MULTI-CELL WCDMA SIGNAL PROCESSING SIMULATION TESTBED

BY VIKRAM KAUL

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor Roy D. Yates

and approved by

New Brunswick, New Jersey

January, 2001

ABSTRACT OF THE THESIS

Multi-cell WCDMA Signal Processing Simulation Testbed

by Vikram Kaul

Thesis Director: Professor Roy D. Yates

Third generation (3G) wireless systems are being developed to increase system capacity and support innovative broadband multimedia services. Wideband CDMA (WCDMA) is one such system where the air interfaces need to support services with different bit-error rates, delay, and rates. This thesis describes the Wireless Propagation and Protocols Evaluation Testbed (WiPPET$_{\text{signal}}$), a virtual testbed for signal processing simulation for the physical layers of CDMA wireless systems. A multi-cell simulation platform implementing user mobility and detailed radio channels has been developed to study the impact of mobile users, fading and propagation effects, and other-cell interference on waveform level processing. The ultimate aim of the study is analyzing and improving algorithms at the link level (interference cancellation) and system level (power control and handoff) for WCDMA systems.

The modules in the simulation model are constructed atop the new object-oriented *Scalable Simulation Framework (SSF)* API that promotes high efficiency and separation of modeling code from the internals of a parallel simulator. As an application of this platform, the reverse link performance of the WCDMA (NTT DoCoMo/ARIB) transceiver design and framing structure has been analyzed and validated against measurements for multiple mobiles in a single cell. Subsequently, we have extended the model to multiple cells, with antenna diversity, SIR based power control and soft-handoff. BER curves for various setups of variables like noise variance, SIR targets, transmitter power limits and spreading gains have been obtained. This reverse link simulation is being used to enhance the pilot assisted coherent multistage interference canceller from NTT DoCoMo. Channel estimation, power control and channel coding are also being investigated to help in the improvement of system capacity. Forward link simulation is being used to assist in soft handoff. Gains in performance due to soft handoff have also been illustrated.

The parallelization problem for the simulation is described, and two possible solution approaches discussed. Speedups obtained from the preferred solution are also illustrated. The validation of the various modules of WiPPET$_{\mathrm{signal}}$ has also been discussed. As part of the thesis, an example SSF program has been elucidated for the benefit of future users.

Acknowledgements

I would like to thank my advisor Dr. Roy D. Yates for his patience and encouragement during the two years at WINLAB. I would also like to thank Dr. Andrew T. Ogielski for introducing me to the world of simulations and SSF, and Dr. Narayan B. Mandayam for providing the theoretical insights to Wireless communications. I also extend my gratitude to NTT DoCoMo and Mr. Kenichi Higuchi and also to Jie Lai and Wenfeng Zhang for supporting and helping me out with lots of the work. Finally, I would like to thank all members of WiPPET group and my friends at WINLAB for all the wonderful memories.

Dedication

To my mother

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

For wireless networks, simulations are vital for validating the analytical and mathematical models of algorithms and protocols. However, these simulations of wireless systems differ considerably from simulations of wired networks in that they have to include modeling of the geographic environment and physical radio channel properties, both of which have a profound effect on network performance. These complex simulations need to be fast and scalable to cater to needs of researchers who wish to validate new and improved algorithms. WINLAB has been actively involved in wireless system simulation and this thesis describes another step into that direction.

1.1 Simulator History at WINLAB

Radio resource management and allocation has been an actively researched topic which aims to provide insight and algorithms for modern radio networks that seek to provide a specified quality of service to the maximum number of users. WINLAB's previous sequential simulation tool, the Mobility And Dynamic Resource Allocation Simulator (MADRAS [7]), was used extensively in research [8, 9, 41, 42]. It has been followed by Wireless Propagation and Protocols Evaluation Testbed (WiPPET), a virtual testbed for parallel simulation of wireless networks. Earlier versions of WiPPET dealt with wireless networks at the packet and session levels [24–26].

1.2 Motivation and Objectives

Emerging third-generation (3G) mobile radio networks intend to provide multimedia and high-bit-rate packet data services. In the International Telecommunications Union (ITU), third generation networks are called International Mobile Telecommunications-2000 (IMT-2000), and in Europe, Universal Mobile Telecommunications System (UMTS). Wideband code division multiple access (CDMA) has emerged as the mainstream air interface solution for the third-generation networks. In Europe, Japan, Korea, and the United States, wideband CDMA systems are currently being standardized.

The promise of high data rate services is the genesis for research into various aspects of 3G systems. Enhancements to the physical layer are among the most interesting and tangible directions for 3G research. However, system complexity dictates the need for simulation tools to evaluate and improve designs. Physical layer network simulation typically requires two main tasks: the calculation of the received signal at each network node and the receiver signal processing for bit detection. Both these classes of signal processing require complex calculations, and thus dominate the total computational effort. Therefore, the goals of efficiency and accuracy require that careful consideration be given to the implementation of these tasks.

Beyond efficiency and accuracy, the other important goal of a simulator must be the ease with which researchers can write and incorporate modules that implement new and better algorithms. In turn, this translates into clear class organization, easy overloading of methods and ability to add to the library of functionality. These simulations also have a high demand for computing resources. Separation of modeling from implementation helps to solve both problems. The Scalable Simulation Framework (SSF) [1] has been designed with the goal of reducing the time required to construct new simulation models by shifting the work to design models rather than low-level programming of customized simulator software and enabling reuse of a growing database of objects for creation of more advanced simulations.

This thesis describes the signal processing version of the Wireless Propagation and Protocols Evaluation Testbed ($\text{WiPPET}_{\text{signal}}$), a virtual testbed for parallel simulation of W-CDMA networks. It has been written in the Scalable Simulation Framework (SSF) that has been developed by the S3 consortium [1]. SSF is a new public-domain standard for discrete-event simulation of large, complex systems. SSF models are compact, flexible, portable, and transparently parallelizable. The SSF API has both C++ and Java bindings, with available high-performance serial and scalable parallel implementations.

The objectives of this work are

- Design and implement WCDMA physical layer wireless network modeling classes in C++ SSF for the purpose of developing a general, versatile simulator; modules of which can be quickly overloaded to provide evaluations of algorithms for various physical layer operations.
- Develop parallelization techniques to speed-up the simulator.
- Demonstrate the simulator in selected applications, in particular, the 3GPP Framing Format with NTT DoCoMo Transceiver.

Chapter 2

Fundamentals of SSF and Discrete Event Simulator

This chapter describes the Scalable Simulation Framework (SSF) developed by the S3 consortium, a collaboration of researchers in networking, parallel simulation, and software engineering. The goal of project S3 is to achieve radical improvements in speed, scalability, and manageability of modeling and simulations of very large scale multi-protocol communications networks. The project investigates and develops novel network modeling techniques and new software architectures for parallel discrete event simulation kernels. Scalability and parallel performance of the S3 software are extensively tested on large scale models of the Internet, ATM networks, and mobile wireless and satellite networks.

The Scalable Simulation Framework programming interface (SSF API) is the next generation core interface common to individual S3 sub-projects. The SSF API promotes high efficiency in the separation of network modeling from the internals of parallel simulators, but without a need for a separate framework language like TeD [28], thus allowing unrestricted use of all facilities of standard C++ or Java object oriented programming. The SSF API supports dynamical modeling, essential for runtime self-organization of very large heterogeneous models, as well as for runtime aggregation of sub-models and other novel techniques. The reference development implementation of

SSF is written in Java. The high performance implementation of SSF is written in C++.

This chapter also gives a basic introduction to Parallel Discrete Event Simulator (PDES) and explains the fundamental concepts and the working of PDES. A brief introduction to the workings of a wireless testbed are also given at the end of the chapter.

2.1 Introduction to SSF

SSF provides a single, unified interface for discrete-event simulation (the SSF API). Object-oriented models that utilize and extend the framework can be portable across SSF compliant simulation environments. In addition to its concrete modeling applications, the API also functions as an abstract target for compilation of models specified in higher-level modeling languages or graphical modeling environments. The framework's primary design goal is to support high performance simulation. SSF makes it possible to build models that are efficient and predictable in their use of space, able to transparently utilize parallel processor resources, and scalable to very large collections of simulated entities.

SSF Abstraction

This subsection gives basic introduction to the various structures and terminology used in SSF. The SSF syntax comprises five base class interfaces:

- Entity
- Process

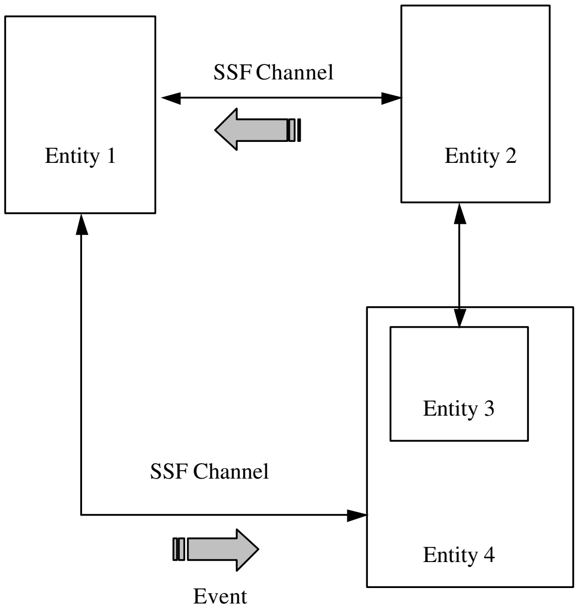


Figure 2.1: Basic SSF framework : SSF Entities communicate over channels through events

- Event
- inChannel
- outChannel

These five classes form a self-contained design pattern for constructing process-oriented, event-oriented, and hybrid simulations. They are sufficient to model any process that can be described as a collection of communicating objects.

In SSF, physical and conceptual objects in the telecommunication domain are modeled as entities. Entities are connected to and interact with each other via channels, called SSF *inChannels* and *outChannels*. SSF channels are the only means of dynamic interaction between entities. Information units, called *events*, flow through SSF channels. An output SSF channel of one entity may connect to the input SSF channel of another entity, or may connect to itself, thus creating *internal* SSF channels. *Processes* inside

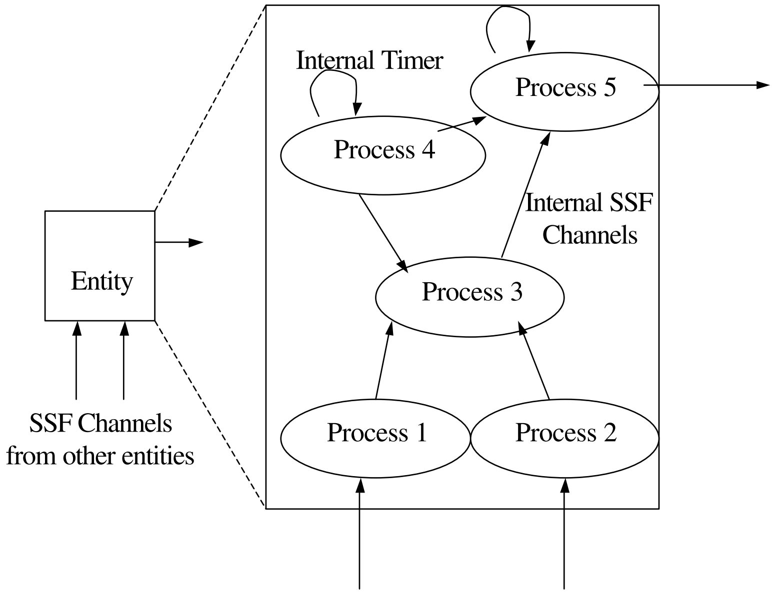


Figure 2.2: Entity and its contents: Processes form the dynamic behaviour of entities

an entity define its dynamic behaviour. Figure 2.1 illustrates the basic framework of SSF where the entities interact with each other. This collection of classes thus allows for modeling an arbitrary structure of connected entities that represent a complex simulation model.

Entity

The Entity base class is used to define a black-box view of an entity. The entity could be a direct model of a real-life object, such as a multiplexer, a Base Station, or of an abstract object, such as a protocol. The behavior of an entity is defined solely in terms of its dynamic reactions to events on its input-mode channels, and the production of events on its output-mode channels. Special semantics are supported for interface inheritance and enhancement.

The dynamic behavior of each entity is described by its *process*. The process is described in terms of the actions of the entity upon event arrival on its input SSF channel, and the production of events on its output SSF channels. Internal SSF channels are used to pass events from one process to another process within an entity. Figure 2.2 illustrates this behaviour. Also, one or more processes can act upon arriving events or wait for events on the input channels of the entity. The processes may generate events on output channels as part of their action. An event generated on an output SSF channel arrives as input on the input SSF channel to which the output SSF channel is *mapped*.

Process

The base class for describing dynamic behavior of an entity is its process. Each instance of process is owned by a specific entity. A process might wait for input to arrive on the channels owned by that entity, wait for some amount of simulation time to elapse, or do both in turn. The simplest process waits for an event to arrive on a channel, responds to it, and then goes back to sleep.

Processes can be defined as dynamic threads of computation acting on behalf of the entities that *own* them. Only the *owner* entity's SSF channels (internal and external) and state variables are accessible to the entity's processes. The functionality of processes consists of combinations of two types of actions: *computation* (using the state variables), and *synchronization* (using SSF channels or time). Computation is a sequence of operations performed on the state variables. Synchronization is a sequence of actions involving events on the channels over time.

Computation actions are specified in the executable language, which is C++ in our system. Synchronization actions are specified using variants of the **wait** statement in the process body. The processes never stop executing, but repeatedly suspend and resume their activity depending on the arrival of events or the timeout of a *wait* statement. SSF processes are conventional thread-like computations with the ability to make nested procedure calls and invoke *wait* statements at arbitrary points in the process and procedure body. These processes can be categorized as :

- *Self-driven process* : This is a self activated process. The process runs periodically after specific time duration.
- *Event-driven process* : This is a process activated by an event. The process listens on input TeD channels, specified in its argument list, and becomes active when an event arrives on the input SSF channels.

Event

The base class for the quantum of information exchange is an *event*. Modelers may create arbitrary derived classes from the event base class provided by the SSF framework. All derived event classes must provide their own public copy constructor. Event storage management is the responsibility of the framework, which may release the storage of an event at any point after its last recipient process has suspended, unless the modeler has explicitly taken over the responsibility for that event.

inChannel and outChannel

The *inChannel* and *outChannel* base classes are the communication endpoints for event exchange. Each instance of inChannel or outChannel is owned by a specific entity.

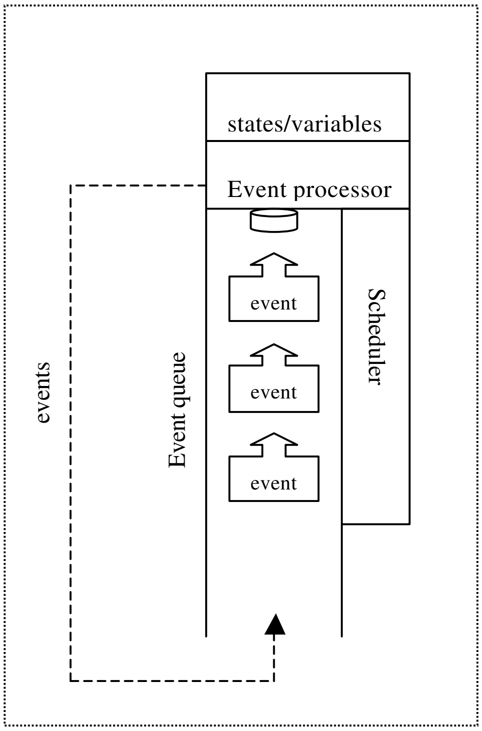


Figure 2.3: DES : Event queues, timeline, state variables

SSF supports multi-cast in-channels (many to one communication) as well as multi-cast out-channels (one to many) and bus-style channel mappings (many to many). Each outChannel might have associated with it an intrinsic minimal transmission delay, which is automatically added to the per-write delays of individual events sent on it.

2.2 Introduction to Parallel Discrete Event Simulator

A Discrete Event Simulation (DES) model expresses changes of state at discrete points in simulated time. The simulation model jumps from one state to another upon the occurrence of an *event*. This subsection first explains the working of DES and then discusses Parallel DES (PDES).

Figure 2.3 shows the main components of a discrete event simulator. The simulator utilizes three principle data structures: 1) the *state variables* that describe the state of the system, 2) an *event queue* containing all pending events that have been scheduled, but have not yet taken effect, and 3) a global *time* variable to denote how far the simulation has progressed. Each event contains a timestamp and some payload data that indicate which change in the state of the simulated system should be executed. The timestamp indicates when this change occurs in the actual system, that is, the time at which the event will be executed. The scheduler attached to the queue arranges the queued events in descending order of their time stamps, so, the earliest time stamped event is at head of the queue and is processed first. The event processing may cause change in the states or variables of the simulation and may generate another events which will be queued. After the first event (which is at the head of the queue) is processed, it is removed from the queue, so the next event in the event queue becomes the head of the queue and will be processed next. The simulation time of the simulator is the timestamp of the last processed event. The simulator advances the simulation time by processing timestamped events.

In this execution paradigm, it is crucial that one always select the earliest timestamped event ($EventT_{min}$) from the event list in the queue as the one to be processed next. This is because if one were to select some other event containing a larger timestamp, say $EventT_x$, it would be possible for $EventT_x$ to modify state variables used by $EventT_{min}$. It would also be possible that $EventT_x$ may use variables that should have been modified earlier by $EventT_{min}$. This would amount to simulating a system which would disregards *causality* conditions and errors of this nature are called causality errors.

Let us now consider parallelization of a simulation program that is based on the above

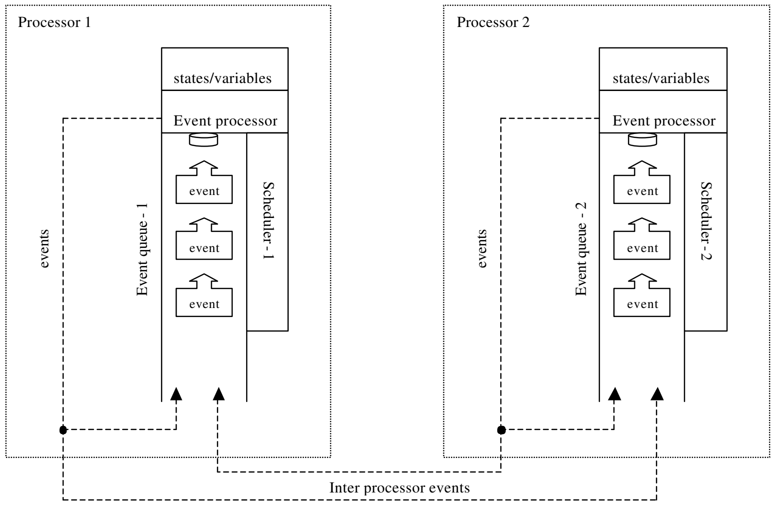


Figure 2.4: PDES : A Two processor parallel discrete event simulators with multi-timeline and event queues

paradigm. The opportunity for parallelism arises from processing events concurrently on different processors. Consider a two-processor PDES shown in Figure 2.4. Assume that the simulation work is distributed between two processors. If there are no inter-processor events (events passing between two processors), the processors run independently of each other. The work should be completed in half the time that one processor would require. Hence, if the work is divided evenly and there are no inter-processor events, then the N processor simulation execution time can be approximated by

$$N \text{ processor execution time} \approx \frac{1}{N} \times (\text{uniprocessor execution time}). \tag{2.1}$$

The interesting case occurs when there are inter-processor events in the two processor simulation. Consider the concurrent execution of two events E_1 sent by processor-1 to processor-2 and E_2 sent by processor-2 to itself, with timestamps T_1 and T_2 respectively. E_1 is the *inter-processor event.* Assume $T_2 > T_1$. Causality will be violated if, say, processor-2 were to process the event E_2 which occurs at timestamp T_2 before processing event E_1. Now, let's consider a scenario in which processor-2 first processed the event E_2 with timestamp T_2. Thus, now, the simulation time of processor-2 is T_2. Then, the E_1 with timestamp T_1 sent by processor-1 is queued in the event queue of processor-2. Since $T_2 > T_1$, E_2 must be executed after E_1 to be sure a causality error does not occur. However, if processors 1 and 2 are not properly synchronized, one can not guarantee causality. Enforcement of causality by synchronization is the basic problem of parallel simulation. Based on the mechanism to avoid or resolve causality errors, there are two basic types of PDES: *conservative* and *optimistic*.

1. *Conservative*: In this case, the simulator avoids the causality error situation. A processor does not process an event unless it is certain that no earlier time event can ever arrive from other processors. There are several algorithms to achieve this; see [11, 16] for a detailed explanation.

2. *Optimistic*: In this case, the simulator allows causality error situations, but it has a mechanism to correct causality errors. Each processor processes events without worry. If an earlier time event arrives, the processor must undo the careless work. In other words, it has to undo the changes made in the simulation states/variables and the effect of processed events. This is called *rollback* of the second-processor. In above example, to resolve causality error situation the second-processor must go backward in time and undo everything done in the simulation up to the present moment T_2. The issue, here, is the possibility of an avalanche of rollbacks. It is measured in terms of *length of rollback* or *rollback distance*, i.e the *number of events* to be rollbacked. See [11, 16] for a details.

The tradeoff between conservative and optimistic simulator is that in conservative simulation one has to estimate the *lookahead time*, which is not easy. On other hand, in case of optimistic simulation, one has to provide mechanism to rollback. A more detailed taxonomy of simulation mechanisms is described in [11, 16, 22].

The SSF framework is based a conservative PDES. Since WiPPET$_{\text{signal}}$ is based on SSF it is a conservative PDES. The parallelization details are further addressed in Chapter 5.

Chapter 3

WCDMA System Model and WiPPET$_{\text{signal}}$ Requirements

WiPPET$_{\text{signal}}$ aims at simulating the physical aspects of a WCDMA system. The framing formats are provided as specifications from the Third Generation Parternership Project (3GPP). The transceiver physical layer signal processing algorithms have to be evaluated; NTT DoCoMo algorithms have been included as a special case. This chapter introduces the WCDMA system model and explains the concept and requirements of virtual testbeds. This chapter and the preceding Section 2.1 are inputs to the design of the simulator which follows in the next chapter.

In this chapter we discuss the system model of WCDMA system with respect to the physical layer characteristics and procedures. First, the DS-CDMA concepts are introduced and then the characteristics that have been added for the 3G WCDMA systems are discussed.

3.1 DS-CDMA Concepts

In DS-CDMA systems, the modulated information-bearing signal (the data signal) is directly modulated by a digital, discrete-time, discrete-valued code signal. This code signal consists of a sequence of 'chips' that can be either +1 or -1. To obtain the desired spreading of the signal, the chip rate of the code signal must be much higher than the chip rate of the information signal. The receiver uses coherent demodulation to despread the spread spectrum (SS) signal, using a locally generated code sequence, which must be synchronized to the received signal. A few important properties of DS-CDMA are:

- *Multiple Access* : If multiple users use the channel at the same time, there will be multiple DS signals overlapping in time and frequency. At the receiver, coherent demodulation is used to remove the code modulation. This operation concentrates the power of the desired user in the information bandwidth. If the crosscorrelation between the code of the desired user and the code of the interfering user is small, coherent detection will capture only a small part of the power of the interfering signal.

- *Multipath interference* : When a code sequence has small autocorrelation, a RAKE receiver can extract the signal energies of all signal components of the multipath.

- *Narrowband interference* : The coherent detection at the receiver involves a multiplication of the received signal by a locally generated code sequence. However, multiplying a narrowband signal with a wideband code sequence spreads the spectrum of the narrowband signal so that its power in the information bandwidth decreases by a factor equal to the processing gain.

We will now review the fundamental elements for understanding DS-CDMA and its application into 3G systems, namely, RAKE receiver, power control, soft handover and multiuser detection.

RAKE Receiver

A spread-spectrum signal waveform is well matched to the multipath channel. In a multipath channel, the original transmitted signal reflects from obstacles such as buildings, and mountains, and the receiver receives several copies of the signal with different delays. If the signals arrive more than one chip apart, the receiver can resolve them. From each multipath signals point of view, other multipath signals can be regarded as interference and they are suppressed by the processing gain. However, a further benefit is obtained if the resolved multipath signals are combined using RAKE receiver. Thus, the signal waveform of CDMA signals facilitates utilization of multipath diversity. Expressing the same phenomenon in the frequency domain means that the bandwidth of the transmitted signal is larger than the coherence bandwidth of the channel and the channel is *frequency selective* (i.e., only part of the signal is affected by the fading).

RAKE receiver consists of a bank of correlators, each receiving a multipath signal components. After despreading by correlators, the signals are combined using, for example, maximal ratio combining. Since the received multipath signals are fading independently, diversity order and thus performance are improved. The RAKE receiver has a receiver finger for each multipath component. In each finger, the received signal is correlated by a spreading code, which is time-aligned with the delay of the multipath signal. After despreading, the correlator outputs are weighted and combined. The combining is usually maximal ratio combining (MRC) wherein each output is weighted by the path gain (attenuation factor). Due to the mobile movement the scattering environment will change, and thus, the delays and attenuation factors will change as well. Therefore, it is necessary to measure the channel profile and to reallocate RAKE fingers as needed.

Power Control

In the uplink of a DS-CDMA system, the requirement for power control is the most serious negative point. The power control problem arises because of the multiple access interference. All users in a DS-CDMA system transmit the messages by using the same bandwidth at the same time and therefore users interfere with one another. If transmit powers were equal, the result of distance attenuation would be that the signal received by the base station from a user terminal close to the base station will be much stronger than the signal received from another terminal located at the cell boundary. Hence, a distant user will be dominated by the nearby user. This is called the *near-far effect*. To counteract the near/far effect, all users transmitting at the same rate to the same base station should arrive at that base station with the same received power. A method to achieve this objective is *power control*, which typically attempts to achieve either a constant received mean power for each user or a target signal to interference ratio (SIR). Therefore, the performance of the transmitter power control (TPC) is one of the several factors in the capacity of a DS-CDMA system. In addition of being useful against interfering users, power control improves the performance of DS-CDMA in a fading channel by compensating the fading dips. If it is possible to track the channel fading perfectly, power control would turn a fading channel into an AWGN channel by eliminating the fading dips completely.

Soft handoff

In non-CDMA cellular systems, a MS performs a handoff when the signal strength of a neighboring cell exceeds the signal strength of the current cell with a given threshold. Since in a CDMA system, the neighboring cell frequencies are the same as in the given cell, this type of approach would cause excessive interference into the neighboring cells and thus a capacity degradation. To reduce this interference, an instantaneous handoff from the current cell to the new cell would be required when the signal strength of the new cell exceeds the signal strength of the current cell. This is not, however, feasible in practice. Instead, CDMA systems have introduced soft handoff, where a MS is connected to more than one BS simultaneously. A MS enters the soft handoff state when the signal strength of neighboring cell exceeds a certain threshold that may be below the current BS's signal strength. Since in soft handoff the MS is connected to either two or more BS, its transmission power can be controlled according to the cell, which the MS receives with the highest signal strength.

The signal structure of CDMA is well suited for the implementation of soft handoff. This is because in the uplink, two or more BS can receive the same signal because of the reuse factor of one. After demodulation and combining, the BSs forward the signal to the combining point, typically the base station controller (BSC). In the downlink the MS can coherently combine the signals from different BS since it sees them as just additional multipath components. A separate pilot channel is usually used for the signal strength measurements for handoff purposes.

In the downlink, however, soft handoff creates more interference to the system since the new BS now transmits an additional signal for the MS. It is possible that the MS cannot catch all the energy that the BS transmits due to a limited number of RAKE fingers. Thus, the gain of soft handoff in the downlink depends on the gain of macro diversity and the loss of performance due to increased interference.

Multiuser detection

The current CDMA receivers are based on the RAKE receiver principle, which considers other users' signals as multiple access interference (MAI). MAI that disturbs a BS or MS is a sum of both intra-cell and inter-cell interference. In an advanced receiver, all signals would be detected jointly or interference from other signals would be removed or suppressed. This is possible when code sequences of interfering users are known (i.e., the interference is deterministic not random).

The joint detection of signals from users to reduce the effect of MAI is called multiuser detection (MUD). In principle, if a MS has a high received power, this MS can be detected first and subtracted from the input signal, so that the other MSs do not see the interference. Since optimal multiuser detection is very complex and in practice impossible to implement for any reasonable number of users, a number of sub-optimum multiuser receivers have been developed. There are a number of multiuser detectors. Linear detectors apply a linear transform into the outputs of the matched filters in order to remove or reduce the multiple access interference (i.e., the interference due to correlations between user codes). Examples of linear detectors are decorrelator [19] and linear minimum mean square error (LMMSE) detectors [20]. In interference cancellation, MAI is first estimated and then subtracted from the received signal. Interference cancellers that use soft outputs are generally linear detectors. When interference cancellation involves hard decisions, the resulting detector is non-linear. Parallel interference cancellation (PIC) [40] and successive (serial) interference cancellation (SIC) [27] are examples of interference cancellation which are well compared in [21].

3.2 WCDMA System Model

Wideband code division multiple access (WCDMA) system is the air interface for multiple access scheme for IMT-2000/UMTS. Emerging requirements for higher rate data services and better spectrum efficiency are the main drivers identified for the third-generation mobile radio systems.

Wideband CDMA has a bandwidth of 5 MHz which is the nominal bandwidth for all 3G proposals. A much more detailed description of CDMA and specifically WCDMA is given in [23]. Larger bandwidths of 10, 15, and 20 MHz have been proposed to support higher data rates more effectively. The main reasons for choosing the 5 MHz bandwidth are:

- Within 5 MHz bandwidth, it is possible to support a significant number of connections at data rates of 144 and 384 Kb/s, the main targets of 3G systems. Even a 2-Mb/s peak rate can be provided under limited conditions.
- Lack of spectrum calls for reasonably small spectrum allocation, especially if the system has to be deployed within the existing frequency bands occupied already by 2G systems.
- The 5-MHz bandwidth can resolve (separate) more multipath components than

Property	*Description*
Channel bandwidth	1.25, 5, 10, 20 MHz
Downlink RF channel structure	Direct spread
Inter-cell operation	Asynchronous (sync possible)
Chip rate	0.96, 3.840, 7.68, 15.36 Mc/s
Frame length	10 ms
Spreading modulation	Balanced QPSK (downlink) Dual channel QPSK (uplink) Complex spreading circuit
Data modulation	QPSK (downlink) BPSK (uplink)
Channel coding	Voice: convolutional (R=1/3, K=9) Data: concatenation of convolutional and RS
Diversity	Rake and Antenna
Coherent detection	User dedicated time or I/Q MUX pilot (up and down) No common pilot in downlink
Channel multiplexing in uplink	Control and pilot channel time multiplexed I/Q multiplexing for data and control channel
Multirate	Variable spreading and multicode
Spreading factors	4–256
Power control	Open and fast closed loop (1.5 kHz)
Spreading (downlink)	Variable length orthogonal sequences for channel sep.; Gold sequences 2^{18-1} for cell and user separation (truncated cycle 10ms)
Spreading (uplink)	Variable length orthogonal sequences for channel sep.; Gold sequences 2^{41-1} for cell and user separation with different time shifts in I and Q channel (truncated cycle 10ms)

Table 3.1: WCDMA 3GPP parameter specification

narrower bandwidths, increasing diversity and thus improving performance.

The important concept of WCDMA is the introduction of intercell asynchronous operation and the pilot channel associated with each data channel. The pilot channel makes coherent detection possible on the reverse link. Furthermore, it makes it possible to adopt interference cancellation and adaptive antenna array techniques. Table 3.1 gives a few parameters of WCDMA operation. The other technical features of WCDMA are summarized below:

- Fast cell search under asynchronous operation.
- Coherent spreading-code tracking.
- Fast transmit power control (TPC) on both the reverse and forward links.
- Coherent Rake reception on both links.
- Orthogonal multiple spreading factors on the forward link.
- Variable rate transmission with blind rate detection.
- Provision of multirate services.
- Packet switched data.
- Seamless interfrequency handoff
- Optional multiuser detection.

We will restrict our discussion to those physical aspects of WCDMA that are relevant to WiPPET$_{\text{signal}}$. The basic physical resource is the code/frequency plane. In addition, on the uplink, different information streams may be transmitted on the I and Q branches. Consequently, a physical channel corresponds to a specific carrier frequency, code, and, on the uplink, relative phase (0 or $\pi/2$). WCDMA physical channels can be classified as *dedicated* or *common*. There are dedicated and common channels for both the *uplink (reverse link)* and the *downlink (forward link)*. We now discuss the relevant channels.

Uplink Dedicated Physical Channels

There are two types of uplink dedicated physical channels, the uplink Dedicated Physi-

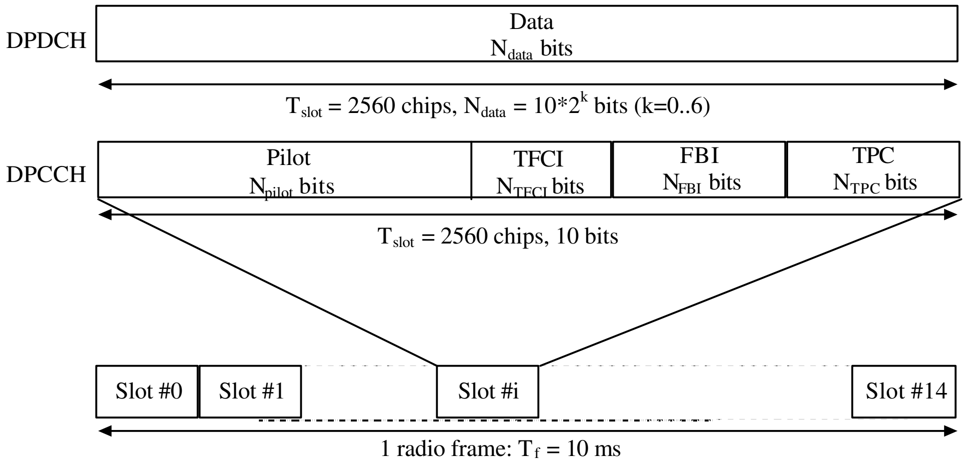


Figure 3.1: Uplink DPDCH/DPCCH frame format

cal Data Channel (uplink DPDCH) and the uplink Dedicated Physical Control Channel (uplink DPCCH). The DPDCH and the DPCCH are I/Q code multiplexed within each radio frame. The uplink DPDCH is used to carry the DCH transport channel. There may be zero, one, or several uplink DPDCHs on each radio link. The uplink DPCCH is used to carry control information generated at Layer 1. The Layer 1 control information consists of known pilot bits to support channel estimation for coherent detection, transmit power-control (TPC) commands, and some other information about instantaneous frame formats and feedback. There is one and only one uplink DPCCH on each radio link. Figure 3.1 shows the frame structure of the uplink dedicated physical channels. Each radio frame of length 10 ms is split into 15 slots, each of length $T_{slot} = 2560$ chips, corresponding to one power-control period. The parameter k in Figure 3.1 determines the number of bits per uplink DPDCH slot. It is related to the spreading factor SF of the DPDCH as $SF = 256/2^k$. The DPDCH spreading factor may range from 256 down to 4. The spreading factor of the uplink DPCCH is always equal to 256, i.e. there are 10 bits per uplink DPCCH slot. The exact number of bits of the uplink DPDCH and the different uplink DPCCH fields are configured by higher layers. Multi-code operation

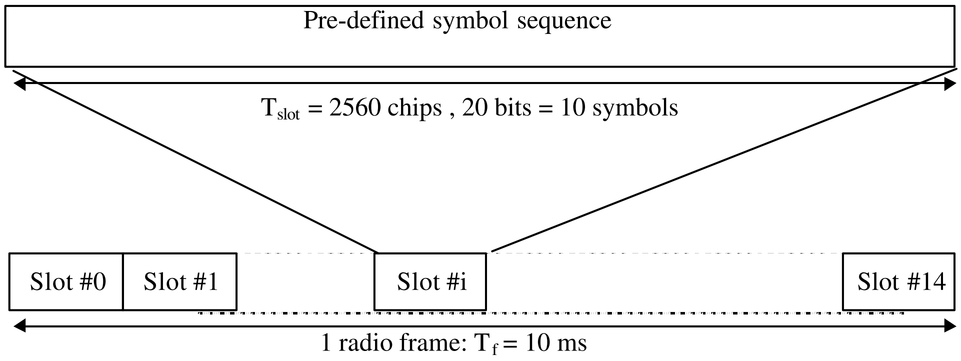


Figure 3.2: Downlink CPICH frame format

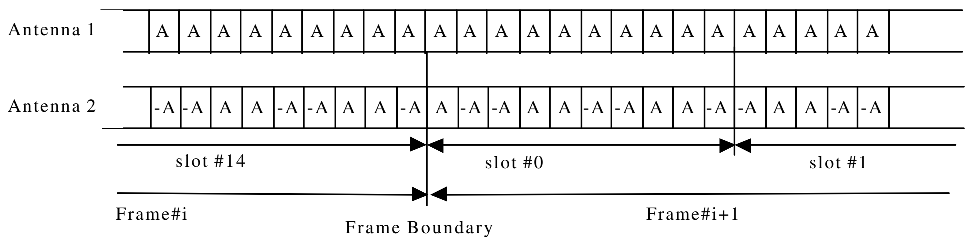


Figure 3.3: Diversity dependent data for Pilots

for the uplink dedicated physical channels is possible by transmitting several parallel DPDCH using different channelization codes.

Downlink Common Physical Pilot Channel

The Common Pilot Channel (CPICH) is a fixed rate (30 kbps, SF=256) downlink physical channel that carries a pre-defined bit/symbol sequence. Figure 3.2 shows the frame structure of the CPICH. In case transmit diversity is used on any downlink channel in the cell, the CPICH shall be transmitted from both antennas using the same channelization and scrambling code. In this case, the pre-defined symbol sequence of the CPICH is different for Antenna 1 and Antenna 2, as illustrated in Figure 3.3. In case of no transmit diversity, the symbol sequence of Antenna 1 in Figure 3.3 is used.

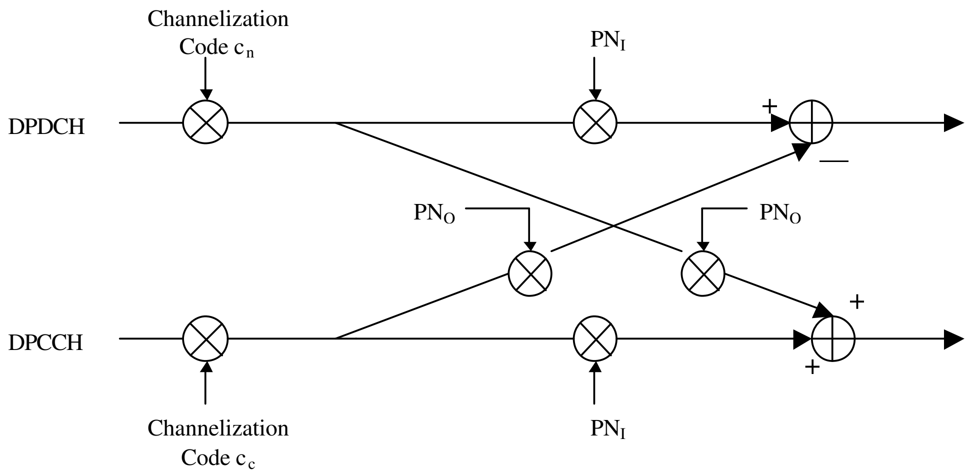


Figure 3.4: The complex spreading circuit is used for IQ/code multiplexing

Complex Spreading

The WCDMA scheme employs long spreading codes for which Gold sequences of length 2^{18-1} and 2^{41-1} are used for downlink and uplink respectively. Both are truncated to form a cycle for a 10-ms frame. For channelization, orthogonal codes are used. Orthogonality between codes with different spreading factors can be achieved by tree-structured orthogonal codes as described in [4].

3.3 WiPPET$_{\text{signal}}$ Requirements

Section 1.1 introduced WINLAB's previous non-parallel and parallel simulators : MADRAS [7] and WiPPET$_{\text{session}}$ and WiPPET$_{\text{packet}}$ [26]. They have been implemented in C and in TeD respectively. WiPPET$_{\text{signal}}$ is is the newest simulator in this series. All simulators share the same characteristics, though the granularity, implementation and points of interests and approach are somewhat different in each. This section introduces components of a virtual signal processing wireless testbed:

1. Geography and Mobility

2. Radio Propagation and Interference

3. Protocols and Algorithms

A signal processing testbed needs to have an efficient implementation of all the above crucial tasks at the finest possible granularity. Once all the fundamental tasks have been implemented, researchers can understand the behaviour and working of the system in various scenarios by changing the parameters.

3.3.1 Geography and Mobility

The motion of the mobile station is of crucial importance to the simulation setup. Mobility is a process describing how and how often a mobile station moves from one point to another within a given geography model. To introduce mobility, a wireless simulator must include a description of the terrain over which the mobile can travel. The set of possible mobile locations in the terrain is the geography on which the simulation will be performed. It is also imperative for the simulation to incorporate geography because the geometry of the terrain determines gross radio propagation characteristics. This way, mobility is connected to geography as well as to radio propagation characteristics of the geography.

WiPPET$_{\text{signal}}$ allows the modeler to import an approximation to a real map, or a synthetic geography. The radio propagation characteristics are precomputed and later incorporated into the simulation. The mobility module is explained in further detail in Section 4.6.

3.3.2 Radio Propagation and Interference

It is imperative that the model of the air channel be capable of simulating phenomena that occur in a typical cellular system. The performance of radio receivers is primarily effected by the propagation and interference effects in the air. Thus, it becomes necessary that in order to properly validate the design of a transceiver, the signal that is arriving at the receiver is put through the rigors of a real channel. This section describes a few of these the air channel phenomena.

Radio reception quality can exhibit very large fluctuations in space and time. Sources of these fluctuations include changes in distance between receiver and transmitter, and changes in environment or terrain and activity of other transmitters. Together these physical conditions produce five main effects:

1. *Distance loss*: signal attenuation that increases with distance
2. *Shadow fading*: signal variations due to environmental features (hills, buildings).
3. *Fast fading*: signal fluctuation due to the relative motion between transmitter and receiver.
4. *Multipath*: signal distortion due to components of the signal traveling along different paths from a transmitter to a receiver.
5. *Interference*: activity of other transmitters sharing the same radio channel, causing an additive disturbance to the received signal of a desired user.

It is necessary to model these spatial and temporal variations in the radio channel for the wireless network simulator to estimate accurately the performance of a wireless system.

There is a vast body of literature on the measurements and modeling of propagation in various environments at various frequencies; for a review, see [15, 30]. The classical models are derived as follows:

For antenna i of transmitter x, and antenna j of receiver y, multipath component l has an associated channel gain $G_{x,i,y,j,l}$ and propagation delay $\tau_{x,i,y,j,l}$. When transmitter x sends signal $s_x(t)$, the corresponding received signal component at antenna j of receiver y is given by

$$r_{x,y,j}(t) = \sum_{i=1}^{A_t} \sum_{l=1}^{L} G_{x,i,y,j,l}(t) s_x(t - \tau_{x,i,y,j,l}) + \mathcal{N}(t) \tag{3.1}$$

where $\mathcal{N}(t)$ is additive white Gaussian noise and A_t and L are the number of transmitting antennas of x, and the number of multi-paths respectively. For a single transmitter antenna and a single receiver antenna, the complex baseband received signal is

$$r_y(t) = \sum_{l=0}^{L-1} r_{x,y,l}(t) + \sum_{x' \neq x} \sum_{l=0}^{L-1} r_{x',y,l}(t) + \omega(t) \tag{3.2}$$

where $r_{x,y,l}(t)$ is the received signal on the lth path for the desired user x and all other users $(x' \neq x)$ and AWGN $\omega(t)$ cause the multi-access interference and noise. The corresponding received power at receiver y for transmitter x is

$$P_{x,y,t} = E[r^2_{x,y}(t)] \tag{3.3}$$

where

$$r_{x,y} = \sum_{l=0}^{L-1} r_{x,y,l}(t) \tag{3.4}$$

A measure of the received signal quality of transmitter x at receiver y is the *signal to interference ratio* (SIR), which is defined by the ratio

$$\Gamma_{x,y}(t) = \frac{P_{x,y}(t)}{\sum_{x' \neq x} P_{x',y}(t) + \eta^2} \tag{3.5}$$

where η^2 is the additive noise power.

The modeling of the channel response $G_{x,i,y,j,l}$ depends on performance assessment needs: it may be site-specific [31], that is based on actual measurements or it may employ stochastic processes with values and correlations mimicking those measured in a class of similar environments (e.g. typical American suburban macrocell, urban micro-cell, or highway). While the design of WiPPET$_{\text{signal}}$ allows the use of site-specific propagation models, in our research we focus on statistical characterization of the wireless network, and thus we limit the discussion to stochastic propagation models.

Signal attenuation in a radio channel is commonly represented as the product of long-scale and short-scale fading

$$G_{x,i,y,j,l}(t) = G^{'}_{x,i,y,j,l}(t) G^{''}_{x,y}(t) \tag{3.6}$$

where the short-scale factor $G^{'}_{x,i,y,j,l}(t)$ represents time-varying effects of interference and delay dispersion of carrier waves arriving over distinct paths. The long-scale or shadow fading, $G^{''}_{x,y}(t)$, represents the time-varying attenuation of the signal averaged due to distance and shadowing by large objects. $G^{''}_{x,y}$ is often not a function of t.

