Snapshot of MAC, PHY and Propagation Models for IEEE 802.11 in Open-Source Network Simulators

Masood Khosroshahy, Thierry Turletti, Katia Obraczka

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Snapshot of MAC, PHY and Propagation Models for IEEE 802.11 in Open-Source Network Simulators

Masood Khosroshahy, Thierry Turletti, Katia Obraczka

N° 6310

Septembre 2007

Thème Com

ISSN 0249-6399

Snapshot of MAC, PHY and Propagation Models for IEEE 802.11 in Open-Source Network Simulators

Masood Khosroshahy, Thierry Turletti, Katia Obraczka

Thème Com – Réseaux et systèmes
Projet Planète

Rapport de recherche n° 6310 – Septembre 2007 - 20 pages

Abstract: Simulation is an essential component of the validation chain in the design of network protocols. Indeed, while simulation is not the only tool used for data networking research, it is extremely useful because it often allows research questions and prototypes to be explored at many orders-of-magnitude less cost and time than that required to experiment with real implementations and networks. In this report, we focus on the simulation of IEEE 802.11 Physical (PHY) and Medium Access Control (MAC) layers and provide a survey of current IEEE 802.11 network simulators. We believe that such survey will help network researchers to select the best simulator according to the requirements of their simulations. Furthermore, we present a detailed description of the YANS prototype network simulator, and especially its physical layer implementation, which will be partly ported in the upcoming NS-3 network simulator.

Keywords: GloMoSim, IEEE 802.11, J-Sim, JiST-SWANS, Network Simulators, NS-2, NS-3, OMNET++, Physical Propagation Models, YANS.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Snapshot des couches MAC, PHY et des Modèles de Propagation IEEE 802.11 implantés dans les Simulateurs Réseaux Open-Source

Résumé: La simulation est un maillon essentiel de la chaîne d'évaluation des protocoles réseaux. En effet, elle permet d'évaluer à moindre coût les nouveaux algorithmes dans un environnement totalement contrôlable. Notre étude se focalise sur la simulation de la couche physique et MAC des réseaux locaux sans fil IEEE 802.11. Nous décrivons une vue d'ensemble des simulateurs réseaux IEEE 802.11 open-source utilisé de nos jours dans la communauté réseaux et fournissons un comparatif détaillé entre eux. Nous pensons qu'une telle étude est utile aux chercheurs désirant sélectionner le simulateur réseau le plus approprié à l'algorithme à évaluer. De plus, nous présentons en détail la couche physique du prototype de simulateur réseau YANS qui servira de base au nouveau simulateur réseau NS-3, successeur du très populaire NS-2.

Mots clés: GloMoSim, IEEE 802.11, J-Sim, JiST-SWANS, Modèles de propagation, NS-2, NS-3, OMNeT++, Simulateurs Réseaux, YANS.

1 Introduction

Simulation is an essential component of the validation chain in the design of network protocols. Indeed, while simulation is not the only tool used for data networking research, it is extremely useful because it often allows research questions and prototypes to be explored at many orders-of-magnitude less cost and time than that required to experiment with real implementations and networks. In this report, we focus on the simulation of IEEE 802.11 Physical (PHY) and Medium Access Control (MAC) layers and provide a survey of network simulators. A number of network simulators exist in research community and the most popular are NS-2, OMNET++, GloMoSim, J-Sim and JiST/SWANS for free open-source simulators and OPNET and QualNet industrial network simulators. The two latter simulators have not been studied in this report due to their commercial nature and the fact that their source codes are either not in the public domain or otherwise unavailable for inspection and modification, compared to the above-mentioned simulators. We believe that such survey will help network researchers to select the best simulator according to the requirements of their simulations. Furthermore, we present a detailed description of the YANS prototype network simulator and especially its physical layer implementation, which will be partly ported in the upcoming NS-3 network simulator.

