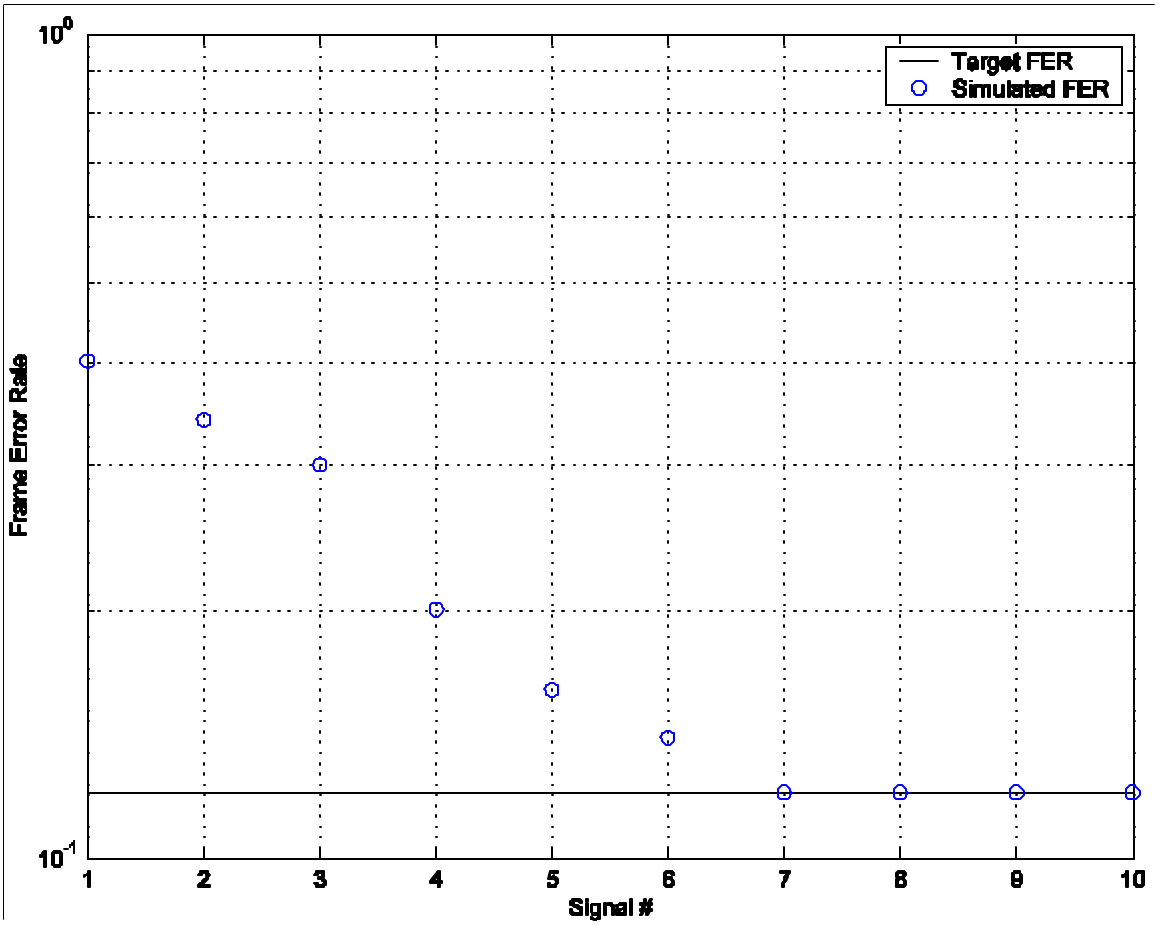

Figure 82 Simulated and Theoretical FERs for 10 Users with Reverse Ordering with respect to Power Limits

Figure 83 and Figure 84 show the power profile and the target FER achieved when we increase the load on the system to K= 40 users.