Equation (3.3) represents the average signal strength, but the power measured at a specific location can be is generally above or below this average. As the receiver moves, it observes that the local average signal slowly fluctuates around a mean value given by equation (3.6) occurring over distance of tens of meters. This phenomenon is referred to as *shadow fading* or *slow fading* or *long-scale fading.* Long-scale fading is treated as constant in time and can be precomputed.

It is important to note that in a simulation the details of the fading model should be matched to the signal processing capabilities of the modeled receiver to obtain meaningful statistics for data loss. WiPPET$_{\text{signal}}$ is implemented with distance loss and shadow fading for chip level simulation.

Distance loss and Shadow fading

The classic stochastic model presents the geographical path losses as

$$G''_{x,y}(t) = G''^{d}_{x,y}(|x-y|)G''^{s}_{x,y} \tag{3.7}$$

where $G''^{d}_{x,y}$ is deterministic function of distance and $G''^{s}_{x,y}$ is a log-normally distributed spatial random process.

Measured $G''^{d}_{x,y}(|x-y|)$ decays as an inverse power of distance, i.e. $G''^{d}_{x,y} \sim |x-y|^{-\alpha}$, where $1 < \alpha \leq 4$ [30]. The propagation exponent $\alpha = 2$ models a *free space propagation environment.* In wireless networks, $\alpha \geq 2$ with the precise value dependent on terrain and other environment factors such as buildings. In urban areas, $\alpha = 4$ is often used [30]. However, it is also possible to find $\alpha < 2$ as a result of wave-guide effects that appear in indoor corridors and in cities when antenna heights are substantially lower than surrounding buildings.

A standard stochastic description of the shadow fading fluctuations due to large obstacles is modeled by a single realization of a two dimensional log-normal random process, $\log G''^{s}_{x,y} \sim \mathcal{N}(0, \sigma^2)$, with exponentially decaying spatial correlations [13] as

$$E[\log G''^{s}_{x,y} \log G''^{s}_{x',y}] = \sigma^2 e^{(\frac{-|x-x'|}{d_0})} \tag{3.8}$$

For WCDMA, WIPPET$_{\mathrm{signal}}$ employs the distance and slow fading represented by path gain matrix $G''(a,b) = G''(x_a, y_b)$ for all $x_a \in \mathcal{M}$ and $y_b \in \mathcal{Y}$, where $\mathcal{M}$ is the set of mobile locations,and $\mathcal{Y}$ is the set of base station locations. In WIPPET$_{\mathrm{signal}}$ the $|\mathcal{M}| \times |\mathcal{Y}|$ gain matrix G'' is a large state constant that may be read from a file or precomputed in a site-specific or stochastic propagation model.

Fast fading

The collection of radio waves that arrives at the receiver from different directions, having traversed distinct reflecting paths from a transmitter, is called *multipath*. The signal arrives at a receiver as a multipath signal or a collection of signal components coming from the different directions with different delays, with different attenuation. As the relative speed between transmitter and receiver changes, each ray undergoes a *Doppler shift* causing the apparent wavelength of the signal to either increase or decrease. Radio waves arriving at right angles to the direction of the motion experience no shift in wavelength. The different Doppler shifts and delays in the many radio waves arriving at the receiver causes the radio waves to arrive with different relative phase. At some locations, the radio waves add or cancel each other and thus the received signal becomes relatively stronger or weaker respectively. These fluctuations occur at a one wavelength scale which is much smaller spatial scale than fluctuations due to *shadow fading*. For example, in an 850 MHz cellular radio band significant changes occur over a distance of the order 30 cm, which is the wavelength of the carrier frequency. In a cellular system with a relative speed at 50 km/hr, the signal exhibits approximately 40 peaks and nulls per second [12]. This effect is called *fast fading* or *short-scale fading*.

The term short-scale fading encompasses the effects of the interference of the components of multipath propagation of radio waves, and their varying travel time from the source to the receiver. Sklar [37] provides a useful taxonomy: First, the relation between the maximum excess delay T_m in impulse response (due to multipath propagation) and the symbol duration time T_s determines two signal degradation categories: i) **Frequency–selective fading** for $T_m > T_s$, producing channel-induced inter-symbol interference (ISI); ii)**Flat fading** for $T_m < T_s$, when multipath components are not resolvable. Second, the relation between the channel coherence time T_0 (characterizing temporal variability due to changing multipath components) and the symbol duration time T_s determines two other signal degradation categories: i) **Fast fading** for $T_0 < T_s$, producing symbol distortion, e.g. due to high Doppler; ii) **Slow fading** for $T_0 > T_s$, SIR fluctuations.

Modeling of frequency-selective fading, e.g., for a RAKE receiver, employs temporal fading (like Rayleigh) separately on each resolved path, together with a stochastic process of arrivals and departures of multipath components, with appropriate delay and power distribution. Such models have been developed for both for urban and indoor environments [13, 30]. There are also various models of temporal fading: Rayleigh and Rice (LOS) are typical. With these, temporal variation appears through relative motion of the transmitter (Tx) and receiver (Rx). However, omnipresent environmental motion introduces temporal fading even if both Tx and Rx are stationary, but that case is difficult to characterize because it is highly environment dependent. The case in which relative motion is the larger effect is characterized by Rayleigh and Ricean fading.

In contrast to long–scale fading, modeling of short–scale fading need not be geographically indexed due to short–range correlations. In WiPPET$_{\text{signal}}$ the stochastically generated short–scale temporal fading is modeled independently for each Tx–Rx pair in the shared radio channel and combined at the receiver. WINLAB has implemented the models of short–scale Rayleigh and Ricean fading [10] with i) wave-superposition method, ii) spectrum-shaping method; and also iii) a quantized channel state Markovian model [6].

3.3.3 Protocols and Algorithms

Wireless systems require actions at the nodes to provide the required quality of service (QoS). These actions are taken on the basis of decisions that are made according to a certain set or rules or a certain logical procedure. The actions are the *protocols* that are followed and the decisions are made by executing particular *algorithms*. In wireless terminology, *radio resource management* is the broad term that defines the signaling protocols that control call admission, channel allocation, power control and handoffs. The algorithms that precede the actions could be either at the system level or at the link level or a combination of the two.

For WCDMA, WiPPET$_{\text{signal}}$ implements the following algorithms as part of the transceiver:

- SIR Estimation

- Power Control
- Channel Estimation
- Equal Gain Combining (EGC) for diversity reception
- Viterbi Soft decoding
- Coherent Multi-stage Interference Cancellation

Each of the above algorithms is described in detail in subsequent chapters.

Chapter 4

WiPPET$_{\text{signal}}$ Design

Chapter 3 and Section 2.1 are the inputs to the design and working of WiPPET$_{\text{signal}}$. This chapter introduces the design of WiPPET$_{\text{signal}}$ and its implementation in the SSF domain. Each of the module is further investigated and sub-modules for each of them are further examined. The interaction of the modules over the SSF channels is illustrated using message sequence charts in C

4.1 Overview

A wireless network consists of mobile stations (MSs), base stations (BSs), and mobiles switching centers (MSCs). A MS is connected to a BS through the air interface (a radio link). A BS is connected to a MSC through a microwave link or an optical fiber link. A MSC has an interface between the wireless network and other public networks like the Public Switching Telephone Network (PSTN). The MSs, BSs and MSCs take part in various protocols and run the system by interacting with each other. Analogously, WiPPET$_{\text{signal}}$ has the following classes that are extended SSF entities:

- Mobile Station : Class `MOBILE` [1]

[1] The names of WiPPET$_{\text{signal}}$ entities that appear in `typewriter` font match exactly the entity name as they appear in the source code of WiPPET$_{\text{signal}}$.

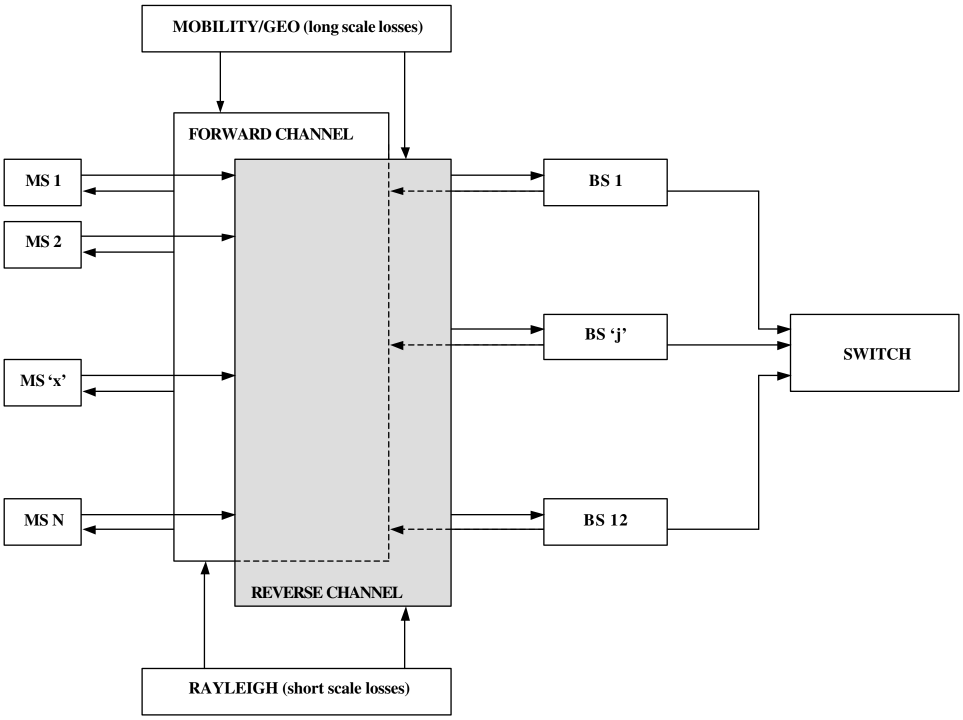


Figure 4.1: WiPPET$_{\text{signal}}$ modules in analogy to the real WCDMA system

- Base Station : Class `BASE`
- Mobile Switching Center : Class `SWITCH`
- Rayleigh Fading : Class `RAYLEIGH`
- Mobility : Class `MOBILITY`
- Radio Channel : Class `CHANNEL`
- Master : Class `MASTER`

Figure 4.1 shows the entities of WiPPET$_{\text{signal}}$ and their basic connectivity through SSF channels. All the basic and other SSF channels further described in Appendix B.

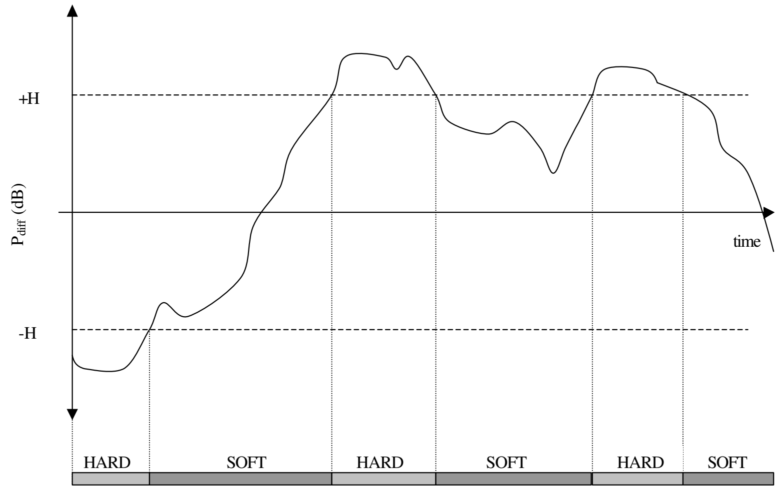


Figure 4.2: Pilot management in MS : This example shows the case for two BS whose power difference is being monitored to decide the soft handoff. A different soft handoff algorithm can also be incorporated into the decision making process

4.2 MOBILE Entity

The mobile station can be regarded as the end user in the WCDMA system. The WiPPET$_{signal}$ MS creates and sends out information bits to the BS where bit detection is done and statistics are created to characterize the performance of the system. It also uses the forward link pilot powers to make decisions on handoff.

Some of the characteristics of the WiPPET$_{signal}$ MS are:

- **Reverse link DPDCH/DPCCH transmission**: The MS creates raw data, pilot data and long codes for each frame. Complex I/Q symbol vectors are sent to the radio channel where they undergo transmission in the presence of interference. Power control commands from the BS are followed so long as the maximum power limit is not crossed. When the limit is reached, the MS does not increase its power even if the power control command indicates a desired increase.

- **Forward link CPICH reception**: The MS receives complex I/Q symbols from the radio channel that have the interfered pilot signals from all the BS. AWGN with appropriate variance is added at each receiver antenna.

- **Pilot Management**: Initially, the MS places the BS with the strongest received pilot tone in the *active* set and all other BSs in the *candidate* set. On the forward link, the BS are continuously sending out the common pilot channel (CPICH). The MS demodulates the CPICH for each BS, and calculate the power level of each BS after every WCDMA slot. This level is averaged over time, and after every WCDMA slot, the MS checks the averaged power levels to make decisions on whether it should initiate soft handoff. Decisions on termination of soft handoff can also be made if the MS is already in soft handoff. At a given time, there is at least one Active BS for a MS. Figure 4.2 illustrates pilot management in the MS. During the time the MS is in soft handoff, the power of the MS is controlled in a slightly different manner than during normal operation.

- **Power Control in soft-handoff**: During a normal state (only one Active BS), only one command arrives at the MS per slot (power control group). The MS complies with the command and increases or decreases transmit power level by some predetermined step. When the MS goes into soft handoff, it now receives more than one command per slot. These commands may or may not be same; the MS has to make an intelligent decision on what it should do about the transmit power. The solution to the problem is very simple. The MS receives and stores all commands from all the active BS. Once it has received all commands, it then checks whether there is at least one command that is asking the MS to reduce the transmit power. If that is the case, the MS reduces its power. This implies that the MS increases its transmit power in the soft handoff state if and only if all active BSs signal the MS for an increase in power.

4.3 `BASE` Entity

The Base Station is the service provider in the WCDMA system. The WiPPET$_{\text{signal}}$ BS receives information signals from all users from which it does bit detection of desired users, and then create statistics like BER.

Some of the characteristics of the WiPPET$_{\text{signal}}$ MS are:

- **Reverse link DPDCH/DPCCH reception**: The MS continuously transmit complex I/Q symbol vectors that represent raw data and dedicated pilot symbols. The BS receives composite signals from all MSs after they have undergone propagation and interference effects. AWGN with appropriate variance is added at each receiver antenna.

- **Forward link CPICH transmission**: Each BS continuously transmits QPSK data modulated pilot signals common for each user. These I/Q complex symbol vectors are sent to the radio channel where they undergo propagation and interference.

- **Connection Management**: The BS needs to maintain the state of the MS. For all the mobiles that are in normal state (this BS is their only active BS), the BS does bit detection makes a hard decision on the data. For MSs that are in soft state, i.e. have more than one active BS, all the active BSs make hard bit detection but don't include that as part of the statistics. The soft data and error rate for these MSs is then forwarded to the switching entity, where the various contributions are either combined and a final hard decision made or the best frame selected.

4.4 SWITCH Entity

When the MS is in a soft-handoff mode, all its active BSs have to forward soft data and the error rate for that WCDMA frame to the switching entity. Once the switch receives all the contributions from all active BSs for a particular MS, it uses, what usually is termed as *macro diversity*. In WiPPET$_{\text{signal}}$, one of the following is used:

- *Selection diversity*: The frame from that active BS is selected as the final frame that has the better BER statistic.
- *Equal gain combining*: The soft data for the frames from all the active BSs is combined using equal gain combining (EGC). Then a final hard decision is made on the bits and corresponding statistics created.

In general, equal gain combining gives better results than selection diversity at the switching entity.

4.5 RAYLEIGH Entity

Short scale (Rayleigh) effects need to be modeled as one of the propagation effects of a cellular system. In WiPPET$_{\text{signal}}$ the Rayleigh Fading for all possible txAnt-path-rxAnt pairs in the system (reverse link and forward link) are calculated in a single `RAYLEIGH` class. We will now discuss the internals of the fading process.

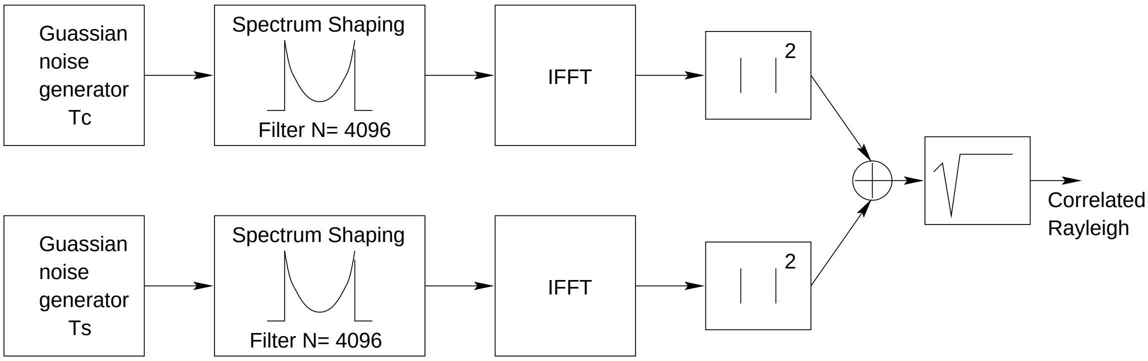


Figure 4.3: Clarke's Model : Implementation using FIR filters

Fader Internals

Clarke's model [10] is one of the best used approaches of implementing a Rayleigh fader. It is based on spectrum shaping method, which we introduce here. We now discuss the approximate Rayleigh fader that we use for WiPPET$_{\text{signal}}$. Section 6.1.1 discusses its accuracy.

Clarke developed a random process model for the received power of random interfering waves. Each incident wave at receiver antenna undergoes a Doppler shift due to the motion of the receiver. In Clarke's model, each incident wave is characterized by amplitude, phase shift, and direction, i.e., the multipath delay is not accounted for. Hence, the model simulates flat fading. Due to the Doppler effect, the received frequency increases if the receiver is advancing into the approaching wavefront and decreases if the receiver is moving away from the wavefront. The maximum Doppler shift is given by

$$f_d = \frac{v f_c}{c} \tag{4.1}$$

where f_d is the maximum Doppler shift, v is the mobile speed, and c is the speed of light in free space. Each arriving wave at the receiver has in-phase and quadrature components T_c and T_s, that are independent stationary Gaussian processes. Assuming angles of arrival to be uniformly distributed from 0 to 2π and assuming an omni-directional receiver, Clarke determined that the power spectral density $S(f)$ of the in-phase and quadrature components are:

$$S_{T_c}(f) = S_{T_s}(f) = \begin{cases} \frac{1}{\pi f_d \sqrt{1-(\frac{f-f_c}{f_d})^2}} & |f - f_c| \leq f_d \\ 0 & |f - f_c| > f_d \end{cases} \tag{4.2}$$

Each component has autocorrelation function ${J_0}^2(2\pi f_d T)$, where J_0 is the zero-order Bessel function of the first kind. Figure 4.3 shows an implementation of the Rayleigh fading generator using the FIR filter (4.2).

Approximate Fader

WiPPET$_{\text{signal}}$ uses an approximation to the filter (4.2), which avoids going through the computation intensive method of calculating the power spectral density (PSD) and then using the inverse fast fourier transform (IFFT) to obtain the correlated Rayleigh value. The approximation is now described. The autocorrelation function for a rayleigh random variable is ${J_0}^2(2\pi f_d T)$, where J_0 is the zero-order Bessel function of the first kind. WiPPET$_{\text{signal}}$ uses an approximation to this function by using the monotonic and periodic properties of negative powers and cosine function respectively. The filtered Gaussian noise method is used to generate the amplitude and phase attenuation. The frequency transform function of the filter is given by

$$G(f) = \frac{1}{\sqrt{\pi}} \cdot \left(f_d^2 - f^2\right)^{-\frac{1}{4}} \quad |f| < f_d \tag{4.3}$$

where f_d is the Doppler frequency shift. The corresponding impulse response function has the closed form as

$$g(t) = \xi \left(2\pi f_d |t|\right)^{-1/4} \cdot J_{\frac{1}{4}} \left(2\pi f_d |t|\right) \tag{4.4}$$

where ξ is the normalizing coefficient so that $\int_{-t}^{t} g^2(\omega)\, d\omega = 1$.

4.6 `MOBILITY` Entity

Long scale (distance attenuation and shadowing) effects also need to be modeled in a cellular system. In WiPPET$_{\text{signal}}$, the distance and shadowing effects for all the possible MS-BS pairs in the system (reverse link and forward link) are modeled in a single `MOBILITY` class. We will now discuss the internals of the mobility module.

A geographical model is required to give a reference to variation in long scale losses for the mobiles. WiPPET$_{\text{signal}}$ contains geographically referenced path loss (distance loss and shadow fading) function $G^{''}(MS_{\text{location}}, BS_{\text{location}})$ via a path loss matrix. This is a two dimensional matrix with one axis representing possible MS locations and other axis representing BS locations in the system. A separate MATLAB program was written to generate the Manhattan geographical model and associated path loss matrix. Figure 4.4 illustrates the output for a 12 cell geography. The program takes distance resolution, correlation distance, variance of shadow fading, etc., as inputs and generates a path loss matrix and stores it in a file. During the simulation parameter initialization, the `MOBILITY` entity reads the file and stores path loss matrix as state information. Manhattan is a special case used in the present work, and `MOBILITY` can be extended to have other types of geographies too, so long as the path loss matrix setup is maintained.

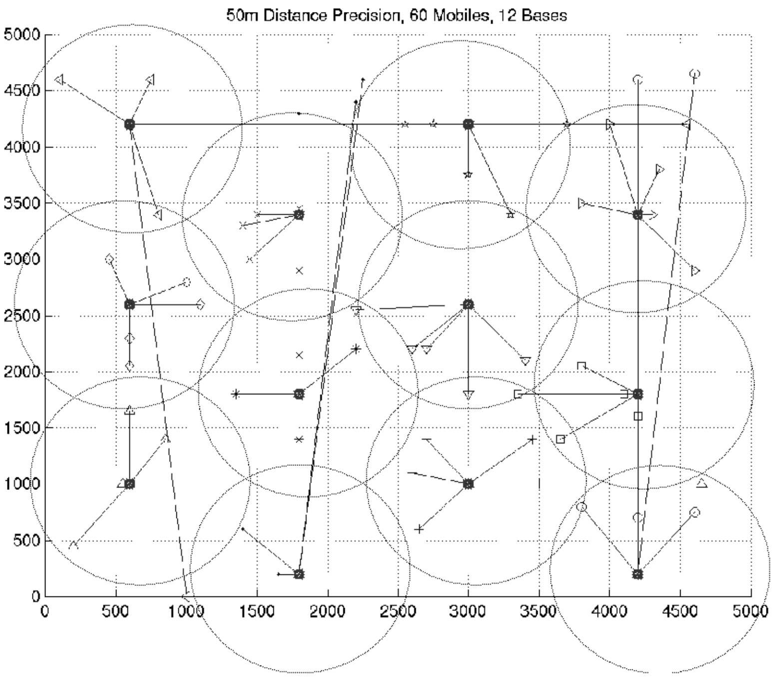


Figure 4.4: Manhattan geography contains streets and avenues. These streets and avenues are modeled as one dimensional lines constructed by discrete points on which mobile station moves step by step

The wrap-around in the model is illustrated in Figure 4.4. It makes sure that there are no boundary mobiles, and hence all BS have interference from all directions. This allows us to collect statistics from all possible BSs rather than just from the central BS which is surrounded by one to two tiers of cells. Figure 4.5 explains the basic implementation of the mobility module. Each point at the corner of a street-avenue cross-section has 4 neighboring points, whereas all other points have 2 neighbors. The frequency of the jumps a mobile makes from one point to another point on the grid is dependent on the speed of the mobile and the precision of the geography.

The simulation results illustrated in Chapter 6 use a Manhattan geography with cells of about 1200 meter diameter. A precision of 0.5 meters is used. For a mobile moving with an average speed of 25 meters/sec, that is about 0.5 meters/20 msecs, which translates to a jump to the neighbouring point every 2 WCDMA frames (each frame is

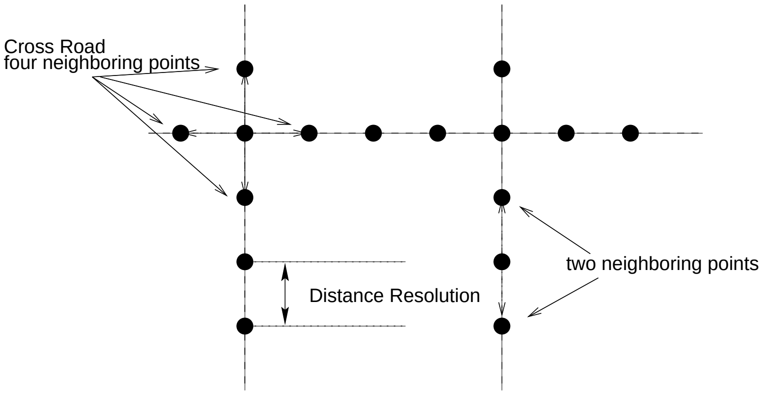


Figure 4.5: Manhattan Street and Avenues : Each point has either 4 or 2 valid neighboring points. The distance between the points depends on the distance precision and the speed of the mobile defines how often a mobile moves from one point to another

of 10 msec duration). Also, the the mobiles in the single-cell or multi-cel environments are uniformly distributed throughout the entire topology. As explained in Section 4.2, the signal power of the received CPICH is measured and averaged after every WCDMA slot. This translates to about 30 periods of measurement per mobility jump for a mobile (each WCDMA frame has 15 slots).

4.7 `CHANNEL` Entity

In WiPPET$_{\text{signal}}$, the air channel is modeled in the `CHANNEL` entity, which is closely coupled with the `RAYLEIGH` and `MOBILITY` entities. We will now discuss how the channel incorporates propagation and interference effects.

The radio channels incorporate waveform level simulations. Waveforms are modeled by a sampled time system with sampling interval Δ. In general, T_c, the duration of one CDMA chip, is a multiple of Δ; however, to reduce the computational requirements, our experiments have employed one sample per CDMA chip. Further, time offsets between transmitted signals are always an integer number of chips. Similarly, the multipath delays of each radio channel and the asynchronism of each base station are always an integer number of chips. Thus, for our purposes, a waveform of duration NT_c is simply a complex I/Q vector of length N.

Let b_x be the symbol that transmitter x has to transmit over antenna i to a receiver antenna j of receiver y over a multipath component l. Let the code sequence for this signal be

$$\mathbf{c}_x = [c_x(0)\ c_x(T_c)\ c_x(2T_c) \ldots\ c_x((N-1)T_c)] \tag{4.5}$$

where $c_x(t)$ is the individual chip. The signal is spread as

$$\mathbf{s}_x = b_x \mathbf{c}_x \tag{4.6}$$

Let the signal path (x, i, y, j, l) have delay $\tau_{x,i,y,j,l}$ and channel gain $\mathbf{G}_{x,i,y,j,l}$ given by

$$\mathbf{G}_{x,i,y,j,l} = [g_{x,i,y,j,l}(0)\ g_{x,i,y,j,l}(T_c)\ g_{x,i,y,j,l}(2T_c) \ldots\ g_{x,i,y,j,l}((N-1)T_c)] \tag{4.7}$$

where $g_{x,i,y,j,l}(t)$ is the channel gain for each individual chip in the most general case where the Rayleigh fading value for each chip is different. For a single path channel, the received chip is given by

$$r(nT_c) = s_x(nT_c - \tau_{x,i,y,j,l}) g_{x,i,y,j,l}(nT_c) \tag{4.8}$$

and the received vector is given by

$$\mathbf{r} = [r(0)\ r(T_c)\ r(2T_c) \ldots\ r((N-1)T_c)] \tag{4.9}$$

For multiple senders, the received signal vector at the receiver antenna is calculated by adding individual chip contributions after the channel gains have been incorporated. In particular, for one chip we must calculate

$$r_{y,j}(nT_c) = \sum_{x=1}^{K} \sum_{i=1}^{T_x} \sum_{l=1}^{L} s_x(nT_c - \tau_{x,i,y,j,l}) g_{x,i,y,j,l}(nT_c) \tag{4.10}$$

to give

$$\mathbf{r}_{y,j} = [r_{y,j}(0)\ r_{y,j}(T_c)\ r_{y,j}(2T_c) \ldots\ r_{y,j}((N-1)T_c)] \tag{4.11}$$

where $\tau_{x,i,y,j,l}$ is the integer chip associated with signal path (x, i, y, j, l).

The following variables are required for the channel implementation as given by equations (4.10) and (4.11):

- Input I/Q chip sequence from all transmitter antennas
- Independent short scale fading values for all (x, i, y, j, l) signal paths
- Long scale (distance loss and shadow fading) values for all (x, y) pairs
- Delay $\tau_{x,i,y,j,l}$ associated with signal path (x, i, y, j, l)

Improvements in the speed of the channel calculations are discussed in Chapter 5.

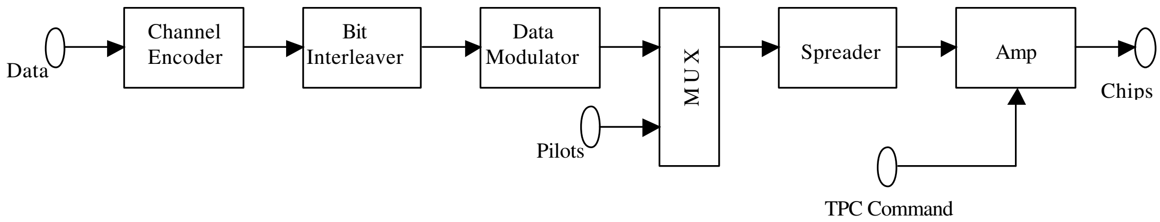


Figure 4.6: MS Transmission: The various blocks in are part of the transceiver that combine to form the end-to-end reverse link transmitter in the MS

4.8 `MASTER` Entity

In WiPPET$_{signal}$ the `MASTER` class encapsulates all the other classes and maps them to each other. It is also responsible for initiating the simulation and collecting statistics.

4.9 Transceiver Functions: the `sCLIENTEXT` Class

The transceiver functionality of the `MOBILE`, `BASE` and `SWITCH` classes is encapsulated in the `sCLIENTEXT` class. The name **CLIENT** has been used to emphasize the fact that any WCDMA transceiver design from an external client can be validated using WiPPET$_{signal}$. In this section, we will discuss some basic characteristics of the NTT DoCoMo transceiver design. This does not mean that WiPPET$_{signal}$ is incompatible with other WCDMA transceiver designs. The DoCoMo modules have only been used to elucidate the working of the `sCLIENTEXT` class.

Figure 4.6 illustrates the main blocks in the single rate reverse link transmitter in the MS. The MS creates a frame of data and then forwards it to the convolutional encoder and then onto the block interleaver. The data is then BPSK modulated and I and Q symbols are created. Pilot I and Q symbols are time multiplexed and a WCDMA frame of symbols is created. Then each symbol is further spread with long codes that have

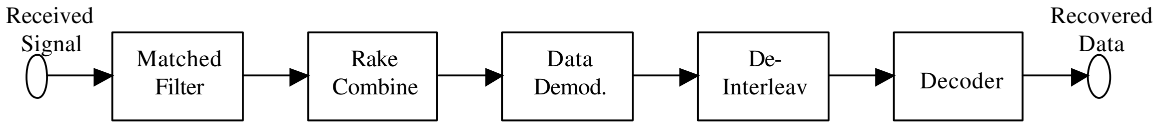


Figure 4.7: Basic BS reception blocks: The various blocks in are part of the transceiver that combine to form the end-to-end reverse link receiver in the BS

been generated using the reverse link polynomial. All MSs are initialized with different seeds for the polynomials which makes sure that no two MS have the same long code for a particular symbol.

Figure 4.7 illustrates the main blocks in the single rate reverse link receiver in the BS. The BS receives the composite signal in form of I and Q chips. For each user, the BS despreads and RAKE combines the signal from each MS by using a simple Matched Filter or a complicated interference canceller (WiPPET$_{\text{signal}}$ uses the COMSIC). Then the data is demodulated and the complementary modules for FEC (deinterleaving and soft Viterbi decoding) are used.

WiPPET$_{\text{signal}}$ employs CPICH on the forward link so that the MS can make decisions on initiating or terminating a soft handoff. The BS uses QPSK modulation on the pilots that are later spread using the forward scrambling code. At the MS, the composite signal is received and each of the BS signal is recreated by using a Matched Filter. Then the power for each BS is calculated on a slot-by-slot basis.

A few of the modules that will be discussed are:

- Forward Error Correction
  - Convolutional Coding and Viterbi Soft Decoding
  - Interleaving and Deinterleaving

- Spreading and Despreading
- Channel Estimation
- SIR Estimation and Power Control
- Coherent Multi-stage Interference Cancellation

4.9.1 Forward Error Correction

The WCDMA system employs Channel Coding and Interleaving as the two main forward error correction methods. WCDMA voice uses convolutional coding whereas data can use turbo coding. In this section, we will discuss the encoding and interleaving that is employed in WiPPET$_{\text{signal}}$. It has to be noted that these techniques are not embedded in the design but can be easily replaced owing to the object oriented character of the `sCLIENTEXT` class.

Convolutional Coding and Viterbi Soft Decoding

A convolutional code is generated by passing the information sequence to be transmitted through a linear finite-state shift register (LFSR). The LFSR consists of K (each with k-bits) stages with n algebraic function generators. The number of output bits for each k bit input sequence is n bits which defines the coding rate R_c of the system as k/n. The parameter K is called the constraint length of the convolutional code. In WCDMA, for voice transmission, a convolutional coding scheme with rate $R_c = 1/3$ and constraint length $K = 9$ is used. Figure 4.8 illustrates the instance of the encoder as provided by the 3GPP [2] specification. The initial value of all the states in the LFSR is 0.

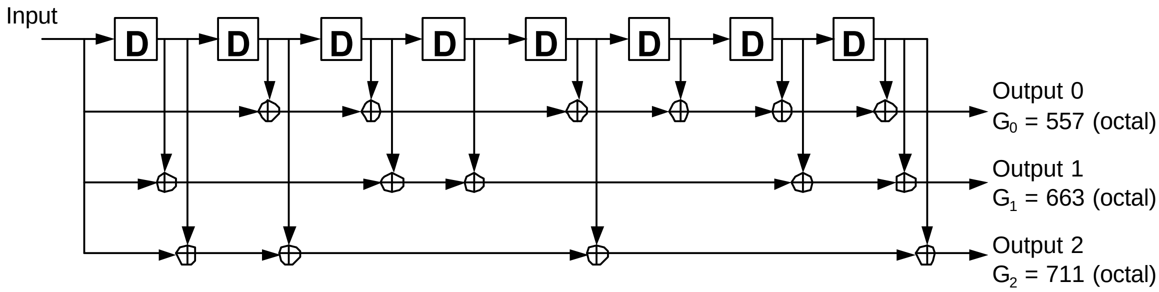


Figure 4.8: Convolutional Encoder in WCDMA and WIPPET$_{\text{signal}}$: The octal codes at each of the outputs represents the algebraic function that is used to generate that particular output

WIPPET$_{\text{signal}}$ employs a rate $R_c = 1/3$ soft Viterbi decoder. The algebraic functions for the LFSR are also configurable and so is the constraint length K which can be 3,5,7 or 9.

Interleaving and Deinterleaving

Most of the well known coding techniques are used to increase the reliability of transmission when the errors caused by the channels are statistically independent. This is the case in AWGN channels. However for a WCDMA system, the presence of multipath and fading causes the channels to exhibit bursty error characteristics.

An effective method of dealing with bursty error channels is to interleave the coded data in such a way that the bursty channel is transformed into a channel having independent errors. WIPPET$_{\text{signal}}$ employs the block interleaving structure. A block interleaver formats the encoded data into a rectangular array of m rows and n columns. These variables (m, n) are also referred as the *horizontal depth* and the *vertical depth* of the block interleaver. The simplest block interleaver that is employed in WIPPET$_{\text{signal}}$ is illustrated in Figure 4.9. The bits are read out column wise and sent for further processing. The corresponding deinterleaving process at the receiver end stores the

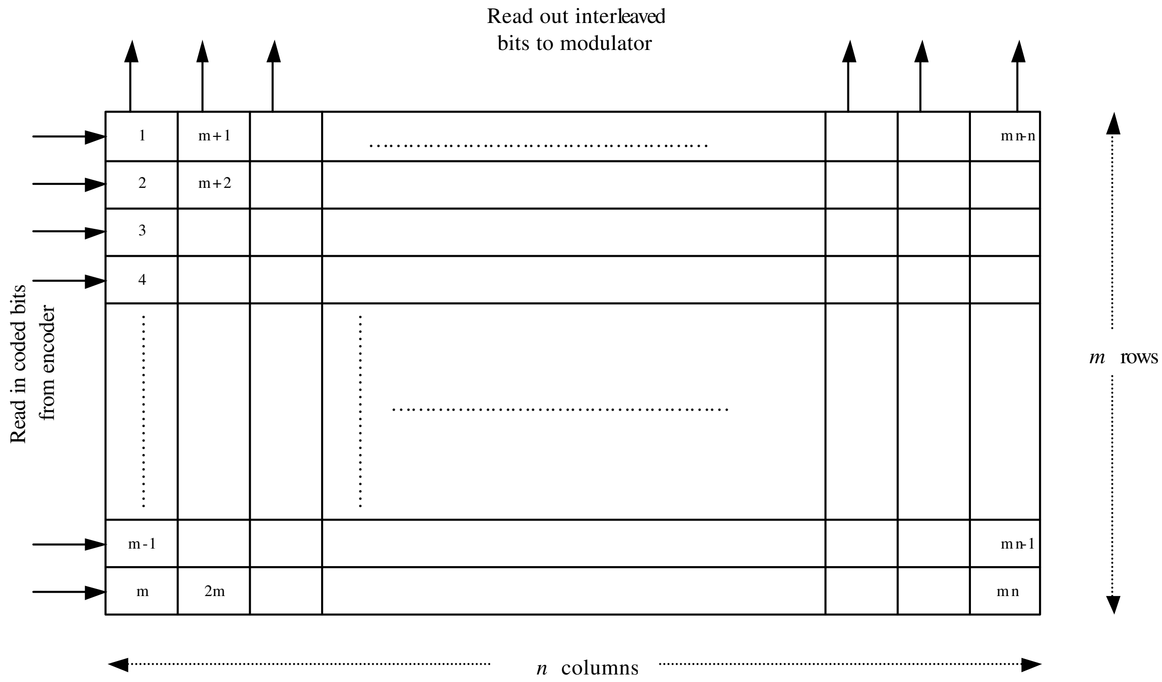


Figure 4.9: Block Interleaver in WCDMA and WiPPET$_{\text{signal}}$

data in the same rectangular array fashion, but data is read by rows. As a result, a burst of errors of length $l = mb$ is broken into m bursts of length b each.

The interleaving and deinterleaving employed in WiPPET$_{\text{signal}}$ has configurable horizontal and vertical depths to correspond to different slot and frame sizes.

4.9.2 Spreading and Despreading

Spreading increases the bandwidth of the signal that is to be transmitted. It is done in the complex domain and different polynomials are used to create these spreading codes.

Reverse link DPDCH/DPCCH spreading

Spreading is applied to the physical channels in two steps. The first step is the *Channelization* operation which transforms the data symbol into a number of chips. The

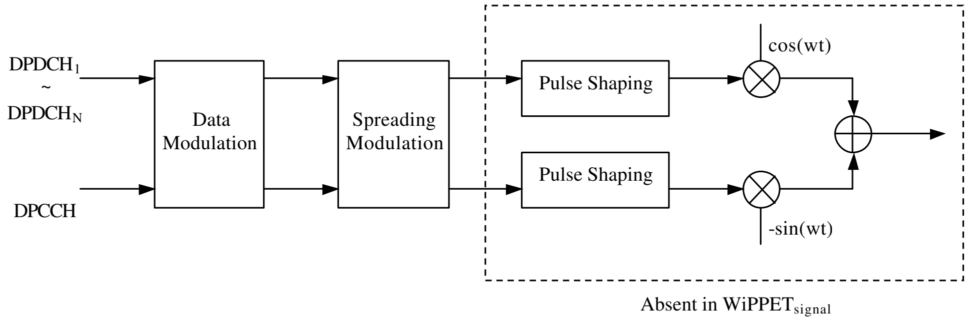


Figure 4.10: Modulation and Spreading in WCDMA

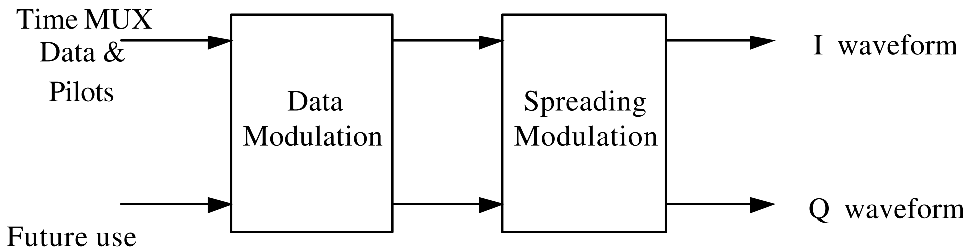


Figure 4.11: Modulation and Spreading in WiPPET$_{\text{signal}}$

channelization codes are Orthogonal Variable Spreading Factor (OVSF) codes and are used to preserve the orthogonality between a user's different physical channels. The next step is the *Scrambling* operation in which the resultant I and Q branches of the first step are further multiplied by a scrambling code. Figure 4.10 gives an illustration of the operation.

We will restrict our discussion to only DPDCH/DPCCH transmissions from the MS. The current version of WiPPET$_{\text{signal}}$ is a single rate system, with only one DPDCH and DPCCH being generated from a MS. Thus, the OVSF codes used in the first step of spreading are not utilized. Instead, only a single stage of spreading is used. Also, the time multiplexed scheme for pilot and data is used on the I channel and the Q channel is used for future use. The spreading modulation only does a scrambling operation. Figure 4.11 gives an illustration of the operation.

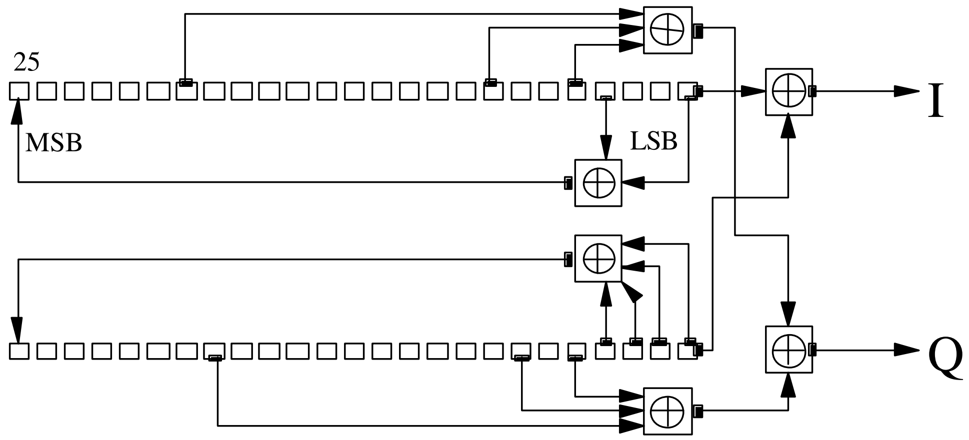


Figure 4.12: Reverse link scrambling generator in WCDMA

The scrambling codes (also called long codes) are the position wise modulo 2 sum of 38400 chip segments of two binary m-sequences generated by means of two polynomials of degree 25 [3]. Let x and y be the two m-sequence respectively. The x sequence is constructed using the primitive polynomial $X^{25} + X^3 + 1$. The y sequence is constructed using the polynomial $X^{25} + X^3 + X^2 + X + 1$. The resulting sequences thus constitute segments of a set of Gold sequences. Figure 4.12 illustrates the generator and polynomials. The quadrature (Q) code is a chip shifted version of the inphase (I) code. The reverse link scrambling code has a period of one radio frame. The binary code words are converted to real valued sequences by transforming 0 to +1 and 1 to -1 respectively.

WiPPET$_{\text{signal}}$ currently employs another version of the long code generator [39]. This is illustrated in Figure 4.13 and uses two polynomials of degree 41. The x and y sequences are constructed using the polynomial $X^{41} + X^3 + 1$ and $X^{41} + X^{20} + 1$ respectively. The current version of WiPPET$_{\text{signal}}$ employs BPSK spreading, so the Q component of the code is set to zero and is not used.