This report is organized as follows. Section 2 provides a global view of IEEE 802.11 PHY and MAC layers. We first start by giving a general introduction to the standard by briefly explaining the features of both PHY and MAC layers. As will be seen, the implementation of PHY Layer is generally very basic in almost all network simulators. The common denominator has been identified as the chosen propagation model, which is not part of IEEE 802.11 standard series; nonetheless, it plays a key role in having a near-realistic PHY layer model. In light of this matter, different propagation models, i.e., Large-scale Path Loss (Free-Space, Two-Ray and Shadowing) and Fading models, are briefly explained in this section as well. In Section 3, we first introduce each chosen simulator and then mention the result of its inspection regarding availability and comprehensiveness in implementation of different aspects discussed in Section 2. Section 4 is dedicated to YANS network simulator's IEEE 802.11a PHY layer and propagation models. In this section, we mention the major features that have been taken into account during the design phase of this part of the simulator. Where worthwhile, implementation-related information has also been provided. We conclude the paper in Section 5, by providing reference tables summarizing all the discussed features of IEEE 802.11 in the set of network simulators presented.

2 IEEE 802.11 PHY/MAC and Propagation Models in a Nutshell

In 1997, IEEE standardized the first Wireless Standard: 802.11. This included both MAC and PHY layers. The motivations behind introducing such a standard were: offering services which up to the time, were only available in wired networks; offering high throughput with acceptable reliability and providing continuous network connectivity to the users.

According to the standard, the stations can communicate in Basic Service Set (BSS) mode using an Access Point (AP); this is called the infrastructure mode. When there is no AP in the network, the BSS is called Independent BSS (IBBS). The term *ad-hoc* refers to the case where we do not have an AP in the network and nodes are communicating directly. When there are multiple Infrastructure BSSs in a network, it is advantageous that access points communicate with each other to facilitate traffic forwarding and mobility of stations among different BSSs. This architecture, where APs are cooperating, is called Extended Service Set (ESS). While the IEEE 802.11 standard and all the later extensions provide extensive information regarding different aspects of the communication, we do not intend to summarize all that information in this introduction. In the coming three sections, we briefly mention the concepts in MAC layer, PHY layer and Propagation Models. For an extensive treatment of the standard (MAC and PHY layers), we refer the reader to the numerous published books and to the IEEE 802.11 standards themselves. Propagation models have been also treated extensively in numerous books on communication topics.

2.1 IEEE 802.11 MAC Layer

MAC layer, as its primary purpose, has the functionality of providing reliable data delivery mechanism over the unreliable wireless air interface. It is the layer that manages station accesses to the shared wireless medium. The original standard utilizes Carrier Sense Medium Access with Collision Avoidance (CSMA/CA) as the access mechanism. This access method, however, wastes a significant percentage of channel capacity, but, it is a necessary feature to provide reliability in data transmission. Among many other features, it also supports the optional Request-To-Send/Clear-To-Send (RTS/CTS) mechanism.

2.1.1 Distributed Coordination Function (DCF)

DCF is the basic 802.11 MAC layer. DCF uses the above-mentioned CSMA/CA method to share the medium between the stations. It may optionally use the RTS/CTS method as well. Under this method, collision rate is relatively high and there is no notion of Quality of Service (QoS) in the network.

2.1.2 Point Coordination Function (PCF)

PCF is an optional coordination function that is defined only in infrastructure mode, where stations are connected to an access point. AP is the element in control of access in the network and it uses two periods to enforce its policies. There is a Conten-

tion Period, in which, DCF method is used. The second period is the Contention Free Period, in which AP basically allows stations, by sending them a special authorization, to send packets.
The recent IEEE 802.11e standard addresses the existing limitations in DCF and PCF. It particularly addresses the problem of QoS provisioning in the network by introducing a new coordination function: Hybrid Coordination Function (HCF).

2.1.3 Enhanced DCF Channel Access (EDCA)

EDCA is a method of channel access within the HCF in IEEE 802.11e. An EDCA is basically a QoS-enabled DCF. This is done by introducing the notion of traffic classes, by giving priority, in channel access, to real-time data, compared to delay-tolerant data.

2.1.4 HCF Controlled Channel Access (HCCA)

HCCA is a QoS-enabled PCF. It also uses EDCA during the Contention Period. Stations transmit the information about their queues status and traffic classes to the AP and, based on this information, AP coordinates access to the medium between the stations.

2.2 IEEE 802.11 PHY Layer

IEEE 802.11 PHY layer is the interface between the MAC layer and the air interface. The frame exchange between Physical layer and MAC is under the control of Physical Layer Convergence Procedure (PLCP). Physical Layer is the entity in charge of actual transmission using different modulation schemes over the air interface. It also informs the MAC layer about the activity status of medium.