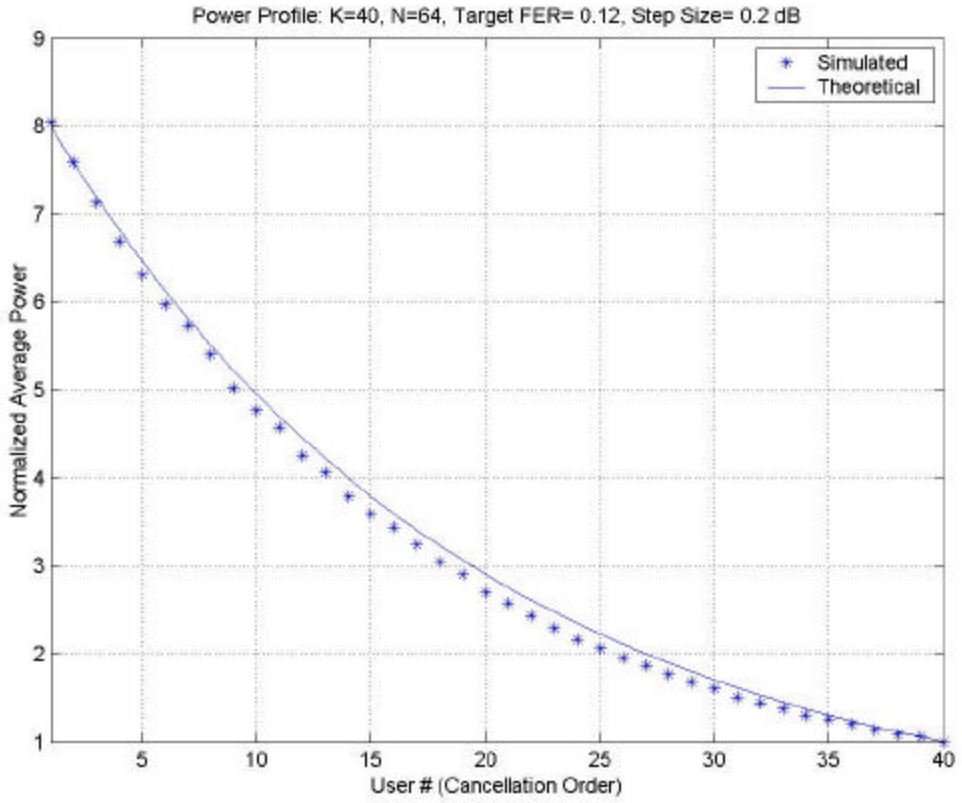

Figure 83 Simulated and Theoretical Power Profile for 40 Users

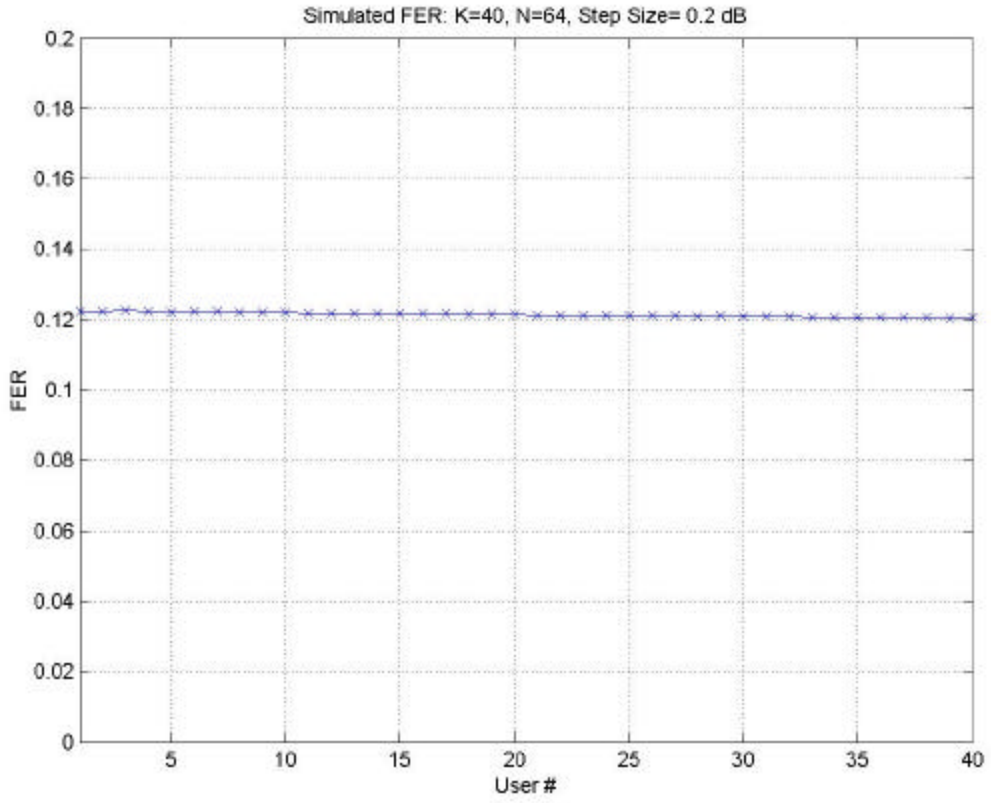

Figure 84 Simulated and Theoretical FERs for 40 Users

Figure 85 shows the power profile for K= 20 users with and without coding. As expected, coding lowers the power required to reach the target FER. It appears that the users earlier in the cancellation order benefit most from the coding in that there is more difference between the coded and un-coded cases earlier in the cancellation order, this is because due to coding, the BER requirements of the system change which changes the geometricity of the power profile (the ratio between consecutive powers).