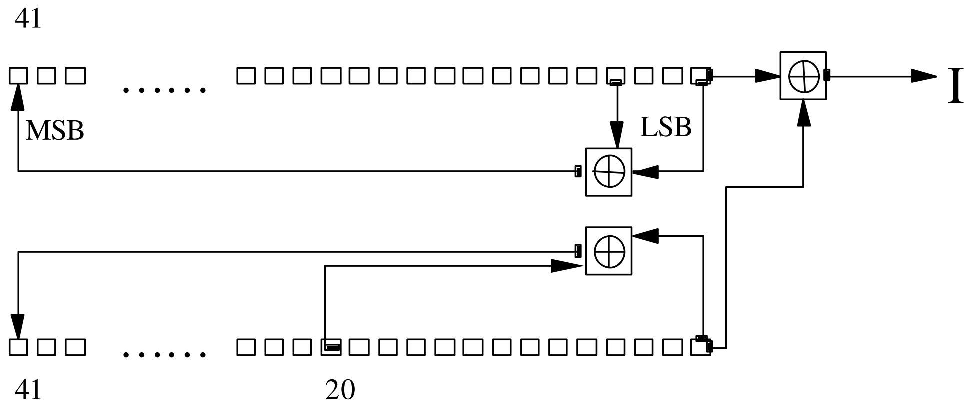


Figure 4.13: Reverse link scrambling generator in WiPPET$_{\text{signal}}$

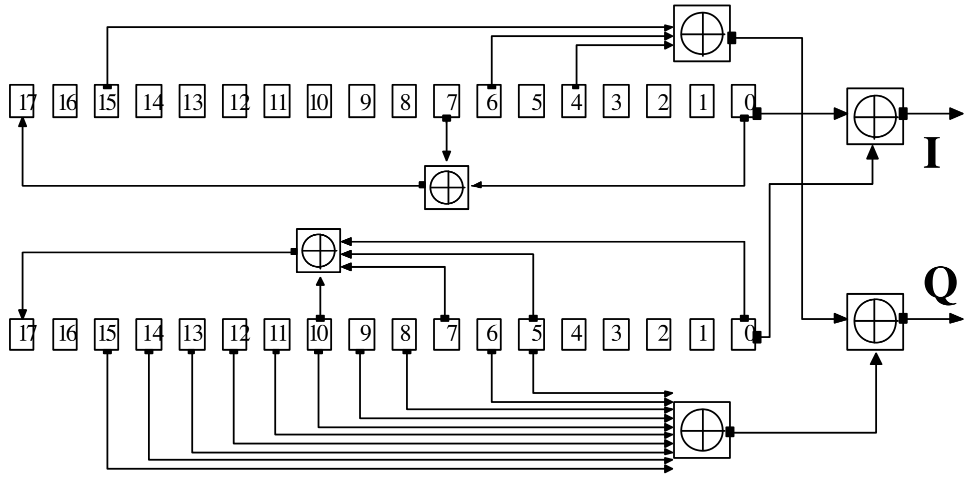


Figure 4.14: Forward link scrambling generator in WCDMA and WiPPET$_{\text{signal}}$

Forward link CPICH spreading

We will restrict our discussion to only CPICH transmissions from the BS. Data modulation is QPSK where each pair of two bits are serial-to-parallel converted and mapped onto the I and Q branches respectively. The I and Q branches are then spread using a scrambling code. This again utilizes the second stage of spreading only since the OVSF codes are not required.

The scrambling codes are the position wise modulo 2 sum of 38400 chip segments of two binary m-sequences generated by means of two polynomials of degree 18 [3] . Let

x and y be the two m-sequence respectively. The x sequence is constructed using the primitive polynomial $X^{18} + X^7 + 1$. The y sequence is constructed using the polynomial $X^{18} + X^{10} + X^7 + X^5 + 1$. The resulting sequences thus constitute segments of a set of Gold sequences. Figure 4.14 illustrates the generator and polynomials. The quadrature (Q) code is a chip shifted version of the inphase (I) code. The reverse link scrambling code has a period of one radio frame. The binary code words are converted to real valued sequences by transforming 0 to +1 and 1 to -1 respectively. Once the long code for one frame is allocated to a BS for the CPICH, it stays the same and is repeated after every frame.

4.9.3 Channel Estimation

Channel estimation filters can be based on linear interpolation, low-order Gaussian interpolation and Wiener filtering [18,32]. However, the Wiener filter requires knowledge of fading channel statistical properties which are difficult to estimate. Furthermore, the Wiener filter can't be applied to a power controlled reverse link as the channel's statistical properties are not preserved. Also, the SNR of the channel estimate of the linear interpolation is not the same over one slot and becomes maximum at the center position. This was pointed out in [14] which also proposed a channel estimation filter that simply averages out the pilot symbols belonging to two consecutive data slots.

WiPPET$_{\text{signal}}$ uses the recently researched channel estimation filter called the Weighted Multi Slot Averaging (WMSA) filter [5]. A slot in WCDMA is defined as the time multiplexed sequence of N_P pilot symbols and N_D data symbols. The observation period of the channel is beyond just one slot, and pilots from more than or equal to two slots are used for channel estimation.

For the mth symbol in the nth slot, the matched filter output for path l of user x at receiver y is given by

$$r_{x,y,l}(m,n) = \sqrt{2P_x(n)}G_{x,y,l}(m,n)b_x(m,n) + w_{x,y,l}(m,n) \tag{4.12}$$

where $P_x(n)$ is the average transmitted power in the nth slot which is constant over the slot, $G_{x,y,l}(m,n)$ is the complex path gain of the lth path, $b_x(m,n)$ is the transmitted symbol and $w_{x,y,l}(m,n)$ is the sum of all the multi-access interference (MAI) including self interference from other paths and background noise. The received despread signal can also be represented as

$$r_{x,y,l}(m,n) = \sqrt{2P_x(n)}\chi_{x,y,l}(m,n)e^{j\phi_{x,y,l}(m,n)} + w_{x,y,l}(m,n) \tag{4.13}$$

where $\chi_{x,y,l}(m,n)$ is the complex path gain and $\phi_{x,y,l}(m,n) \in \{i\pi/2 : i = 0,1,2,3\}$ is the modulation phase of the lth path and the mth symbol in the nth slot. The first N_P symbols are pilot symbols with phase $\phi_{x,y,l}(m,n) = 0$. The value of $\eta_{x,y,l}(n) = \sqrt{2P_x(n)}G_{x,y,l}(m,n)$ is estimated for the slot n by the WMSA channel estimation filter. The weights used to compensate for channel effects are denoted by $\widetilde{\eta}^*_{x,y,l}(n)$, which is the complex conjugate of the channel estimate. The N_P pilot symbols of the slot $[b_x(0,n)\ b_x(1,n)\ldots\ b_x(N_{P-1},n)]$ are all unity. The instantaneous channel estimate $\widetilde{\eta}_{x,y,l}(n)$ for the nth slot is performed using the N_P pilot symbols belonging to the same slot as

$$\widetilde{\eta}_{x,y,l}(n) = \frac{1}{N_P} \sum_{m=0}^{N_{P-1}} r_{x,y,l}(m,n) \tag{4.14}$$

Another version of the WSMA channel estimation filter takes two consecutive slots to estimate the channel. Pilots from each slot are used, and a combined estimate is created for the first slot. The N_P pilot symbols of slot n, $[b_x(0,n)\ b_x(1,n) \ldots\ b_x(N_{P-1},n)]$ are all unity. The N_P pilot symbols of slot $(n+1)$, $[b_x(0,n+1)\ b_x(1,n+1) \ldots\ b_x(N_{P-1},n+1)]$ are also all unity. The instantaneous channel estimate for the nth slot is performed using the N_P pilot symbols belonging to slots n and $(n+1)$ as

$$\widetilde{\eta}_{x,y,l}(n) = \frac{1}{2N_P} \sum_{i=n}^{n+1} \sum_{m=0}^{N_{P-1}} r_{x,y,l}(m,i) \tag{4.15}$$

A detailed analysis of the WSMA filter is given in [5]. WiPPET$_{\text{signal}}$ employs both versions of the WSMA channel estimation filter. During SIR calculations (section 4.9.4), only one slot is used for the estimation whereas during bit detection, two slots are used. This design choice is also discussed in [5].

The coherent RAKE combiner output sample for the symbol m in slot n is given by

$$\widehat{r}_{x,y}(m,n) = \sum_{l=0}^{L-1} r_{x,y,l}(m,n)\widetilde{\eta}^{*}_{x,y,l}(n) \tag{4.16}$$

Figure 4.15 illustrates the block diagram of the coherent RAKE receiver. The matched filter is equivalent to a bank of synchronous correlators, one for each multipath. In order to perform RAKE combining of all the resolved signal sample sequences based on

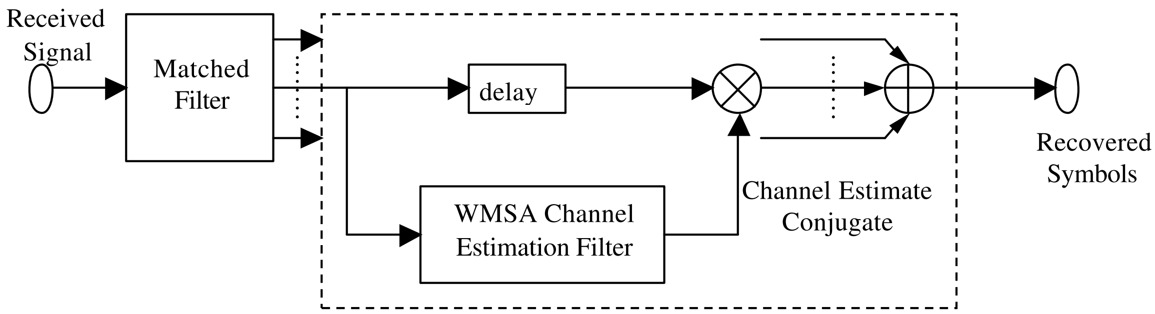


Figure 4.15: RAKE receiver using WMSA channel estimation : The received multipath signal is despread by a matched filter and resolved into L signals

maximal ratio combining (MRC) method, the value of $\eta_{x,y,l}(n)$ is first estimated by the WMSA channel estimation filter. The signal is then compensated by $\widetilde{\eta}^*_{x,y,l}(n)$, which is the complex conjugate of the channel estimate and subsequently combined with other signals to recover the symbols.

4.9.4 SIR Estimation and Power Control

To overcome the power variations due to the distance dependent loss and and shadowing, slow transmit power control (TPC) is used. WCDMA uses RAKE reception and FEC to mitigate the effects of fast multipath Rayleigh fading, but still the instantaneous received power varies significantly. This makes it necessary to implement a fast TPC scheme based on the measured instantaneous signal-to-average interference plus noise (SIR) [36]. The transmit power is adaptively controlled to attempt to keep the instantaneous received SIR at a prescribed target.

WiPPET$_{\text{signal}}$ uses the WSMA channel estimation filter and subsequent RAKE combining by using only one slot which consists of N_P pilot symbols and N_D data symbols in a time-multiplexed mode. One slot is used to reduce the SIR measurement delay. The channel estimation is first performed for the current slot n and the received signal

$r_{x,y,l}(m,n)$ is weighted by the complex conjugate of the channel estimate $\widetilde{\eta}_{x,y,l}(n)$. The reconstructed RAKE combiner output is given by Equation (4.16), and the estimate used is given by Equation (4.14). The estimated SIR $\lambda_{x,y}(n)$ of user x at receiver y for the nth slot is given by

$$\lambda_{x,y}(n) = \frac{\widehat{S}_{x,y}(n)}{\widehat{I}_{x,y}(n)} \tag{4.17}$$

where $\widehat{S}_{x,y}(n)$ is the instantaneous signal power measured over the interval of N_P symbols and $\widehat{I}_{x,y}(n)$ is the average MAI plus noise power after RAKE combining. At the base, the estimated $\lambda_{x,y}(n)$ is compared to the target SIR λ_0. In WIPPET$_{\text{signal}}$ the TPC command is sent to the mobile via an error free SSF channel to lower its transmit power by Δ dB if $\lambda_{x,y}(n) > \lambda_0$ and to raise it by Δ dB otherwise. Figure 4.16 gives an illustration of this algorithm. WIPPET$_{\text{signal}}$ is currently employing a fixed common SIR target for all mobiles in all cells.

The SIR measurement block diagram is shown in Figure 4.17. First, we describe the measurement of $\widehat{S}_{x,y}(n)$. We assume a measurement period of N_M symbols from the beginning of the slot ($N_M \geq N_P$). $\widehat{S}_{x,y}(n)$ is given by

$$\widehat{S}_{x,y}(n) = \left| \frac{1}{N_M} \left(\sum_{m=0}^{N_{P-1}} \widehat{r}_{x,y}(m,n) + \sum_{m=N_P}^{N_{M-1}} r_{x,y}(m,n) \cdot e^{-j\widehat{\phi}_{x,y}(n,m)} \right) \right|^2 \tag{4.18}$$

where $\widehat{r}_{x,y}(m,n)$ is given by equation (4.16) and

$$\widehat{\phi}_{x,y}(n,m) = \arg \max_{\phi \in \{i\pi/2 : i=0,1,2,3\}} Re\left[\widehat{r}_{x,y}(m,n) \cdot e^{-j\phi} \right] \tag{4.19}$$

is the tentatively detected phase of the received QPSK signal. $Re[\cdot]$ denotes the real part

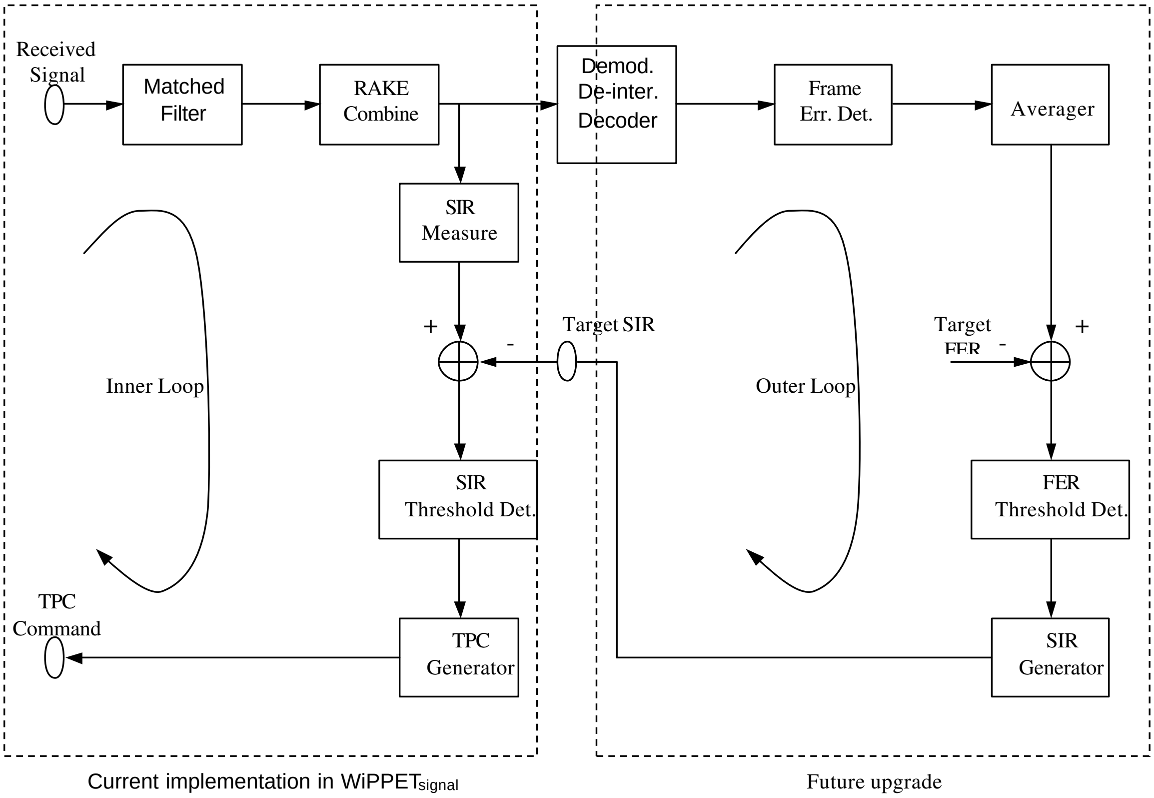


Figure 4.16: Power Control : A fixed target λ_0 is used in WIPPET$_{\text{signal}}$. This inner loop can also be modified to incorporate outer loop power control with variable target SIR

of the complex value. For example, for BPSK data modulation, if $\widehat{r}_{x,y}(m,n) = -1+j0$, then $\widehat{\phi}_{x,y}(n,m) = -\pi$ and $e^{-j\tilde{\phi}_{x,y}(n,m)} = -1$. As discussed earlier, the first N_P symbols are pilot symbols with phase $\phi_{x,y}(m,n) = 0$. As the signal power measurement period N_M increases, SIR measurement error decreases.

The MAI, which includes Gaussian noise, is measured as follows. First, the instantaneous power $\widetilde{I}_{x,y}(n)$ associated with the nth slot is measured using only the pilot symbols belonging to that slot. $\widetilde{I}_{x,y}(n)$ is defined as the average of squared errors of the received pilot signal samples of slot n after RAKE combining. $\widetilde{I}_{x,y}(n)$ is given by

$$\widetilde{I}_{x,y}(n) = \frac{1}{N_P} \sum_{m=0}^{N_{P-1}} \left| r_{x,y}(m,n) - \sum_{l=0}^{L-1} \widetilde{\eta}_{x,y,l}(n) \right|^2 \tag{4.20}$$

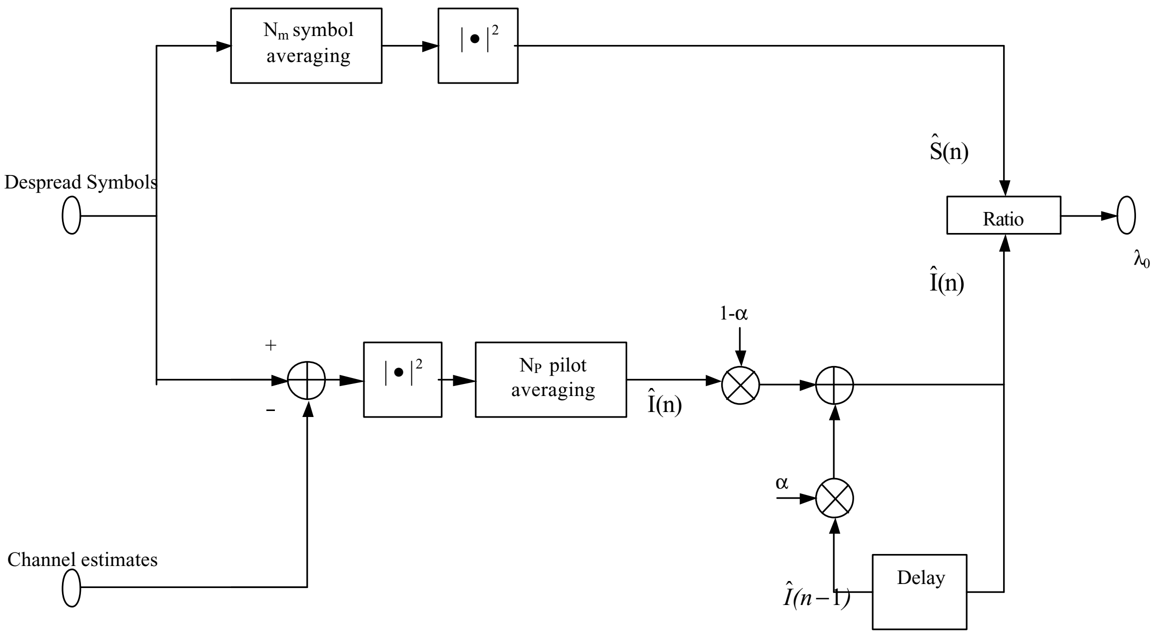


Figure 4.17: SIR Measurement : The signal $\widehat{S}_{x,y}(n)$ and average interference $\widehat{I}_{x,y}(n)$ are calculated using N_M pilots and data and N_P pilots respectively

The following first order linear filter is then applied with forgetting factor $\alpha < 1$ to obtain the average interference plus noise power as:

$$\widehat{I}_{x,y}(n) = \alpha \widetilde{I}_{x,y}(n-1) + (1-\alpha)\widetilde{I}_{x,y}(n) \tag{4.21}$$

The method for calculation of SIR can also be modified to calculate the SIR for each path and then summing it over all paths. The following equations represent this method.

$$\widehat{\phi}_{x,y,l}(n,m) = \arg \max_{\phi \in \{i\pi/2; i=0,1,2,3\}} Re\left[\widehat{r}_{x,y,l}(m,n) \cdot e^{-j\phi}\right] \tag{4.22}$$

$$\widehat{S}_{x,y,l}(n) = \left| \frac{1}{N_M} \left(\sum_{m=0}^{N_{P-1}} \widehat{r}_{x,y,l}(m,n) + \sum_{m=N_P}^{N_{M-1}} r_{x,y,l}(m,n) \cdot e^{-j\widehat{\phi}_{x,y,l}(n,m)} \right) \right|^2 \tag{4.23}$$

$$\widetilde{I}_{x,y,l}(n) = \frac{1}{N_P} \sum_{m=0}^{N_{P-1}} \left| r_{x,y,l}(m,n) - \widetilde{\eta}_{x,y,l}(n) \right|^2 \tag{4.24}$$

$$\widehat{I}_{x,y,l}(n) = \alpha \ \widetilde{I}_{x,y,l}(n-1) + (1-\alpha)\widetilde{I}_{x,y,l}(n) \tag{4.25}$$

$$\lambda_{x,y}(n) = \sum_{l=0}^{L-1} \frac{\widehat{S}_{x,y,l}(n)}{\widehat{I}_{x,y,l}(n)} \tag{4.26}$$

where $\widehat{r}_{x,y,l}(m,n)$ is given by

$$\widehat{r}_{x,y,l}(m,n) = r_{x,y,l}(m,n)\widetilde{\eta}^*_{x,y,l}(n) \tag{4.27}$$

and $\widetilde{\eta}^*_{x,y,l}(n)$ is given by equation (4.14).

In WiPPET$_{\text{signal}}$, simulations the forgetting factor α is set to 0.999; the equivalent measurement interval is 1000 slots = (666.66 ms). The reason for such a long measurement period is that we need to average out the instantaneous variations in the MAI due to TPC, otherwise the TPC operation may become unstable. A qualitative study of SIR-based TPC of reverse link for coherent DS-CDMA systems is done in [36].

4.9.5 Coherent Multistage Interference Cancellation

In interference cancellation, MAI is first estimated and then subtracted from the received signal. In this section we adopt the notation and outline the method of the pilot symbol assisted coherent multistage interference canceller (COMSIC) [35] as proposed by NTT DoCoMo. Figure 4.18 shows the operational flow of the COMSIC using

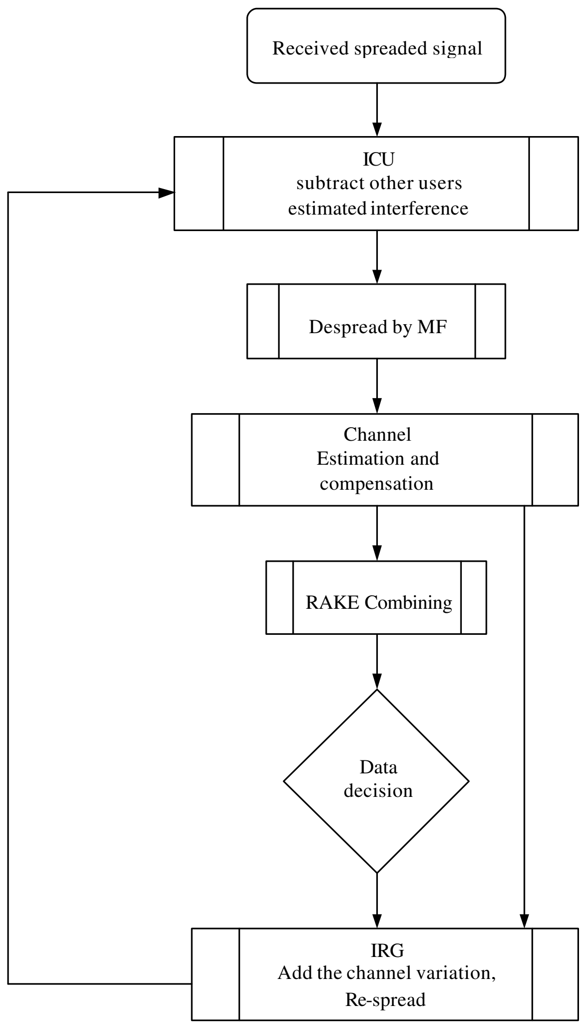


Figure 4.18: COMSIC Flow Chart : The various stages in interference cancellation are shown in the flow-chart. The implementation of the ICU and IRG is design specific

recursive channel estimation. The COMSIC contains multistage interference cancellation units (ICUs). In the ICU, the received signal is first despread. Then the phase variations due to the fading are compensated by estimating the channel. Tentative decisions are made after RAKE combining. By using the recovered data and the estimated complex fading envelope, the estimated interference of each path is regenerated at the interference replica generation unit (IRG) and combined in each received chip timing. Channel estimation is recursively performed and updated at each interference cancelling stage. The accuracy of the channel estimate is successively improved as the

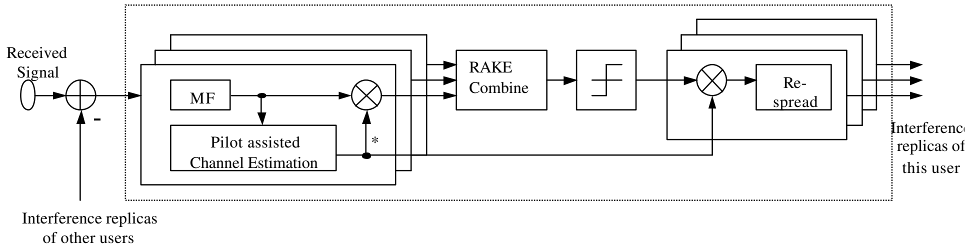


Figure 4.19: CEIGU : Channel Estimation and Interference replica Generation Unit

number of cancelling stages increases.

Figure 4.20 shows a schematic diagram of the COMSIC using recursive channel estimation. The channel ranking block ranks users according to their average received power levels, and the strongest user is ranked first. This ranking is done after despreading using matched filters (MFs), and is updated every slot. At each stage, there are K channel estimation and interference replica generation units (CEIGUs) [34,35]. The block diagram of the CEIGU is shown in Figure 4.19. Each CEIGU performs channel estimation, data-decision and replica regeneration of the MAI for each user in order of the decreasing powers.

Let $\widehat{I}_{x,y,l,p}(n)$ be the generated replica, at stage p of user x spread signal which is associated with path l and slot n. At each stage the MAI from $K-1$ users is subtracted from the received composite signal $r(t)$. The MAI from higher ranked users ($1 \leq u \leq x-1$) is given by:

$$\widehat{I}_{u,y,p}(n) = \sum_{l=0}^{L-1} \widehat{I}_{u,y,l,p}(n) \tag{4.28}$$

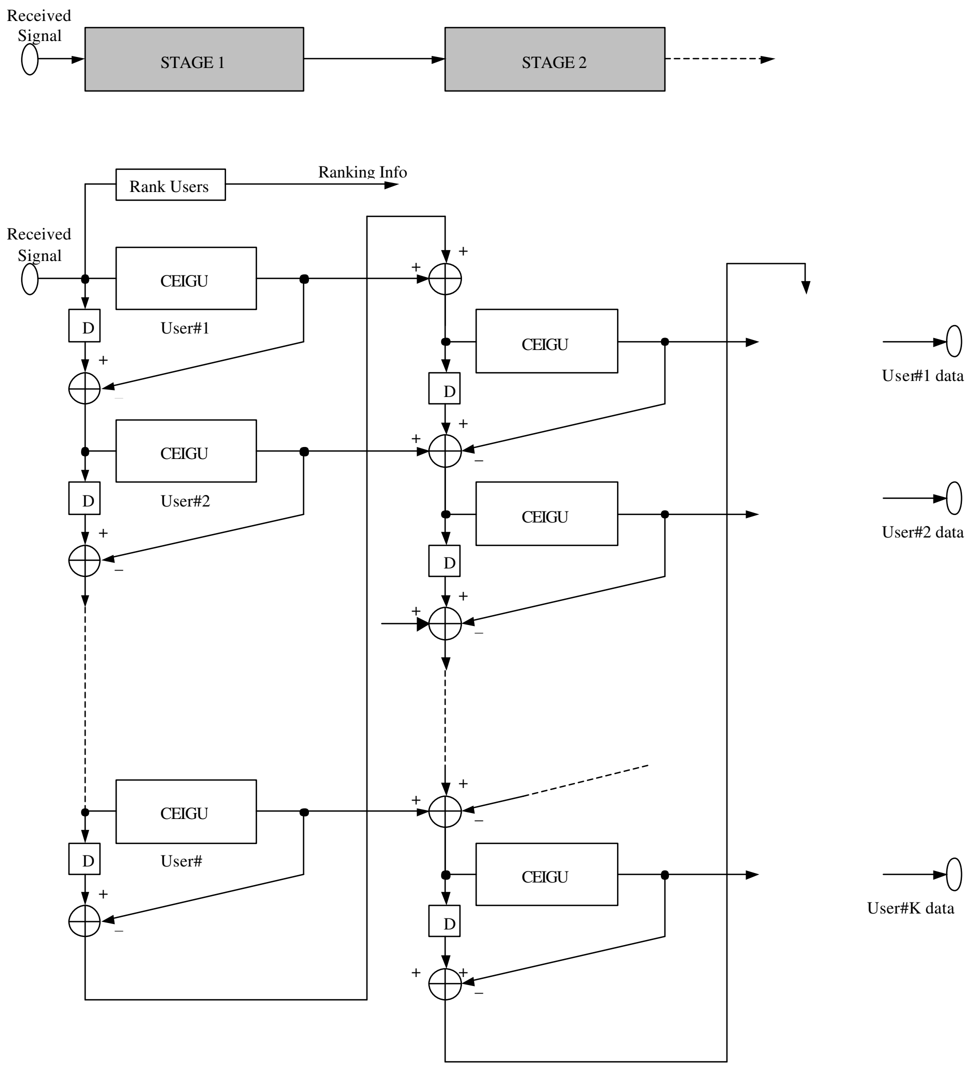


Figure 4.20: COMSIC : Coherent Multi-Stage Interference Canceller. This is the generic design of an IC which does serial interference cancellation in multiple stages. The CEIGU is the Channel Estimation and Interference replica Generation Unit

For lower ranked users ($x+1 \leq \ u \leq \ K$)), the interference contribution is:

$$\widehat{I}_{u,y,p-1}(n) = \sum_{l=0}^{L-1} \widehat{I}_{u,y,l,p-1}(n) \tag{4.29}$$

Then the output of the CEIGU is given by:

$$r_{x,y,l,p}(n) = r(n) - \left[\sum_{i=1}^{x-1} \sum_{j=0}^{L-1} \widehat{I}_{i,y,j,p}(n) + \sum_{i=x+1}^{K} \sum_{j=0}^{L-1} \widehat{I}_{i,y,j,p-1}(n) \right] \tag{4.30}$$

WiPPET$_{\text{signal}}$ currently employs a COMSIC receiver with three stages. WiPPET$_{\text{signal}}$ also incorporates an interference rejection control (IRC) factor β in order to reduce the error of the generated interference replica due to channel estimation error and data decision error [33]. All the estimated interference replicas are weighted by this factor β before being subtracted from the composite signal. Equation 4.30 in this case changes to:

$$r_{x,y,l,p}(n) = r(n) - \beta\left[\sum_{i=1}^{x-1}\sum_{j=0}^{L-1}\widehat{I}_{i,y,j,p}(n) + \sum_{i=x+1}^{K}\sum_{j=0}^{L-1}\widehat{I}_{i,y,j,p-1}(n)\right] \tag{4.31}$$

Reference [33] discusses the design in detail and gives average values of β in various situations. In all the results described in this thesis, a nominal value of 0.99375 has been used for β in the simulations involving interference cancellation.

Chapter 5

WiPPET$_{\text{signal}}$ Complexity, Speed, Parallelization and Speedups

WiPPET$_{\text{signal}}$ is a computationally intensive simulation, and has significant processing power and memory requirements. The two main components of WiPPET$_{\text{signal}}$ are:

- *Radio Channel Processing* : This involves the calculation of the received signal at each of the receivers in the system. This processing is done in the `RevChannel` for reverse link and in the `FwdChannel` for the forward link. In both cases, the flow is gathering information from all transmitters, incorporating the phenomena of propagation (short and long-scale channel effects) and interference.

- *Bit Detection* : Once the computed signal arrives at the receivers, it has to be decoded to extract the transmitted bits. This processing is part of the `client` that resides in the `Base`, `Mobile` and the `Switch`. We will term this computation as *Detector - MF* signifying a *Matched Filter* receiver and a *Detector - IC3* signifying a NTT DoCoMo *3 stage Interference Canceller.*

This chapter details their relative complexities, and possible approaches for simulation speedup through parallelization.

5.1 Expected Complexity Gradients

In this section, we will discuss the kind of processing involved in the main components of WiPPET$_{\text{signal}}$. This exercise will paint a picture of the nature of results we will encounter when we establish the complexity (Section 5.2) and speeds (Section 5.3) by actual experiments.

5.1.1 Radio Channel Processing

The radio channels incorporate waveform level with one sample per CDMA chip. Further, time offsets between transmitted signals are always an integer number of chips. Similarly, the multipath delays of each radio channel and the asynchronism of each base station are always an integer number of chips. Thus, for our purposes, a waveform of duration nT_c is simply a complex I/Q vector of length n.

Let us focus our attention on a waveform of one symbol duration. For a spreading factor of N, this waveform is a complex I/Q vector $\mathbf{S}$ of length N given by

$$\mathbf{S} = [s(0)\ s(T_c)\ s(2T_c) \dots\ s((N-1)T_c)] \tag{5.1}$$

where $s(nT_c)$ represents the chip signal at time nT_c. We will build our logic starting from the simplest situation. Let us assume a single transmitter with just one transmitting antenna and a single receiver with just one receiving antenna. Further assume that there is only a single path in the system. The received signal vector $\mathbf{R}$ at the receiver is

$$\mathbf{R} = [r(0)\ r(T_c)\ r(2T_c) \ldots\ r((N-1)T_c)] \tag{5.2}$$

where $r(nT_c)$ is the output corresponding to the chip $s(nT_c)$ and is given by

$$r(nT_c) = s(nT_c)G(nT_c) \qquad 0 \leq n \leq (N-1) \tag{5.3}$$

and $G(nT_c)$ is the channel gain at time nT_c. Each $s(nT_c)G(nT_c)$ operation comprises a complex multiplication with a Rayleigh fading value and multiplication with the long scale gain. This, in effect is equivalent to six multiplication and two addition operations. The result is also to be added to a pre-stored value in the buffer, which involves two more addition operations. Let us denote this total of six multiplication and four addition operations by ρ. For the calculation of $\mathbf{R}$ in equation (5.2), there were ρN operations. An increase in the sreading factor from N to aN will increase the processing by the same multiplicative factor a.

Now, let us assume that for this single receiver, there are A_r receiver antennas. The same operation given in equations (5.3) and (5.2) has to be performed A_r times, once for each receiving antenna j. In particular, for the nth chip we must calculate

$$r_j(nT_c) = s(nT_c)G_j(nT_c) \qquad 0 \leq n \leq (N-1) \tag{5.4}$$

in order to generate the signal vector $\mathbf{R}_j$ as

$$\mathbf{R}_j = [r_j(0)\ r_j(T_c)\ r_j(2T_c) \ldots\ r_j((N-1)T_c)] \tag{5.5}$$

Of course, the channel gain G_j will be different for each antenna. Now there are $A_r\rho N$ operations.

Similarly, let us assume that now there are L paths in the system, with path l having a delay $\tau_{j,l}$. The operations in equation (5.4) have to be performed L times, once for each path. For the nth chip, we must calculate

$$r_j(nT_c) = \sum_{l=0}^{L-1} s(nT_c - \tau_{j,l})G_{j,l}(nT_c) \qquad 0 \leq n \leq (N-1) \tag{5.6}$$

in order to generate the signal vector $\mathbf{R}_j$ as given by equation (5.5). The channel gain $G_{j,l}$ will be different for each antenna and each path. Now there are $LA_r\rho N$ operations.

The same can be extended to the number of receivers (BS) B, number of transmitters (MS) per cell K, and the corresponding number of transmitting antennas A_t. Since the total number of mobiles in the system is KB, the total workload in the Radio Channel can be expressed as

$$W_{\mathrm{R,symbol}} = KBBA_tA_rLN\rho \tag{5.7}$$

per symbol period. A WCDMA slot with M symbols will have a workload of

$$W_{\mathrm{R}} = KB^2A_tA_rLMN\rho \tag{5.8}$$

5.1.2 Bit Detection - Matched Filter

The bit detection processing has been explained in Sections 4.9.2 and 4.9.3. The following general steps lead to bit detection

1. Despreading
2. Tentative Channel estimation
3. Tentative Rake and data decision
4. Decision feedback channel estimation
5. Hard Rake and data decision
6. Error comparison

Let us focus our attention on one slot (M symbols). Section 4.9.3 gives details about the WSMA Channel estimation where the pilots of two consecutive slots are used. Let us assume that a fraction p of the symbols in a slot are pilots. Hence, there is a $p:(1-p)$ ratio between the pilots and data symbols. For a slot with M symbols, pM are pilots and the other $(1-p)M$ are data. The pM pilot symbols of the next slot will also be used. Let us call these $(1+p)M$ symbols as an *extended slot.*

Despreading

The extended slot ($(1+p)M$ symbols) of received signal is an I/Q complex vector $\mathbf{R}$ of length $S_e = (1+p)MN$. The mth symbol of length N is given by

$$\mathbf{r_m} = [r_m(mT_c + 0)\ r_m(mT_c + T_c)\ r_m(mT_c + 2T_c) \ldots\ r_m(mT_c + (N-1)T_c)] \tag{5.9}$$

where each r_m is given by equation (5.3). The extended slot of length S_e chips is given by

$$\mathbf{R} = [\mathbf{r}_0 \ \mathbf{r}_1 \ \mathbf{r}_2 \ldots \ \mathbf{r}_{(1+p)M-1}] \tag{5.10}$$

We will again build our logic starting from the simplest situation. Let us assume a single transmitter with just one transmitting antenna and a single receiver with just one receiving antenna. Further assume that there is only a single path in the system. The received signal is despread by the corresponding $\mathbf{C}$. The code for the mth symbol is given by

$$\mathbf{c}_m = [c_m(mT_c + 0) \ c_m(mT_c + T_c) \ c_m(mT_c + 2T_c) \ldots \ c_m(mT_c + (N-1)T_c)] \tag{5.11}$$

where c_m is the individual chip. The code sequence $\mathbf{C}$ for the extended slot is

$$\mathbf{C} = [\mathbf{c}_0 \ \mathbf{c}_1 \ \mathbf{c}_2 \ldots \mathbf{c}_{(1+p)M-1}] \tag{5.12}$$

The $(1+p)M$ despread symbols are given by

$$\widetilde{r}_m = \frac{1}{N} \sum_{v=0}^{N} r_m(mT_c + v) c_m(mT_c + v) \qquad 0 \leq m \leq (1.1M - 1) \tag{5.13}$$

The processing in equation (5.13) to despread one symbol involves N complex multiplicatons and N complex additions. This translates to $4N$ multiplications and $4N$ additions. Let us denote this total by 2ξ. The processing following despreading is independent of the spreading factor in a matched filter.

Now, let us assume that for this single receiver, there are A_r receiver antennas. Since the same operation given in equation (5.13) has to be performed A_r times, once for each receiving antenna j, there are $A_r 2\xi$ operations required for despreading. Similar relationship exists between despreading and the number of paths L, the number of transmitters (MS) per cell K and the corresponding number of transmitting antennas A_t. The total workload in despreading an extended slot at one receiver is

$$W_{\text{despread}} = KA_tA_rL[(1+p)MN2\xi] \tag{5.14}$$

Tentative channel estimation

After the $(1+p)M$ symbols are created using the processing in equation (5.13), the $2pM(= pM + pM)$ pilots have to be used to estimate the channel for each diversity path for each user. The symbols are also I/Q valued since data modulation in WCDMA could be either BPSK or QPSK.

The instantaneous channel estimate for the slot is performed by using the processing in equation (4.15). In our current nomenclature the estimate is given by

$$\widetilde{\eta} = \frac{1}{2pM} \sum_{m=0}^{pM-1} \left\{ \widetilde{r}_m + \widetilde{r}_{M+m} \right\} \tag{5.15}$$

where $\widetilde{r}_m$ is given by equation (5.13). Each pilot I/Q symbol is weighed and added to find a cumulative value for the pilots for that slot. That involves $pM\xi$ operations. The same processing is done for the pilots of the next slot to give a combined processing of $2pM\xi$ operations. Then the two estimates are averaged which involves another ξ operations. Hence, the number of operations for estimating the channel on one path for one user is $2pM\xi + \xi$.

Now, assume that there are L paths in the system. The processing described by equation (5.15) increases L fold. The same relationships exist between the processing for channel estimation and the number of paths L the number of transmitters (MS) per cell K and the corresponding number of transmitting antennas A_t. The total workload in the tentative channel estimation of a slot can be expressed as

$$W_{\text{tentChest}} = KA_tA_rL[2pM\xi + \xi] \tag{5.16}$$

Tentative rake and data decision

Once the channel is estimated, the $(1-p)M$ data symbols $\widetilde{r}_m$ have to be compensated by multiplying the despread data with the complex conjugate of the channel estimate $\widetilde{\eta}_{l,m}$as described in equation (4.16). Let us assume that there are L paths. In our current nomenclature, the rake combiner output is

$$\hat{r}_m = \sum_{l=0}^{L-1} \widetilde{r}_{l,m}\widetilde{\eta}^*_{l,m} \tag{5.17}$$

where $\widetilde{r}_{l,m}$ and $\widetilde{\eta}_{l,m}$ are given by equation (5.13) and equation (5.15) respectively. Multiplying a complex number to another complex number and adding the results to another complex number involves 2ξ operations. Hence, for L paths and $(1-p)M$ data symbols, the number of operations are $L(1-p)M2\xi$.

Now, assume that there are A_r receiving antennas. Hence the diversity expands to A_rL, and the processing for rake combination changes to $A_rL(1-p)M2\xi$. A similar relationship exists between the processing for rake combination and the number of transmitters (MS) per cell K and the corresponding number of transmitting antennas A_t. The total workload in the tentative rake and data decision for a slot can be expressed as

$$W_{\text{tentRake}} = KA_tA_rL[2(1-p)M\xi] \tag{5.18}$$

Decision feedback channel estimation

Decision feedback channel estimation is done on the tentative data to create a better estimate of the radio channel. For a single path, the $(1-p)M$ data symbols $\widetilde{r}_m$ are multiplied by the complex conjugate of their hard decisions to get a hard estimate of the channel. Each multiplication involves 2ξ operations. The feedback channel estimate and the tentative channel estimate are then averaged which requires another ξ operations. Hence, for one path and $(1-p)M$ data symbols, the number of operations are $2(1-p)M\xi + \xi$.

Now, assume that there are A_r receiving antennas and L paths. Hence the diversity expands to A_rL, and the processing for decision feedback channel estimation changes to A_rŁ$[(1-p)M2\xi + \xi]$. Similar relationship exists between the processing for decision feedback channel estimation and the number of transmitters (MS) per cell K and the corresponding number of transmitting antennas A_t. The total workload in the decision feedback channel estimation for a slot can be expressed as

$$W_{\text{hardChest}} = KA_tA_rL[2(1-p)M\xi + \xi] \tag{5.19}$$

Hard rake and data decision

The channel estimate after decision feedback is used in a similar manner as in equation (5.17) to create the hard Rake and data decision. The workload is the same as in equation (5.18). The total workload in hard rake and data decision for a slot can be expressed as

$$W_{\text{hardRake}} = W_{\text{tentRake}} = KA_tA_rL[(1-p)M2\xi] \tag{5.20}$$

Error Comparison

This processing follows the hard rake and data decision and is independent of the number of diversity paths. If the system has coding, then Soft Viterbi decoding is also a component of this calculation. Let us assume that the processing for checking errors for one user is some constant $C_1\xi$. As the number of users increase, this processing also increases. In general, the workload for this processing for one cell and number of transmitters (MS) per cell K can be given by

$$W_{\text{error}} = KC_1\xi \tag{5.21}$$

Total processing - Matched Filter

We use equations (5.14), (5.16), (5.18), (5.19), (5.20) and (5.21) to describe the total workload in the Matched Filter bit detection for a slot in one cell as

$$\begin{aligned} W_{\text{OneMF}} &= W_{\text{despread}} + W_{\text{tentChest}} + W_{\text{tentRake}} + W_{\text{hardChest}} + W_{\text{hardRake}} + W_{\text{error}} \\ &= KA_tA_rL\Bigg[[(1+p)MN2\xi] + [2pM\xi + \xi] + 2[(1+p)M2\xi] + [(1-p)M2\xi + \xi]\Bigg] + KC_1\xi \\ &= KA_tA_rL\Bigg[[p(2N-4) + 6 + 2N]M + 2\xi\Bigg] + KC_1\xi \end{aligned} \tag{5.22}$$

For B cells, this complexity is

$$W_{\text{MF}} = BW_{\text{OneMF}} \tag{5.23}$$

5.1.3 Bit Detection - Interference Cancellation

Interference Cancellation has been discussed in detail in Section 4.9.5. The following general steps lead to bit detection with and interference canceller

1. Rank the users
2. Perform interference cancellation and bit detection in multiple stages

Each stage of bit detection for one user can further be divided into

1. Interference subtraction from other users
2. Perform bit detection

Bit detection using interference cancellation has two more steps than that done for matched filter bit detection as enumerated in Section 5.1.2. The extra steps are in the beginning and towards the end of the processing. The general steps are

1. Interference subtraction from own multipath (new step)
2. Despreading
3. Tentative Channel estimation
4. Tentative rake and data decision
5. Decision feedback channel estimation
6. Hard rake and data decision
7. Interference replica generation (new step)
8. Error comparison

Let us focus our attention on the extended slot with $(1+p)M$ symbols.