Currently, there are four standards defining the physical layer: IEEE 802.11a, 802.11b, 802.11g and 802.11n. Among these, IEEE 802.11n is the newest which is still under standardization. It utilizes Multiple-input-multiple-output (MIMO) technology to achieve significantly higher rates. These Physical Layer standards define their operating frequency band, number of available channels, possible transmission rates, modulation and Forward Error Correction (FEC) coding schemes. Propagation models are not part of the Physical Layer standards; however, as mentioned before, they form an integral part of physical layer implementation in the simulators.

2.3 Propagation Models: Overview and Description of Respective Scenarios

In this section, we explore both concepts of Large-scale Path Loss and Fading. We introduce three models of Large-scale Path Loss that account for the large-scale attenuation of signal based on distance: Free-Space, Two-Ray and Log-normal Shadowing. As will be presented hereafter, however, the level of sophistication and the inclusiveness of the models increase from the simple model of Free-space to the more realistic model of Shadowing. On the other hand, Fading is the phenomenon responsible for rapid fluctuations of signal over a short period of time or distance.

2.3.1 Free-Space Model

This model is used to predict the signal strength when the transmitter and the receiver have a clear, unobstructed line-of-sight (LOS) path between them. It predicts that the received power decays as a function of Transmitter-Receiver distance raised to some power – typically to the second power. The well-known Friis equation is used to calculate the received power.

2.3.2 Two-Ray Model

This model, which is a more realistic model than the Free-Space model, addresses the case when we consider a ground-reflected propagation path between transmitter and receiver, in addition to the direct LOS path. This model is especially useful for predicting the received power at large distances from the transmitter and when the transmitter is installed relatively high above the ground. It is interesting to note that at far distances, the received power becomes independent of the frequency. Also, the received power attenuates much more rapidly with distance, compared to the Free-Space model, i.e., attenuates to the fourth power of the distance.

2.3.3 Log-normal Shadowing

The empirical approach for deriving radio propagation models is based on fitting curves or analytical expressions that recreate a set of measured data. Adopting this approach has the advantage of taking into account all the known and unknown phenomena in channel modeling. A widely-used model in this category is Log-normal Shadowing. In this model, power decreases logarithmically with distance. The average loss for a given distance is expressed using a Path Loss Exponent. For taking into account the fact that surrounding environmental clutter can be very different at various locations having the same Transmitter-Receiver distance, another parameter is incorporated in the calculation of path loss. According to measurement results, this parameter, called Shadowing hereafter, is a zero-mean Gaussian distributed random variable (in dB) with a standard deviation, also expressed in dB. Shadowing accounts for the fact that measured data are sometimes significantly different from the average power at a given distance from the transmitter. For calculating the received power based on this model, we first calculate the received

power at a reference distance using the Friis formula. Then, we incorporate the effect of path loss exponent and shadowing parameters.

2.3.4 Fading Model

The term *Fading* is used to describe the rapid fluctuations of the amplitudes, phases, or multipath delays of a signal over a short period of time or distance. It is caused by interference between multiple versions of the transmitted signal which arrive at the receiver at slightly different times. Hence, the resulting signal at the receiver may have a wide-varying amplitude and phase. In short, the effects of multipath are rapid changes in signal strength over a small travel distance or time interval, random frequency modulation due to varying Doppler shifts on different multipath signals and time dispersion caused by multipath propagation delays. The multipath components combine vectorially at the receiver which causes the signal to distort, to fade or even to strengthen at times.

Type of fading experienced by the signal going thorough a channel depends on the nature of the signal and the characteristics of the channel. If the bandwidth of the signal is smaller than the bandwidth of the channel, or equally in the time domain, the delay spread of the channel is smaller than the symbol period, the fading is considered to be flat. Otherwise, the fading channel is considered to be frequency-selective. If the Doppler spread is far smaller than the signal bandwidth, or alternatively, the coherence time of the channel is greater than the symbol transmission period, then the fading is considered to be slow. Otherwise, we have a fast fading channel. Rayleigh distribution is commonly used to describe the statistical time varying nature of the received envelope of a flat fading signal, or the envelope of an individual multipath component. When there is a dominant stationary, non-fading signal component present, such as a line-of-sight propagation path, the fading envelope distribution is *Rician*. However, the Rician distribution degenerates to a *Rayleigh* distribution when the dominant component fades away.