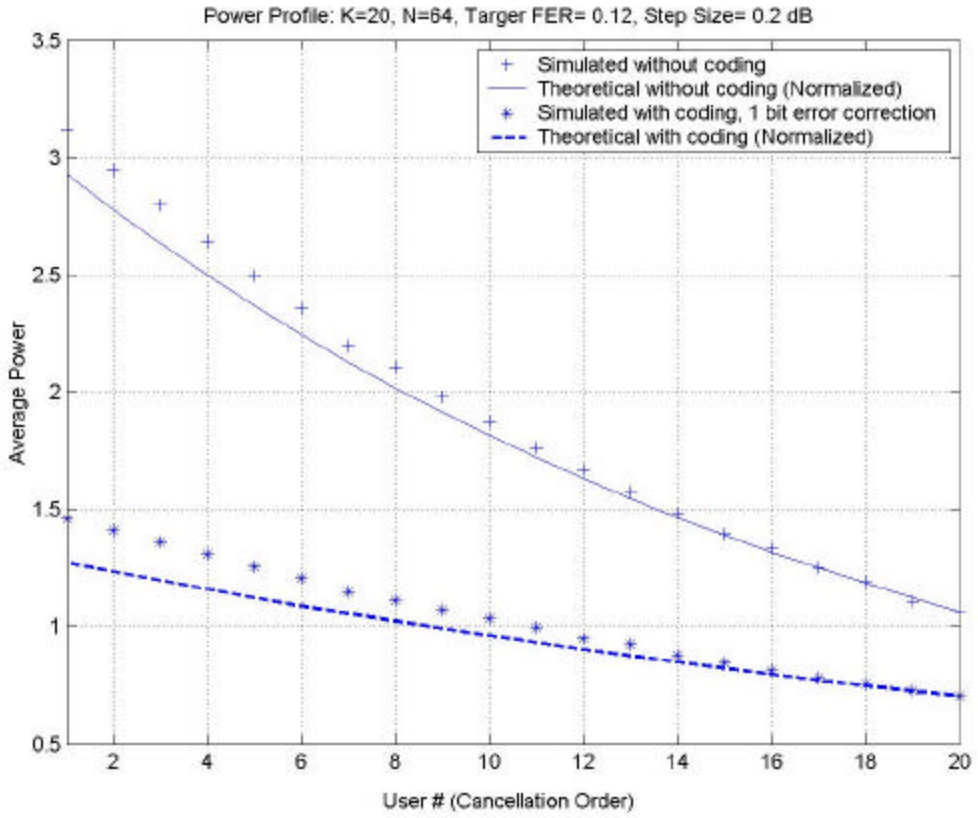

Figure 85 Simulated and Theoretical Power Profile for 20 Users with and without Coding

Figure 86 shows the power profiles achieved when we incorporate varying QoS requirements with K= 30 users. Power profile 1 is ordered such that the first 10 users have a target FER of 10%, the next 10, 30% and the last 10, an FER of 5 %. If we reverse the cancellation order, such that the first 10 users have an FER requirement of 50% and the last 10, an FER requirement of 10%, power profile 2 is obtained. Figure 87 and Figure 88 show the target FER for power profile 1 and power profile 2 respectively.

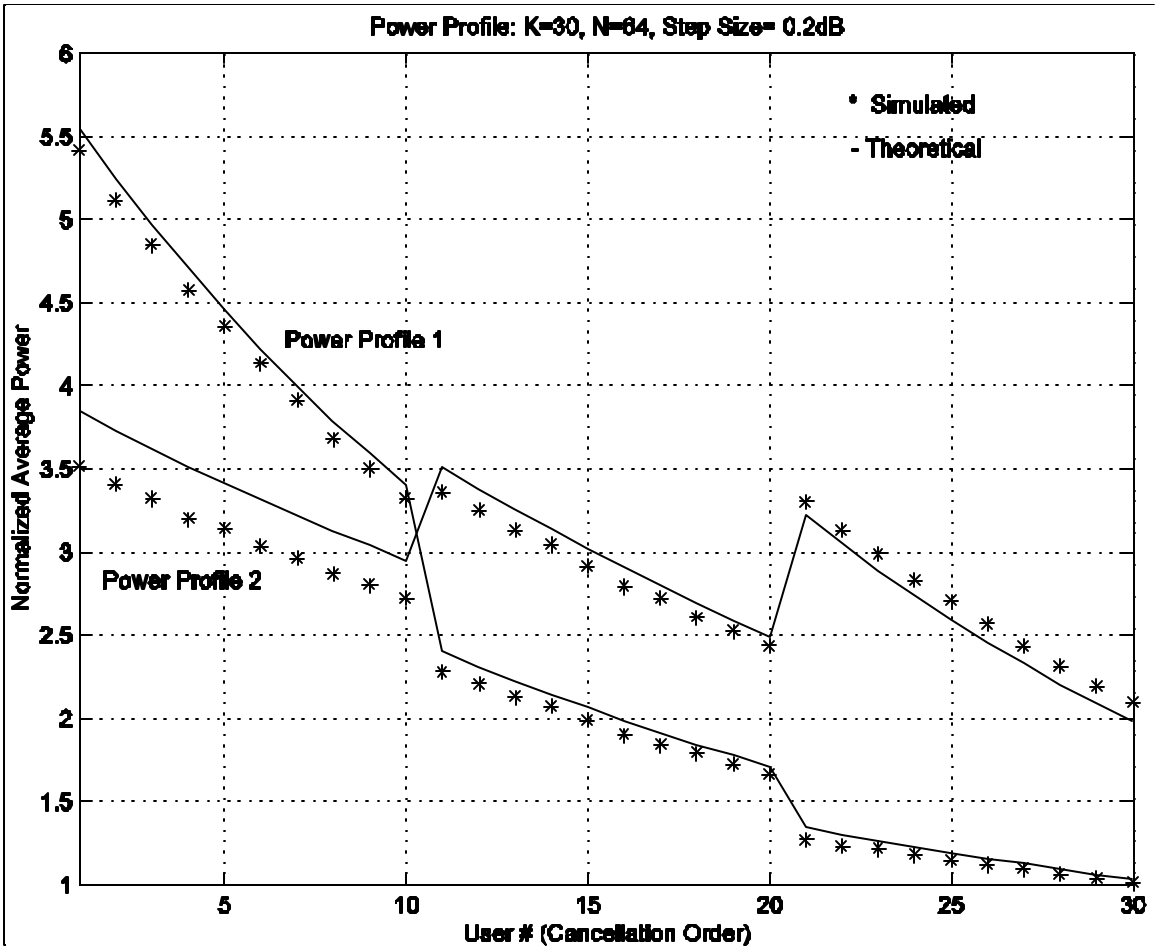

Figure 86 Simulated and Theoretical Power Profiles for 30 Users with Varying QoS Requirements and Different Ordering