User ranking and sorting

The first step in interference cancellation is the ranking of users. The received signal is despread and then the power is calculated for the slot. The processing in equation (5.13) to despread one symbol involves N complex multiplications and N complex additions equivalent to 2ξ operations. Next, the power is calculated which requires another ξ operations. For one slot of M symbols, the processing for despreading and then calculation of power requires $M(2\xi+\xi)$ operations. For A_r receiver antennas and

L paths, there is a $A_r L$ fold increase in the diversity, which increases the processing to $A_r LM(2\xi + \xi)$. For K users per cell, this processing is

$$W_{\text{ranking}} = KA_r L[M(2\xi + \xi)] \tag{5.24}$$

The next step is the sorting of the list of powers to decide the order of interference cancellation. The *quicksort* algorithm is used. For a cell with K users, the complexity is of the order of $O\Bigg(K\log(K)\Bigg)$. Although this represents a non linearity in sorting processing as K increases, in a single cell $K < M$ and $\log(K) \ll M$. Thus we neglect the cost of sorting. This step is followed by multiple stages of interference cancellation and bit detection.

Single stage Interference Subtraction from other users

Interference subtraction has been explained in Section 4.9.5. Equation (4.31) details the processing involved in the cancellation of interference from all other users. Let us consider the simplest case of two users, where the interference of one has to be subtracted from the received signal. In the current nomenclature, the processing for the mth symbol is given by

$$\mathbf{r}_{2,m} = \mathbf{r}_m - \beta\widetilde{r}_{1,m}\ \mathbf{c}_{1,m} \tag{5.25}$$

where $\mathbf{r}_m$ is the received signal given by equation (5.9) and $\mathbf{r}_{2,m}$ is the modified received signal to be used for bit detection of user 2. β is the interference rejection weight, $\widetilde{r}_{1,m}$ is the soft decision mth symbol for user 1 after decision feedback channel estimation and $\mathbf{c}_{1,m}$ is the spreading code for user 1 for the mth symbol.

The processing in equation (5.25) requires ξ operations for each chip; which translates to $N\xi$ for the entire symbol. For an extended slot, the number of operations are $(1+p)MN\xi$. Separate signals are received on each receiver antenna. If the receiver has A_r receiver antennas, the processing for cancelling one user from the other is

$$W_{\text{cancelOneUser}} = A_r[(1+p)MN\xi] \tag{5.26}$$

Now assume that there are K users per cell. The processing load in equation (5.27) has to be performed $K(K-1)$ times yielding a total workload

$$W_{\text{cancelUser}} = K(K-1)A_r[(1+p)MN\xi] \tag{5.27}$$

The K^2 term in equation (5.27) causes the processing in the interference canceller to increase in a non-linear fashion as K increases.

Single stage Bit detection - Interference Cancellation

The bit detection using interference cancellation involves two more steps than that using matched filter. For each diversity path, the interference generated by a user's own multipath is subtracted before any processing. Also, before the processing of a stage is completed, interference replicas are generated.

For a single user with one diversity path, there is no multipath, and hence no subtraction is necessary. Now. let us assume that there are 2 paths. Subtraction of the interference of one path from the other requires ξ operations for one chip. This increases to $(1+$

$p)MN\xi$ for an extended slot with $(1+p)1M$ symbols and a spreading factor of N. Now, assume that there are L paths. The processing has to be performed $L(L-1)$ times, yielding $L(L-1)(1+p)MN\xi$ operations. For A_t transmitting antennas and L paths, the number of multipaths are A_tL. Hence the processing is $A_tL(A_tL-1)(1+p)MN\xi$. The processing load for K users per cell and A_r receiver antennas is given by

$$W_{\text{subOwn}} = KA_rA_tL[(A_tL-1)(1+p)MN\xi] \tag{5.28}$$

The A_tL^2 term in equation (5.28) causes the processing in the interference canceller to increase in a non-linear fashion as either A_t or L or both increase.

Interference replica are generated by multiplying the despread symbol by the corresponding spreading code. For one symbol with N chips, this requires $N\xi$ operations. For an extended slot, the processing load for K users per cell, A_t transmitting antennas, L paths and and A_r receiver antennas for one cell is given by

$$W_{\text{intfGen}} = KA_rA_tL[(1+p)MN\xi] \tag{5.29}$$

Single stage total processing

We use equations (5.27), (5.28), (5.14), (5.16), (5.18), (5.19), (5.20) and (5.29) to describe the total workload in a single stage of bit detection for a slot in one cell as

$$\begin{aligned}
W_{\text{OneStage}} &= W_{\text{cancelUser}} + W_{\text{subOwn}} + W_{\text{despread}} + W_{\text{tentChest}} + W_{\text{tentRake}} + \\
&\quad W_{\text{hardChest}} + W_{\text{hardRake}} + W_{\text{intfGen}} \\
&= K(K-1)A_r[(1+p)MN\xi] + KA_rA_tL(A_tL-1)[(1+p)MN\xi] + \\
&\quad KA_tA_rL\Big[[(1+p)MN2\xi] + [2pM\xi + \xi] + 2[(1-p)M2\xi] + [(1-p)M2\xi + \xi]\Big] + \\
&\quad KA_rA_tL[(1+p)MN\xi] \\
&= K(K-1)A_r[(1+p)MN\xi] + \\
&\quad KA_tA_rL\Big[[p(3N-4+A_tL) + 6 + 3N + A_tL]M + 2\xi]\Big]
\end{aligned} \tag{5.30}$$

Total processing - Interference Cancellation

Now, we use equations (5.30), (5.24) and (5.21) to describe the total workload in the interference cancellation bit detection. For a slot in one cell with S stages of cancellation, the total workload is

$$
\begin{aligned}
W_{\text{OneIC}} &= W_{\text{ranking}} + SW_{\text{OneStage}} + W_{\text{error}} \\
&= KA_rLM(2\xi + \xi) + \\
&\quad + S\Big[K(K-1)A_r[(1+p)MN\xi] \\
&\quad + KA_tA_rL\Big[[p(3N - 4 + A_tL) + 6 + 3N + A_tL]M + 2\xi]\Big]\Big] \\
&\quad + KC_1\xi
\end{aligned} \tag{5.31}
$$

where S is the number of stages. For a system with B receivers (BS), the workload is

$$W_{\text{IC}} = BW_{\text{OneIC}} \tag{5.32}$$

5.1.4 Conclusions

Assuming that the transmitter antennas A_t, receiver antennas A_r and interference cancellation stages are held constant, the components of WIPPET$_{\text{signal}}$ show the following change in their processing requirements with a change in configurable variables.

- Radio Channel Processing : $W_{\text{R}} = O\Big(KB^2LMN\Big)$
- Bit Detection - Matched Filter : Linear $W_{\text{MF}} = O\Big(BKLMN\Big)$
- Bit Detection - Interference Canceller : Non Linear $W_{\text{IC}} = O\Big(BK^2MNL^2\Big)$

Before we proceed onto the experimental calculations of complexity and speeds of the various setups, we can make the following observations

- The ratio of the relative complexity of radio channel processing to that of bit detection with a matched filter should settle down at some constant values as K increases. This will occur because of the fact that both the radio channel and the detector will require a linear increase in their processing requirements as the number of users increase

- The ratio of the relative complexity of radio channel processing to that of bit detection with an interference canceller will change with the offered load. This will occur because of the fact that the radio channel will require a linear increase in its processing requirements but the detector will have a non-linear (dominantly quadratic) requirement as the number of users increase

- The time taken by the simulation will increase linearly as the number of users increase for a Matched Filter bit detector

- The time taken by the simulation will increase non-linearly (dominantly quadratic) as the number of users increase for an Interference Canceller bit detector

The next section deals with the experiments and their results.

5.2 Complexity: Experimental Results

Profiling allows us to learn where the program spent its time and which functions called which other functions during its execution. This information can show you which pieces of the program are slower than expected, and what is the relative time that was spent in each module of the program. We have used the GNU profiler, *gprof* to determine which parts of WiPPET$_{\text{signal}}$ are taking up most of the time. All source files are compiled

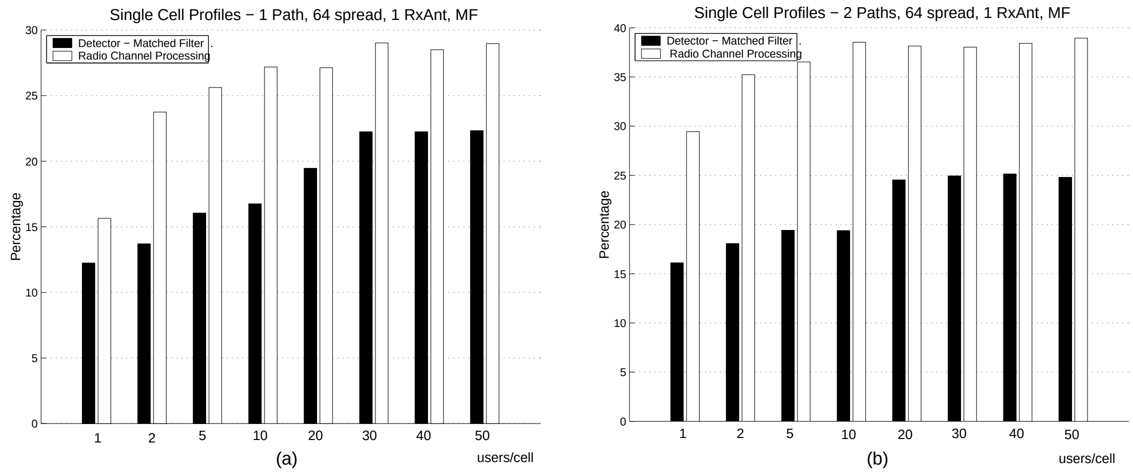


Figure 5.1: Single cell (reverse link) complexity - MF : The bar graphs (a) and (b) represent the percentage of time spent for radio channel processing in the `RevChannel` entity and for bit detection (MF) in the `Base` entity for one and two path Rayleigh respectively. There is only one transmitting and one receiving antenna per node.

with an additional *-pg* compile time option, and the program is executed normally. However, apart from the WiPPET$_{\text{signal}}$ program processing, extra processing is done by the profiler that is interlaced with the original processing. A file (*gmon.out*) is created after execution and that is profiled. The functions that are part of profiling are neglected as their time gives a measure of the amount of overhead due to profiling. All necessary statistics are up-scaled to compensate for this reason.

5.2.1 Single Cell Reverse Link

We will discuss the relative complexity of the radio channel processing vs. bit detection in a single cell. The computation in the radio channel increases linearly as the number of users increase. The same behaviour is seen for a Matched Filter (MF) bit detector. However, the 3-stage Interference Canceller (IC3) bit detector requires a non-linear increase in computational requirement.

Figure 5.1 illustrates the relative complexity of the radio channel and the matched filter receiver in a single cell system. For both the single path setup in Figure 5.1(a) and the two path setup in Figure 5.1(b), the same behaviour of attaining a constant complexity is observed.

Let us refer to equations (5.8) and (5.23) which give the approximate workloads of the radio channel processing and of bit detection using matched filter. For the above setup in Figure 5.1(a), the corresponding parameters are $B = 1$, $A_t = 1$, $A_r = 1$, $L = 1$, $M = 40$, $p = 0.1$ and $N = 64$. The quantity ρ in equation (5.8) and other operational overhead can be approximated to roughly 5ξ, which is the same ξ in equation (5.23). By substituting these values in equations (5.8) and (5.23) we get

$$\begin{aligned} W_{\mathrm{R}} &= K(64)(40)5\xi \\ W_{\mathrm{MF}} &= K\Big[[0.1(2(64)-4)+6+2(64)]40+2\xi\Big]+KC_1\xi \end{aligned} \tag{5.33}$$

From equation (5.33), the ratio $\frac{W_{\mathrm{MF}}}{W_{\mathrm{R}}}$ is $\frac{5874+C_1}{12800}$ which is approximately equal to $0.4589+\frac{C_1}{12800}$. Let us now refer to Figure 5.1(a) and choose the values of K for which the complexity of radio channel processing and bit detection are constant and stabilized. The values are around 28.5 and 22.5 for W_{R} and W_{MF} respectively, which gives a ratio of $\frac{W_{\mathrm{MF}}}{W_{\mathrm{R}}} = \frac{22.5}{28.5} = 0.7894$. We now compare the values of the ratio from equation (5.33) and the experiment to get

$$\begin{aligned} 0.7894 &= 0.4589 + \frac{C_1}{12800} \qquad \text{which gives} \\ C_1 &= 4230.4 \end{aligned} \tag{5.34}$$

Now, let us refer to Figure 5.1(b) which gives the relative complexity for radio channel processing and bit detection for a 2 path Rayleigh setup. All the parameters used to

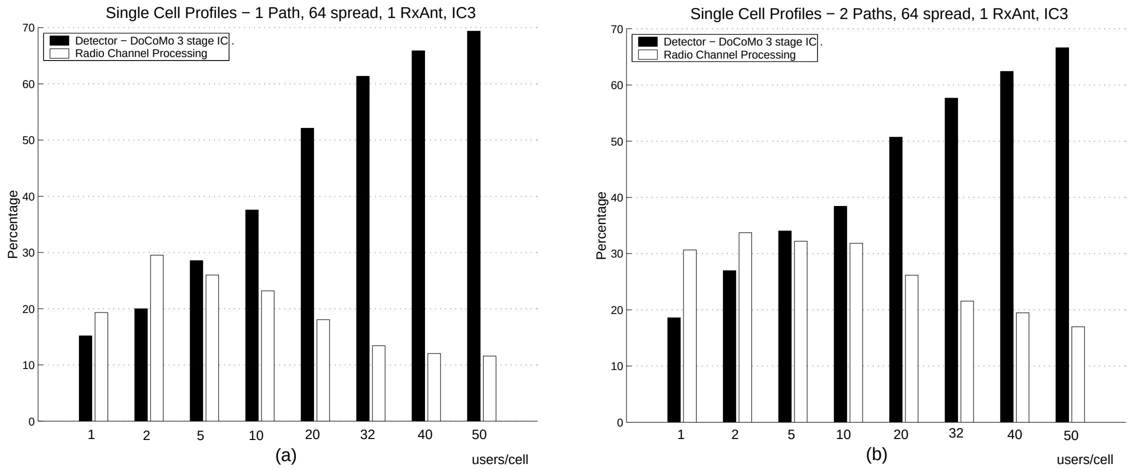


Figure 5.2: Single cell (reverse link) complexity - IC3 : The bar graphs (a) and (b) represent the percentage of time spent for radio channel processing in the `RevChannel` entity and for bit detection (IC3) in the `Base` entity for one and two path Rayleigh respectively. There is only one transmitting and one receiving antenna per node.

find equation (5.33) remain the same except L which is now 2. We use the calculated value of C_1 from equation (5.34) and calculate

$$\begin{aligned} W_{\mathrm{R}} &= K(2)(64)(40)5\xi \\ W_{\mathrm{MF}} &= K(2)\Big[[0.1(2(64)-4)+6+2(64)]40+2\xi\Big]+K(4230.4)\xi \end{aligned} \tag{5.35}$$

which gives $\frac{W_{\mathrm{MF}}}{W_{\mathrm{R}}} = \frac{11748+4230.4}{25600}$ which is 0.6255. Now, let us refer to Figure 5.1(b) and choose the values of K for which the complexity of radio channel processing and bit detection are constant and stabilized. The values are around 38.5 and 25.0 for W_{R} and W_{MF} respectively, which gives a ratio of $\frac{W_{\mathrm{MF}}}{W_{\mathrm{R}}} = \frac{25.0}{38.5} = 0.6493$ which is quite near the estimate of 0.6255 from equation (5.35).

Figure 5.2 illustrates the growing complexity of the 3-stage interference canceller receiver in a single cell system. Since the complexity of the bit detection in the interference canceller increase quadratically with an increase in number of users, and the radio channel increases linearly, a divergent trend is seen for both a single path setup

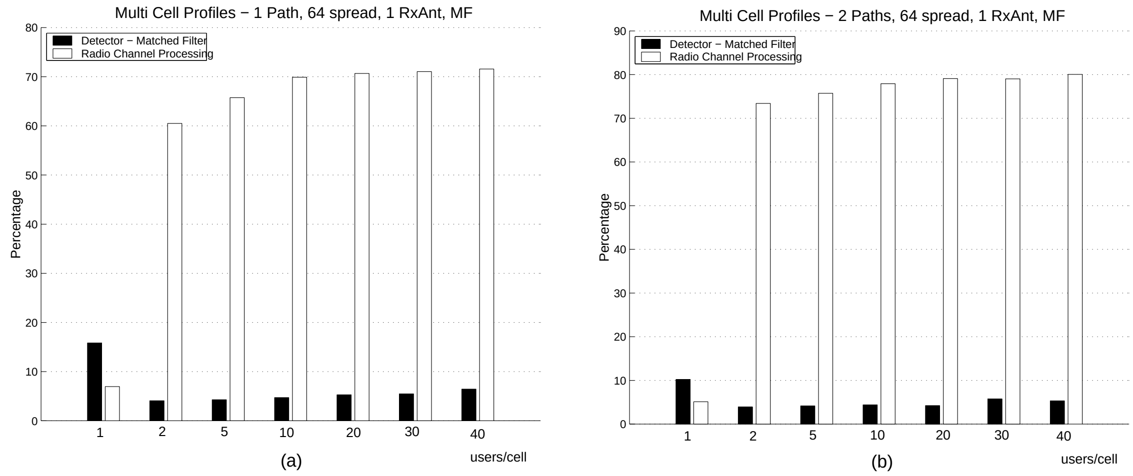


Figure 5.3: Multi cell (reverse link) complexity - MF : The bar graphs (a) and (b) represent the percentage of time spent for radio channel processing in the `RevChannel` entity and for bit detection (MF) in the `Base` entity for one and two path Rayleigh respectively. There is only one transmitting and one receiving antenna per node.

in Figure 5.2(a) and a two path setup in Figure 5.2(b).

5.2.2 Multi Cell Reverse Link

We now discuss the relative complexity of the radio channel processing vs. bit detection in a multi cell system. We again will discuss cases of one or two Rayleigh paths for a matched filter (MF) and a 3-stage interference canceller (IC3) respectively.

Figure 5.3 illustrates the relative complexity of the radio channel and the matched filter receiver in the multi cell system. For both the single path setup in Figure 5.3(a) and the two path setup in Figure 5.3(b), the same behaviour of attaining a constant complexity is observed.

Let us refer to equations (5.8) and (5.23) which give the approximate workloads of the radio channel processing and of bit detection using matched filter. For the above setup in Figure 5.3(a), the corresponding parameters are $B = 12$, $A_t = 1$, $A_r = 1$, $L = 1$,

$M = 40$, $p = 0.1$ and $N = 64$. The quantity ρ in equation (5.8) and the operational overload can be approximated to roughly 5ξ, which is the same ξ in equation (5.23). By substituting these values in equations (5.8) and (5.23) we get

$$\begin{aligned} W_{\mathrm{R}} &= K(12)(64)(40)5\xi \\ W_{\mathrm{MF}} &= K\Big[[0.1(2(64)-4)+6+2(64)]40+2\xi\Big]+K4230.4\xi \end{aligned} \tag{5.36}$$

From equation (5.36), the ratio $\frac{W_{\mathrm{MF}}}{W_{\mathrm{R}}}$ is $\frac{5874+4230.4}{12800(12)}$ which is approximately equal to 0.0678. Let us now refer to Figure 5.3(a) and choose the values of K for which the complexity of radio channel processing and bit detection are constant and stabilized. The values are around 71.0 and 5.0 for W_{R} and W_{MF} respectively, which gives a ratio of $\frac{W_{\mathrm{MF}}}{W_{\mathrm{R}}} = \frac{5.0}{71.0} = 0.0704$ which is quite near the estimated value of 0.0678 from equation (5.36).

Now, let us refer to Figure 5.3(b) which gives the relative complexity for radio channel processing and bit detection for a 2 path Rayleigh setup. All the parameters used to find equation (5.36) remain the same except L which is now 2. We get

$$\begin{aligned} W_{\mathrm{R}} &= K(12)(2)(64)(40)5\xi \\ W_{\mathrm{MF}} &= K(2)\Big[[0.1(2(64)-4)+6+2(64)]40+2\xi\Big]+K(4230.4)\xi \end{aligned} \tag{5.37}$$

which gives $\frac{W_{\mathrm{MF}}}{W_{\mathrm{R}}} = \frac{11748+4230.4}{25600(12)} \approx 0.0520$. Now, let us refer to Figure 5.3(b) and choose the values of K for which the complexity of radio channel processing and bit detection are constant and stabilized. The values are around 80.0 and 4.5 for W_{R} and W_{MF} respectively, which gives a ratio of $\frac{W_{\mathrm{MF}}}{W_{\mathrm{R}}} = \frac{4.5}{80.0} = 0.0562$ which is quite near the estimated value of 0.0520 from equation (5.37).

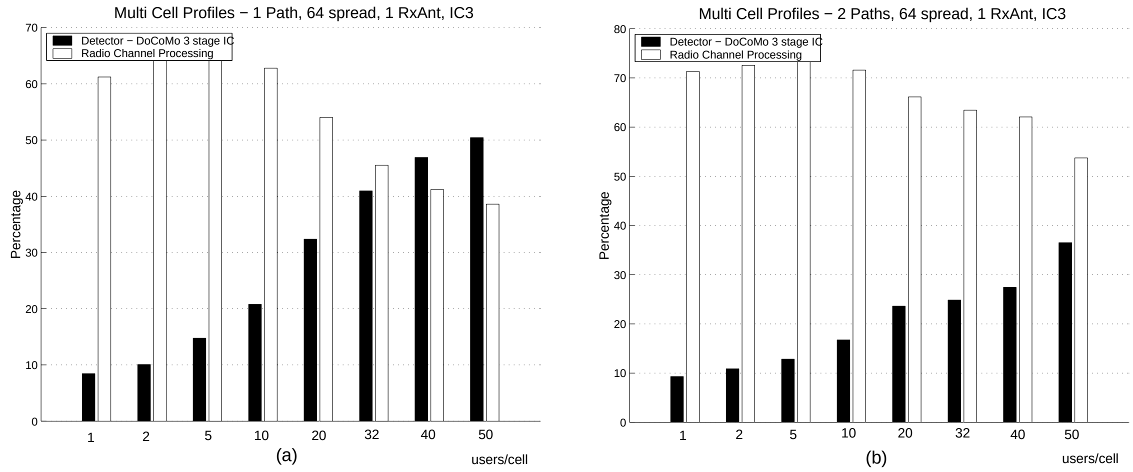


Figure 5.4: Multi cell (reverse link) complexity - IC3 : The bar graphs (a) and (b) represent the percentage of time spent for radio channel processing in the `RevChannel` entity and for bit detection (IC3) in the `Base` entity for one and two path Rayleigh respectively. There is only one transmitting and one receiving antenna per node.

Figure 5.4 also illustrates the growing complexity of the 3-stage interference canceller receiver in a multi cell system. Since the complexity of bit detection in the interference canceller increase quadratically with an increase in number of users, and the radio channel increases linearly, a divergent trend is seen for both a single path setup in Figure 5.2(a) and a two path setup in Figure 5.2(b).

5.2.3 Multi Cell Forward and Reverse Link

In the current version of WiPPET$_{\text{signal}}$, the forward link is used only to transmit pilot frames for soft handoff measurements. When there is only one BS, there is no soft handoff and thus no need for a forward link. We will not discuss the complexity in case of a single cell with forward link also present because there is no practical sense in forward link pilot channel simulations when there are no other BSs to have soft-handoff with. We will discuss the case of 2 path Rayleigh, for a 3-stage Interference Canceller (IC3) only.

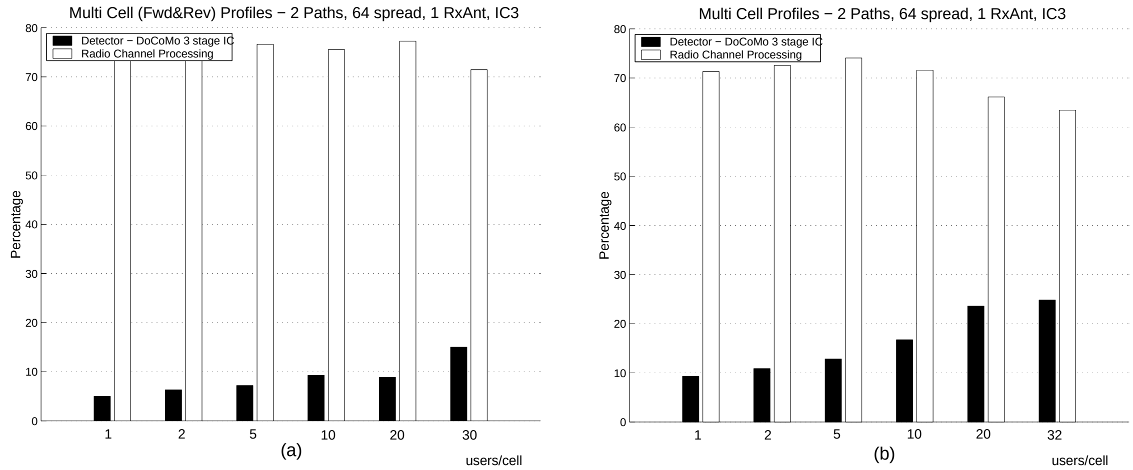


Figure 5.5: Multi cell (forward and reverse link vs. reverse link) complexity (IC3, 2 paths): The bar graphs - reverse and forward link (a) and reverse link (b) provide a comparison for the relative complexities for signal computation and bit detection. The bar graph(b) is the same as in Figure 5.4(b) and have been repeated for highlighting the comparison only.

Let us compare the complexities of Multi cell reverse only (Section 5.2.2) and the Multi cell forward and reverse link cases. The complexity of the radio channel processing is done is doubled as an equivalent amount of work is done in the forward channel as well. However, the computation in the reverse link detector in the BSs stays the same. Also, the computation in the forward link pilot detectors is very simple as compared to the detectors in the reverse link. The overall effect is that the total complexity of the radio channel in the multi cell case with both forward and reverse links is greater than the case with only the reverse link as illustrated in Figure 5.5.

5.2.4 Effects of Rayleigh Generation Resolution

The resolution of the Rayleigh generation process effects the overall runtime of the simulation because more time has to be devoted to generating the correlated Rayleigh random variables. Assume that the spreading factor of a WCDMA symbol is N in a single cell system with p multipaths and x transmitters. For the one receiver with one antenna, the number of independent Rayleigh faders are xp. Now, we could use an extreme resolution of one Rayleigh sample per chip; i.e. xpN Rayleigh samples per symbol of simulation in the system. However, since the Rayleigh fading process is correlated in time, a rate of one Rayleigh sample per chip may not be needed. If we chose a much lower resolution, say 1 sample per symbol, that would mean the need to generate only xp Rayleigh samples per symbol of simulation. The same approximation in a multi cell system with total x transmitters would give the same numbers as $12xpN$ and $12xp$ respectively. In either case, this implies an N fold decrease in the number of Rayleigh random variables generated. Here, we will concentrate on how the choice of resolution effects complexity. The qualitative effects, i.e. how the BER is effected with this choice, will be discussed in Section 6.1.1.

Figure 5.6 illustrates the effect of the full resolution Rayleigh generation on a single cell system with x transmitters, 1 path and 1 receiver with a 3-stage interference canceller detector. As is evident, although the detector dominates the processing, it is no longer the only heavy module in Figure 5.6(a). The generation of Rayleigh variables also takes an appreciable amount of processing time. However, workload of Rayleigh fading becomes negligible when it is reduced by the spreading factor N as shown in Figure 5.6(b).

5.3 Speed

In this and the following sections, we will discuss the time taken by the simulation. The period over which time will be measured is one WCDMA frame. As discussed in

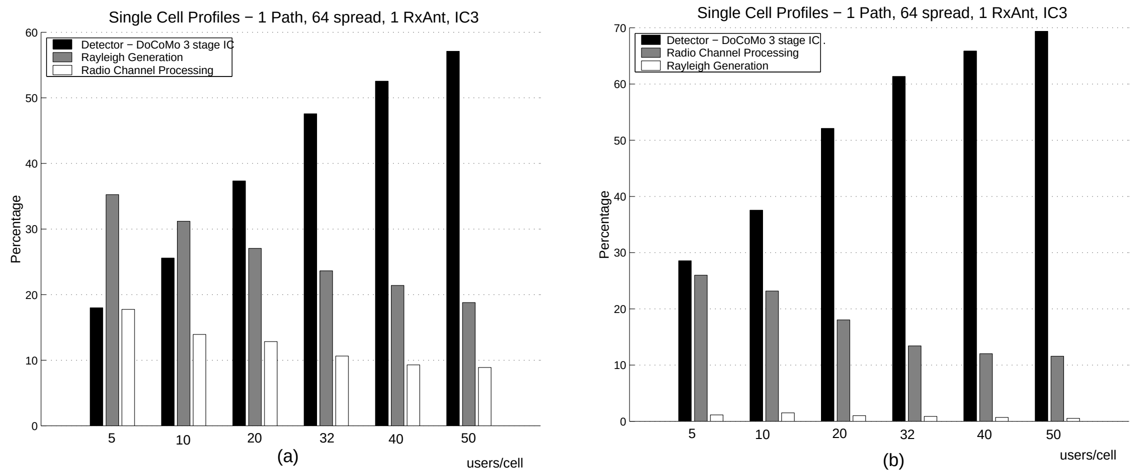


Figure 5.6: The effect of Rayleigh resolution on single cell (reverse link) complexity comparisons using the interference canceller and 1 path: The relative complexities of a single cell system are compared to illustrate the effect of Rayleigh precision on the simulation. Bar graph(a) shows the percentage of time spent for radio channel processing in the `RevChannel` entity, the time spent for generating Rayleigh fading values for every chip (maximum resolution) in the `Rayleigh` entity using interference cancellation in the `Base` entity. The bar graph(b) has minimum precision Rayleigh generation, the load for which is negligible as compared to other processing.

section 5.2, we know that the two main computational efforts are in the radio channel for signal calculation, and in the receiver nodes for bit detection. The overhead, as illustrated by the bar graphs of figures in section 5.2 clearly shows that the overhead becomes less significant as the simulation becomes larger. We will initiate the timing measurement at the beginning of the first WCDMA frame and terminate it at the beginning of the $(T+1)$th WCDMA frame in the receiver node. Then we will average the time by T to find out the time required for one frame. This will guarantee that all radio channel processing, bit detection and other overhead (SSF event transfer, fading generation, memory and class management) is incorporated in one frame. This frame is representative as one block of the simulation, as the same sequence of events is repeated every time.

Since WiPPET$_{\text{signal}}$ can run in linux or solaris environments, we have chosen an Ultra-30 Solaris Sun workstation to do all our measurements. We will only discuss serial execution in this section. Parallel execution speeds will be discussed in section 5.4.2.

5.3.1 Expected Speed and Rate Gradients

We now discuss the trends in simulation time and WiPPET$_{\text{signal}}$ processing rates in B cells as the number of transmitters K per cell increase for a given setup. We have already introduced these concepts in Section 5.1.4. Let T_{R} be the time required for simulating one WCDMA frame when only the radio channel processing is present and the detector is absent. This time will be given by

$$T_{\mathrm{R}} = C_2 W_{\mathrm{R}} \tag{5.38}$$

where C_2 is some constant and W_{R} is given by equation (5.8). Recall that W_{R} is directly proportional to K and B.

Let $T_{\mathrm{R+MF}}$ be the time required for simulating one WCDMA frame when the radio channel processing and bit detection using matched filter is used. This time will be given by

$$T_{\mathrm{R+MF}} = C_3 [W_{\mathrm{R}} + W_{\mathrm{MF}}] \tag{5.39}$$

where C_3 is some constant and W_{MF} is given by equation (5.23). Recall that W_{MF} is also directly proportional to K and B. Similarly, the time $T_{\mathrm{R+IC}}$ required for simulating one WCDMA frame when the radio channel processing and bit detection using interference canceller is used is given by

$$T_{\mathrm{R+IC}} = C_4[W_{\mathrm{R}} + W_{\mathrm{IC}}] \tag{5.40}$$

where C_4 is some constant and W_{IC} is given by equation (5.32).

The total number of bits simulated is same in all three cases given by

$$N_{\mathrm{bits}} = C_5 K B \tag{5.41}$$

where C_5 is a constant denoting the bits in a frame.

The WiPPET$_{\mathrm{signal}}$ processing rates for the setup without any bit detection is given by

$$R_{\mathrm{R}} = \frac{N_{\mathrm{bits}}}{T_{\mathrm{R}}} = \frac{C_5 K B}{C_2 W_{\mathrm{R}}} \tag{5.42}$$

where N_{bits} and W_{R} are given by equations (5.41) and (5.8) respectively. As the number of users per cell K increases, both the numerator and denominator increase linearly, thus resulting in the same constant processing rate.

The WiPPET$_{\mathrm{signal}}$ processing rates for the setup with bit detection using matched filter is given by

$$R_{\mathrm{R+MF}} = \frac{N_{\mathrm{bits}}}{T_{\mathrm{R+MF}}} = \frac{C_5 K B}{C_3[W_{\mathrm{R}} + W_{\mathrm{MF}}]} \tag{5.43}$$

where N_{bits}, W_{R} and W_{MF} are given by equations (5.41), (5.8) and (5.23) respectively. As the number of users per cell K increases, both the numerator and denominator increase linearly, thus resulting in the same constant processing rate.

For bit detection using interference canceller, a different trend is seen. The WiPPET$_{\text{signal}}$ processing rates for the setup with bit detection using interference canceller is given by

$$R_{\text{R+IC}} = \frac{N_{\text{bits}}}{T_{\text{R+IC}}} = \frac{C_5 K B}{C_4 [W_{\text{R}} + W_{\text{IC}}]} \tag{5.44}$$

where N_{bits}, W_{R} and W_{IC} are given by equations (5.41), (5.8) and (5.32) respectively. As the number of users per cell K increases, both the numerator increases linearly. However, the denominator increases quadratically, which causes the rate to decrease as K increases.

When the number of BSs is held constant, the following trends in WiPPET$_{\text{signal}}$ bit processing rates will be seen as the number of users increase

- Radio channel processing only: $O\left(1\right)$
- Radio channel processing and bit detection using matched filter: $O\left(1\right)$
- Radio channel processing and bit detection using interference canceller: $O\left(\frac{1}{K}\right)$

5.3.2 Single Cell Reverse Link Serial Execution

Let us assume that there are K MSs in the cell, and only single antenna receivers and transmitters. For a channel with L multipaths in its profile, the number of transmitted contributions are KL and received signals is 1. We calculate the time required to completely process (create, interfere and propagate, and re-create) one WCDMA frame from each of the K users. The only detector in the system is used to decode all the K users. Figure 5.7(a) illustrates the time in seconds to process one complete WCDMA

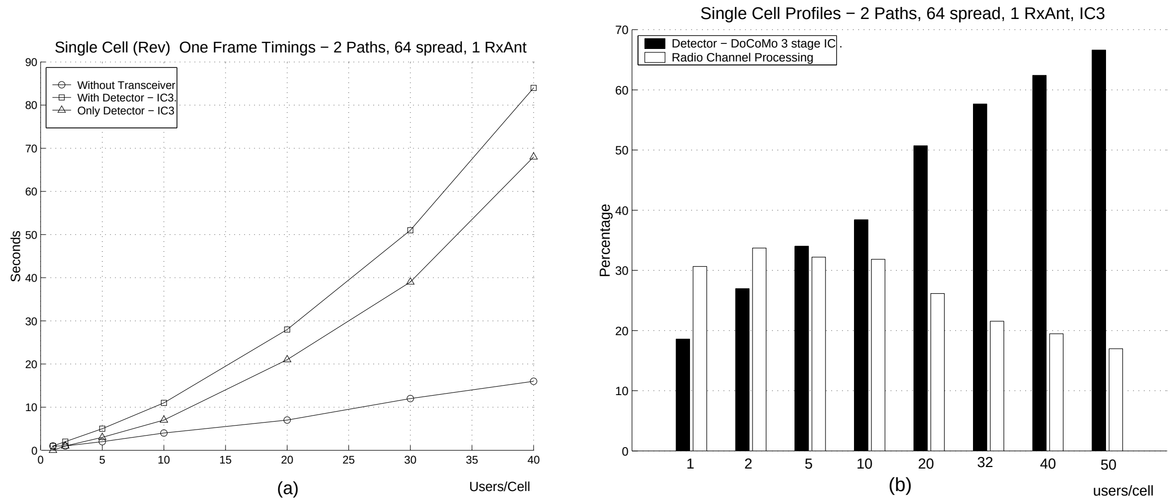


Figure 5.7: Single cell (reverse Link) speeds (IC3, 2 paths) : The curves(a) show the time taken to completely process one frame for each of the x users (x WCDMA frames in total) in WIPPET$_{\text{signal}}$. The bar graph(b) is the same as in Figure 5.2(b), and is given as a reference for the relative complexities of the radio channel and the detector. It is clear that the detector takes up most of the time in the simulation.

frame for a two path Rayleigh channel with a 3-stage interference canceller. It is evident that as the number of MSs increases, the time required by the detector increases at a much faster rate than the radio channel processing. This is in accordance with the conclusions drawn in Section 5.3.1 about the quadratic increase in workload with the number of users.

Figure 5.8 gives the total number of bits processed by WIPPET$_{\text{signal}}$ in one second. For a WCDMA frame with 16 slots, each having 40 symbols (36 data + 4 pilots) and spreading factor of 64, there are $640 = 16 * 40$ bits in one frame. For a $36 : 4 :: 9 : 1$ ratio of data to pilot bits, the information rate is 0.9 times the processing rate. In Figure 5.8 we observe that the processing rate for two of the three cases is constant. These setups are when the transceiver processing is absent and when matched filter is being used for bit detection. As explained earlier in Sections 5.1.1 and 5.1.2 and then concluded in Sections 5.1.4 and 5.3.1, the processing in the radio channel and the matched filter

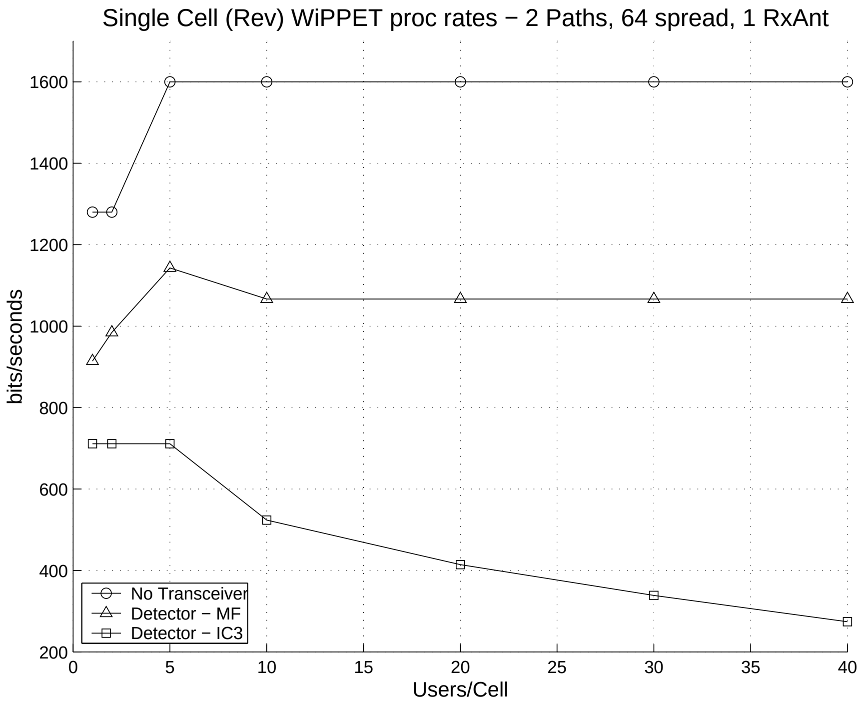


Figure 5.8: Single cell (reverse Link) processing rate : The processing in the Radio Channel and the Matched Filter detector increases linearly with the number of users, whereas the increase is non-linear for Interference Cancellation detector. This results in constant processing rates for the setups when the transceiver is absent or when the Matched Filter bit detector is used

detector increases linearly with the number of users, and as a result the processing rate stays constant.

However, the interference canceller bit detector has a quadratic increase in processing requirement as discussed in Sections 5.1.3 and 5.1.4. The number of bits increase linearly. Hence, as concluded in Section 5.3.1, the processing rate drops roughly proportional to $\frac{1}{K}$ the number of users increase.

5.3.3 Multi Cell Reverse Link Serial Execution

Let us assume that there are K MSs in one cell, and only single antenna receivers and transmitters. For a channel with L multipaths in its profile, the number of transmitted

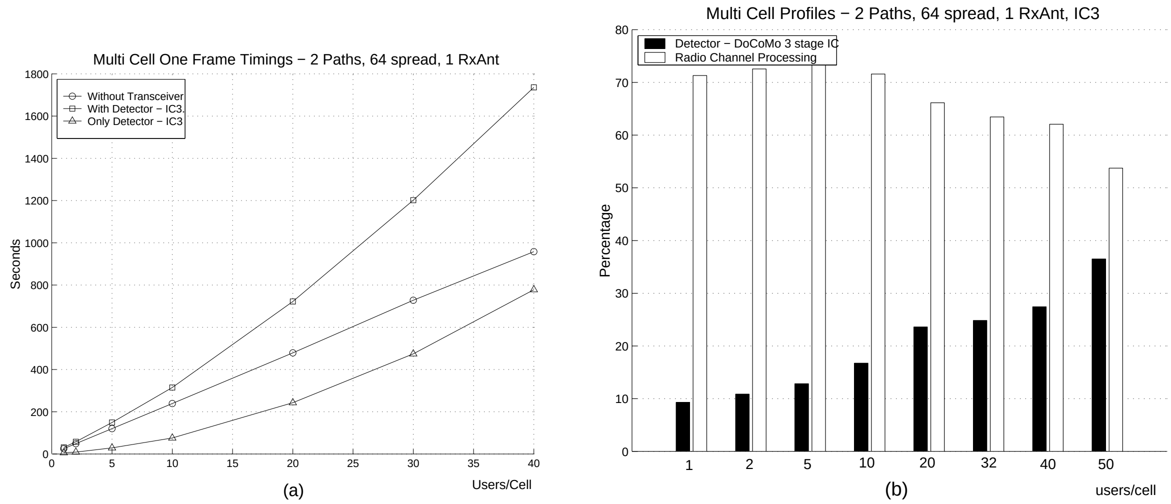


Figure 5.9: Multi cell (reverse link) speeds (IC3, 2 paths) : The curves(a) show the time taken to completely process one frame for each of the x users per cell ($12x$ WCDMA frames in total) in WiPPET$_{\text{signal}}$. The bar graph(b) is the same as in Figure 5.4(b), and is given as a reference for the relative complexities of the radio channel and the detector. The radio channel takes up most of the time, though the detector catches up in later stages.

contributions are $12KL$ and received signals are 12. We calculate the time required to completely process (create, interfere and propagate, and re-create) one WCDMA frame from each of the $12K$ users. There are 12 detectors in the system; each one used to decode K users. Figure 5.9(a) illustrates the time in seconds to process one complete WCDMA frame for a two path Rayleigh channel with a 3-stage interference canceller. It is evident that for cases where there are fewer MSs, the radio channel processing takes up most of the time. However, as the number of MS increase, the time required by the detector catches up. This is in accordance with the conclusions drawn in Section 5.3.1 about the quadratic increase in workload with the number of users. increase.