3 Snapshot of IEEE 802.11 Implementation in Network Simulators

In this section, we report on the state-of-the-art of the implementation of each of the aspects discussed in the previous section in the set of network simulators. In each of the following sections, we first briefly introduce the simulator and then go on to discuss its major implementation features.

3.1 NS-2

NS-2 [1] is by far the most popular network simulator within the research community. It has been around since 1989 and it has emerged as a variant of REAL network simulator. Its development has been supported by various grants over the years and it has received substantial contributions from the researchers all over the world. As will be clear in the coming few sections, most of the IEEE 802.11-related modules have been contributed by researchers not directly associated with the project, hence, these modules are not bundled into the main distribution.

3.1.1 MAC

For the MAC layer, there have been five major contributions to NS-2. However, these five contributed modules have been developed separately and have not been built on the work of the preceding efforts. So, the user has no choice but to select one over the other, considering the needs of the current project at hand.

There were two early efforts to develop an original IEEE 802.11 MAC. In the first one, a Distributed Coordination Function (DCF) was developed by Carnegie Mellon University [2]. Their extension to NS-2 was intended to simulate mobile nodes connected by wireless network interfaces, including the ability to simulate multi-hop wireless ad hoc networks. In a later project [3], a Point Coordination Function (PCF) was added to the simulator. The module allowed a station to become a Point Coordinator and send beacons. The station could initiate Contention Free Periods and poll other stations during these periods in order to provide different levels of priority.

After standardization of IEEE 802.11e, up to now, there have been three major contributed modules to NS-2 implementing issues discussed in this standard. The most comprehensive module is that of INRIA-Planète Group [4]. In this module, both HCF Controlled Channel Access (HCCA) and Enhanced DCF Channel Access (EDCA) have been implemented. It is worth mentioning, however, that this module later served as the basis of IEEE 802.11 module in YANS(NS-3) Network Simulator (Introduced later in the survey) and has undergone major improvements and bug-fixing since then.

The other major contributed module is that of University of Pisa [5]. They have developed an HCF Controlled Channel Access (HCCA) module which allows for a flexible integration of different scheduling algorithms. In their module, a classifier tags incoming packets with the appropriate traffic stream identifier. The HCCA scheduler is used at both QoS AP and QoS stations.

The last MAC module for NS-2 that we would like to mention is that of “Technische Universität Berlin” [6]. Their work extends the wireless and mobility code, which has been developed in the CMU Monarch project. They have added the contention free bursting (CFB), or TXOP bursting, to their model, which allows the transmission of a train of small packets without intermediate contention.

3.1.2 PHY-Propagation Models

Like the MAC module, there have been several PHY-Propagation Model modules contributed to the NS-2 Network Simulator. However, as far as we know, there is only one model implementing some features of an IEEE 802.11 Physical Layer specification and that is the aforementioned model of INRIA-Planète Group [4]. This model implements an IEEE 802.11a physical layer. As mentioned before, this model served as the basis in the YANS(NS-3) project and among other improvements, non-occurrence of packet collisions in the original NS-2 module has been fixed.

As for the propagation models, NS-2 enjoys a complete set of known models: Based on the work of CMU Monarch project, there is a Free-Space model; the same project has also contributed a Two-Ray model. USC/ISI has contributed a Shadowing model, resulting NS-2 having a good set of Large-Scale Path Loss models. Antenna and Radio Communications Group of Carnegie Mellon University has contributed a Fading Channel model [7]. In their work, the fading process has been computed once and saved in a text file, distributed in their package, according to an algorithm published by them in a paper. This text file is read during a simulation and the elements therein serve as multiplicative factors to simulate the effect of signal power level fluctuations.

3.1.3 License

GPLv2 is NS-2's current license, but since the simulator has numerous contributors, the license of each specific module should be checked as a result. However, there is a specific exception added to GPLv2 which states that the module copyright holder gives the right that the model can be combined with free software programs or libraries that are released under the GNU LGPL license. Pre-existing software in the project are mostly governed by Original BSD license. Some new codes are under Apache 2.0 license. As recommended by NS-2 developers, new code should preferably use GNU GPL, with the specific exception, and if not possible, should use the Modified BSD license, Apache 2.0 license or the original BSD license.