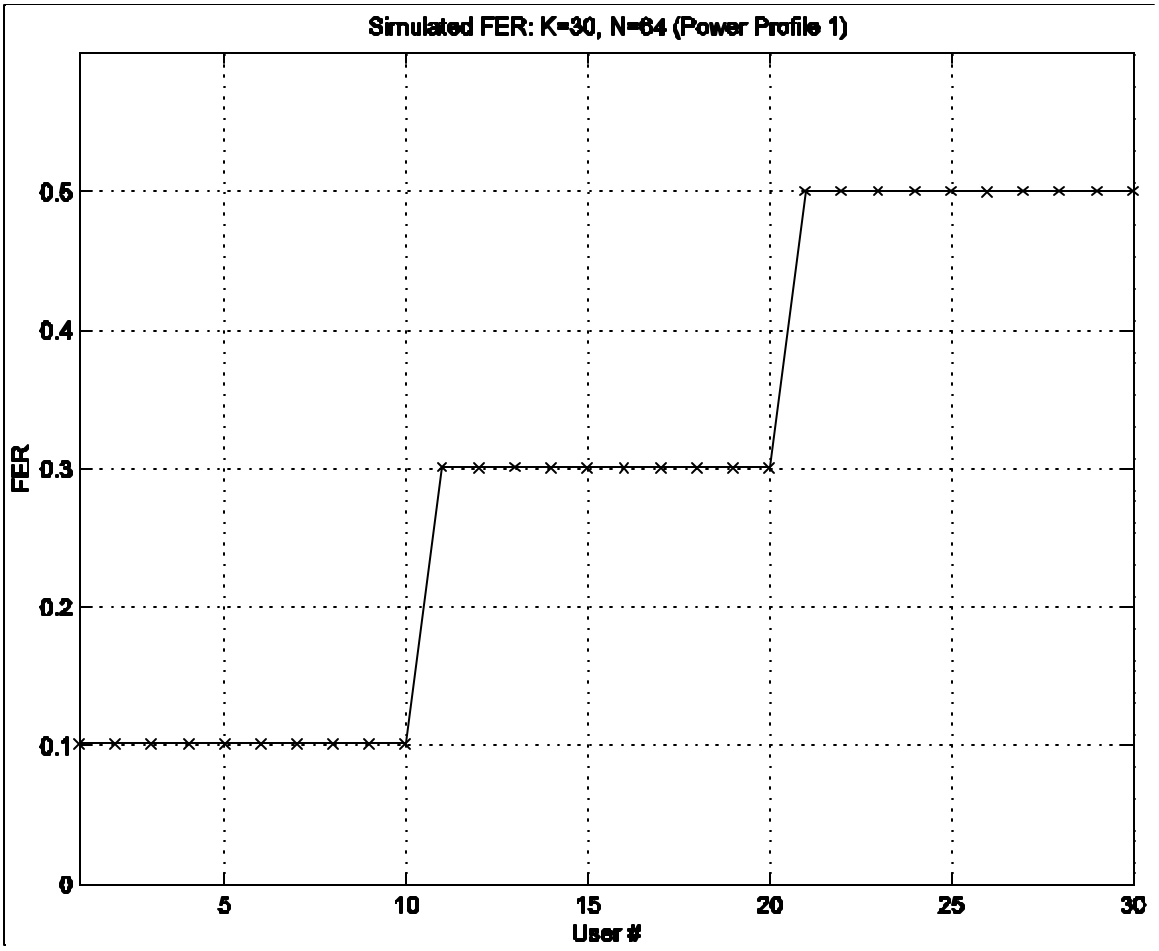

Figure 87 Simulated and Theoretical FERs for 30 Users with Varying QoS Requirements Ordered with Increasing FER