Figure 5.10 gives the total number of bits processed by WiPPET$_{\text{signal}}$ in one second. We observe that the processing rate for two of the three cases is constant. These setups are when the transceiver processing is absent and when matched filter is being used for bit detection. As explained earlier in Sections 5.1.1 and 5.1.2 and then concluded

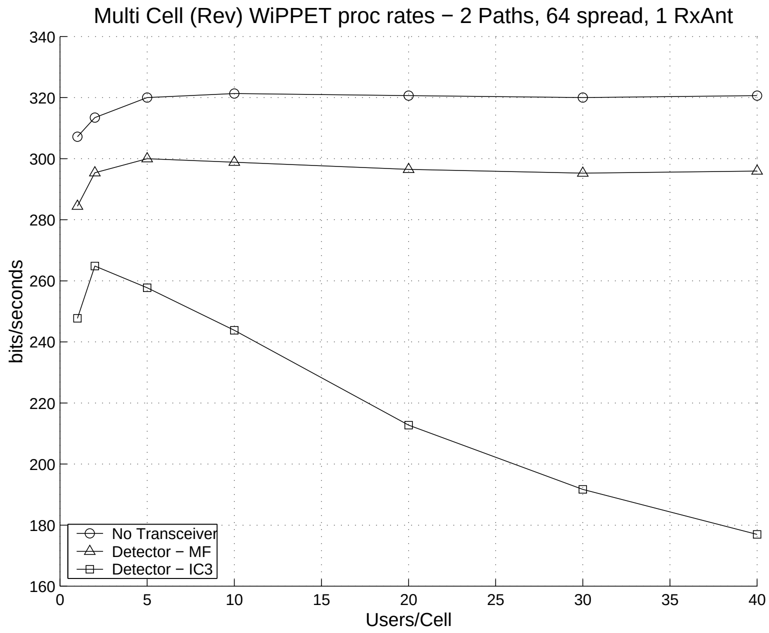


Figure 5.10: Multi cell (reverse Link) processing rate : The processing in the Radio Channel and the Matched Filter detector increases linearly with the number of users, whereas the increase is non-linear for Interference Cancellation detector. This results in constant processing rates for the setups when the transceiver is absent or when the Matched Filter bit detector is used

in Sections 5.1.4 and 5.3.1, the processing in the radio channel and the matched filter detector increases linearly with the number of users, and as a result the processing rate stays constant. However, for the interference canceller bit detector, the processing rate decreases linearly as the number of users increase.

5.3.4 Multi Cell Forward and Reverse Link Serial Execution

Once again, we will not discuss the complexities in case of a single cell with forward link. Let us assume that there are K mobiles per cell in the system, and only single antenna receivers and transmitters. For a channel with L multipaths in its profile, the number of transmitted contributions and received signals are KL and 12 respectively for the reverse link. These numbers are $12L$ and $12KL$ respectively for the forward link.

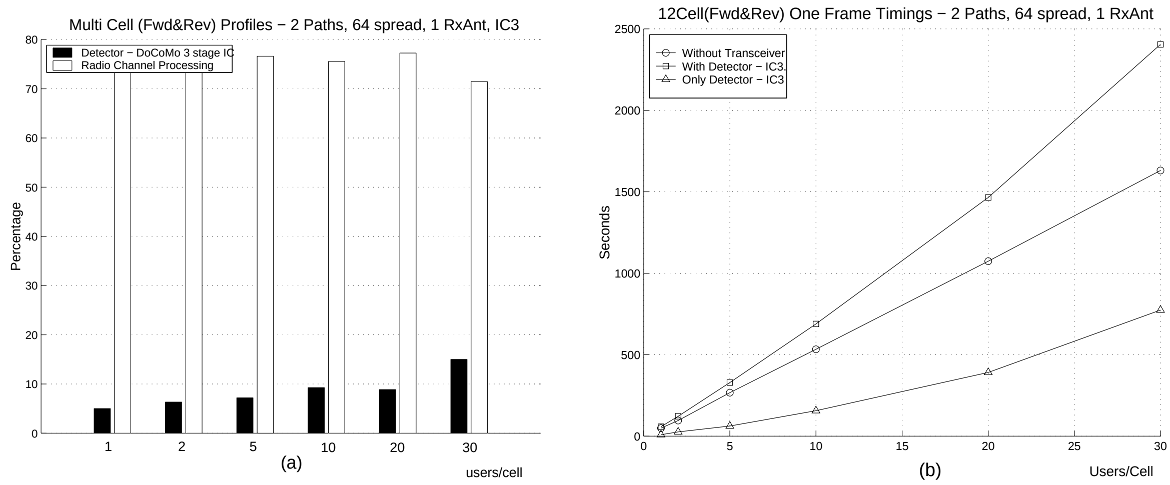


Figure 5.11: Multi cell (forward and reverse link) speeds (IC3, 2 paths) : The curves(b) show the time taken to completely process one frame for each of the x users per cell ($12x$ WCDMA frames in total) in WiPPET$_{\text{signal}}$. The bar graph(a) is the same as in Figure 5.5(a), and is given as a reference for the relative complexities of the radio channel and the detector. It is clear that the radio channel takes up most of the time in the simulation

We calculate the time required to completely process (create, interfere and propagate, and re-create) one WCDMA frame from each of the $12K$ users. On the reverse link, there are 12 detectors in the system; each one used to decode K users. On the forward link, there are $12K$ receivers, but they only despread the pilots of the forward link and hence the receivers are far less complicated than than the reverse link detectors. Figure 5.11(b) illustrates the time in seconds to process one complete WCDMA frame for a two path Rayleigh channel with a 3-stage interference canceller. It is evident that for all cases the radio channel processing takes up most of the time.

Figure 5.12 gives the total number of bits processed by WiPPET$_{\text{signal}}$ in one second. We observe that the processing rate for the case of Radio Channel Processing is constant. This is due to the linear increase in processing requirement for the Radio Channel as compared to a quadratic increase in the case of an Interference Canceller.

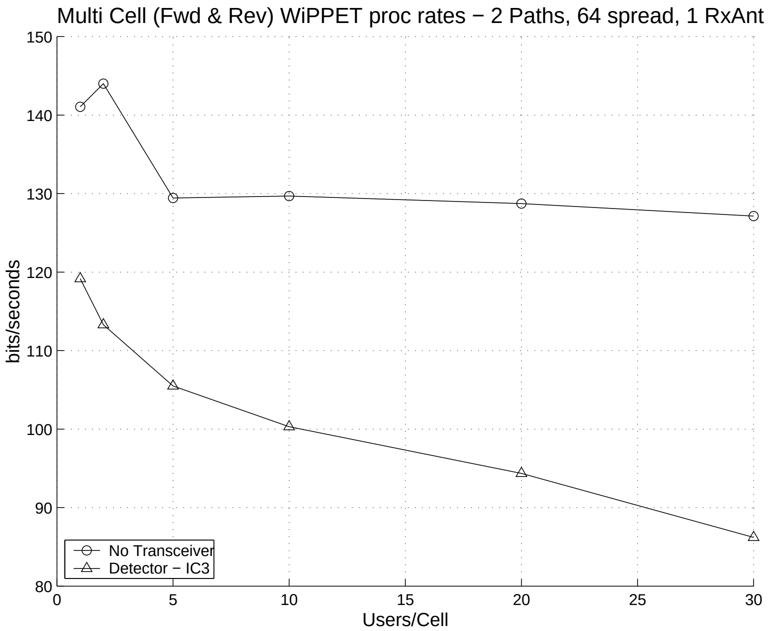


Figure 5.12: Multi cell (forward and reverse Link) processing rate : The processing in the Radio Channel increases linearly with the number of users, whereas the increase is non-linear for Interference Cancellation detector. This results in constant processing rates for the setups when the transceiver is absent

5.4 Parallelization

Parallel simulation performance is measured in terms of execution time speedup and efficiency based on experiments performed in a multi-processor environment. This section introduces two possible methods of parallelization of SSF based simulations. Their merits and restrictions are discussed and the reasons for choosing one over the other for WiPPET$_{\text{signal}}$ are developed.

5.4.1 Parallelization Schemes

A successful parallelization of a simulation should have the property of scalability, i.e., the execution time on N processors should be approximately N times smaller than the execution time on one processor. In practice, however, the reduction of simulation time

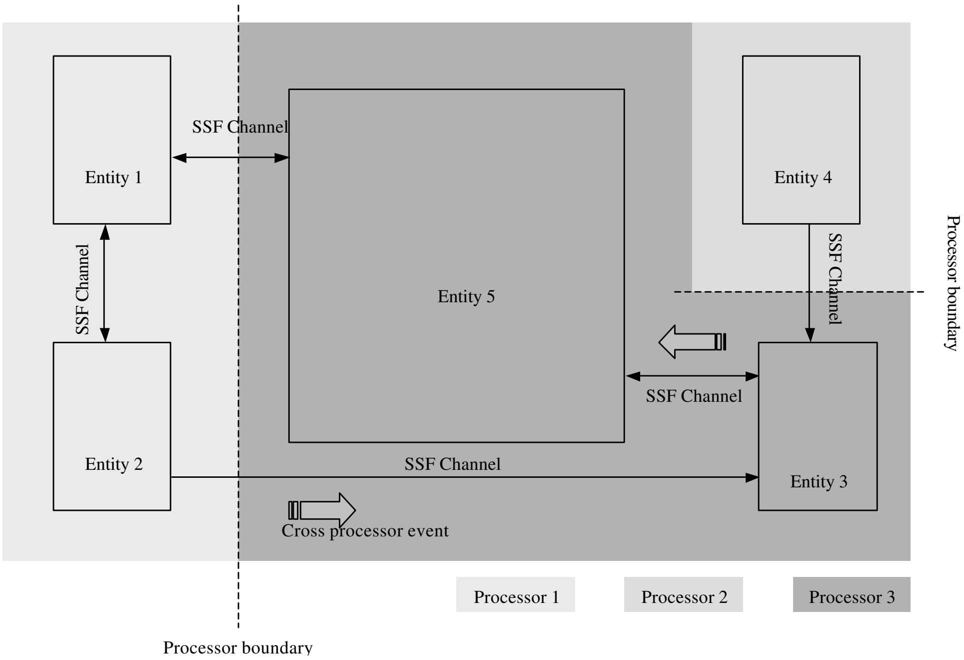


Figure 5.13: Entity parallelization scheme : SSF entities are grouped and aligned to different timelines that execute on different processors. Load balancing is an important criterion that decides the speedup.

is smaller due to reasons varying from load balancing and cross processor events. We will discuss two main parallelization schemes: a) Entity parallelization and b) Per-event parallelization.

Entity Parallelization - Optimal partitioning

Entities in a serial SSF are *aligned* to one common *timeline*. This means that all the processing that has to take place at a particular SSF instant is done in all the entities, and only then does the simulation proceed to the next simulation instant. Entity parallelization utilizes parallel processors to align entities or groups of entities to different timelines, which parallelizes the simulation. Figure 5.13 illustrates one such setup where 5 entities have been divided onto 3 groups to run on 3 processors.

In order to have appreciable speedups and to minimize parallelization costs, it is desirable that the simulation time progresses at the same rate on every processor. This is called *synchronization* of simulation time among processors, which is directly related to load balancing. It maybe possible before starting a simulation to distribute simulation load approximately equally among processors by mapping equal number and same type of SSF entities to each processor. However, the method is only approximate because the dynamic nature of the simulation allows that the same entity can transit between idle or compute-intensive behavior. Besides, if one particular entity dominates the computation all the time, it is not possible to partition that entity into multiple timelines using this scheme. The other major factors are the number of *cross-processor* events and the time between two cross-processor events coming into a timeline (processor). If a simulation is parallelized such that there are a lot of cross-processor events, and the time between the events is small, there is too much synchronization overhead. As a result, a very negligible speedup might be observed in extreme cases.

In order to solve parallelization problems, we must adopt partial empirical methods with the following aims:

- *Computational load* must be balanced: The number of cycles spent processing events or timers on each processor should be roughly the same for all processors
- *Communications load* must be minimized: The rate of cross-timeline events should be as small as possible. This essentially means that the time the processors wait for other processors or the number of cycles spent spinning on locks while sending/receiving events sent on channels between timelines should be minimized

The first step is to time the per-event workload, and figuring out how many events each

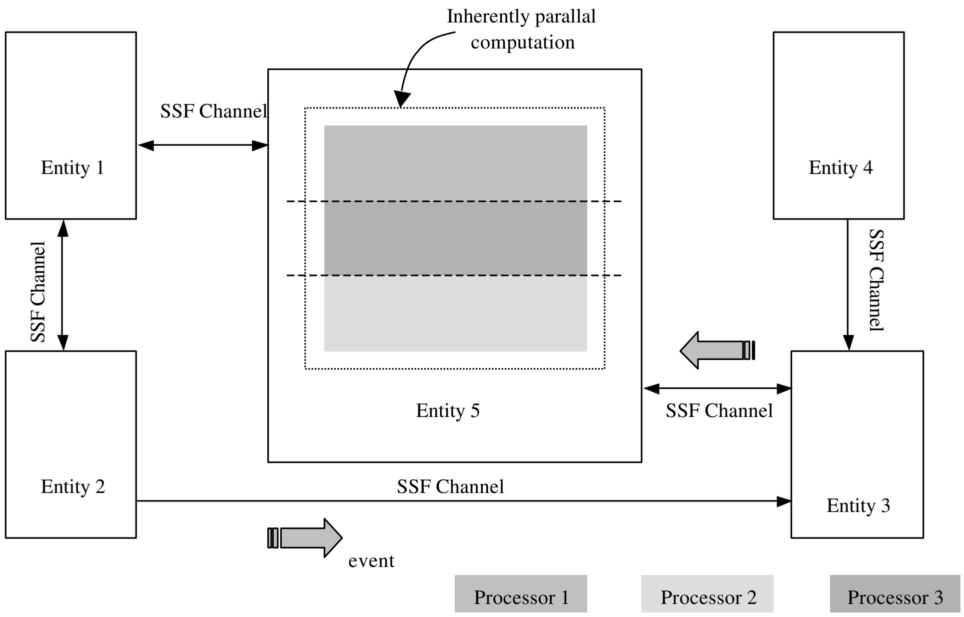


Figure 5.14: Per-event parallelization scheme : For a classically parallel workload, the computation of entity 5 is divided into multiple processors. The heavier the workload, the better the speedup performance.

entity will process locally. Then the entities have to be organized into partitions so that the minimum delay on a channel connecting two timeline partitions is maximized; this maximizes the number of events that can be transferred per window of synchronization between timelines, and minimizes the number of synchronization barriers that the workload needs to amortize to get good speedup. This process almost certainly requires a long period of tuning and analysis, especially for complex models.

Per-Event Parallelization

Another approach to get speedups from a simulation is applicable only if the processing in certain entities is suited to parallelization. If the per-event processing is heavy enough (consumes many cycles per event), it is possible to use a *per-event* parallelization scheme in SSF. Figure 5.14 illustrates the situation in which the processing (could be one or more SSF processes) of one of the entities is parallelized using this scheme.

The scheme utilizes system threads and allocates them to different processors. The following sequence is followed:

1. Spin multiple concurrent threads from within the SSF process body
2. Do the computation in parallel
3. Join those threads back together
4. Return control to the SSF framework

If the per-event workload is *classically parallel* it might be a better use of parallel resources (easier to tune) than trying to come up with the optimal partition for the discrete-event simulation itself as described in the previous subsection.

5.4.2 WiPPET$_{\text{signal}}$ in Per-Event SSF parallelization

WiPPET$_{\text{signal}}$ employs the per-event parallelization scheme. Using optimal partitioning for SSF simulations can't be used for a few reasons. Firstly, as stated earlier, workload balancing for complex simulations (like WiPPET$_{\text{signal}}$) is a highly sophisticated and non trivial problem. After a lot of tuning and analysis, it is possible to partition the entities in a certain way so that the workload is somewhat balanced. However, the more critical factor is the need to maximize the minimum time delay in cross processor events. This is difficult to achieve in WiPPET$_{\text{signal}}$ because of design tradeoffs in the radio channel entity. Data events in WiPPET$_{\text{signal}}$ arrive to and from the radio channel. These events carry a payload which is stored in the channel. A separate buffer has to be provided for each transmitting and receiving entity that sends and receives events from the channel. In order to maximize the window between arrival of events, the payload size could be increased, thus transferring more data per event. However, this in turn causes an increase in the memory requirement, which is also a system resource that is not in abundance. The payload size had thus been fixed; but it still is not large enough to have a large delay between event arrival.

`CHANNEL` parallelization

The radio channel implementation (`CHANNEL` class) gets instantiated as the `RevChannel` for reverse link simulations and `FwdChannel` for forward link simulations. The following sequence of processing is used:

1. Receive contributions from all transmitters. Create multipath components and chip delay timings.
2. Compute the composite signals at each of the receivers. Dispatch the created signals to individual receivers.
3. Repeat cycle from step 1.

The computation of composite signals at each of the receivers (step 2) is a classically parallel problem. After all transmitter contributions are received, the composite signal at each of the receivers is computed using equation (3.1). Let us again refer to equations (5.3) and (5.2) which give the received signal for a single transmitter x with one transmitting antenna and a single receiver y with one receiving antenna in a one path system. Now, assume that there are K transmitters and L multipaths. Each received chip at receiver y is given by

$$r_y = \sum_{x=0}^{K-1} \sum_{l=0}^{L-1} s_x(nT_c - \tau_{x,y,l}) G_{x,y,l}(nT_c) \qquad 0 \leq n \leq (N-1) \tag{5.45}$$

where $G_{x,y,l}$ and $\tau_{x,y,l}$ are the channel gain and integer chip timing offset for signal component (x, y, l). The signal for the symbol with N chips is given by

$$= [r_y(0)\ r_y(T_c)\ r_y(2T_c) \ldots\ r_y((N-1)T_c)] \tag{5.46}$$

where r_y is given by equation (5.45). Now, assume that there are B receivers. In this case, the radio channel has to do the computation in equations (5.45) and (5.46) B times, once each for each receiver. Once the transmitter contributions s_x and the channel gains $G_{x,y,l}$ are received, the computation of the composite signal at one received node is **NOT** dependent on the result of the computation of the composite signal at some other receiver node. The same logic can be extended to multiple receiver antennas and multiple paths.

The implementation of this parallelization is illustrated in Figure 5.15. The receivers for the reverse and forward links are divided into groups. One processor handles two groups (one from the MS and one from the BS). The simulation tries to load all the threads equally, which maximizes the speed. In case that is not possible, all but one threads are allocated a maximum equal number of groups, and the last remaining one is given the remaining receivers. Speedups for this setup are detailed in the next section.

As described, the parallel problem is partitioned into multiple threads and each partition is solved independently by each thread. However, there is overhead involved in the

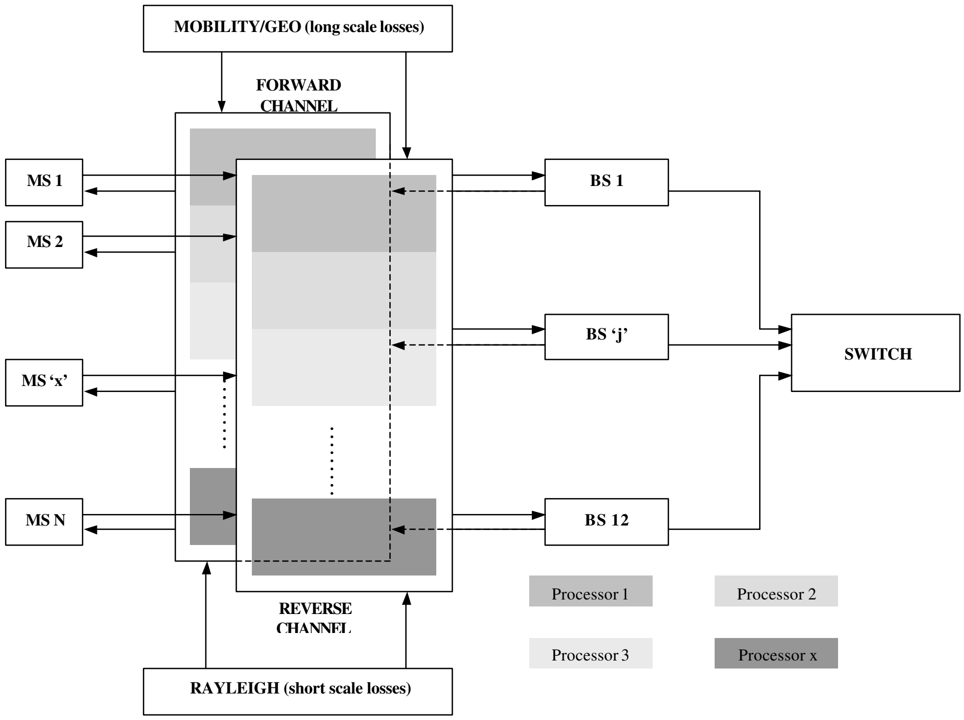


Figure 5.15: WiPPET$_{\text{signal}}$ `CHANNEL` parallelization scheme : `RevChannel` and `FwdChannel` entities are parallelized.

spinning of the threads before the computation can begin, and after the computation ends. Figure 5.16 depicts the overhead involved in the processing of one symbol. If the load of the parallel problem is large enough to show appreciable gains while going from a certain number of processors to a higher number of processors, then the overload may have negligible effect. However, if the load is small, then appreciable speedups might not be seen as the speedup obtained from the parallelization will be reduced by the non-negligible overload. Instances of this effect will be discussed in Section 5.5.

Transceiver parallelization

The bit detection implemented in WiPPET$_{\text{signal}}$ is executed serially in each of the BSs. Though these computations in the BSs are independent of each other and form

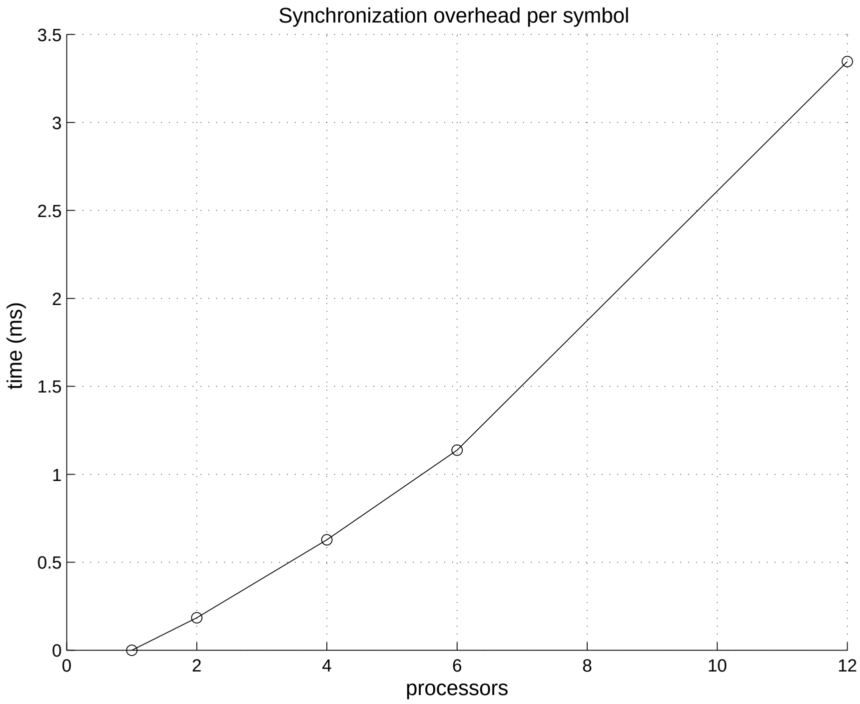


Figure 5.16: WIPPET$_{\text{signal}}$ thread synchronization overhead: The spinning and joining of threads requires an overhead that increases non-linearly as the number of threads increase

a classical problem in parallel computation, the per-event parallelization scheme can't be used because the `Base`s are different SSF entities.

We have modified the design so that the bit detection is now done in another SSF entity named `PARALLEL`. This entity acts as a *server* to the various BSs that are its *clients*. Once the BSs create the information that is needed to decode the frame, they transfer this information and the data frame to the server. On the reception of all *jobs*, the server entity then uses per-event parallelization as shown in Figure 5.17. The jobs are divided into groups, and each thread handles one group. Speedups for this arrangement are detailed in the next section.

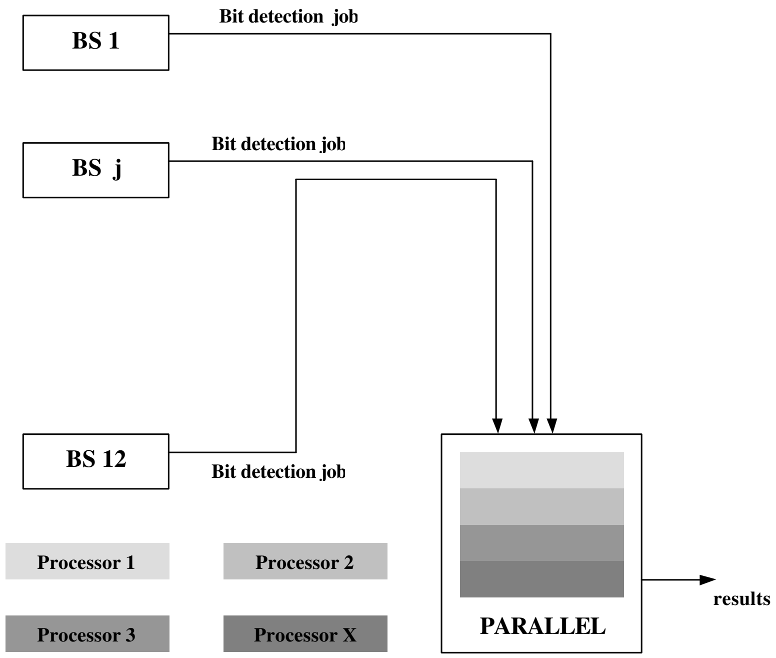


Figure 5.17: WiPPET$_{\text{signal}}$ Transceiver parallelization scheme : The BER computation jobs in the BSs are parallelized by forwarding them to another entity that uses per-event parallelization.

5.5 Speedups

The *speedup* of N processor simulation is defined as the ratio of computing time or execution time on a uniprocessor to the computing time on N processors. We will not parallelize any single cell simulation with the receiver having only one receiver antenna because the problem is not classically parallel. This section deals with multi cell (12 cell for WiPPET$_{\text{signal}}$) cases, although a single cell case with multiple receiver antennas can also be parallelized using the same concept.

The experiments have been done on an Ultra-Enterprise Sparc running Sun Solaris OS with 14 processors, each roughly equivalent to an ultra-30. Results for parallelization on 2, 4, 6 and 12 processors have been illustrated.

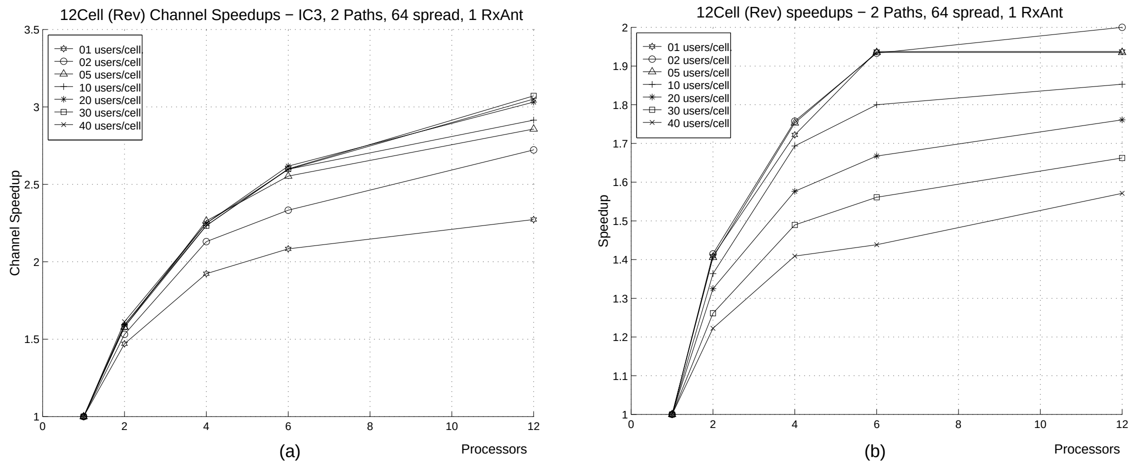


Figure 5.18: Speedups - reverse link (only `CHANNEL` parallelized), 12 cells, 2 paths, 1 RxAnt : Channel speedups (a) increase as the number of processor and users increases. System speedups (b) decrease when the users increase

5.5.1 `CHANNEL` Parallelization Speedups

This section details the speedups achieved when only the `CHANNEL` entity is parallelized. Transceiver parallelization speedups are discussed separately in section 5.5.2.

Multi Cell Reverse Link

Only the `RevChannel` is parallelized in this scenario. In order to observe the speedups due to just the channel parallelization, we have done the timing calculations for WiPPET$_{\text{signal}}$ when the receiver and transmitter signal processing is absent.

Figure 5.18(a) gives the speedups for the case when the transceiver processing is absent. As the number of users/cell increase, the speedups also increase because for higher number of users, the channel processing dominates the simulation, and all other entities and SSF processing reduces. Also, as expected, as the number of processors increase, the speedup also increases, although less than linearly. Figure 5.18(b) shows the system speedups, that is, the speedup when the simulation includes the transceiver processing

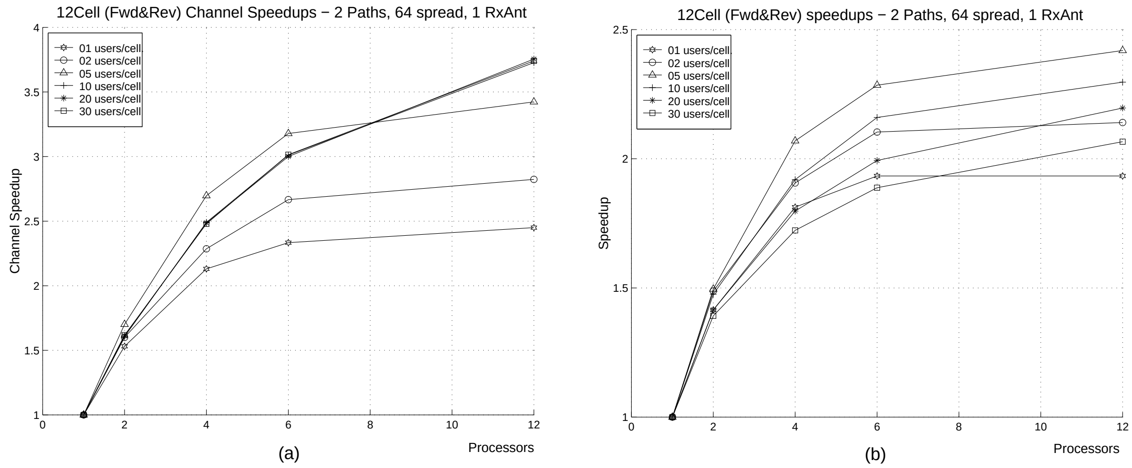


Figure 5.19: Speedups - forward and reverse link (only `CHANNEL` parallelized), 12 cells, 2 paths, 1 RxAnt : Channel speedups (a) increase as the number of processor and users increases. System speedups (b) decrease when the users increase

and the parallelized radio channel for reverse link simulation. The following observations can be made for system speedups in Figure 5.18(b) vis-a-vis the channel speedups in Figure 5.18(a).

- As the number of users increase, the system speedup decreases. This is unlike the case for channel speedups in Figure 5.18(a) because here, the detector processing, which has **NOT** be parallelized, grows heavier as the load increases.

- As the number of processors increase, the system speedup in Figure 5.18(b) is not as much as in the case of channel speedups in Figure 5.18(a). This is again due to the fact that when the detector is present, the time used in detector processing reduces the speedup.

Multi Cell Forward and Reverse Link

Both the `RevChannel` and the `FwdChannel` is parallelized in this scenario. In order to observe the speedups due to just the channel parallelization, we have done the timing calculations for WiPPET$_{\text{signal}}$ when the receiver and transmitter signal processing is absent.

Figure 5.19(a) shows the channel speedups for the case when the transceiver processing is absent. As the number of users/cell increase, the speedups also increase because for higher number of users, the channel processing dominates the simulation, and all other entities and SSF processing reduces. Also, as expected, as the number of processors increase, the speedup also increases, but again not linearly. Figure 5.19(b) shows the system speedups for forward and reverse channel simulation. Again, because of the presence of the detector and transmitter signal processing, the system speedups are lower than the channel processing speedups. Also, as discussed in Section 5.4.2, for lesser number of users, since the processing load is small, negligible speedups are seen as higher number of threads are used for parallelization.

5.5.2 Transceiver Parallelization Speedups

This section details the speedups achieved when both the `CHANNEL` and the transceiver are parallelized.

Multi Cell Reverse Link

The `RevChannel` and the transceiver bit detection functionality is parallelized in this scenario. Figure 5.20(a) gives the system speedups for reverse channel simulation. The following observations can be made for this case vis-a-vis the system speedups in Figure 5.18(b) where only the `CHANNEL` is parallelized.

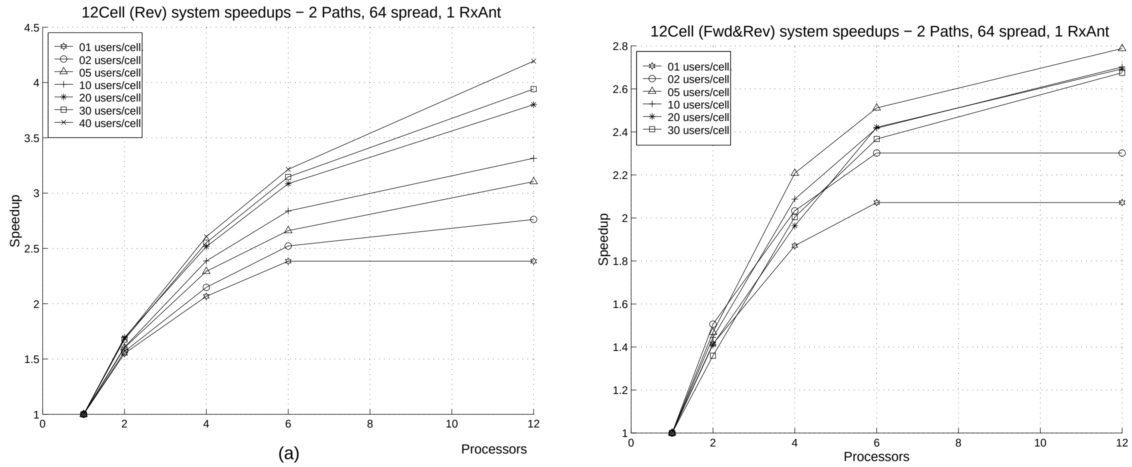


Figure 5.20: Speedups - (both `CHANNEL` and transceiver parallelized), 12 cells, 2 paths, 1 RxAnt : System speedups for reverse link (a) increase as the number of processor and users increases. System speedups for both forward and reverse link (b) decrease when the users increase

- The speedup for each combination of users and processors is higher than its corresponding value for Figure 5.18(b) because now the bit detection in the transceiver has been parallelized.

- As the number of users increases, the system speedup in Figure 5.20(a) increases. This is unlike the case in Figure 5.18(b) where the situation is the opposite. This is due to the fact that for a multi cell (rev only) case, as the users increase, the complexity of the bit detector increases. We exploit this higher complexity to obtain better speedups.

Multi Cell Forward and Reverse Link

The `RevChannel`, `FwdChannel` and the transceiver bit detection functionality is parallelized in this scenario. Figure 5.20(b) gives the system speedups for forward and reverse channel simulation. The speedups are higher than the corresponding system speedups in Figure 5.19(b) because now the transceiver parallelization has also been exploited. Also, as discussed in Section 5.4.2, for lesser number of users, since the processing load is small, negligible speedups are seen as higher number of threads are used for parallelization.

Chapter 6

WiPPET$_{\text{signal}}$ Validation and Sample Results

Before a simulator can be used to extract statistics, some kind of system validation has to be performed to make sure that the results are representative of the actual system characteristics. This validation can be done on a sub-system level basis (unit testing) or an end-to-end system basis. WiPPET$_{\text{signal}}$ has been validated using both techniques. This chapter will describe some of this validation and also sample results that have been obtained from WiPPET$_{\text{signal}}$.

6.1 Validation

Qualitative validity of the various modules in WiPPET$_{\text{signal}}$ will be discussed in this section. Some modules have been unit tested, while other modules have been validated using end-to-end system statistics.

6.1.1 Rayleigh Fading

The `Rayleigh` module in WiPPET$_{\text{signal}}$ is used to generate the short scale Rayleigh fading values for all the possible signal paths in the system. We have done the validation on one instance of the fader. Since all Rayleigh faders are identical instances, the results

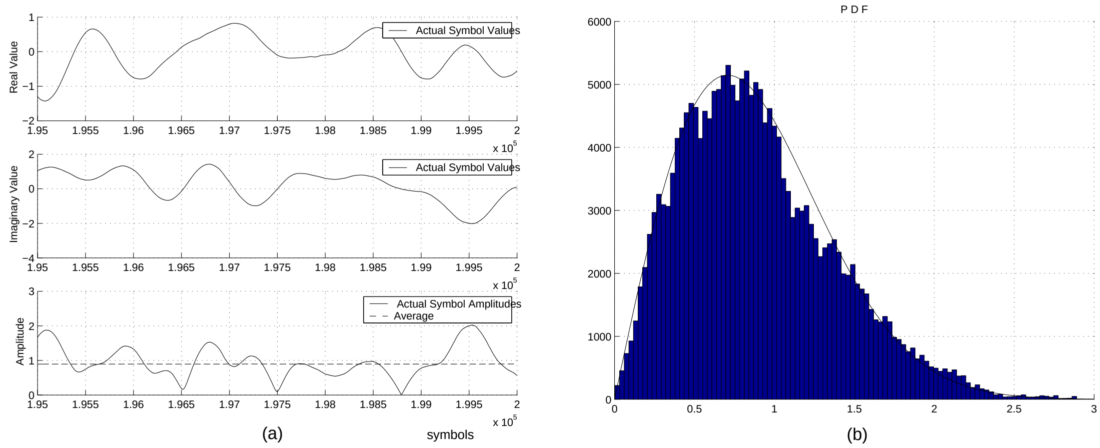


Figure 6.1: Rayleigh Fader Values and PDF : Plot (a) gives the real, imaginary and amplitude values of the rayleigh fading values that have been generated. Plot (b) gives a histogram of the real component of the fader values and the corresponding PDF of the real component

are representative of all faders in the system.

The performance of the fader was evaluated by running the fader for 200,000 samples at a sampling frequency of 64 ksamples/s and a doppler frequency of 80 Hz. This roughly gives us 3.125 seconds of real time fading data. Figure 6.1(a) gives a plot of the real component, the imaginary component and the amplitude of the complex fading values generated by the fader. The average value of the amplitude has also been plotted in Figure 6.1(a). Figure 6.1(b) gives a histogram of the real component samples. The PDF of a Rayleigh distribution has also been plotted in the figure for the purpose of comparison. The autocorrelation of the Rayleigh fading is a Bessel function of order zero. Figure 6.2(a) plots the auto correlation of the real component and of the imaginary component. The Bessel function is also plotted to give an indication of the error in correlation of fader values. These autocorrelation results were then passed through a Kaiser Window and the logarithmic version of the FFT is plotted in Figure 6.2(b). As is expected, the curve should peak at the Doppler frequency of

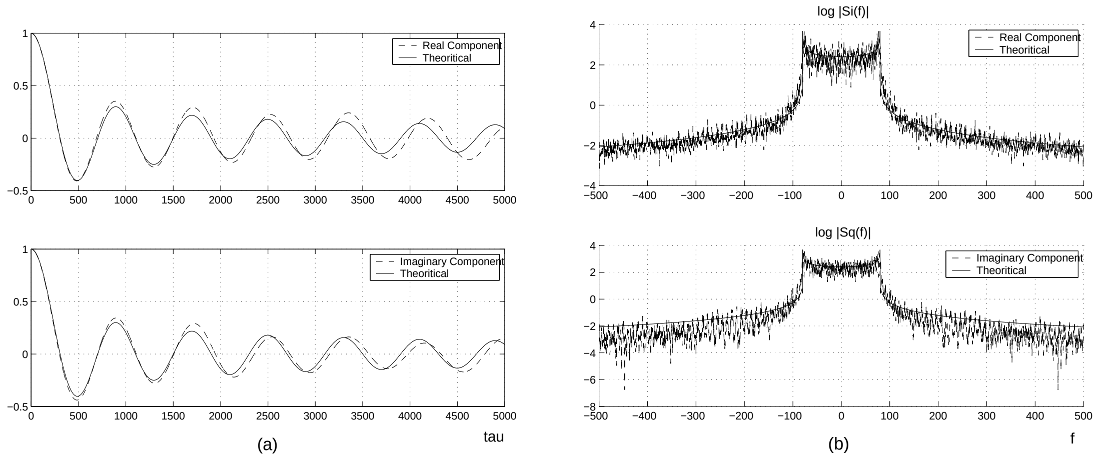


Figure 6.2: Rayleigh Fader Correlations and PSD : Plot (a) gives the auto correlation of the real and imaginary component. Plot (b) gives the spectral density of the real and imaginary component

Seed	*Receiver*	*Chip Resolution Errors*	*Symbol Resolution Errors*	*Bit Mismatch*
A	16-MF	1045	1044	3
A	64-IC	227	227	0
B	16-MF	1068	1069	1
B	64-IC	252	253	1
C	16-MF	1011	1011	4
C	64-IC	228	228	0

Table 6.1: Qualitative effects of resolution in Rayleigh fader : Experiment performed on 1 Cell, 5 users/cell, 10 frames/user ,1 RxAnt, 2 Paths

80 Hz. This exercise validates the reasonableness of the approximate fader that has been utilized in WiPPET$_{\text{signal}}$.

In section 5.2.4, we discussed how reducing the number samples of the Rayleigh fader increases the speed of the simulation. However these Rayleigh fader quantitative results are justified only if qualitative results are not compromised. To examine this issue, let us first define

- *Symbol resolution*: In this case, the same complex valued Rayleigh sample is used for all chips of a symbol on a signal path in the system. A new sample for each signal path is generated every N chips, where N is the spreading factor.

- *Chip resolution*: In this case, a separate complex valued Rayleigh sample is generated and used for every chip on a signal path.

Note that N times more samples are generated for chip resolution than for symbol resolution for each signal path. We also define a quantity called *bit mismatch* as the number of bit positions where the two methods (chip resolution and symbol resolution) give different bit decisions over all receivers. We claim that the approximation in precision is called for when *bit mismatch* rate is far less than the actual bit error rate. That would mean that even if we use the symbol approximation (which saves a LOT of time and memory) we will still be able to come to that same conclusions about the BER statistics. In particular, we are not compromising on the quality of the results to gain speed.

Table 6.1 illustrates sample experiments performed for 10 frames per user for a single cell system with 5 users. In Table 6.1, a receiver type of 16-MF denotes a matched filter bit detector with spreading factor 16 while 64-IC denotes an interference canceller detector with spreading factor 64.A total of 28800 bits were processed for each run. Let us now understand the implications of the (Seed C, 16-MF) run. The number of bits in error for both the symbol resolution and chip resolution are the same. That implies that both cases provide the same BER statistic. However, that is not the only benchmark for making sure our approximation didn't introduce any errors. The *bit mismatch* for this case is 4, which means that out of the 28800 bits processed, the hard decisions in the two cases differed only in 4 places. That low ratio is the benchmark for our check for the quality of the approximation. Similarly, the (Seed C, 64-IC) run did

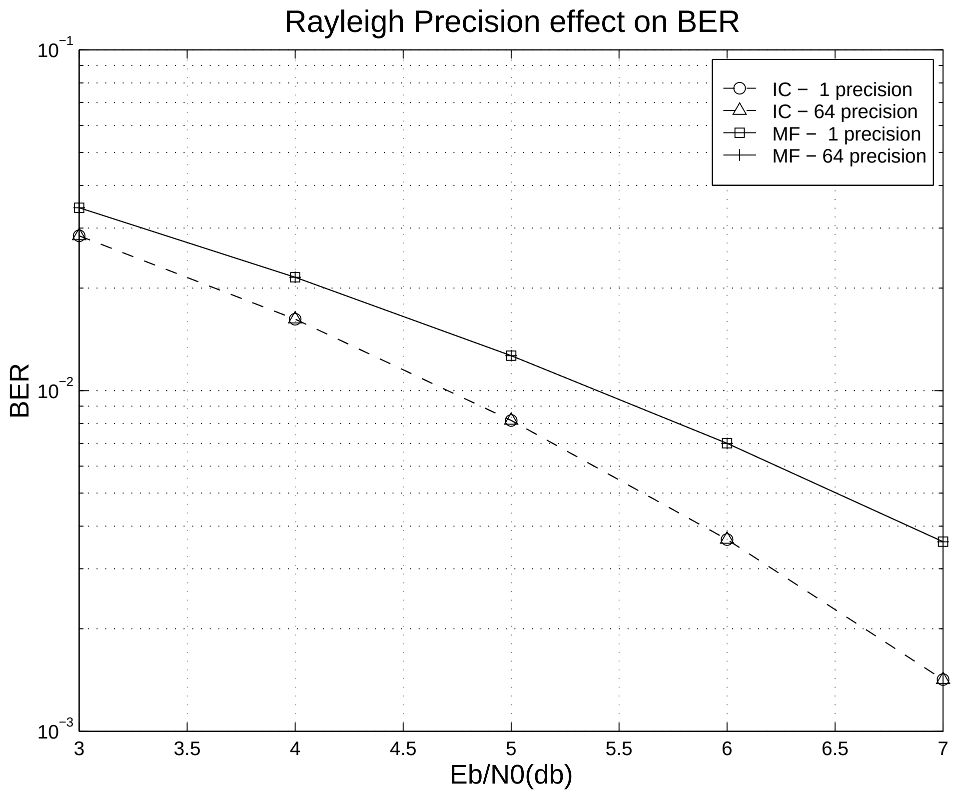


Figure 6.3: Effect of Rayleigh precision on BER : The change in the error rates because of maximum (64) or minimum (1) precision is negligible.

not introduce any significant *bit mismatch.*

End-to-end BER statistics for the symbol resolution vs. chip resolution case for a 3-stage interference canceller(IC) and a matched filter (MF) are illustrated in Figure 6.3. As is evident, the change in precision has almost negligible effect on the BER. The actual values that have been plotted are given in Table 6.2.