3.2 OMNET++

OMNeT++ [8] is a simulation environment which has become quite popular recently. It is not a network simulator by itself, but has served as the basis of some communication network simulators. Due to its generic nature, it has also found application in simulation of IT systems, queuing networks and even hardware architectures. As for IEEE 802.11 simulation, the implementations are in three different projects which are based on the OMNET++ simulation framework: INET Framework [9], Ipv6SuiteWithINET [10] and Mobility Framework [11]. As is unfortunately the case in many other open-source projects, the development efforts have not been coordinated, so the user needs to choose one of these packages for their simulations, considering the features needed. Hereafter, we mention what is available in each package.

3.2.1 MAC

Regarding the supported MAC modes, among the three packages, INET Framework and Mobility Framework both support Ad-hoc operation, but, Ipv6SuiteWithINET falls short of offering this possibility. However, for Infrastructure operation, it is just the Mobility Framework which does not support this feature.

All three packages support some form of Distributed Coordination Function (DCF) feature, albeit with some differences. Mobility Framework offers CSMA/CA with RTS/CTS, however, the support of INET Framework does not include RTS/CTS. Also, the DCF implementation in Ipv6SuiteWithINET only works in the context of Infrastructure mode.

As for the Point Coordination Function (PCF) feature, it is just the Mobility Framework which has no implementation at all. Unfortunately, up to this date, none of the packages has support for IEEE 802.11e MAC, or more specifically, for Hybrid Coordination Function (HCF).

3.2.2 PHY-Propagation Models

On the Physical Layer side, OMNeT++ based packages have performed poorly. All of the three packages have implementations based on IEEE 802.11b specification, but, the only implemented propagation model is the basic Free-Space.

3.2.3 License

OMNeT++ is governed by GPL license for academic use. However, for commercial use, a Commercial License from SimulCraft has to be obtained.

3.3 GloMoSim

GloMoSim [12], developed by Parallel Computing Laboratory at UCLA, is a scalable simulation environment for wireless network systems, and has utilized a parallel discrete-event simulation capability provided by Parsec, a C-based simulation language developed in the same group. Unfortunately, the simulator is no longer under active development in the context of the original project, but is under development in the name of another commercial simulator called QualNet.

3.3.1 MAC

The only supported MAC mode is that of Ad-hoc mode. Distributed Coordination Function (DCF) has been implemented by integrating a CSMA/CA with RTS/CTS. There is no support for Point Coordination Function (PCF), nor is there support for the new IEEE 802.11e MAC, i.e., Hybrid Coordination Function (HCF) which was non-existent at the time of the last release of the simulator.

3.3.2 PHY-Propagation Models

On the Physical Layer side, there is a partial implementation of 802.11-1997. As for the Propagation Models, there are implementations of classical formulas for Free-Space and Two-Ray Large-Scale Path Loss models. Shadowing model has not been implemented though, but, there is an implementation of Rician Fading channel for use.

3.3.3 License

GloMoSim is free for educational use (Access to download only granted to academic Top Level Domains). It is not covered by a standard well-known license though. The user has the right to copy and modify the software at the condition that the resulting software is offered at no charge to research community and the original copyright notice should be included in any derivative work. Commercial license can also be obtained from UCLA.

3.4 J-Sim

J-Sim [13] Network Simulator has been developed in the context of a PhD thesis in Ohio State University. Illinois University has also been significantly involved in the project. J-Sim (formerly known as JavaSim) is a component-based simulation environment which has been built upon the notion of Autonomous Component Programming Model. On top of the autonomous component architecture, a generalized packet switched network model has been designed in order to be able to do network modeling and simulation. Similar to NS-2, J-Sim is a dual-language simulation environment in which classes are written in Java and glued together using Tcl/Java. Unfortunately, J-Sim does not have a feature-rich IEEE 802.11 module; the existing features are explained hereafter.

3.4.1 MAC

In J-Sim, there is only support for Ad-hoc MAC, i.e., there is no implemented Point Coordination Function (PCF) functionality. In the Distributed Coordination Function (DCF) implementation, CSMA/CA with support for RTS/CTS and Power Saving Mode are the major available features. There is also no support for the new IEEE 802.11e MAC, i.e., for Hybrid Coordination Function (HCF).