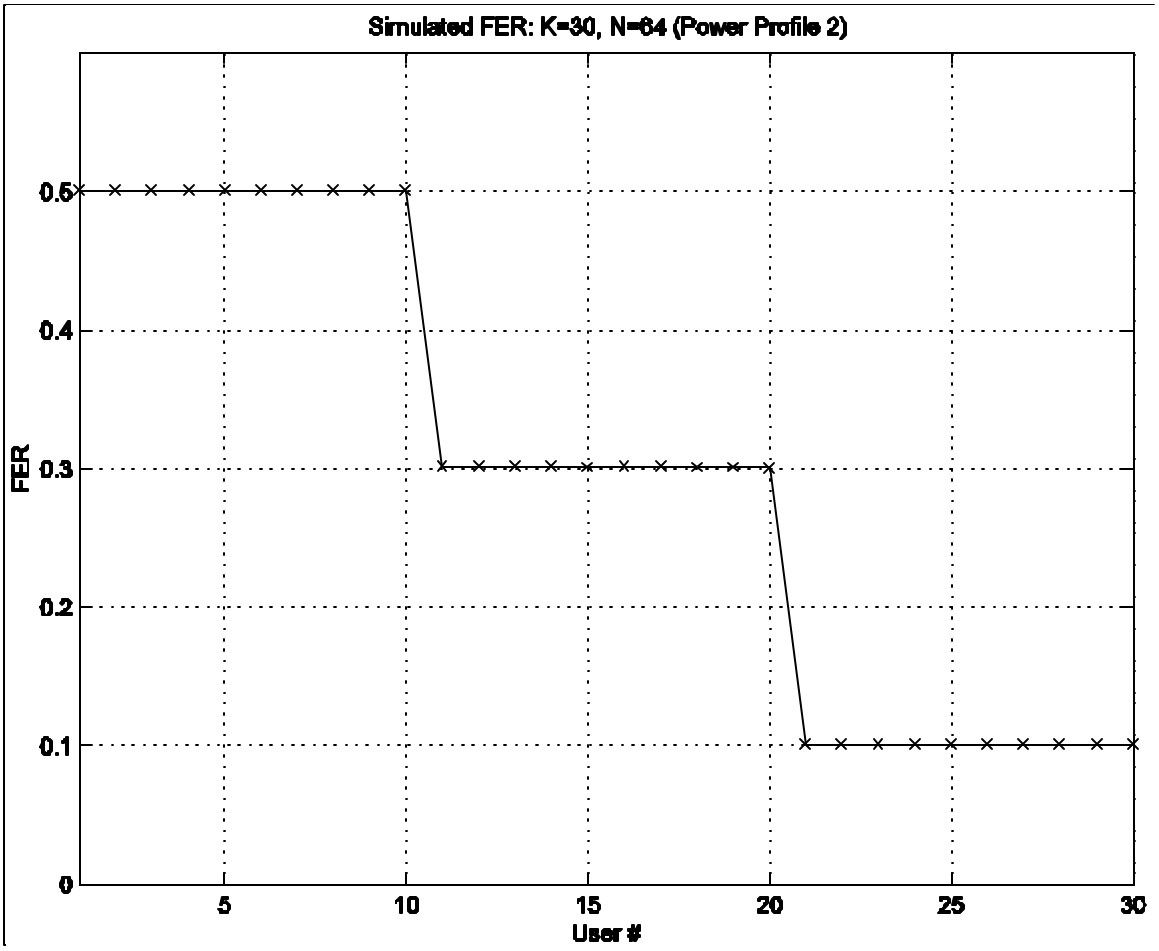

Figure 88 Simulated and Theoretical FERs for 30 Users with Varying QoS Requirements Ordered with Decreasing FER

From Figure 87 and Figure 88 we notice that when we are combining varying QoS requirements with Successive Interference cancellation, it makes sense to order the users with increasing QoS requirements (users with a higher target FER should be cancelled last), so that the power limit on each mobile is less. Hence, in a system with mixed traffic, users more tolerant to a higher BER such as voice should be assigned to the initial cancellation spots while those with data traffic should be assigned later and thus benefit most from the cancellation. These signals would then, despite the higher QoS requirement, require less power than if assigned earlier.

All the results discussed above assume perfect inner loop power control. On modeling the inner-loop power control error with a lognormal distribution, the power profiles shown in Figure 89 and Figure 90 are obtained.

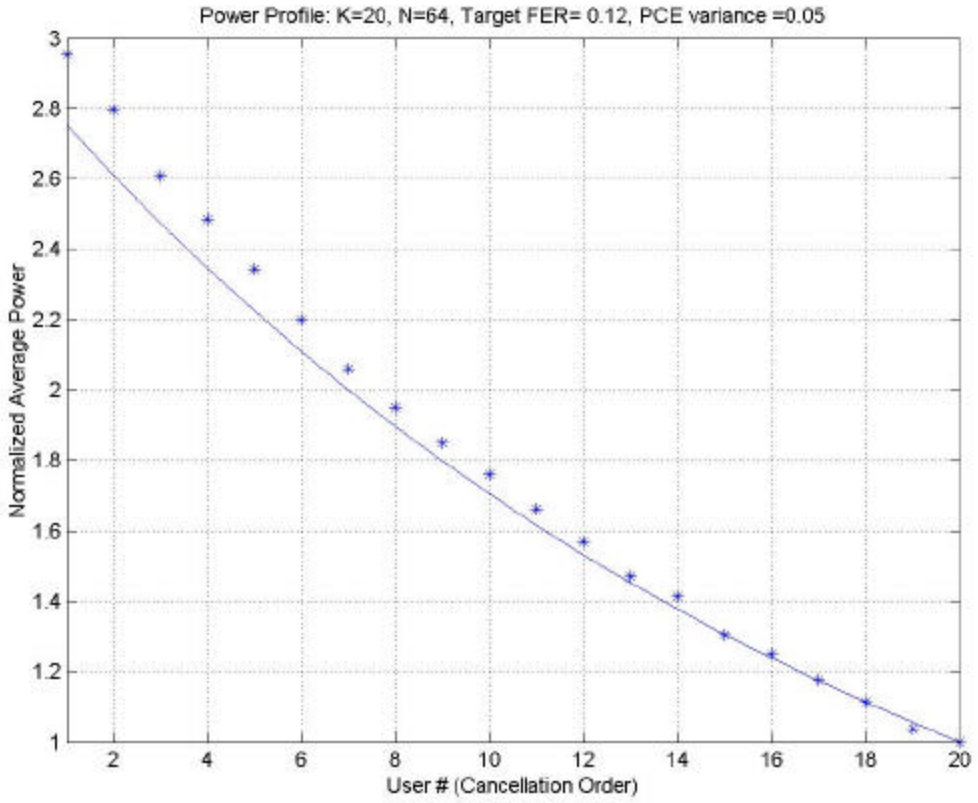

Figure 89 Simulated and Theoretical Power Profile with 0.05 dB Inner-Loop Power Control Error