6.1.2 Shadow Fading

The `Mobility` module in WiPPET$_{\text{signal}}$ is coupled with the geography to precompute the shadow fading losses for all possible points on the grid. We have done the validation by generating the shadow fading samples for a part of the geography.

EbN0 (dB)	*Precision*	*Detector*	*Error Bits*	*Total Bits*	*BER*
3.0	1	IC	60007	2108160	2.846416e-02
3.0	64	IC	60062	2111040	2.845138e-02
3.0	1	MF	60045	1745280	3.440479e-02
3.0	64	MF	60062	1745280	3.441912e-02
4.0	1	IC	34014	2099520	1.620085e-02
4.0	64	IC	34038	2099520	1.621228e-02
4.0	1	MF	40013	1860480	2.150682e-02
4.0	64	MF	40021	1860480	2.151112e-02
5.0	1	IC	17156	2099520	8.171868e-02
5.0	64	IC	17171	2099520	8.179012e-02
5.0	1	MF	26577	2099520	1.265908e-02
5.0	64	MF	26582	2099520	1.266147e-02
6.0	1	IC	7677	2099520	3.656550e-02
6.0	64	IC	7679	2099520	3.657503e-02
6.0	1	MF	14709	2099520	7.005887e-02
6.0	64	MF	14687	2099520	6.995408e-02
7.0	1	IC	2980	2099520	1.419372e-02
7.0	64	IC	2975	2099520	1.416991e-02
7.0	1	MF	7561	2099520	3.601299e-02
7.0	64	MF	7557	2099520	3.599394e-02

Table 6.2: BER deltas due to Rayleigh resolution: 1 Cell, 5 users per cell, 1 RxAntenna, 2 Paths

In particular, $40,000$ shadow fading values were generated with a distance precision of 0.5 m and a correlation distance of 50 m. The standard deviation of the log normal variable was 7.0 dB. Figure 6.4(a) gives a sample plot of the correlated shadow fading values seen over a distance of about 250 m. in the dB scale. A histogram of the fading values in dB scale is plotted in Figure 6.4(b). A Gaussian marginal PDF overlaid in Figure 6.4(b) shows good agreement.

6.1.3 Single User BPSK Error Rates

Single user BPSK characteristics and error rates for static channels and in presence of Rayleigh fading and path diversity have been analyzed in reference [38]. In particular,

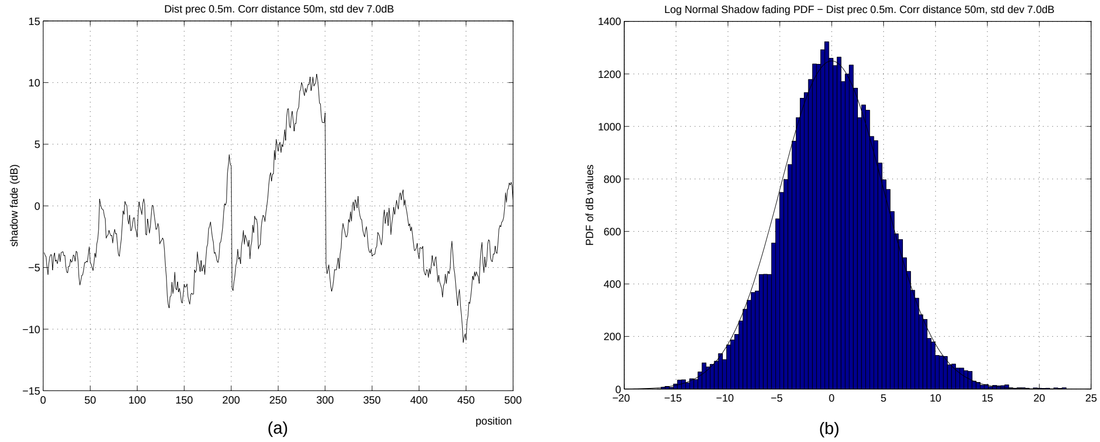


Figure 6.4: Shadow fading validation - Values and PDF : Plot a) gives a sample of shadow fading values generated over 250 m in the dB scale. Plot (b) gives a histogram of the fading values in dB scale and the corresponding PDF

the analytical results are given by

$$\mathrm{BER} = \begin{cases} Q\left[\sqrt{2\nu}\right] & \text{Static 1 path} \\ 0.5\left[1 - \sqrt{\frac{\nu}{1+\nu}}\right] & \text{Rayleigh 1 path} \\ 0.5\left[1 - \sqrt{1 - \frac{1}{(1+\nu)^2}}\right] & \text{Rayleigh 2 paths EGC} \end{cases} \tag{6.1}$$

where ν is the symbol SIR.

We ran WiPPET$_{\text{signal}}$ for a single MS and BS to find the error rates for various values of ν in a perfect power control setup. Figure 6.5(a) demonstrates the agreement between simulation and analytical results.

WiPPET$_{\text{signal}}$ has also been validated for static 1 path BPSK with soft Viterbi decoding with constraint length $K = 3$ and code rate $R_c = 1/3$. The algebraic expressions used are $g1 = [100], g2 = [101]$ and $g3 = [111]$. Reference [29] gives detailed analysis of the same rate coder and soft Viterbi decoder for the same polynomials. Figure 6.5(b) illustrates the theoretical bounds and the simulation results for the setup.

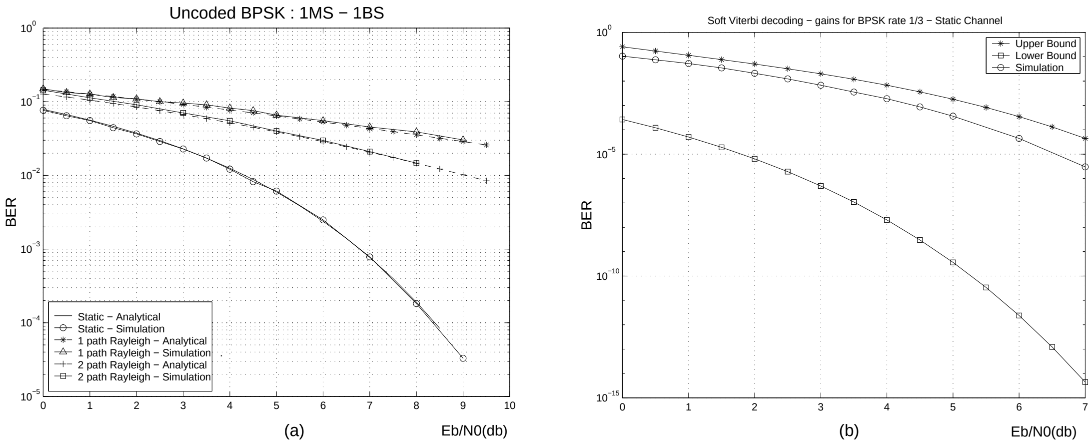


Figure 6.5: Single user BPSK error rates : The simulation results have been compared with the theoretical results to validate the end-to-end correctness of WIPPET$_{\text{signal}}$. The curves in (a) are for the uncoded scheme and the ones in (b) are for the coded scheme

6.1.4 Comparison with NTT DoCoMo Results

Multi user cases for uncoded single cell system for various configurations were simulated, and results checked against those calculated by NTT DoCoMo for the same configurations. This validated the end-to-end transmitter-channel-receiver design for WIPPET$_{\text{signal}}$.

Figure 6.6(a) illustrates the BER for a 10 user uncoded BPSK single cell system with a spreading of 64 when Rayleigh fading is absent. The results for the simulation results at WINLAB for the DoCoMo 3 stage interference canceller and matched filter closely the results obtained from NTTDoCoMo for the same cases. Similarly, Figure 6.6(b) illustrates the BER for a 10 user uncoded BPSK single cell system with a spreading of 64 when 2 path 80Hz Rayleigh fading is present.

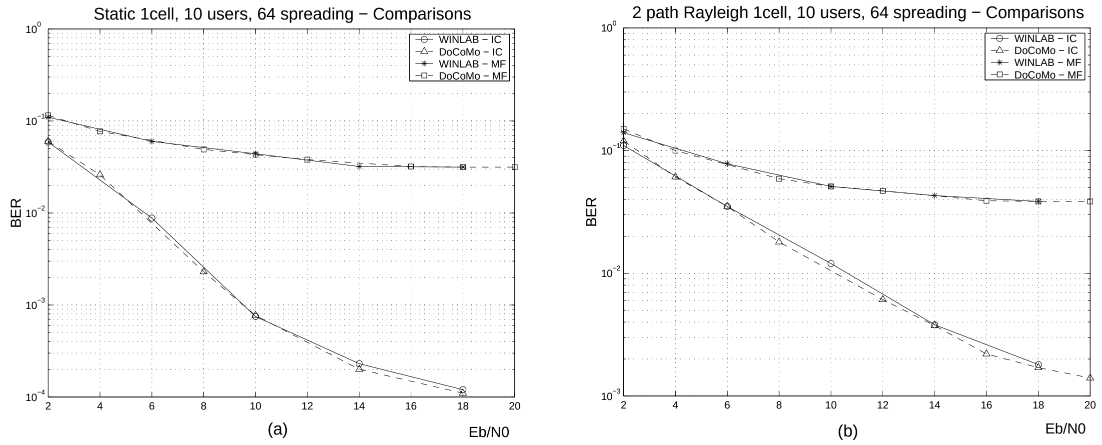


Figure 6.6: BER (WINLAB vs. DoCoMo) : BER curves match each other to validate the design and implementation of WiPPET$_{\text{signal}}$ for (a) static and (b) 2 path Rayleigh fading

6.2 Sample Results

WiPPET$_{\text{signal}}$ gives a number of choices in parameters that can be used to simulate the WCDMA system. They are given in Table 6.3.

Combinations of the above parameters creates simulation setups. As a sample, a certain combination of parameters has been given in Table 6.4(A) These parameter choices would correspond to a multi cell simulation with $3(= 36/12)$ mobiles per cell transmitting chips at a rate of 3.840 Mcps which would correspond to 60 Ksymbols/s for a spreading factor of 64. The simulation will only be for the reverse link, as the forward link simulations have been turned off. The base station receiver will be a simple matched filter and the transmitted power will be controlled for a SIR target of 6.0 dB and noise PSD of -150.0 dBm/Hz. The transmitted power has an upper limit of 30 dBm which is equal to 1 W. Also, rate 1/3 coding will be present. We will now discuss some sample cases.

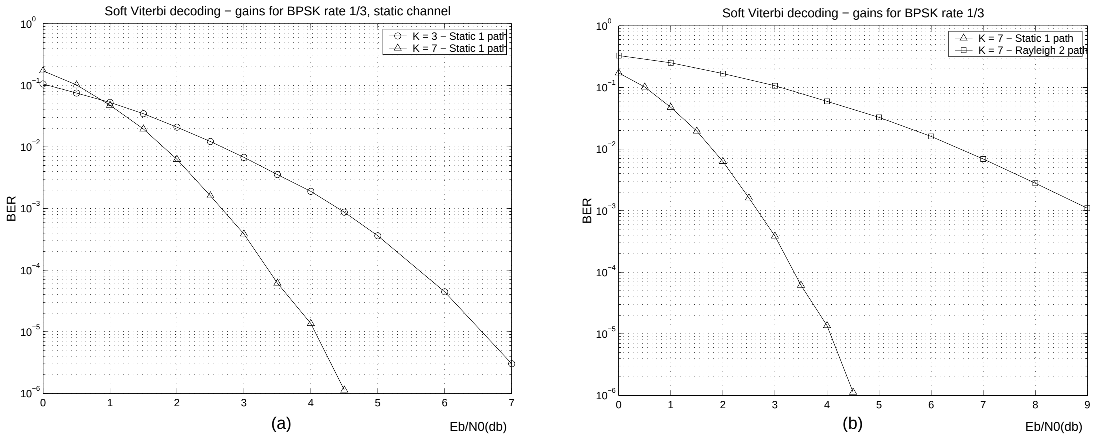


Figure 6.7: Single user BPSK (Rate 1/3 coding): Error rates for a perfectly power controlled single user system in (a) static and (b) Rayleigh fading

6.2.1 Error Rate Evaluation

Figure 6.7(a) gives the error performance of a single user BPSK system in a static channel for rate 1/3 coding. The polynomials used for constraint length $K = 3$ run are $g1 = [100]$, $g2 = [101]$ and $g3 = [111]$and for the constraint length $K = 7$ run are $g1 = [1111001]$, $g2 = [1011011]$ and $g3 = [1110101]$. The experiments have been performed for perfect transmit power control mode and for 1 path. Figure 6.7(b) gives the error performance for the $K = 7$ coding in a 2 path Rayleigh fading environment.

Figure 6.8(a) gives the error performance for a single cell with 10 users for an uncoded system. The matched filter (MF) and NTT DoCoMo 3 stage interference canceller (IC3) have been compared for perfect transmit power control. Experiments have been performed in a 2 path Rayleigh fading environment. The curves indicate the improvements in error rates given by the interference canceller. Figure 6.8(b) compares the error performance of the IC3 when rate 1/3 coding is used with constraint length $K = 7$ and polynomials $g1 = [1111001]$, $g2 = [1011011]$ and $g3 = [1110101]$.

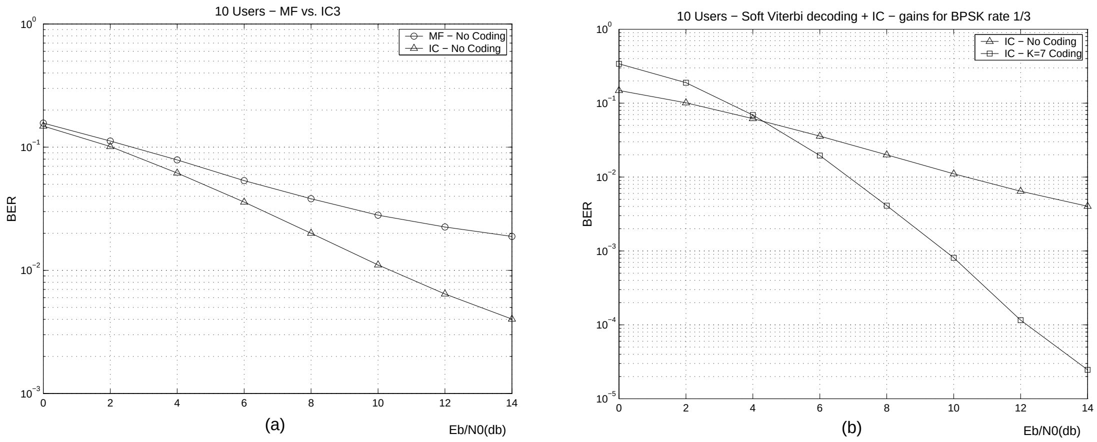


Figure 6.8: Multi user results (Rate 1/3 coding): Curve (a) provides error rates for a perfectly power controlled multi user system for a matched filter and interference canceller. Curve (b) compares the interference canceller when coding is used

6.2.2 Feasibility Study of Transmit Power Control

Transmit power control based on SIR is implemented in WiPPET$_{\text{signal}}$. In what is commonly called perfect transmit power control, the transmit power of the mobiles is adjusted so that a base station sees the same received power level for all mobiles in the cell. Every time the mobiles move, the change in the long scale loss is compensated by increasing or decreasing the transmitted power. Hence, the received power is roughly the same at all times. However, for the SIR based transmit power control, power is adjusted to meet an SIR objective based on feedback from the BS.

Section 4.9.4 introduced the logic and flow of SIR measurements and power control. To evaluate the transmit power control, we now define the following statistics:

- $\tilde{S}_t$: The measured average transmitted signal power.
- $\tilde{S}_r$: The measured average received signal power.
- $\widetilde{SIR}$: The measured ratio of average received signal power to average interference

power.

- $\widetilde{SNR}$: The measured ratio of average received signal power to AWGN power spectral density.

WiPPET$_{\text{signal}}$ calculates and updates these statistics every frame. If a particular mobile is below the target SIR, the regular feedback of power control commands (at 1500 b/s) causes more increase than decrease in transmitted power. However, once the maximum limit on transmitted power is reached, the commands for further increase have no effect at the MS, which in turn implies that that particular MS will not be able to attain the SIR objective. In this case, we say that the target SIR is *infeasible*. Feasibility depends on the spatial distribution of users and may change as users move.

As a sample study, we have chosen a one path Rayleigh single cell system with the following parameters of Table 6.4(B). The experiments have been done for 10 and 20 users. The target SIR varies from −4 dB to 14 dB and a maximum transmit power limit of 30 dBm has been put into place. The AWGN has a power spectral density of −150 dBm/Hz. Bit detection is being done using a NTT DoCoMo 3 stage interference canceller.

Figure 6.9 gives a plot between the fraction of users that have reached the maximum power limit of 30 dBm and the target SIR. As the target SIR requirement increases, the average transmitted power also increases and the number of users that have reached their maximum power limit increases. Figure 6.10(a) gives a plot between the average transmitted power $\hat{S}_t$ and the target SIR. The maximum upper limit of 30 dBm is also plotted. For higher target SIRs, more users approach the transmit power limit which is reflected in the average transmit power. A saturation is observed for very high target

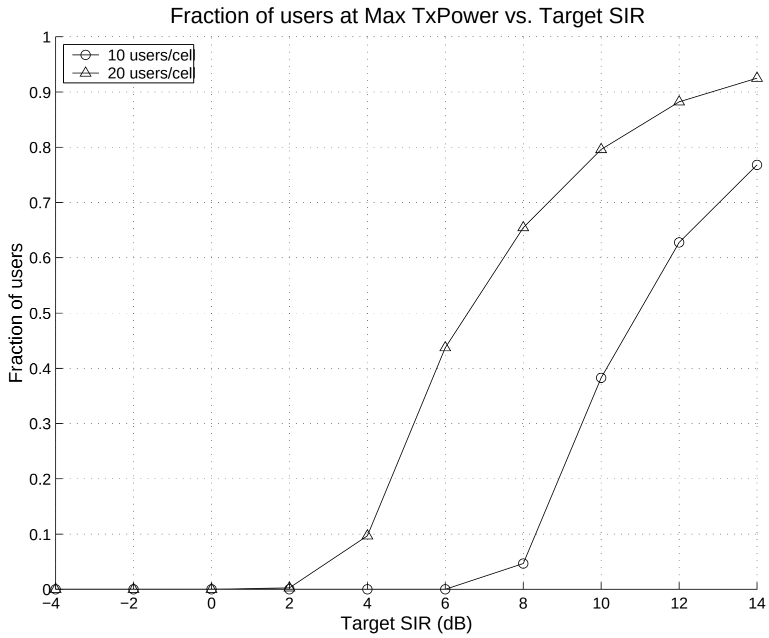


Figure 6.9: Fraction of users reaching maximum power limit: As the target increases, more users approach the target limit of 30dBm.

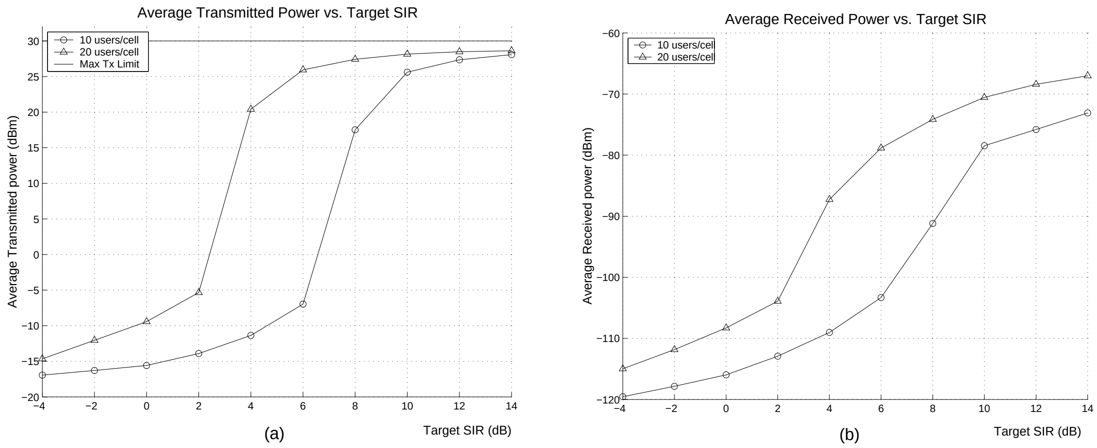
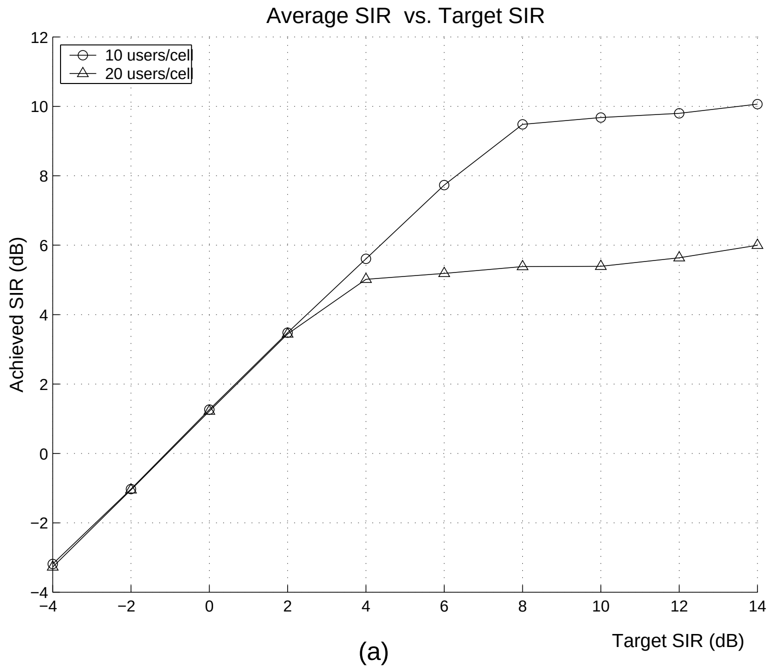


Figure 6.10: Average transmitted and received powers for a range of target SIRs: Plot (a) shows a saturation in the average transmit power of the system because more MSs have approached their maximum limit as the target increases. Plot (b) gives the average received power which also increases.

SIRs because for those targets, a large fraction of users have attained the maximum limit as shown in Figure 6.9. Figure 6.10(b) plots the average received power $\hat{S}_r$ vs. the target SIR. The trend in received powers is similar to that of transmitted powers in Figure 6.10(a).



(a)

Figure 6.11: Average $\widetilde{SIR}$ for a range of target SIRs: A linear increase in the average received signal to interference ratio indicates that the target SIR is being achieved. Saturation indicates infeasibility.

Figure 6.11 plots the measured average SIR $\widetilde{SIR}$ as a function of the target SIR. As the target SIR increases, a linear increase in $\widetilde{SIR}$ indicates that the target is being met by all the users. However, as the target increases further, a saturation in the values of $\widetilde{SIR}$ is seen. This indicates that very often the target SIR is infeasible. This is consistent with Figure 6.9 and Figure 6.10(a) which show that most of the users have reached their maximum power limit.

To examine the impact if choosing the target SIR too high, let us choose two points (target SIR 4dB and target SIR 10dB) on the curve for 20 users in Figure 6.11(a). It is evident that the target is being met in the 4dB case, whereas for the 10dB case, it is not being achieved and has saturated to around 5dB. Figure 6.12(a) gives the histogram of users' SIR samples for a target SIR of 4dB. A narrow bell around the target indicates that the target is being met and the system is feasible. Figure 6.12(b)

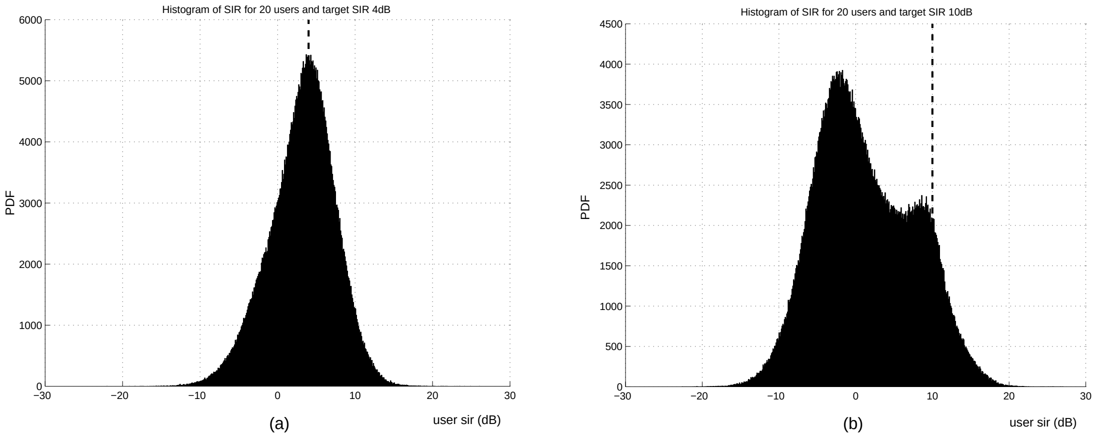


Figure 6.12: User SIR histograms indicating infeasibility for 20 users in single cell: Plot (a) gives the histogram of user SIRs for a target SIR of 4dB. A narrow bell around the target indicates that the target is being met and that the target SIR is typically feasible. Plot (b) gives the histogram of user SIRs for a target SIR of 10dB. The values for this case are not around the target SIR, which indicates frequent infeasibility.

gives the histogram of users' SIR samples for a target SIR of 10dB. As expected, the peak does not lie at the 10dB value, but at a much smaller value, in this case, near -4 dB. Also, the distribution is more spread indicating that there is a lot of variation in the users' SIR.

The 20 user case for 6dB (feasible) and 10dB (infeasible) target SIRs is also evaluated by comparing the average transmit power of each of the 20 users against the average SIR they achieve in Figure 6.13(a). It is clear that for the feasible target, almost all users are able to achieve it, whereas the infeasible target is achieved only by a few users. The transmitter powers for the infeasible case are more towards the upper limit of 30dBm. A result of the inability of the users to attain the 10dB target is given in Figure 6.13(b) where the users with bas average SIR have very bad BER. As a contrast, the 6dB target SIR case has all users around the SIR and almost the same BER performance.

Figure 6.14(a) gives a plot between the BER and the target SIR. Initially, as the target SIR increases, the BER decreases. However, for higher targets, a saturation

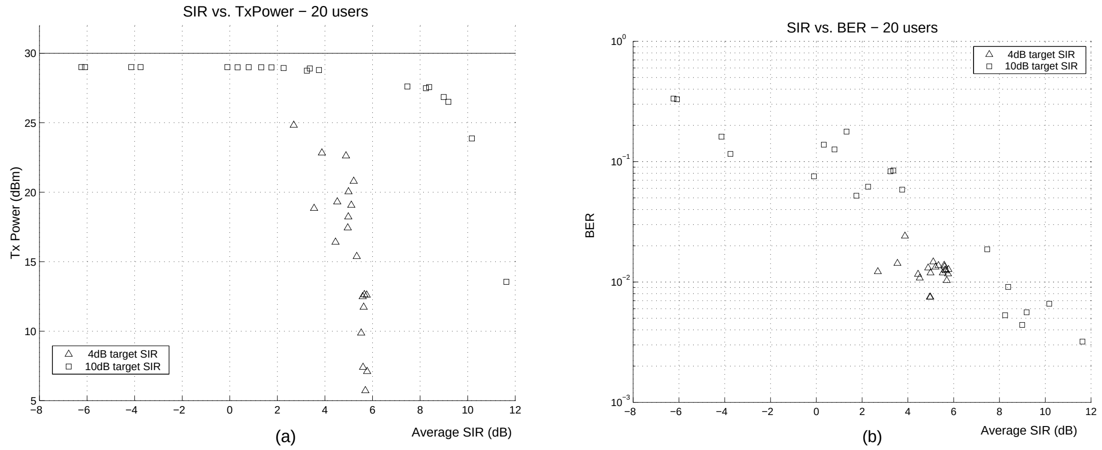


Figure 6.13: User Tx. powers and BERs indicating infeasibility for 20 users in single cell: Plot (a) gives the average transmit power of each of the 20 users against the average SIR they achieve for the 6dB feasible and 10dB infeasible target SIR. Plot (b) gives the average BER against average SIR for each of the 20 users for the 6dB feasible and 10dB infeasible target SIR.

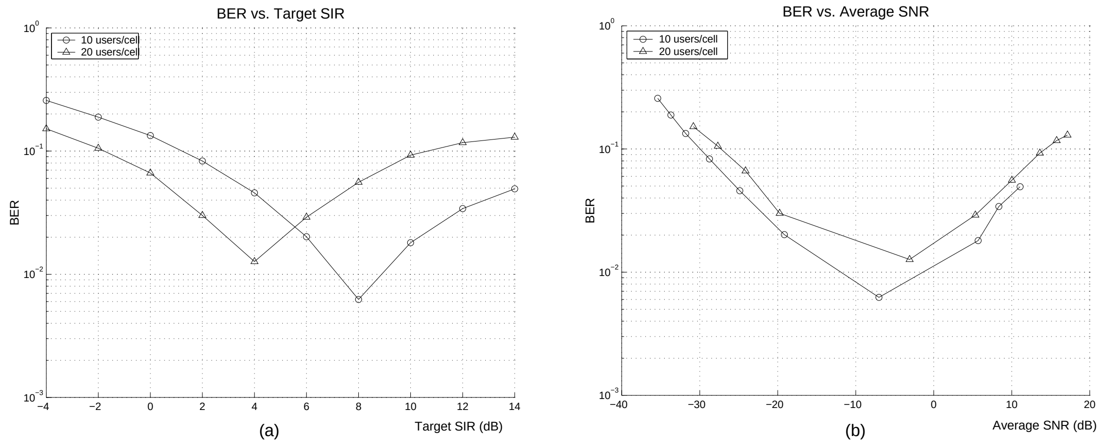


Figure 6.14: BER curves indicating infeasibility: Plot (a) gives the BER against the target SIR. Plot (b) gives the BER against $\widetilde{SNR}$. Both curves decrease to a minimum BER indicative of the highest SIR target that is feasible. An increase in BER indicated increasing levels of infeasibility

in the average transmitted powers (Figure 6.10(a)) is observed. The users with very low SIRs cause a degradation of the average performance. Figure 6.14(b) gives a plot between the BER and the average received signal to interference ratio (Figure 6.11(b)).

6.2.3 Soft Handoff Gains

WiPPET$_{\text{signal}}$ implements soft handoff which has been discussed in Sections 4.2, 4.4 and C. Each MS is allocated an *Active* BS and all the other remaining BS are *Candidate* BS. Each MS continuously monitors the power levels of the CPICH on the forward link. When any candidate BS is within a particular power level range from the currently active BS, soft handoff is initiated with that candidate BS. Going off that range results in a termination of soft handoff.

Feasible SIR Target

In order to have simulations involving forward link CPICH and soft handoff, we first make certain to run experiments for a reasonable SIR target that is a *feasible* target. Section 6.2.2 included a detailed sample results and deductions about SIR feasibility. A similar exercise is done for a multi cell system with parameters given in Table 6.4(C). The experiments have been done for 5 (= 60/12) users per cell. The target SIR varies from 2 dB to 8 dB and a maximum power limit of 30 dBm has been put into place. The AWGN has a power spectral density of −150 dBm/Hz. Bit detection is being done using the NTT DoCoMo 3 stage interference canceller.

Figure 6.15(a) gives the average transmitted power in the multi cell system. Figure 6.15(b) gives the fraction of users that have attained their maximum power limit of 30dBm. Both the curves increase with an increase in SIR target. However, unlike the curves in Section 6.2.2, these curves do not saturate. An increase indicates that the mobiles along the boundary of the cell have reached their limits, and might be ideal candidates for soft handoff with other BSs. Figure 6.16(a) gives the average received

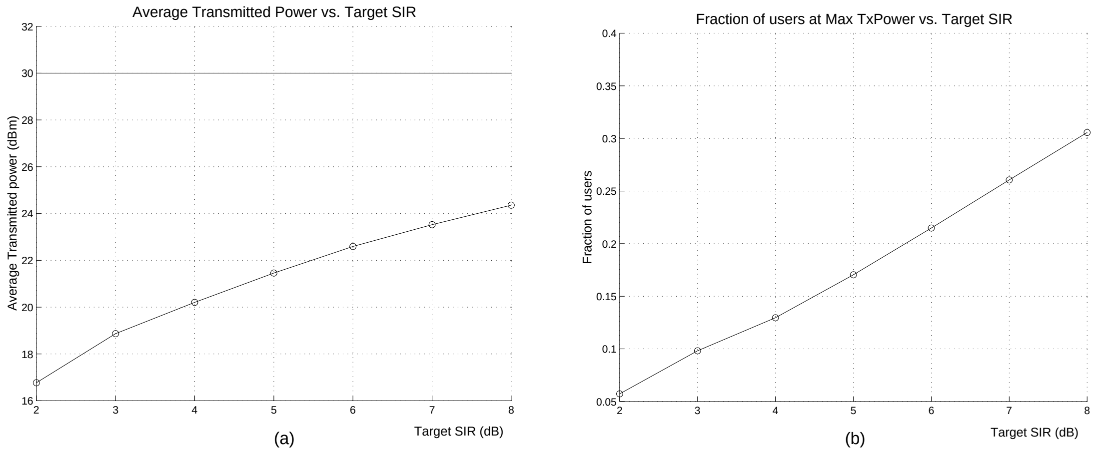


Figure 6.15: Feasibility check for multi-cell system: Plot (a) shows that the average transmitted power increases with an increase in the target SIR. Plot (b) gives the fraction of users that have attained the maximum power limit.

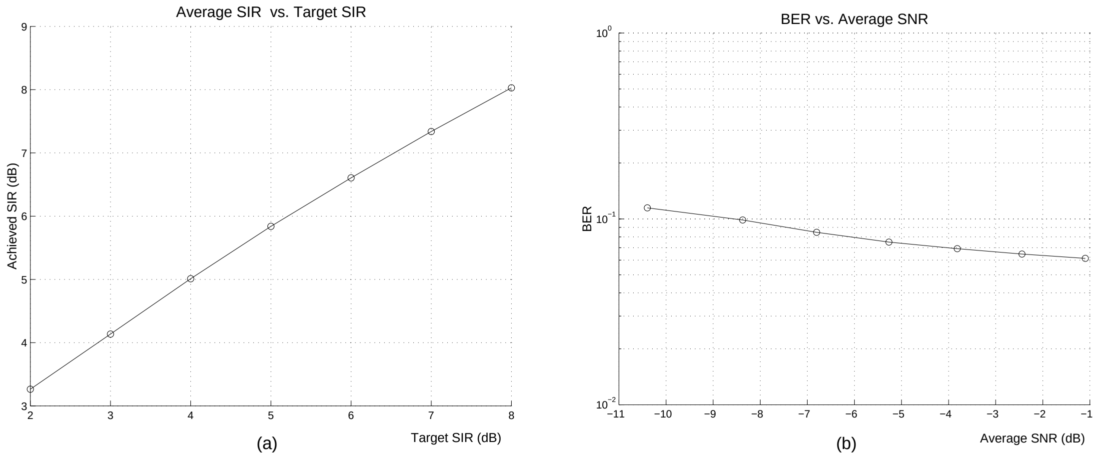


Figure 6.16: Average $\widetilde{SIR}$ and BER for multi cell system: The linear increase in plot (a) indicates that the system is feasible at all the target SIRs for that plot.

signal to interference ratio for the multi cell system. Figure 6.16(b) gives the BER against $\widetilde{SNR}$. The linear increase in Figure 6.16(a) indicates that the system is feasible for all target SIRs.

We are now in a position to select a target SIR, and then perform forward link simulations to understand the effect of soft handoff on this configuration. A nominal SIR target of 4.0 dB has been chosen as a sample study.

Soft Handoff experiments

WiPPET$_{\text{signal}}$ uses a dynamic range H for soft handoff for making decisions about soft handoff at the MS. All candidate BSs whose received pilot power is within H dB of the strongest BS are upgraded to active BSs by that MS and a soft handoff is initiated. When any active BS falls below this level of H dB from the strongest BS, it is downgraded to a candidate BS again, and soft handoff is terminated. An increase in this range is responsible for bringing more MSs into the *coverage* of a particular BS.

The simulation for forward link CPICH and soft handoff uses the same parameters given in Table 6.6. However, the target SIR is fixed at 4.0 dB and `FwdLinkPresent` is set to 1. All active BSs for a particular MS in soft handoff forward the soft data as well as a hard BER statistic to a switch where the soft data is combined using equal gain combining.

Figure 6.17 shows that the fraction of users in soft handoff increases as the soft handoff range increases. When soft handoff is absent ($H = 0$dB), each MS is in connection with just one BS. As the soft handoff range increases, the BSs increase their *coverage* area. This means that more MSs can *detect* BSs other than the one they are in connection with. For higher values of the soft handoff range, the possibility that an MS that would have otherwise regarded some candidate BS as *weak*, but now will regard it as *strong enough*, increases. This results in soft handoff with this candidate BS and its upgrade to an active BS.

Figure 6.18 gives a plot between the fraction of users that have reached the maximum power limit of 30 dBm and the soft handoff range. As the soft handoff range increases, the average transmitted power decreases. When soft handoff is absent, a MS around

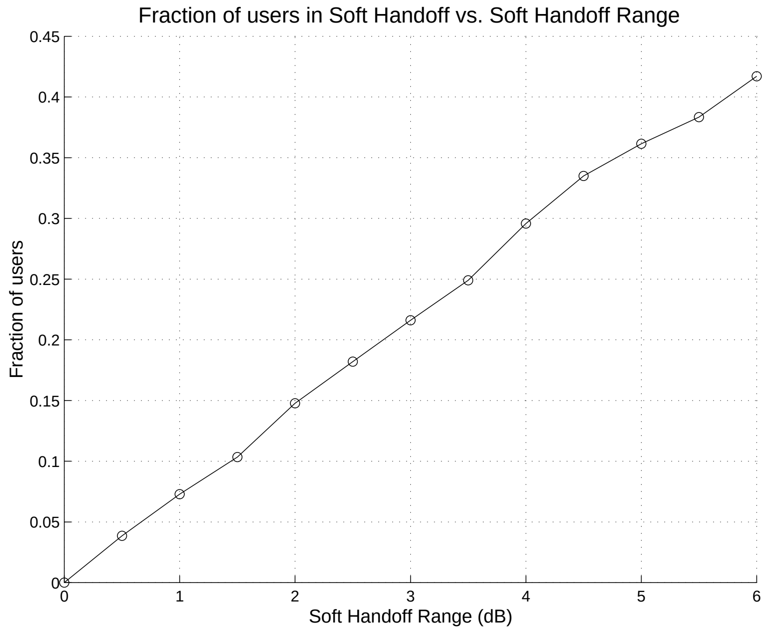


Figure 6.17: Fraction of users in soft handoff: As the soft handoff range increases, the fraction of users in soft handoff increases linearly.

the boundary of the cell must maintain acceptable SIR at its assigned BS. However, when a MS is in soft handoff, it receives power control commands from all its active BSs. The power of the MS is increased only if all the active BSs command an increase, otherwise it is decreased. Hence, there are lesser increases than in the case without soft handoff. In effect, power control takes advantage of the diversity mechanism of soft handoff. Figure 6.18 illustrates the decreasing trend in the fraction of users that have reached the maximum power limit.

Figure 6.19(a) gives a plot between the average transmitted power $\hat{S}_t$ and the soft handoff range. A maximum upper limit of 30 dBm is also plotted. For higher handoff ranges, lesser users approach their transmit power limit as shown in Figure 6.18. This is reflected in the average transmit power. Figure 6.19(b) gives a plot between the average received power $\hat{S}_r$ and the soft handoff range. The trend in received powers is similar to that of transmitted powers in Figure 6.19(a).

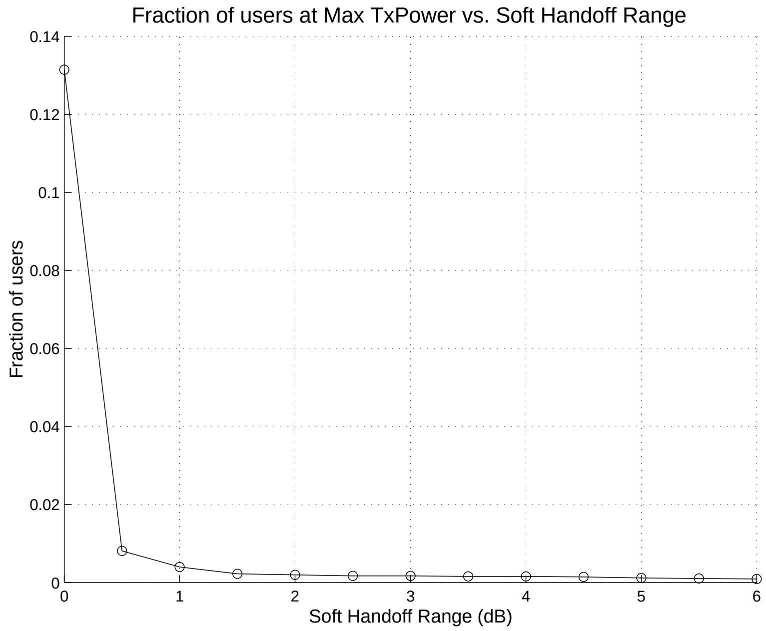


Figure 6.18: Fraction of users reaching maximum power limit with soft handoff: As the soft handoff range increases, a lesser fraction of users approach the maximum limit of 30dBm.

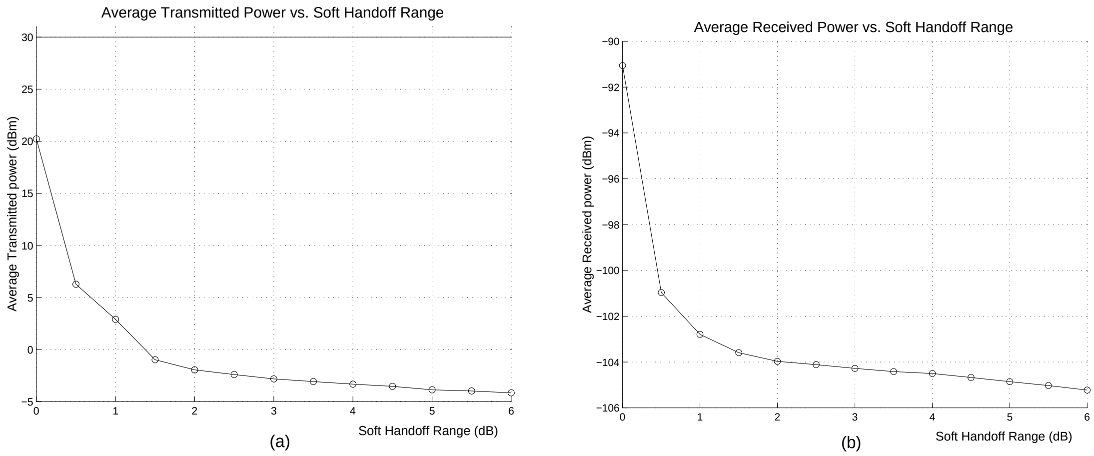
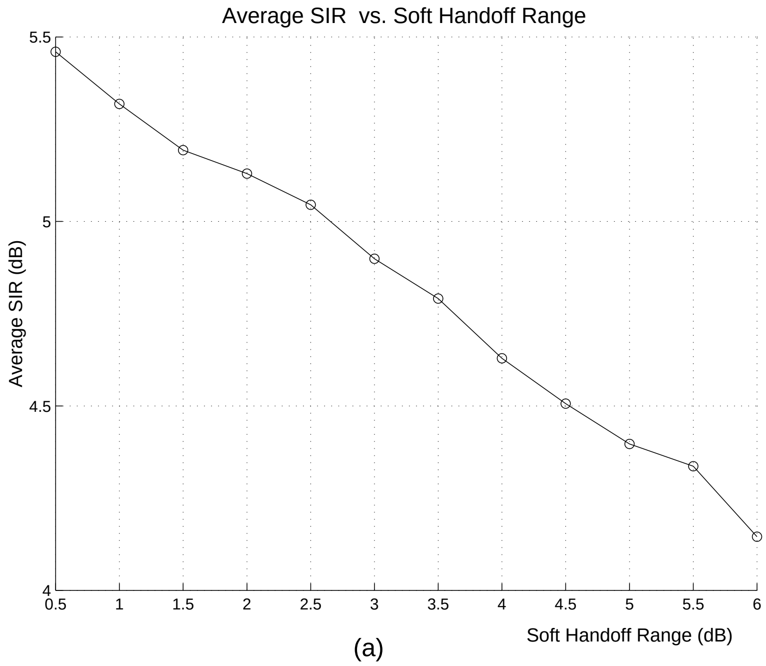


Figure 6.19: Average transmitted and received powers for a multi cell system with soft handoff: Plot (a) shows a decrease in the average transmit power of the system because lesser number of MS approach their maximum limit as the soft handoff range increases. Plot (b) gives the average received power which also decreases.