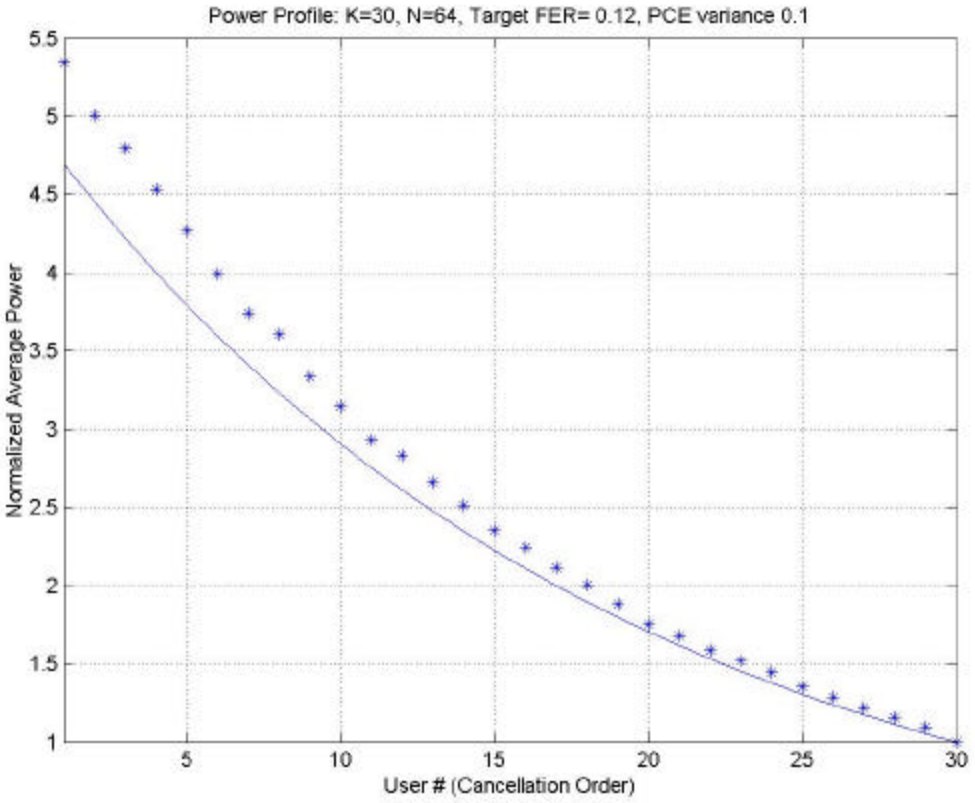

Figure 90 Simulated and Theoretical Power Profile with 0.1 dB Inner-Loop Power Control Error

5.4 Conclusions

In this chapter the role successive interference cancellation receivers at the physical layer play on power control which is a layer 2 radio resource management issue is studied. With the increasing need to differentiate classes of service among users, the deployment of applications that lead to mixed traffic QoS (Quality of Service) is a very important issue in wireless systems. In this section, a power control algorithm was studied, which drives the average received powers to those required, based on the QoS requirements of the individual users for a cellular CDMA system using SIC receivers.

It has been shown that FER-based outer-loop power control can be used with successive interference cancellation to provide stable error rate performance without having sophisticated power threshold tables. An FER-based (or equivalently BER-based) power control algorithm was shown to converge for SIC and the performance was demonstrated in several simulations. Further it was shown that the effect of making errors in the assignment of cancellation order does not have an adverse effect on the whole system. Rather it affects only the signal which is limited. This effect can be detected at the base station by monitoring FER and making corrections in the cancellation order.

The power control algorithm was extended to include varying QoS requirements for each user in terms of the target FER. The algorithm remained stable and the power profile converged to that required for each user to meet its target FER.

Intelligent assignment of the cancellation order is required, as it does not make sense to assign the user furthest from the base station to the first cancellation slot. This will not only result in more interference to other users, but also that mobile may have insufficient power to achieve its target FER.

An important aspect of adding varying QoS requirements to the system is that users should be ordered such that those with more stringent QoS (lower FER) requirements should be assigned places lower in the cancellation order. This is because these users would benefit more from the SIC receivers and would require a lower power to reach their higher QoS requirement. Hence the power limit on each mobile would be less.

Several loading factors were examined, as well as the case of multiple target error rates. Also the impact of coding and inner-loop power control error was briefly examined. The system was stable with inner-loop errors of up to 0.1 dB.

The performance benefits obtained though a cross-layer design applied to a power control algorithm was studied in this section. Future work could further investigate the impact of increased power control errors and convergence time.

6 Conclusions

6.1 Cross-Layer Simulation methodologies

While simulating a cross-layer scenario such as a MANET equipped with smart antennas abstractions of the physical layer hold the greatest promise for achieving high fidelity simulations with short simulation run times. So in general, abstracted physical layers should be used when possible. However, all abstractions make certain simplifying assumptions. Before using an abstraction, it is imperative that the assumptions of the abstraction are known and the modeled system can be shown to satisfy the abstraction's assumptions.

When this does not occur, significant simulation fidelity can be lost as was seen in Section 2.3 However, had all nodes in that modeled system been stationary, then the abstracted beam pattern would have been appropriate and acceptable fidelity would have been achieved.

Full physical layer simulations should be performed whenever any of the following conditions apply:

- no abstractions exist, as can happen when modeling a new waveform
- the conditions for the abstraction are not satisfied
- no specific knowledge of whether or not an abstraction holds, *i.e.*, when in doubt err on high fidelity rather than simulation run-time

The exact nature of the channel realization can significantly impact the fidelity of an algorithm. For a slowly varying channel, it should be safe to assume a static channel and the same SNR (presumably used by different stages) can be applied across an entire packet. However, for a quickly changing channel (high Doppler), the static channel assumption would be erroneous and fidelity would be sacrificed, this is especially true for coded systems

For mobile systems, a full implementation of the adaptive antenna structure is a necessity. Unlike modulation and coding where expressions exist, there are no equations that describe the gain of an antenna array as an instantaneous function of mobile positions and transmit powers. With no acceptable abstraction, the full behavior must be implemented.

For convolutionally encoded systems, consideration needs to be given to whether the channel is relatively static or is dynamic. For a static system, simulation should be performed with an abstracted representation of the performance of the encoder, such as the union bound used in Section 3.3.3.3.2. This provides both excellent fidelity and excellent run-time. However, for a dynamic channel (significant changes in SNR over the length of a packet), the encoder needs to be fully implemented.

While co-simulation with MATLAB is attractive from a development point of view, it runs significantly slower than C implementations. The use of mex-files do little to reduce

the additional overhead as the primary simulation time penalty results from the operation of copying information between the OPNET memory space and MATLAB's memory space as was shown in Section 2.1.3.

While initial algorithm development can be performed in MATLAB, code intended for long-term usage, possibly in other simulations, should be ported to C. This simplifies algorithm debugging (for which MATLAB is superior to OPNET) and then the C implementation provides the speed necessary to allow the programmer to focus on other aspects of the simulation.

OPNET provides a convenient platform for the modeling of network protocols. Process models may be written to model the behavior of a link layer scheme such as a MAC protocol. However, as established earlier, systems with dynamic physical channels need to be simulated for a cross-layer perspective in order to achieve a sufficient level of fidelity even though this results in an increase in simulation run time.

Cross-layer simulations are best achieved by building process models to simulate higher-layer behavior and using C code to modify the pipeline stages used by OPNET to model channel effects.

We demonstrated that mixed C and OPNET simulation was superior to mixed MATLAB and OPNET simulation in terms of simulation time with only a small added degree of complexity. Complex systems containing significant physical layer and link layer behavior can be simulated by a mix of OPNET and C code. In this approach, physical layer behavior is simulated in C by introducing modifications to the pipeline stages and higher layer behavior is implemented using OPNET process models. This is an optimal strategy as C provides the level of detail required for physical layer simulations and OPNET provides excellent support for higher layer simulations.

6.2 A Medium Access Control scheme for MANETs equipped with smart antennas

We studied a MAC scheme for use in MANETs with smart antennas. We found that the 802.11 based MAC scheme with omnidirectional antennas performs better than the MAC scheme studied in this research *with omnidirectional antennas*. This is as expected as the MAC scheme studied and implemented in meant for use with smart antennas. This scheme when used in a system along with adaptive beamforming performs better than the 802.11 MAC scheme based on CSMA/CA. Using transmitter beamforming only does not hinder system performance substantially. While transmitter and receiver beamforming achieves marginally better performance, it contributes to an increase in required processing power.

This study demonstrated the interaction between the physical layer and the link layer MAC protocol. This interaction arose from coupling the MAC addressing scheme with a MMSE beamforming algorithm. This also demonstrates the feasibility and advantage of integrating smart antenna algorithms into a MANET.

The true benefit of this MAC scheme using smart antennas is seen on the increase in network capacity which reinforces the importance of system design from a cross-layer perspective.

6.3 A Power Control Scheme with Successive Interference Cancellation to satisfy Varying QoS Requirements for CDMA systems

The role successive interference cancellation receivers at the physical layer play on power control which is a layer 2 radio resource management issue was studied, a power control algorithm, which drives the average received powers to those required, based on the QoS requirements of the individual users for a cellular CDMA system using SIC receivers was implemented.

An FER-based (or equivalently BER-based) power control algorithm was shown to converge for SIC and the performance was demonstrated in several simulations. Further it was shown that the effect of making errors in the assignment of cancellation order does not have an adverse effect on the whole system. Rather it affects only the signal which is limited. This effect can be detected at the base station by monitoring FER and making corrections in the cancellation order.

The power control algorithm was extended to include varying QoS requirements for each user in terms of the target FER. The algorithm remained stable and the power profile converged to that required for each user to meet its target FER.

Intelligent assignment of the cancellation order is required, as it does not make sense to assign the user furthest from the base station to the first cancellation slot. This will not only result in more interference to other users, but also that mobile may have insufficient power to achieve its target FER.

An important aspect of adding varying QoS requirements to the system is that users should be ordered such that those with more stringent QoS (lower FER) requirements should be assigned places lower in the cancellation order. This is because these users would benefit more from the SIC receivers and would require a lower power to reach their higher QoS requirement. Hence the power limit on each mobile would be less.

Several loading factors were examined, as well as the case of multiple target error rates. Also the impact of coding and inner-loop power control error was briefly examined. The system was stable with inner-loop errors of up to 0.1 dB.

.

6.4 Future Directions

This work provides an excellent starting point for additional research on cross-layer systems. For instance, the impact of the MAC scheme detailed in Section 4.1.1 on multi-hop message transfer could be studied to further study the system benefits of smart antennas in a MANET and the interaction between the MAC and the network layer.

Another issue that needs addressing is the periodicity of the RTR packets in the MAC scheme.

Further investigation on the impact of power control errors and convergence time could be carried out on the power control scheme described in Section 5.