Figure 6.20 gives a plot between the $\widetilde{SIR}$ and the soft handoff range. As the range increases, power of more MSs is controlled more appropriately based on their current positions and state of handoffs. As a result, the average SIR approaches the target SIR



(a)

Figure 6.20: Average $\widetilde{SIR}$ in a multi cell system with soft handoff: As the soft handoff range is increased, a better power control causes the $\widetilde{SIR}$ to approach the target SIR.

of 4.0 dB.

BER improvements due to soft handoff are illustrated in Figure 6.21 which gives a plot between the bit error rate and the soft handoff range. The $H = 0$ dB value indicates that soft handoff is absent, and this value corresponds to the value in Figure 6.15(b) for a target SIR of 4.0 dB. Note that as the soft handoffrange increases, BER decreases, even though average SIR also decreases. This demonstrates the diversity benefit of soft handoff. The switching entity uses *macro* diversity by using equal gain combining or selection diversity.

6.2.4 Capacity Plots for Perfect Power Control

WiPPET$_{\text{signal}}$ evaluates BER statistics for WCDMA systems. For example, Figure 6.8(b) in Section 6.2.1 gives the BER for a perfectly power controlled single cell

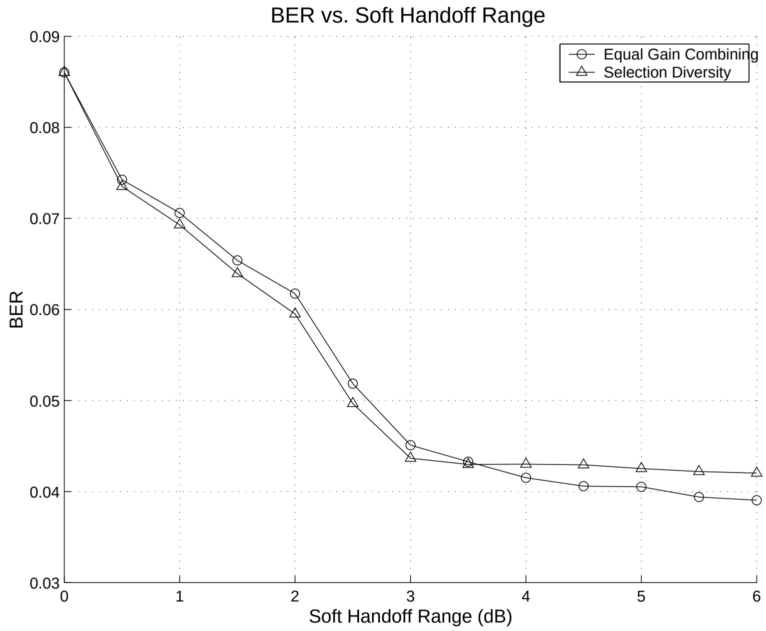


Figure 6.21: BER for soft handoff: BER decreases as the soft handoff range increases.

system with 10 users in presence of rate 1/3 coding. Similar runs need to be done for different number of users and curves corresponding to the same setup have to be derived for each choice of number of users. Then a family of curves for that particular setup is created in which the x-axis is the required SNR and the y-axis is the BER.

Figure 6.22(a) gives one such family of curves, for a multi cell system with receiver antenna and path diversity, NTT DoCoMo 3 stage interference canceller, a chip rate of 4.096 Mcps and a spreading factor of 64. Coding is absent in this scenario, and the system is perfectly power controlled. An operational BER of 0.001 has been chosen to be the QoS requirement. Then the SNR required to achieve that BER for particular number of users is obtained by drawing perpendiculars onto the x-axis from the points where the curves intersect the BER=0.001 horizontal line. The capacity curve is the plot of these values of SNR against the normalized number of users ($= K/N$) where K is the number of users/cell and N is the spreading factor. Figure 6.22(b) gives the

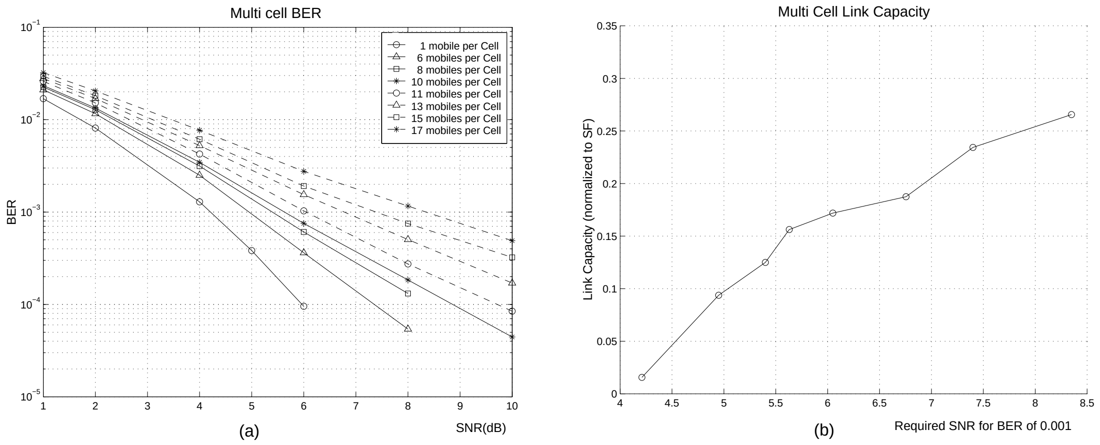


Figure 6.22: Multi cell BER and link capacity: Curves in plot (a) give the BER for various values of signal to noise ratio in an uncoded and perfectly power controlled multi cell system with receiver antenna and path diversity. Plot (b) gives the normalized link capacity for a BER of 0.001

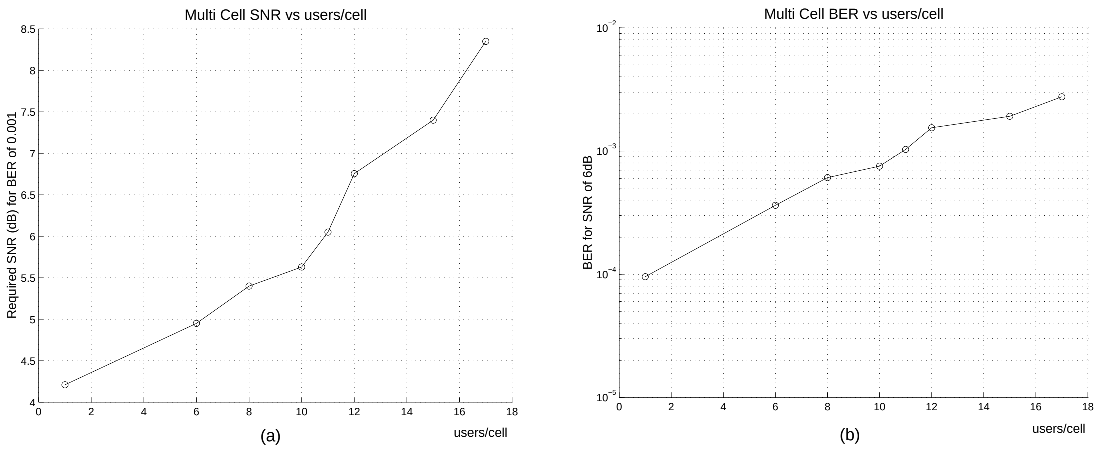


Figure 6.23: Multi cell capacity: Plot (a) gives the SNR required to achieve an average BER of 0.001 for various values of the offered load. Plot (b) gives the achieved BER at a nominal SNR of 6 dB for various values of the offered load.

corresponding normalized link capacity curve for the BER obtained in Figure 6.22(a).

The capacity of the system can also be illustrated via plots for BER and SNR against the load in the system. Figure 6.23(a) gives the SNR required to achieve an average BER of 0.001 for various values of the offered load (users/cell) in the multi cell system. Figure 6.23(b) gives the achieved average BER at a nominal SNR of 6 dB for various values of the offered load.

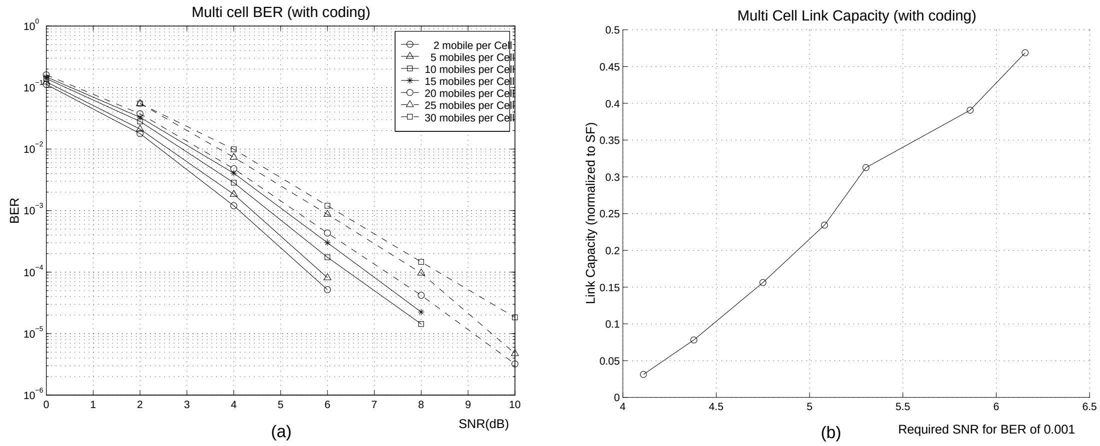


Figure 6.24: Multi cell BER (with coding) and link capacity: Curves in plot (a) give the BER for various values of signal to noise ratio in a rate 1/3 coded and perfectly power controlled multi cell system with receiver antenna and path diversity. Plot (b) gives the normalized link capacity for a BER of 0.001

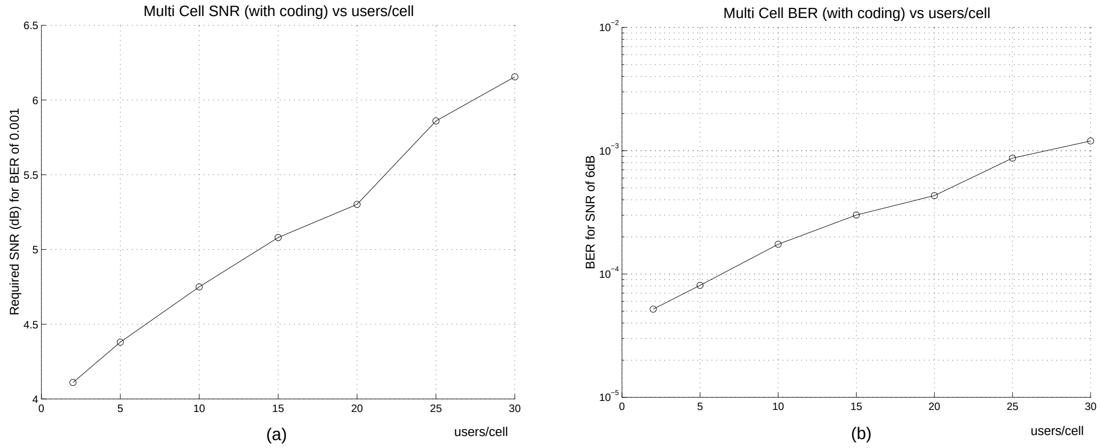


Figure 6.25: Multi cell capacity (with coding): Plot (a) gives the SNR required to achieve an average BER of 0.001 for various values of the offered load. Plot (b) gives the achieved BER at a nominal SNR of 6 dB for various values of the offered load.

Figure 6.24(a) gives another family of curves with same specifications as for Figure 6.22, but with rate 1/3 coding. Figure 6.24(b) gives the corresponding normalized link capacity curve for the BER obtained in Figure 6.24(a).

Figure 6.25(a) gives the SNR required to achieve an average BER of 0.001 for various values of the offered load (users/cell) in the multi cell system with coding. Figure 6.25(b) gives the achieved average BER at a nominal SNR of 6 dB for various values of the

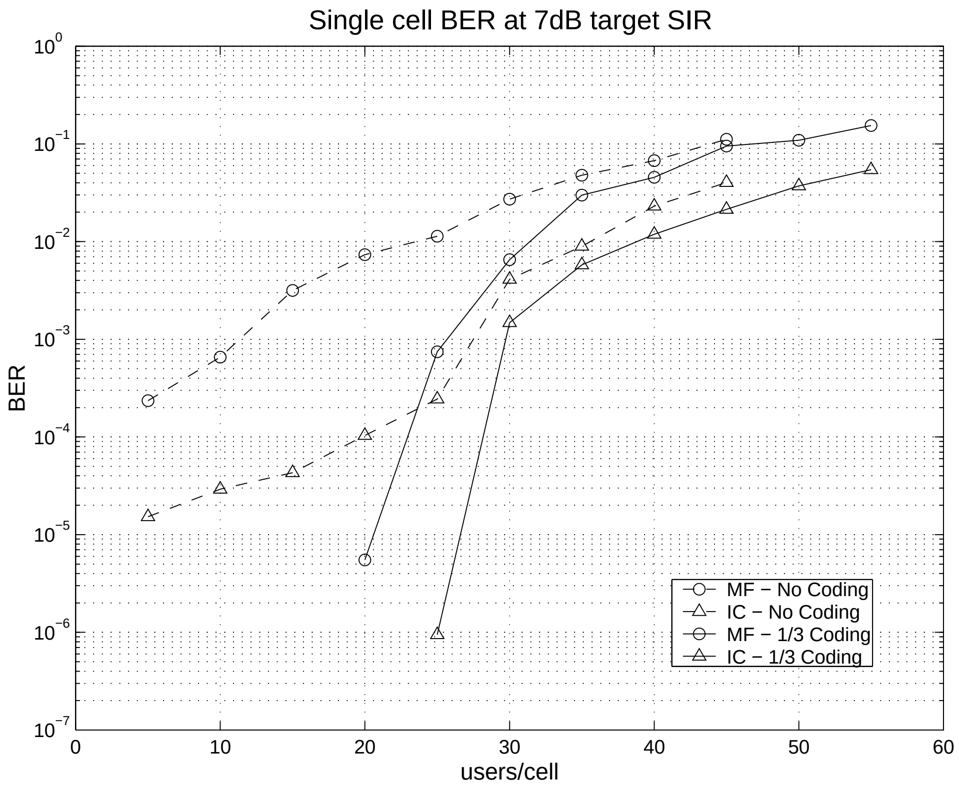


Figure 6.26: Single Cell Capacity with real TPC: As the load in the cell increases, the BER increases. A nominal target SIR of 7 dB has been selected for the simulation with 2 receiver antennas and 2 fadingr paths.

offered load.

6.2.5 Capacity Plots for Real Power Control

The previous Sectoin 6.2.4 gave capacity plots for systems that have perfect power control where all users are received with the same power. For systems with real power control (as explained in Section 6.2.2), a target SIR is required to be achieved.

Figure 6.26 gives the average BER in a single cell system with 2 receiver antennas and 2 paths at a nominal target SIR of 7 dB. As the load in the cell increases, the BER also increases. The improvements due to the interference canceller and rate 1/3 coding are also illustrated. The spreading factor the same in all experiments. For example, comparing *MF-uncoded* and *MF-rate 1/3 coded*, the coded system is transmitting data bits at 1/3 the rate of the uncoded system over the air. Also, the definition of the SIR target the same for the coded and uncoded symbols. This effectievly means that the target SIR of of 7 dB is aimed for using the measurements at the output of the rake correlator. The power control loop ingores whether the incoming symbols are coded or uncoded.

Parameter	*Range*	*Definition*
numBS	1 or 12	1 for single cell 12 for multi cell
numMS	any positive value	Number of MS in the system
BSTiming	any non negative value	BS Timing in multiples of symbols
chipRate	any power of 2 in KHz	WCDMA system chip rate
multipathProfile	any	Delay of paths in nanoseconds Relative Power profiles of paths
multipathChannel	A or B	Rounding off to nearest chips Channel_B default
spreadingFactor	any 2 power	spreading for one WCDMA symbol
rxAntenna	any positive value	number of receiving antennas in a node
txAntenna	any positive value	number of transmitting antennas in a node
LSGainPresent	0 or 1	0 (absent) or 1 (present)
FadingPresent	0 or 1	0 (absent) or 1 (present)
FwdLinkPresent	0 or 1	0 (absent) or 1 (present)
codingPresent	0 or 1	0 (absent) or 1 (present)
perfectTPCPresent	0 or 1	0 (absent) or 1 (present)
interfCanceller	0 or 1	0 (MF) or 1 (3 stage DoCoMo IC)
datumPresent	0 or 1	0 (absent) or 1 (present)
perfectTPCEbN0	any value in dB	EbN0 for perfect TPC
Noise	any value in dBm/Hz	AWGN for real TPC
SIRTarget	any value in dB	Target for real TPC
maxTxPower	any value in dBm	Maximum power limit for transmission
minTxPower	any value in dBm	Minimum power limit for transmission
softHandoffRange	any value in dB	Power range at MS within which BSs need to be to maintain soft handoff for that MS
TPCStep	any value in dB	Increase in transmit power
forgetFactor	between 0 and 1	TPC measurement filter averager
IRWeight	between 0 and 1	Interference rejection weight
startFrameForMeas	any positive value	WCDMA frame starting with which statistics are collected
maxErroBits	any positive value	Total max limit for error bits to stop simulation
simulationTime	any positive integer	Total SSF tick for simulation
channelParallelize	0 or 1	0 (absent) or 1 (present)
transceiverParallelize	0 or 1	0 (absent) or 1 (present)
numParallelProcs	any positive integer	Number of threads

Table 6.3: WiPPET$_{\text{signal}}$ Parameters

Parameter Sets	*A*	*B*	*C*
numBS	12	1	12
numMS	36	variable	60
chipRate	3840 kHz	3840 kHz	3840 kHz
spreadingFactor	64	64	64
FwdLinkPresent	0	0	1
codingPresent	1	0	0
interfCanceller	0	1	1
perfectTPCPresent	0	0	0
Noise	-150 dBm/Hz	-150 dBm/Hz	-150 dBm/Hz
SIRTarget	6.0 dB	variable	4.0 dB
maxTxPower	30 dBm	30 dBm	30 dBm
minTxPower	-80 dBm	-80 dBm	-80 dBm

Table 6.4: WiPPET$_{\text{signal}}$ parameters for (A) Sample study in Section 6.2, (B) SIR based power control sample study in Section 6.2.2 and (C) Multi cell soft handoff in Section 6.2.3

Chapter 7

Conclusions

The C++ flavor of SSF, a software program for parallel simulation was reviewed. The requirements of a WCDMA physical layer simulator were defined and a model of the simulation was discussed. This thesis also discussed the possible ways to parallelize the simulation and implemented one of them. Complexity of the various modules was highlighted and speedups obtained in various scenarios were illustrated. The integrity of the simulator was verified by unit level and system level tests. Sample results about BER, target SIR feasibility, soft-handoff gains and capacity were also illustrated. Some initial results have been published in [17].

WiPPET$_{\text{signal}}$ has been shown to be a versatile physical layer WCDMA simulator that can cater to researchers interested in system level statistics like BER or very specific details like the effect of the interference rejection weight. The flexibility of the configuration allows for a variety of combinations that can be simulated in accordance to the physical conditions in the system. WiPPET$_{\text{signal}}$ is also integrated to *DATUM*, a WINLAB Java program for viewing and run-time plotting of graphs for SIR, BER and other statistics.

WiPPET$_{\text{signal}}$ can be extended to include multi-rate operation for WCDMA. It can also be modified for cdma2000 simulations. Other applications like effect of antenna beamforming and sectorization can also be incorporated.

Appendix A

SSF Example Program

We will describe the main characteristics of a SSF program by running through an example program. Source code has been provided and the functionality of statements has been detailed.

Consider a system as shown in Figure A.1. It has 3 types of SSF *entities*:

- `Sender` : This entity sends out messages on its *outChannel*
- `Receiver` : This entity receives messages on its *inChannel*
- `Master` : This entity encompasses all the `Sender` entities and the `Receiver` entity and does the channel mapping

We now describe the system in more detail.

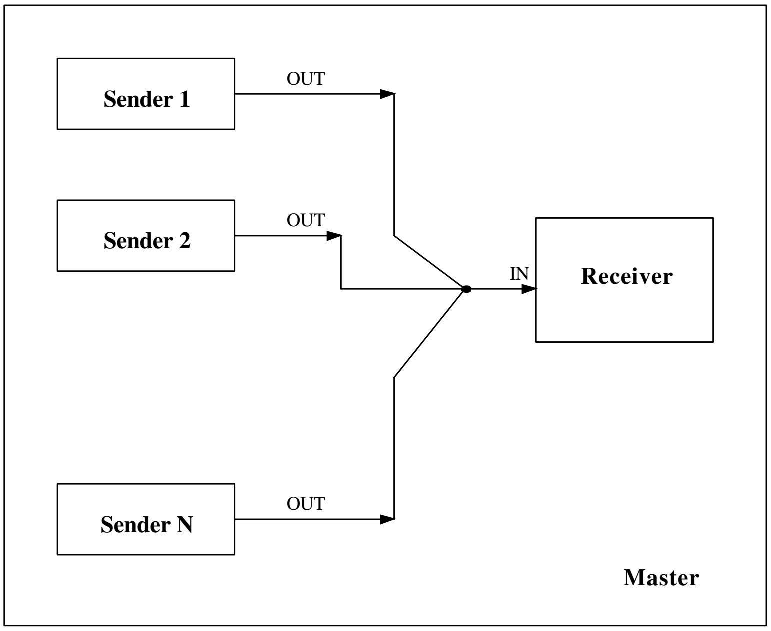


Figure A.1: Example SSF system : Multiple sender entities transmitting to the same receiver entity

Information Event

```
class Info : public Event
{
public :
  int  sender;
  int  x;
  char y;
};

class Msg
{
public:
  Info info;

  Msg(int _s, int valX, char valY)  // Constructor
  {
    info.sender = _s;
    info.x = valX;
    info.y = valY;
  }
};
```

Information is passed from the `Sender` entities to the `Receiver` entity in an SSF *event*. The event is named `Info` and it extends the base SSF event. Class `Msg` encompasses this extended event.

Sender Entity

```
class Sender : public Entity
{
public:
  outChannel* OUT;
  int         entityNum;
  int         sent;

  Sender(int i)
  {
    sent      = 0;
    entityNum = i;
    OUT       = new outChannel(this,0);
    process* send = new process(this,
                        (ACTION)&tSender::sendProc, true);
  };

  void sendProc(process* sender)
  {
    Msg *message;
    message = new Msg(entityNum, now()+entityNum,'Y');
    OUT->write(&(message->info), entityNum);
    sent++;
    sender->waitFor( 10);
  };
};
```

The `Sender` entity is defined with with one *outChannel* named `OUT`. Since there are multiple senders, it also carries an identification by the name `entityNum` and a state variable `sent` which keeps track of how many messages have been sent out. This entity has a SSF *process* by the name `sendProc` and a constructor. The constructor of the entity initializes the outChannel `OUT` with a zero delay. It also sets up the process `sendProc`. This process executes the following logic:

1. Declare a message and initialize it

2. Use the SSF *write()* statement to send out the information event in the message on the `OUT` channel . The delay (second parameter in *write()*) is set to the entity number

3. Increment the `sent` state variable by 1

4. Use the SSF *waitFor()* statement to go to step 1 after 10 SSF ticks

Receiver Entity

```
class Receiver : public Entity
{
public:
  inChannel *IN
  int       numSenders;
  int       rcvdTot;
  int       *rcvd;

  Receiver(int _numSenders)
  {
    numSenders = _numSenders;
    rcvdTot    = 0;
    rcvd       = new int[numSenders];
    IN         = new inChannel(this);
    for(int i=0; i < numSenders; i++)
      rcvd[i] = 0;

    process* recvTick = new process(this,
                            (ACTION)&tReceiver::recvProcTick, true);
    process* recv     = new process(this,
                            (ACTION)&tReceiver::recvProc, true);
  };

  void recvProcTick(process* receiverTick)
  {
    int a = rcvdTot;
    // any other processing
    receiverTick->waitFor(30);
  };

  void recvProc(process* receiver)
  {
    Event** pe;
    Event** active = (IN)->activeEvents();

    for(pe= active; *pe; pe++)
    {
      int i = ((tInfo *)(*pe))->sender;
      int x = ((tInfo *)(*pe))->x;
      char y= ((tInfo *)(*pe))->y;
      rcvd[i]++;
      rcvdTot++;
    }
    receiver->waitOn(IN);
  };
};
```

The `Receiver` entity is defined with with an *inChannel* named `IN`. Each *outChannel* from the `Sender` entities will be mapped onto this inChannel. Since there are multiple senders, the `Receiver` it also carries a constant `numSenders`, a state variable array `rcvd` which keeps track of how many messages have been received from each sender and also another state variable `rcvdTot` which keeps track of the total received events. This entity has two *processes*; (a) `recvProc` and (b) `recvProcTick` and also a constructor. The constructor of the entity initializes the various constants and state variables, and sets up the two processes. The `recvProcTick` is internally triggered and executes the following logic:

1. Do some processing based on the state variables
2. Use the SSF *waitFor()* statement to go to step 1 after 30 SSF ticks

The `recvProc` process is triggered by the arrival of events on the `IN` channel. It executes the following logic:

1. Declare an array of pointers to SSF *events*
2. De-queue all *active* events from the `IN` channel into this array of pointers
3. Go through each event and extract the `Sender` entity ID and the content of the information
4. Increment the state variables
5. The SSF statement *waitFor()* indicates that the process should get triggered only if events arrive on the `IN` channel that has been provided as the parameter

Master Entity

```
class Master : public Entity
{
public:
  Sender** S;
  Receiver* R;

  Master(int numSenders)
  {
    S = new Sender*[numSenders];
    R = new Receiver(numSenders);

    for(int i=0; i< numSenders; i++)
    {
      S[i] = new tSender(i);
      (S[i]->OUT)->mapto(R->IN);
    }
  };
};
```

The `Master` entity encapsulates all the entities. It is defined with with an array `S` of `Sender` entities and one `R` `Receiver` entity. The constructor initializes the entities by calling their corresponding constructors where the SSF channels get initialized. The SSF statement *mapto()* is used to map the `OUT` *outChannel* from each sender onto the `IN` *inChannel* of the receiver. Now the simulation is all set to start.

main() Function

```
int main()
{
  int numSenders = 3;
  Master*  M = new Master(numSenders);

  cout << ``START'' << M->now() << endl;

  M->startAll(100000);
  M->joinAll();

  cout << ``END'' << M->now() << endl;

  cout << ``TOTAL RECEIVED'' << M->R->rcvdTot << endl;

  for(int i=0; i < numSenders; i++)
  {
    cout << ``SENT     by'' << i `` = `` << M->S[i]->sent << endl;
    cout << ``RECEIVED by'' << i `` = `` << M->R->rcvd[i] << endl;
  }
}
```

The *main()* function initializes `M` as the `Master` entity with 3 senders. The SSF statement *now()* gives the current instant of the simulation. The *startAll()* statement passes the parameter that indicates the number of SSF ticks for which the simulation will last. The *joinAll()* statement starts all the processes aligned to the same timeline.

Appendix B

WiPPET$_{\text{signal}}$ SSF Abstraction

The design of WiPPET$_{\text{signal}}$ was introduced in Section 4.1. We now introduce the connectivity in WiPPET$_{\text{signal}}$ and the SSF abstractions of each entity.

Connectivity in WiPPET$_{\text{signal}}$

Figure B.1 illustrates the connectivity among the various entities. Part (a) shows the WCDMA test data flow from the `Mobile`s to the `Base`s via the `RevChannel` which simulates the WCDMA DPCCH/DPPDCH on the reverse link, and from the `Base`s to the `Mobile`s via the `FwdChannel` that simulates the WCDMA CPICH on the forward link. Part (b) shows the channels used to update the `FwdChannel` and the `RevChannel` to introduce time variations due to long and short scale fading effects. Part (c) shows the protocol message flow from the `Base`s to the `Mobile`s and vice-versa. Every `Base` is connected to every `Mobile` and the same is true for the opposite direction. The `Base`s are also connected to the `Switch`.

WiPPET$_{\text{signal}}$ follows the following convention for naming a connecting SSF channel: $<Direction>\_<Type>\_<Source>To<Destination>$. So, if a channel is named `O_PHYS_StationToRevChannel` that means that this SSF channel is an **output** channel used to transfer **physical** simulation data from the **Station** (`Mobile` or `Base`)

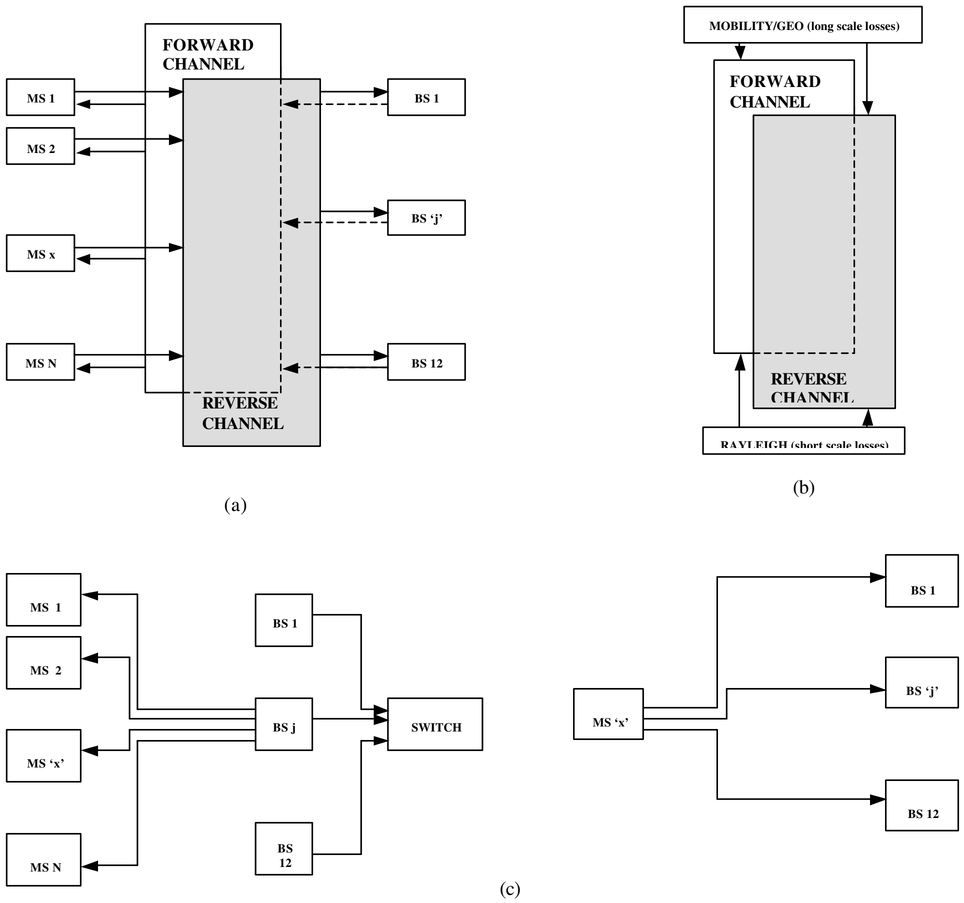


Figure B.1: The various entities are connected to each through SSF Channels. They can be classified as (a) WCDMA Physical, (b) Channel Variation and (c) WCDMA Protocol SSF channels

to **Channel** (`RevChannel` in this case). Similarly, `I_PROT_BaseToMobile` is the SSF channel used as an **input** port for an error free reception of **protocol** messages from the `Base` to the `Mobile`. The following sections provide an introduction of each entity.

`MOBILE` SSF Abstraction

In WIPPET$_{\text{signal}}$ the `MOBILE` class extends the SSF Entity API. Figure B.2 illustrates the block diagram of the `MOBILE` class with the following SSF channels:

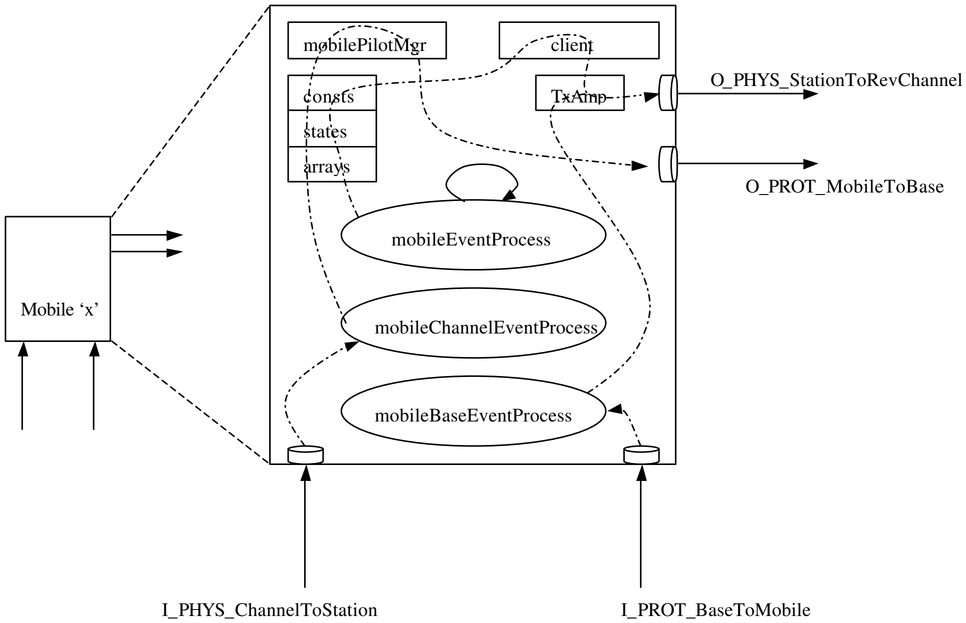


Figure B.2: The various processes and main data structures in the `MOBILE`

- `O_PHYS_StationToRevChannel` : Connects `Mobile` to `RevChannel` for simulation data to travel on the WCDMA DPCCH/DPPDCH reverse link.
- `I_PHYS_ChannelToStation` : Connects `FwdChannel` to `Mobile` for simulation data to travel on the WCDMA CPICH forward link.
- `O_PROT_MobileToBase` : Connects `Mobile` to each `Base` for providing an error free channel to pass protocol messages from MS to BS.
- `I_PROT_BaseToMobile` : Connects `Mobile` to each `Base` for providing an error free channel to pass protocol messages from BS to MS.

The following SSF processes are part of the `MOBILE` class:

- `mobileEventProcess` : This process is internally timed by SSF and gets activated after every WiPPET$_{\text{signal}}$ processing loop[1]. After every activation, the process

[1] The length of the WCDMA reverse link symbol in chips

sends out the next reverse link DPDCH/DPCCH symbol as a complex valued array of I/Q chips on the `O_PHYS_StationToRevChannel` SSF outChannel. It also keeps track of WCDMA reverse link slots and frames and changes the transmitting amplitude, the long code and the fake MAC data frame on pre-determined boundaries.

- `mobileBaseEventProcess` : This process gets activated every time there is a control message from the BS to the MS on the `I_PROT_BaseToMobile` SSF inChannel. Currently only the power control command is received. Any further upgrades that need more control commands from the BS to MS will be handled by this process.

- `mobileChannelEventProcess` : This process gets activated every time a simulation data message (WiPPET$_{\text{signal}}$ processing loop worth of I/Q chips) arrives from the `FwdChannel` on the `I_PHYS_ChannelToStation` SSF inChannel. The WCDMA CPICH channel is handled in this process.

`BASE` SSF Abstraction

In WiPPET$_{\text{signal}}$ the `BASE` class extends the SSF Entity API. Figure B.3 illustrates the block diagram of the `BASE` class with the following SSF channels:

- `O_PHYS_StationToFwdChannel` : Connects `Base` to `FwdChannel` for simulation data to travel on the WCDMA CPICH forward link.

- `I_PHYS_ChannelToStation` : Connects `RevChannel` to `Base` for simulation data to travel on the WCDMA DPDCH/DPCCH reverse link.

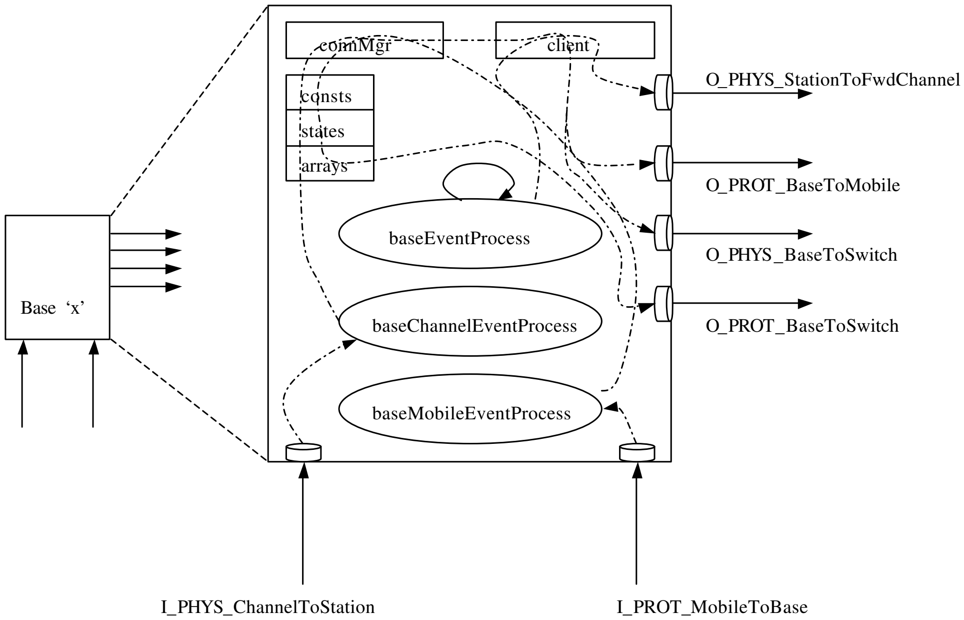


Figure B.3: The various processes and main data structures in the `BASE`

- `O_PROT_BaseToMobile` : Connects `Base` to each `Mobile` for providing an error free channel to pass protocol messages from BS to MS.

- `I_PROT_MobileToBase` : Connects `Base` to each `Mobile` for providing an error free channel to pass protocol messages from MS to BS.

- `O_PHYS_BaseToSwitch` : Connects `Base` to `Switch` for simulation data to travel on the WCDMA DPDCH reverse link to MSC.

- `O_PROT_BaseToSwitch` : Connects `Base` to `Switch` for providing an error free channel to pass protocol messages from BS to MSC.

The following SSF processes are part of the `BASE` class:

- `baseEventProcess` : This process is internally timed by SSF and gets activated after every WiPPET$_{\text{signal}}$ processing loop. After every activation, the process sends out the next forward link CPICH symbol as an array of I/Q chips on the

O_PHYS_StationToFwdChannel SSF outChannel. It also keeps track of WCDMA forward link slots and frames and changes the long code on pre-determined boundaries.

- baseMobileEventProcess : This process gets activated every time there is a protocol message from the MS to the BS on the I_PROT_MobileToBase SSF inChannel. Currently only the soft-handoff activation and termination commands are received. Any further upgrades that need more control commands from the MS to BS will be handled by this process.
- baseChannelEventProcess : This process gets activated every time a simulation data message (WiPPET$_{\text{signal}}$ processing loop worth of I/Q chips) arrives from the RevChannel on the I_PHYS_ChannelToStation SSF inChannel. The WCDMA DPDCH/DPCCH channel is handled in this process.

SWITCH SSF Abstraction

In WiPPET$_{\text{signal}}$ the SWITCH class extends the SSF Entity API. Figure B.4 illustrates the block diagram of the SWITCH class with the following SSF channels:

- I_PHYS_BaseToSwitch : Connects Base to Switch for simulation data to travel on the WCDMA DPDCH reverse link to MSC.
- I_PROT_BaseToSwitch : Connects Base to Switch for providing an error free channel to pass protocol messages from BS to MSC.

The following SSF processes are part of the SWITCH class:

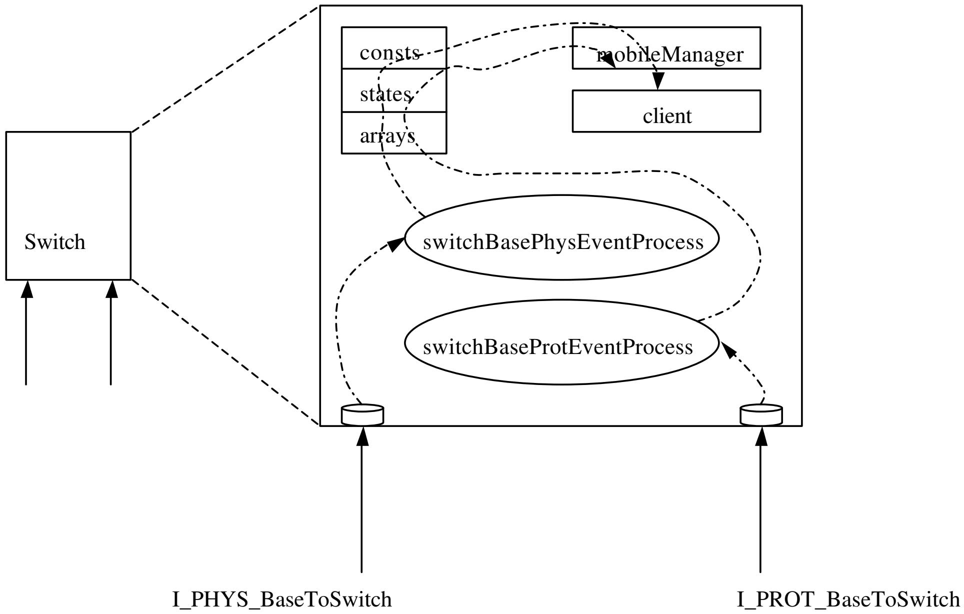


Figure B.4: The various processes and main data structures in the `SWITCH`

- `switchBasePhysEventProcess` : This process gets activated every time there is a soft simulation data event from the BSs to the MSC on the `I_PHYS_BaseToSwitch` SSF inChannel. This process keeps accumulating the soft contributions, and when all contributions are received, they are combined and a hard decision about the MS made.

- `switchBaseProtEventProcess` : This process gets activated every time there is a protocol message from the BSs to the MSC on the `I_PROT_BaseToSwitch` SSF inChannel. Currently only the soft-handoff activation and termination commands are received. Any further upgrades that need more protocol messages from the BS to MSC will be handled by this process.

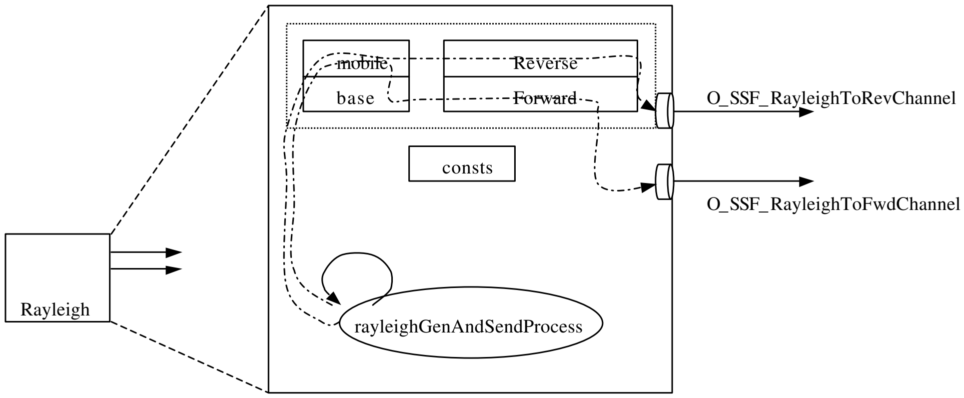


Figure B.5: The various processes and main data structures in the RAYLEIGH

RAYLEIGH SSF Abstraction

In WiPPET$_{\text{signal}}$ the `RAYLEIGH` class extends the SSF Entity API. Figure B.5 illustrates the block diagram of the `RAYLEIGH` class with the following SSF channels:

- `O_SSF_RayleighToRevChannel` : Connects `Rayleigh` to `RevChannel` for providing an error free SSF channel to pass short-scale Rayleigh fading values.
- `O_SSF_RayleighToFwdChannel` : Connects `Rayleigh` to `FwdChannel` for providing an error free SSF channel to pass short-scale Rayleigh fading values.

The following SSF processes are part of the `RAYLEIGH` class:

- `rayleighGenAndSendProcess` : This process is internally timed by SSF and gets activated after every WiPPET$_{\text{signal}}$ processing loop. After every activation, the process generates the next set of fading values for the forward and reverse links, packs them in events and delivers them to the `RevChannel` and `FwdChannel` over `O_SSF_RayleighToRevChannel` and `O_SSF_RayleighToFwdChannel` SSF outChannels respectively.

MOBILITY SSF Abstraction

In WiPPET$_{\text{signal}}$ the MOBILITY class extends the SSF Entity API. Figure B.6 illustrates the block diagram of the MOBILITY class with the following SSF channels: with it:

- O_MobilityToRevChannel : Connects Mobility to RevChannel for providing an error free SSF channel to pass long-scale fading values.
- O_MobilityToFwdChannel : Connects Mobility to FwdChannel for providing an error free SSF channel to pass long-scale fading values.
- O_SSF_MobilitySelf : Output port to connect Mobility to itself to provide internal events for triggering mobility events for mobiles.
- I_SSF_MobilitySelf : Input port to connect Mobility to itself to provide internal events for triggering mobility events for mobiles.