7 References

[1] **OPNET user documentation**

[2] **A novel MAC layer protocol for space division multiple access in wireless ad hoc networks**
Lal, D.; Toshniwal, R.; Radhakrishnan, R.; Agrawal, D.P.; Caffery, J., Jr.;
Computer Communications and Networks, 2002. Proceedings. Eleventh International Conference on , 14-16 Oct. 2002

[3] **Performance, optimization, and cross-layer design of media access protocols for wireless ad hoc networks**
Toumpis, S.; Goldsmith, A.J.;
Communications, 2003. ICC '03. IEEE International Conference on , Volume: 3 , 11-15 May 2003

[4] **Cross-layer design for wireless networks**
Shakkottai, S.; Rappaport, T.S.; Karlsson, P.C.;
Communications Magazine, IEEE , Volume: 41 Issue: 10 , Oct. 2003

[5] **Equal BER performance in linear successive interference cancellation for CDMA systems**
Buehrer, R.M.;
Communications, IEEE Transactions on , Volume: 49 Issue: 7 , July 2001

[6] **Power allocation for a simple successive interference cancellation scheme in a multi-rate DS-CDMA system**
Berggren, F.; Ben Slimane, S.;
Communications, 2002. ICC 2002. IEEE International Conference on , Volume: 1 , 28 April-2 May 2002

[7] **Throughput enhancement in wireless ad hoc networks with spatial channels a MAC layer perspective**
Lal, D.; Gupta, R.; Agrawal, D.P.;
Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on , 1-4 July 2002

[8] **A MAC protocol for mobile ad hoc networks using directional antennas**
Nasipuri, A.; Ye, S.; You, J.; Hiromoto, R.E.;
Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE , Volume: 3 , 23-28 Sept. 2000

[9] **Spatially divided channel scheme using sectored antennas for CSMA/CA "directional CSMA/CA"**
Kobayashi, K.; Nakagawa, M.;
Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on , Volume: 1 , 18-21 Sept. 2000

[10] **A simple and effective cross layer networking system for mobile ad hoc networks**
Wing Ho Yuen; Heung-no Lee; Andersen, T.D.;
Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on , Volume: 4 , 15-18 Sept. 2002

[11] **Medium access control protocols using directional antennas in ad hoc networks**
Young-Bae Ko; Shankarkumar, V.; Vaidya, N.H.;
INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Volume: 1 , 26-30 March 2000

[12] **MAC protocol for mobile ad hoc network with smart antennas**
Jun Yang; Jiandong Li; Min Sheng;
Electronics Letters , Volume: 39 Issue: 6 , 20 March 2003

[13] **A comparison study of omnidirectional and directional MAC protocols for ad hoc networks**
Zhuochuan Huang; Chien-Chung Shen;
Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE , Volume: 1 , Nov 17-21, 2002

[14] **IEEE 802.11 over multi-hop wireless networks: problems and new perspectives**
Hung-Yun Hsieh; Sivakumar, R.;
Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th , Volume: 2 , 24-28 Sept. 2002

[15] **Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks**

Mineo Takai, Jay Martin; Rajive Bagrodia;
Proceedings of the 2001 ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc 2001), October 2001

[16] **On the Usefulness of Outer-Loop Power Control with Successive Interference Cancellation**
Buehrer, R.M; Mahajan, Rahul
Communications, IEEE Transactions on, December 2003

[17] **Analysis of a simple successive interference cancellation scheme in a DS/CDMA system**
Patel, P; Holtzman, J;
Communications, 1994. IEEE Journal on Selected Areas, Volume: 12 , June 1994

[18] **Reduction of other-cell interference with integrated interference cancellation/power control**
Hatrack, P.; Holtzman, J.M.;
Vehicular Technology Conference, 1997 IEEE 47th, Volume: 3 , 4-7 May 1997

[19] **Power control for a spread spectrum system with multi-user receivers**
Kumar, P. S.; Holtzman, J.;
Personal, Indoor and Mobile Radio Communications, 1995. PIMRC'95. 'Wireless: Merging onto the Information Superhighway'., Sixth IEEE International Symposium on , Volume: 3 , 27-29 Sept. 1995

[20] **Stochastic power control for nonlinear multiuser receivers in cellular radio networks**
Varanasi, M.K.;
Information Theory and Communications Workshop, 1999. Proceedings of the 1999 IEEE , 20-25 June 1999

[21] **Optimal sequences, power control, and user capacity of synchronous CDMA systems with linear MMSE multiuser receivers**
Viswanath, P.; Anantharam, V.; Tse, D.N.C.;
Information Theory, IEEE Transactions on , Volume: 45 Issue: 6 , Sept. 1999

[22] **Joint power control, multiuser detection and beamforming for CDMA systems**
Yener, A.; Yates, R.D.; Ulukus, S.;
Vehicular Technology Conference, 1999 IEEE 49th , Volume: 2 , 16-20 May 1999

[23] **Power control algorithm for MMSE receiver based CDMA systems**
Almutairi, A.F.; Miller, S.L.; Latchman, H.A.; Wong, T.F.;
Communications Letters, IEEE , Volume: 4 Issue: 11 , Nov. 2000

[24] **Power control for an asynchronous multirate decorrelator**
Saquib, M.; Yates, R.D.; Ganti, A.;
Communications, IEEE Transactions on , Volume: 48 Issue: 5 , May 2000

Vita

Rahul Mahajan was born in Mumbai, India in 1979. He received the Bachelor of Engineering degree in Electronics and Telecommunication from the University of Pune in 2001. He received his M.S. degree in Electrical Engineering at Virginia Tech in 2003. From 2001 to 2002 he served as the Outreach Chair for the Virginia Tech IEEE Chapter. He was a graduate research assistant at the Mobile and Portable Radio Research Group while at Virginia Tech where his interests included cross-layer optimization of systems, CDMA systems, radio resource management and mobile ad-hoc networks.