The following SSF processes are part of the MOBILITY class:

- mobilityInitProcess : This process gets activated only once during the initialization of the simulation. It sets up the state for each point on the grid and also for each station (MS or BS) in the system.
- mobilityEventProcess : This process is internally timed by SSF by using an internal event. A different event is used for each mobile as they get activated at different times and also might have different speeds. After every activation, the mobile moves to the next neighboring point, and its long-scale losses to the 12 bases are packed into an event and sent to the RevChannel and to the FwdChannel

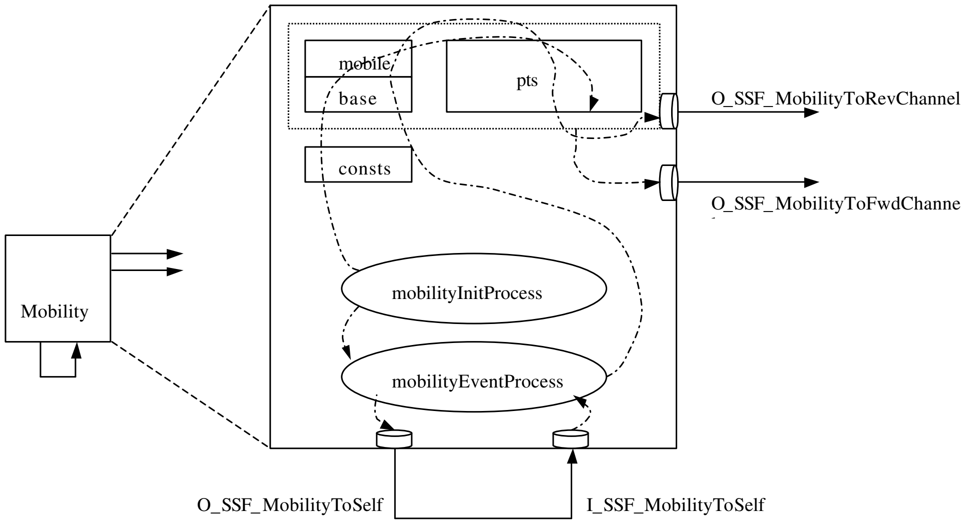


Figure B.6: The various processes and main data structures in the `MOBILITY`

over the `O_MobilityToRevChannel` and `O_MobilityToFwdChannel` SSF outChannels respectively.

`CHANNEL` SSF Abstraction

In WiPPET$_{\text{signal}}$ the `CHANNEL` class extends the SSF Entity API. Figure B.7 illustrates the block diagram of the `CHANNEL` class. The following SSF channels are associated with it:

- `O_PHYS_ChannelToStation` : Connects `RevChannel` to `Base` for simulation data to travel on the WCDMA DPDCH/DPCCH reverse link, and `FwdChannel` to `Mobile` for simulation data to travel on WCDMA CPICH forward link.

- `I_PHYS_StationToChannel` : Connects `RevChannel` to `Mobile` for simulation data to travel on the WCDMA DPDCH/DPCCH reverse link, and `FwdChannel` to `Base` for simulation data to travel on WCDMA CPICH forward link.

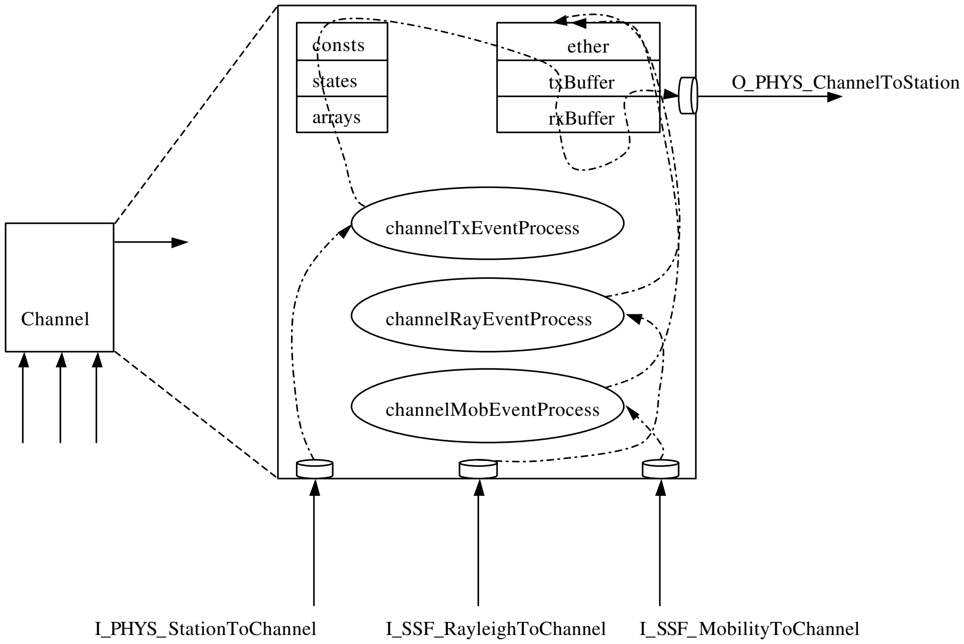


Figure B.7: The various processes and main data structures in the `CHANNEL`

- `I_SSF_RayleighToChannel` : Connects `Rayleigh` to `FwdChannel` and to `RevChannel` for providing an error free SSF channel to pass short-scale rayleigh fading values to the air channels.

- `I_SSF_MobilityToChannel` : Connects `Mobility` to `FwdChannel` and to `RevChannel` for providing an error free SSF channel to pass long-scale distance and shadow fading values to the air channels.

The following SSF processes are part of the `CHANNEL` class:

- `channelTxEventProcess` : This process gets activated when simulation data events arrive over `I_PHYS_StationToChannel` SSF inChannel from the transmitters (`Mobiles` for `RevChannel` and `Bases` for `FwdChannel`). All events are set up to arrive at the same SSF time, and once they are dequed, the buffers are set filled with simulation data (I/Q chips from all transmitters). Then the process uses the short scale and long scale fading values to compute the received chips at the receivers after incorporating the propagation and interference effects. Once the entire calculation is done for all receivers, the resultant (channel-effected) test data is packed into SSF events and sent to the receivers (`Bases` for `RevChannel` and `Mobiles` for `FwdChannel`) over the `O_PHYS_ChannelToStation` SSF outChannel.

- `channelMobEventProcess` : This process gets activated every time there is a control message from the `Mobility` object on the `I_SSF_MobilityToChannel` SSF inChannel. The control message updates the long-scale loss from a MS to a BS for both the `RevChannel` and `FwdChannel`.

- `channelRayEventProcess` : This process gets activated every time there is a control message from the `Rayleigh` object on the `I_SSF_RayleighToChannel` SSF inChannel. The control message updates the short-scale loss from a transmitter antenna to a receiver antenna for each multipath for both the `RevChannel` and `FwdChannel`.

`MASTER` SSF Abstraction

The `MASTER` class extends the SSF entity API. Figure B.8 illustrates the block diagram of the `MASTER` class. The following SSF processes are part of the `MASTER` class:

- `masterBERProcess` : This process is internally timed by SSF and after each activation, the `Master` accesses the statistics from `Base`[0] to `Base`[11] and created statistics averaged over all BS and over time. These averaged statistics are a final output of the simulation.

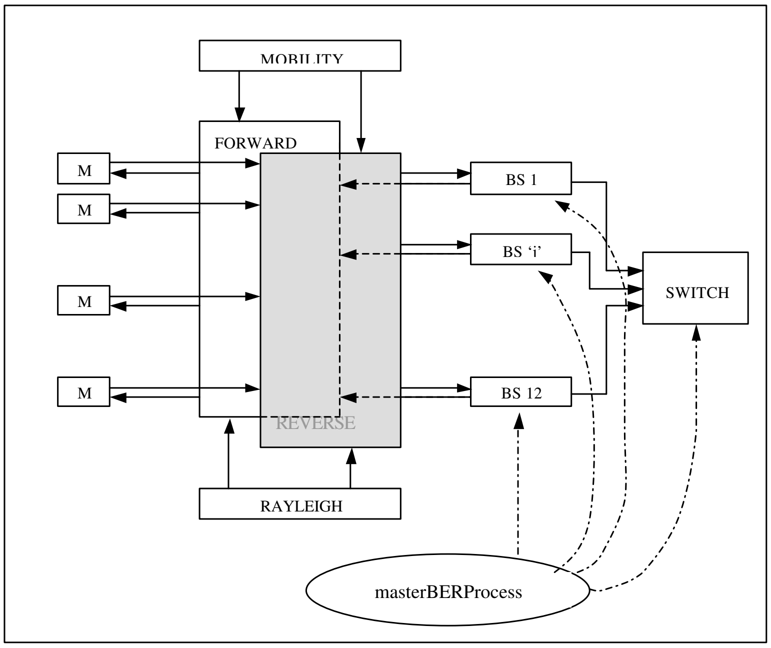


Figure B.8: The various processes and main data structures in the `MASTER`

Appendix C

WiPPET$_{\text{signal}}$ Event Flow

Message sequence charts are given to illustrate the flow of events in WiPPET$_{\text{signal}}$ and the interactions among the various entities. The objects in WiPPET$_{\text{signal}}$, viz. `Mobile`, `Base`, `Switch`, `RevChannel`, `FwdChannel`, `Rayleigh`and `Mobility`are connected to each other through SSF channels. All simulation data and maintainence information transfer is done on these SSF channels during the simulation except during initialization, when pointers to each object are used.

Initialization

WiPPET$_{\text{signal}}$ initialization involves the calling of the constructors of each of the classes that are public SSF entities. The last step in the constructors of `CHANNEL`, `RAYLEIGH`, `MOBILITY`, `BASE`, `MOBILE` and `MASTER` is the initialization of all the SSF processes in the entity. The `Master` class also maps the SSF channels from one entity to another before calling the SSF process. The processes get activated only once all the constructors have been executed. The `Master` object calls a public SSF method called *startAll()* which starts all the processes that are aligned to the `Master`.

Figure C.1 illustrates the flow of information from one object to another during initialization. This transfer is done using pointers. Note that this is done before the

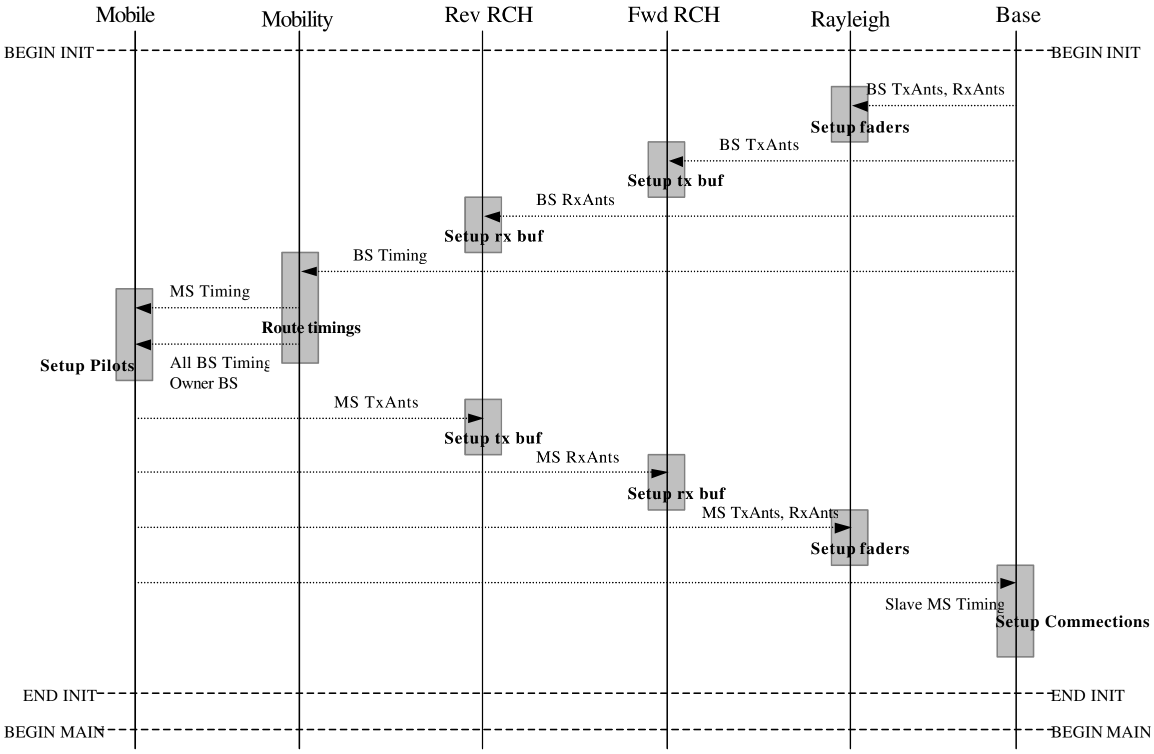


Figure C.1: Initialization flow in WIPPET$_{\text{signal}}$: Pointer referencing is used for all registrations and information transfer among the entities

channels have been mapped. This method is called **Registration** and there are public methods in the *Target Entity* which are called by the *Registering Entity*. For example, all the `Base`s and the `Mobile`s register with the `Rayleigh` and the `FwdChannel` and the `RevChannel` entities to update them about the number of hardware transmitting and receiving antennas they have. This lets the target entities initialize their arrays and states. The `Base`s register with the `Mobility` entity and update it about their timings. The `Mobility` in turn generates random timings for the `Mobile`s and during `Mobile` registration updates them about their and BS timings so that the `Mobile`scan set up the Pilot Manager. The `Mobile`s also register with their designated BS with their timing and IDs.

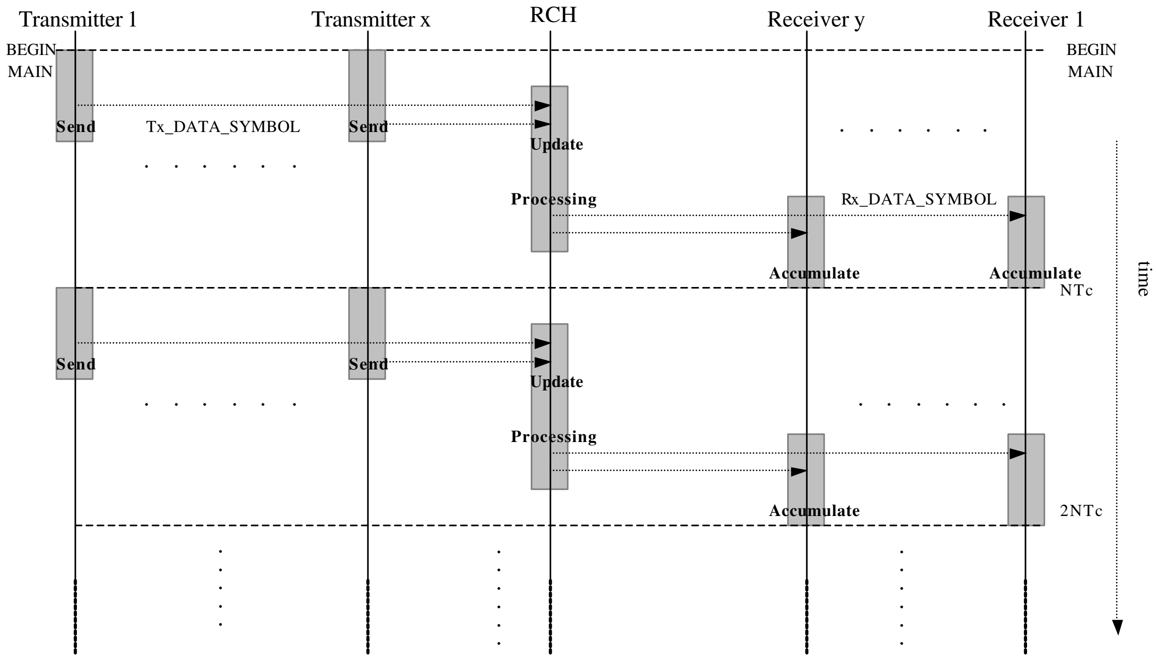


Figure C.2: Steady state flow in WiPPET$_{\text{signal}}$: Simulation data comprises of spreaded symbols (I and Q) and is received by the RCH (`RevChannel` or `FwdChannel`) from their respective transmitters, processed by the RCH and then sent to the designated receivers

Steady State

Once all the entities are initialized, the *startAll()* call triggers all processes in all the entities. The `Mobiles` start sending events to the `RevChannel` in the reverse link and the `Bases` start sending events to the `FwdChannel` in the forward link. The `Rayleigh` and `Mobility` entities also keep updating the `RevChannel` and `FwdChannel` about the short-scale and long-scale losses.

Channel

Figure C.2 illustrates the flow of simulation data from the transmitters to the receivers via the Radio Channels. Each event (Tx_DATA_SYMBOL) and (Rx_DATA_SYMBOL) comprises of a sequence of I and Q chips that represent one spreaded symbol. All

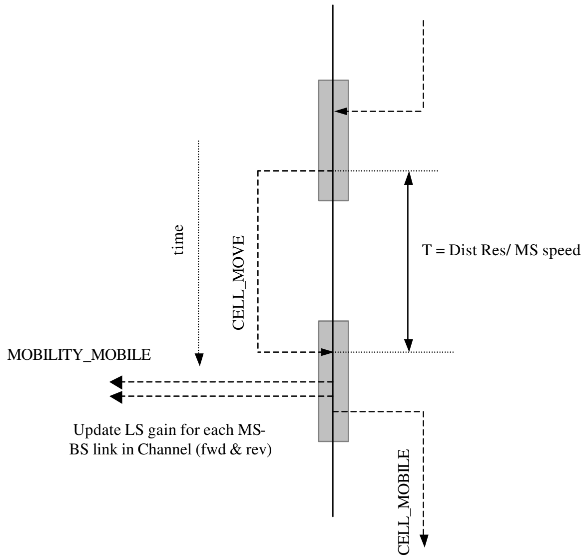


Figure C.3: Mobility flow in WiPPET$_{\text{signal}}$: Updates about the long-scale losses are sent to both the `RevChannel` and `FwdChannel` based on the speed of the mobile and distance precision of the geography

Tx_DATA_SYMBOL events arrive at the RCH (`RevChannel` or `FwdChannel`) at the same time. They are dequeued, and then processed; i.e. the channel effects of propagation and interference are incorporated, and signals as received at the receivers are created. Once one symbol worth of chips are created to be delivered, they are sent to the designated receiver inside the R_DATA_SYMBOL event. In Figure C.2 the events from `Mobility` and `Rayleigh` are not shown for purpose of clarity. Reverse link simulation flow means that the transmitters are `Mobiles`, the RCH is `RevChannel` and the receivers are `Bases`. Similarly, for forward link simulation flow, the transmitters are `Bases`, the RCH is `FwdChannel` and the receivers are `Mobiles`.

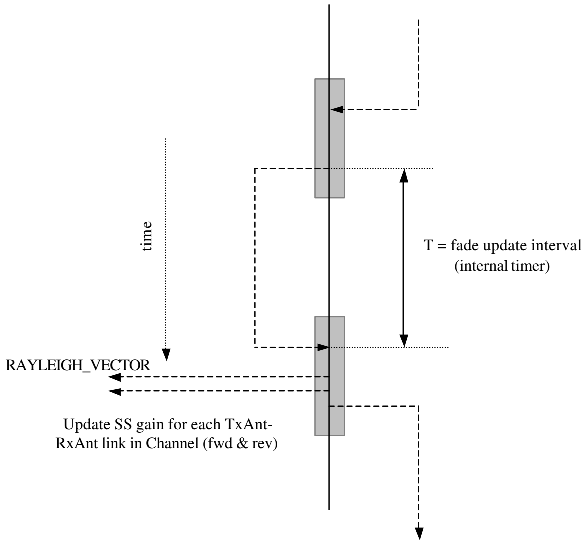


Figure C.4: Rayleigh flow in WiPPET$_{\text{signal}}$: Updates about the short-scale losses are sent to both the `RevChannel` and `FwdChannel` based on the precision of the rayleigh faders

Mobility

The `Mobility` entity sends updates about long-scale losses from each mobile position to all base station positions. The rate at which these updates are sent depends on the speed of the mobile and the distance precision of the geography model. Figure C.3 illustrates the flow. An internal event (CELL_MOVE) is sent with the mobile designation, and on its reception, the next location of the mobile is chosen. Losses corresponding to that position are encapsulated in a MOBILITY_MOBILE events and dispatched to both the `RevChannel` and the `FwdChannel`.

Rayleigh

The `Rayleigh` entity sends updates about short-scale (Rayleigh) losses for every possible (x, i, m, y, j) signal path in the system. The rate at which these updates are sent depends on the precision of the Rayleigh faders. WiPPET$_{\text{signal}}$ normally uses a

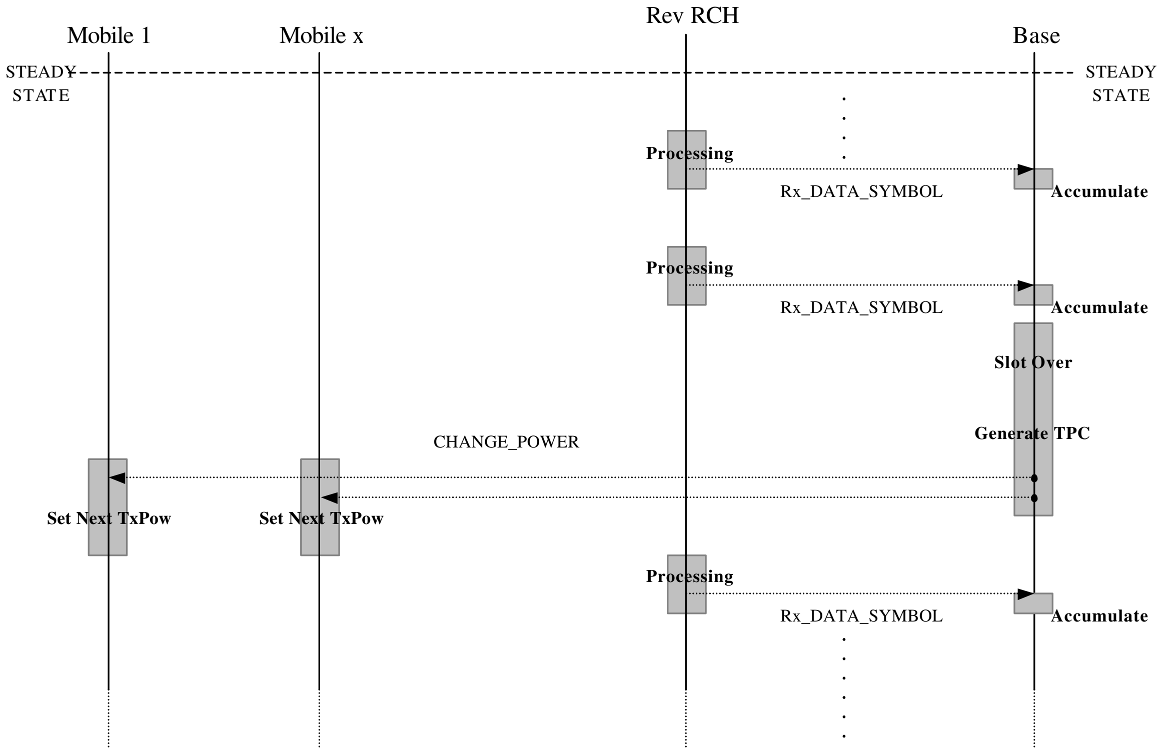


Figure C.5: Power control clow in WIPPET$_{\text{signal}}$: The BSs estimate the SIR of each of their MS and compare it with a target level. An error free feedback is sent over to the MS which increase or decrease their transmit power level by a constant factor.

precision of 1 Rayleigh sample per symbol, although it is possible to go to as fine a precision as one sample per chip. Figure C.4 illustrates the flow. An internal timer in the entity expires every symbol, on which the next fading value for each signal path in the system is generated, and stored in the RAYLEIGH_VECTOR event. Once all values are generated and stored, the events are dispatched to the `RevChannel` and the `FwdChannel`.

Power Control

WIPPET$_{\text{signal}}$ implements outer closed loop power control. Figure C.5 illustrates a simple case of the power control loop. The `Base` is constantly receiving Rx_DATA_SYMBOL events from the `RevChannel` which carries the received signal spreaded symbol. The `Base`s accumulate this data and add AWGN to it and also keeps track of when the power control group (a WCDMA slot) gets over. At that time, the `Base` estimates the SIR of each of its `Mobile`s and then sends a feedback encapsulated in a CHANGE_POWER protocol message event on an error free SSF channel for the `Mobile` to increase or decrease its transmit power for the next slot. This feedback can also be delayed by some amount to analyze the effect of protocol delay on the performance of the system.

Mobiles that are in soft-handoff with two or more BSs receive CHANGE_POWER protocol message events from all the active `Base`s. The `Mobile` collects all commands, and increases the transmit power only if all the BSs have send an increase command. Otherwise, it decreases its power.

Bit Detection

Figure C.6 illustrates the flow during bit detection and the subsequent calculation of BER in WiPPET$_{\text{signal}}$. Let us first focus on `Base`-1. It receives Rx_DATA_SYMBOL events from `RevChannel` which it accumulates while adding AWGN. It also keeps track of the number of symbols it has received. We now describe a *fake frame* as data that has been pre-stored, received and accumulated which contains 15 WCDMA slots of simulation data received from all MS via the `RevChannel`. This fake frame has a sufficient window to contain data from all MS, even if their timing with reference to this BS is different. So, in fact, this frame will have one WCDMA frame and a few more symbols to provide a window for all MS to contribute their one WCDMA frame. Once the fake frame is over, the BS detects the bits for all its MS, and makes hard decisions

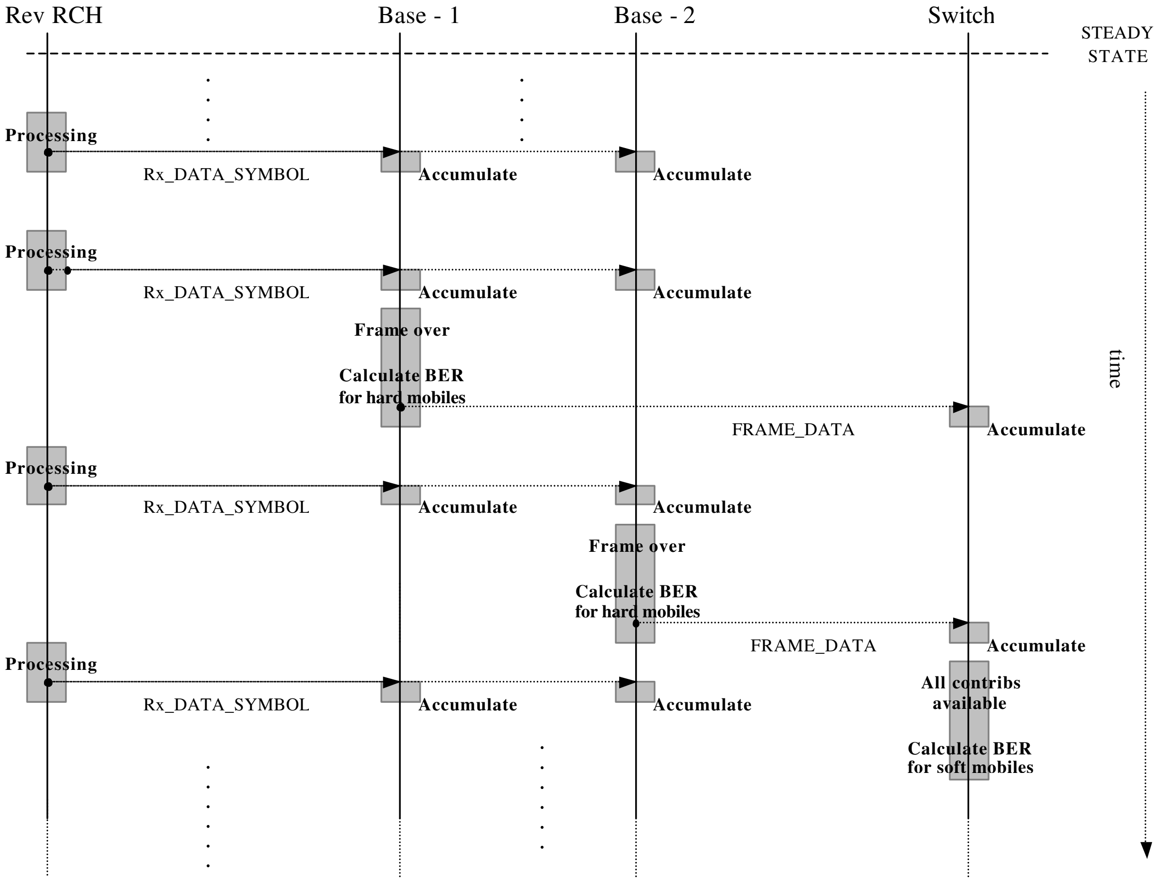


Figure C.6: Bit detection flow in WiPPET$_{\mathrm{signal}}$: The BSs calculate BER for all MSs that are not in soft-handoff. The Switch entity receives soft data for soft-handoff MS and combines that to calculate BER

on all except the ones that are in soft-handoff with other BSs. Similarly, `Base`-2 detects the bits for its own MS.

Now, let us assume that there are one or more `Mobile`s that are in soft-handoff with these two `Base`s. For that MS, when the bits are being detected at the respective `Base`s, the hard decision taken for the data is not included in the average statistics. Instead, the *soft data* is encapsulated into a FRAME_DATA event that is dispatched to the `Switch`, which has already created a setup for receiving these soft data from BS for particular MS. Once the `Switch` receives all contributions for the MS, it combines them, and then makes a hard decision to detect the bits for that MS.

Soft Handoff

The `Mobile`s in WIPPET$_{\text{signal}}$ make decisions about initiating and terminating soft-handoff with BS other than their active BS. The MS divides all BSs into active, candidate and neighbor sets. Initially one BS is designated active and all others are candidates. Based on the criterion for soft-handoff, the BSs switch between candidate and active sets during the simulation.

The protocol messages from the `Mobile` to the `Base` and also from the `Base` to the `Switch` can be sent with delays to simulate the effect of higher layer protocol transmission delays in the real system.

Initiation

Figure C.7 illustrates the initiation of a soft-handoff for `Mobile`-1 with `Base`-1 and `Base`-2. This `Mobile` gets the Rx_DATA_SYMBOL events from `FwdChannel`, which it accumulates. The mobile is given the relative timings of all bases, so that it triggers itself to start measuring the power level for each base after each slot. This level is averaged over the course of the simulation. After every slot, the mobile ranks the BSs in decreasing order of their current average power levels and makes a decision on whether to upgrade a candidate BS to an active BS. Once this decision is made, the `Mobile` sends a protocol message HARD_TO_SOFT to its currently active BSs (`Base`-1 in Figure C.7). It also sends another protocol message CONFIG_SOFT to the new active BS (`Base`-2 in Figure C.7), which is told of the long code state and the timing of this mobile, so that it can start decoding it. `Base`-1 and `Base`-2 update their current *hard* and *soft* mobile list and send protocol messages CONFIG_SOFT_BASE to the

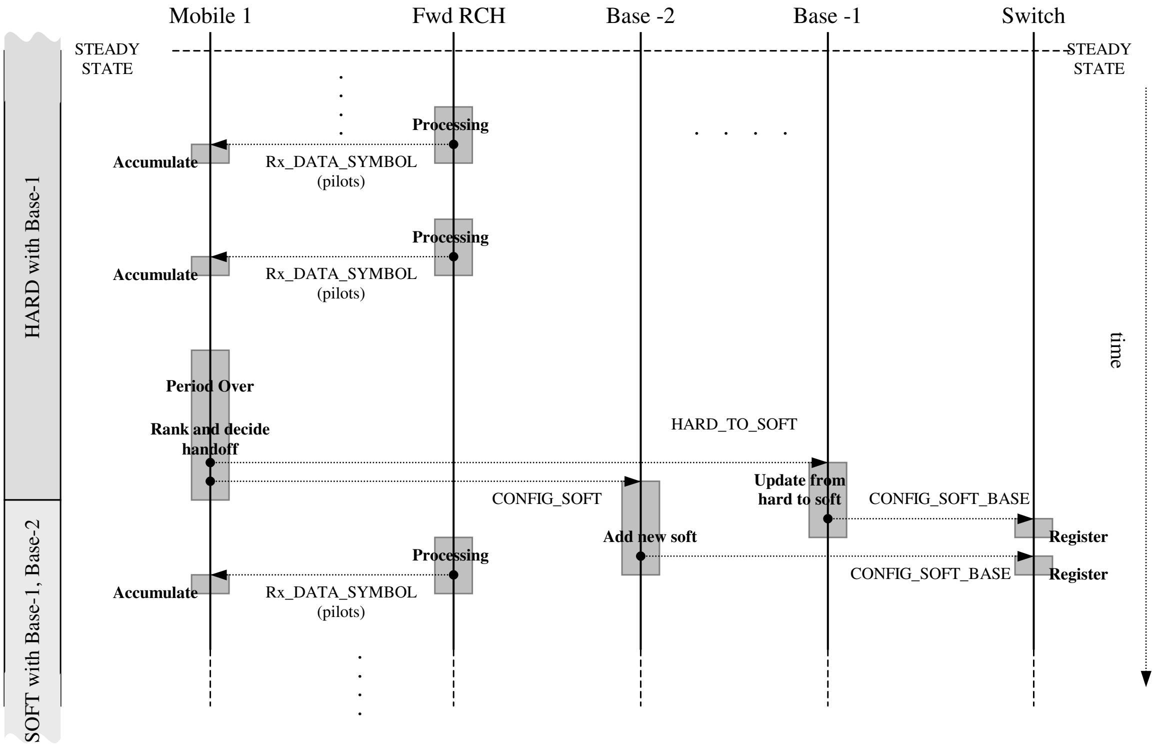


Figure C.7: Soft handoff initiation flow in WIPPET$_{\text{signal}}$: The MS, BSs and the Switch exchange protocol messages which initiates the soft-handoff

`Switch` which contains information about the mobile that is in soft-handoff with them. The switch then establishes a setup where it awaits soft data from these two BSs which can be combined and BER calculated for the mobile (`Mobile`-1).

Termination

Write here Figure C.8 illustrates the termination of a soft-handoff for `Mobile`-1 with `Base`-1 and `Base`-2. As described in the previous section, the MS makes decisions about upgrading or downgrading of BSs between active and candidate sets. Let us assume that the mobile (`Mobile`-1 in Figure C.8) decides to terminate the soft-handoff with `Base`-1 and revert to a normal connection with `Base`-2. Once this decision is made, the `Mobile` sends a protocol message DECONFIG_MOBILE to `Base`-1. It also sends

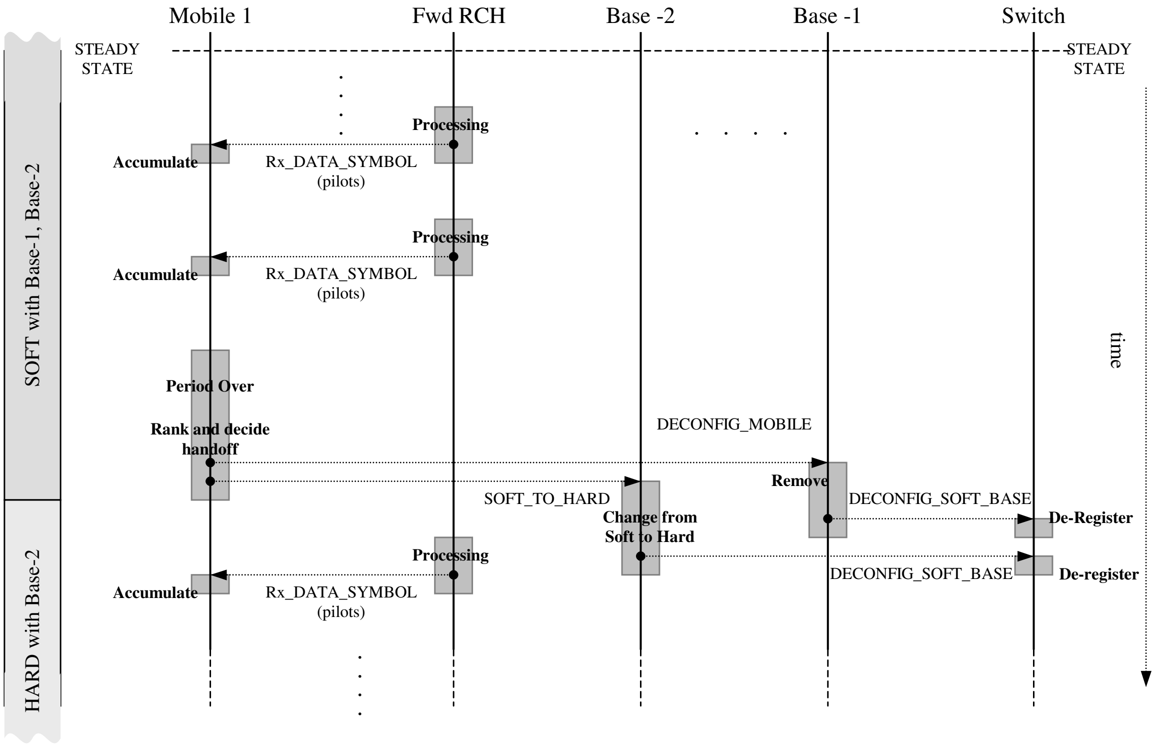


Figure C.8: Soft handoff termination flow in WIPPET$_{\text{signal}}$: The MS, BSs and the Switch exchange protocol messages which terminates the soft-handoff

another protocol message SOFT_TO_HARD to `Base`-2. The bases update their current *hard* and *soft* mobile lists and send protocol messages DECONFIG_SOFT_BASE to the `Switch` which contains information about the mobile that will no longer be in soft-handoff with them.

References

[1] SSFNET website : http://www.ssfnet.org.

[2] 3GPP. Technical Specification Group Radio Access Network; Multiplexing and channel coding (FDD). 3G TS 25.212 version 3.2.0.

[3] 3GPP. Technical Specification Group Radio Access Network; Spreading and modulation (FDD). 3G TS 25.213 version 3.2.0.

[4] F. Adachi, K. Ohno, A. Higashi, T. Dohi, and Y. Okumura. Coherent multicode DS-CDMA mobile radio access. E79-B:1316–1325, Sep 1996.

[5] H. Andoh, M. Sawahashi, and F. Adachi. Channel Estimation Filter using time-multiplexed pilot channel for Coherent RAKE combining in DS-CDMA mobile radio. E81-B:1517–1525, Jul 1998.

[6] F. Babich, O. E. Kelly, and G. Lombardi. A variable-order discrete model for the mobile propagation channel. In M. Luise and S. Pupolin, editors, *Broadband Wireless Communications (Proc. 9th Tyrrhenian Internat. Workshop on Digital Comm., Lerici, Italy, 1997)*, pages 259–266. Springer–Verlag, London, 1998.

[7] K. G. Chen. Integrated dynamic radio resource management of wireless communication systems. Masters thesis, Dept. of ECE, Rutgers, The State University of New Jersey, New Brunswick, NJ, May 1996.

[8] C.N. Chuah and R. Yates. Evaluation of a minimum power handoff algorithm. In *Sixth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC'95*, pages 623–627, 1995.

[9] C.N. Chuah, R. Yates, and D. Goodman. Integrated dynamic radio resource managemant. In *Proc. IEEE Vehicular Technology Conference, VTC-95*, pages 584–588, 1995.

[10] R. H. Clarke. A statistical theory of mobile-radio reception. *Bell System Technical Journal*, 47:957–1000, 1968.

[11] R. M. Fujimoto. Parallel discrete event simulation. *Communications of the ACM*, 33(10):30–53, October 1990.

[12] D. J. Goodman. *Wireless Personal Communication Systems*. Addison-Wesley, first edition, 1997. pages 350–355.

[13] M. Gudmundson. Correlation model for shadow fading in mobile radio systems. *Electron. Lett.*, 27(23):2145–2146, 1991.

[14] Y. Honda and K. Jamal. Channel estimation based on time-multiplexed pilot symbols. IEICE Technical Report.

[15] W. C. Jakes, editor. *Microwave Mobile Communications.* Wiley, 1974.

[16] P. F. Reynolds Jr. A spectrum of options for parallel simulation. In *Proceedings of the 1988 Winter Simulation Conference, WSC'88*, pages 325–332, December 1988.

[17] V. Kaul, W. Zhang, and R.D. Yates. Multi-cell WCDMA Signal Processing Simulation. Sep 2000.

[18] F. Ling. Coherent detection with reference-symbol based estimation for direct sequence CDMA uplink communications. *Proc. IEEE Vehicular Technology Conference, VTC-98*, pages 400–403, May 1993.

[19] R. Lupas and S. Verdú. Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channels. *IEEE Transactions on Information Theory*, 35:123–136, Jan 1989.

[20] U. Madhow and M. Honig. MMSE Interference Suppression for Direct-Sequence Spread-Spectrum CDMA. *IEEE Transactions on Communications*, 42(12):3178–3188, Dec 1994.

[21] S. Moshavi. Multi-user detection for DS/CDMA communications. *IEEE Communications Magazine*, 34(10):124–136, Oct 1996.

[22] D. M. Nicol and R. M. Fujimoto. Parallel simulation today. *Annals of Operations Research*, 53(3):249–286, 1994.

[23] Ojanpera and R. Prasad. An Overview of CDMA Evolution towards Wideband CDMA. *IEEE Communications Surveys*, 1998.

[24] J. Panchal, O. Kelly, J. Lai, N. Mandayam, A. T. Ogielski, and R. Yates. Parallel simulations of wireless networks with TED: Radio propagation, mobility and protocols. *Performance Evaluation Review*, 25(4):30–39, Mar 1998.

[25] J. Panchal, O. Kelly, J. Lai, N. Mandayam, A. T. Ogielski, and R. Yates. Wippet, a virtual testbed for parallel simulations of wireless networks. In *Parallel And Distributed Simulations - PADS*, 1998.

[26] Jignesh Panchal. Wippet : Wireless propagation and protocol evaluation testbed. Masters thesis, Dept. of ECE, Rutgers, The State University of New Jersey, New Brunswick, NJ, Sep 1998.

[27] P. Patel and J. Holtzman. Analysis of a Simple Successive Interference Cancellation Scheme in DS/CDMA Systems. *IEEE JSAC, Special Issue on CDMA*, 12(4):796–807, Jun 1994.

[28] K. Perumalla and R. Fujimoto. A C++ instance of TeD. TR GIT-CC-96-33, College of Computing, Georgia Institute of Technology, 1996.

[29] John G. Proakis. *Digital Communications.* McGraw-Hill International Editions, 1995.

[30] T. S. Rappaport. *Wireless Communications*. Prentice Hall, first edition, 1996. pages 69–192.

[31] T. S. Rappaport, Scott Y. Seidel, and Koichiro Takamizawa. Statistical channel impulse response models for factory and open plan building radio communication system design. *IEEE Transactions on Communications*, 39(5):794–807, May 1991.

[32] S. Sampei and T. Sunaga. Rayleigh fading compensation for QAM in land mobile radio communications. *IEEE Transactions on Vehicular Technology*, VT-42:137–147, May 1993.

[33] M. Sawahashi, H. Andoh, and K. Higuchi. Interference Rejection Weight Control for pilot sumbol-assisted Coherent Multistage Interference Canceller using Recursive Channel Estimation in DS-CDMA mobile radio. E81-A No. 5:957–972, May 1998.

[34] M. Sawahashi, H. Andoh, K. Higuchi, and F. Adachi. Experiments on Coherent Multistage Interference Canceller for DS-CDMA mobile radio. Sep 1998.

[35] M. Sawahashi, Y. Miki, H. Andoh, and K. Higuchi. Pilot Symbol-assisted Coherent Multistage Interference Canceller using Recursive Channel Estimation for DS-CDMA mobile radio. E79-B:1262–1270, Sep 1996.

[36] S. Seo, T. Dohi, and F. Adachi. SIR-based transmit power control for reverse link DS-CDMA mobile radio. E81-B:1508–1516, Jul 1998.

[37] B. Sklar. Rayleigh fading channels in mobile digital communication systems part 1. *Communications Magazine*, 35(9):136–146, 1997.

[38] Gordon L. Stuber. *Principles of Mobile Communications*. Kluwer Academin Publishers, 1996.

[39] SWG2, ARIB IMT-2000 Study Committee. Specifications of Air-Interface for 3G Mobile System. 3, Dec 1998. 1.0-1.0.

[40] M. K. Varanasi and B. Aazhang. Multistage Detection for Asynchronous Code-Division Multiple-Access Communications. *IEEE Transactions on Communications*, 38:509–519, Apr 1990.

[41] R. Yates, S. Gupta, C. Rose, and S. Sohn. Soft dropping power control. In *Proceedings of The IEEE Vehicular Technology Conference VTC 97*, pages 1694–1698, May 1997.

[42] R. Yates and C.Y. Huang. Integrated power control and base station assignment. *IEEE Transactions on Vehicular Technology*, 44(3):638–644, August 1995.