3.4.2 PHY-Propagation Models

On the Physical Layer side, things look even grimmer. There are only few basic functionalities of the Physical Layer; hence, not adhering to any particular standard. As for the available Propagation Models, Free-Space and Two-Ray models have been implemented, but, there is neither Shadowing model, nor Fading Channel in the implemented Physical Layer. However, there is an interesting Propagation Model which seems to be a distinctive feature of J-Sim Physical Layer: Irregular Terrain Model. This model is based on electromagnetic theory and on statistical analyses of both terrain features and radio measurements, and predicts the median attenuation of a radio signal as a function of distance and the variability of the signal in time and in space. The model requires altitude on each point of the earth which can be obtained from Globe data that can be downloaded from a mentioned URL. When using Irregular Terrain Model, one must use ellipsoidal latitude and longitude coordinates instead of Cartesian coordinates.

3.4.3 License

J-Sim developers have released their code under the BSD license.

3.5 JiST-SWANS

Another relatively new project is that of Cornell University: JiST-SWANS [14]. It is a high-performance discrete event simulation engine that runs over a standard Java virtual machine. They have proposed a concept called “Virtual machine-based simulation” and JiST serves as a prototype of this idea. SWANS is a scalable wireless network simulator built on top of JiST platform. A complete wireless network configuration can be formed by SWANS's independent software components. However, implementation of IEEE 802.11 module is not exhaustive. Here are the details:

3.5.1 MAC

In SWANS, there is support for Ad-hoc MAC mode, but not for Infrastructure mode, i.e., Distributed Coordination Function (DCF) is implemented with good detail, but, there is no Point Coordination Function (PCF) functionality. The DCF implementation is according to the modifications of IEEE 802.11b and it has support for features such as RTS/CTS, retransmission, NAV and backoff. The current MAC implementation misses support for the new IEEE 802.11e MAC, i.e., for Hybrid Coordination Function (HCF).

3.5.2 PHY-Propagation Models

On the Physical Layer side, there is an implementation of few basic functionalities of IEEE 802.11b. As for the Large-scale Path Loss Models, it has support for Free-Space and Two-Ray propagation models, but not for Shadowing model. It has also an implementation of Rayleigh/Rician Small-scale Fading for the physical layer, establishing the simulator's place among the simulators with relatively good physical layer implementation.

3.5.3 License

JiST-SWANS is not governed by a well-known license. The important aspects of the license are: Cornell Research Foundation is the copyright holder; it is free for non-commercial academic use; any derivative work should acknowledge the original work and be released under the same license; usage of software outside the United States may require approval from the U.S. Government.

3.6 YANS

YANS [15] is a prototype network simulator developed by the INRIA's Planète group. The primary goal of the development of Yet Another Network Simulator , YANS for short, has been to build a clean, solid-core event-based simulator. Its development decision has been taken due to short-comings of the existing open-source network simulators, and its code base, due to the partnership of Planète group with NS-3 project initiative [16], will be ported to the upcoming NS-3 Network Simulator. The primary module in YANS, due to the research interests of the Planète group, is the IEEE 802.11 module. Hereafter, we summarize the existing features.

3.6.1 MAC

The implemented MAC has support for both Ad-hoc and Infrastructure modes. Distributed Coordination Function (DCF) feature has been implemented, but, there is no implementation for the Infrastructure counterpart, i.e., for Point Coordination Function (PCF). The distinctive feature of YANS is that there is a complete and feature-rich implementation of the new IEEE 802.11e MAC, i.e., for Hybrid Coordination Function (HCF). Both HCF Controlled Channel Access (HCCA) and Enhanced DCF Channel Access (EDCA) have been implemented enabling the user to conduct simulations in both Ad-hoc and Infrastructure modes.

3.6.2 PHY-Propagation Models

IEEE 802.11a specification has been faithfully adhered to, in the implementation of IEEE 802.11 Physical Layer. All the well-known propagation models are also precisely integrated into the simulator: The classical formulas for Free-Space and Two-Ray propagation models have been implemented. The Shadowing model is also integrated into the simulator with great detail. In the implementation of Shadowing mode, a reference power, at a reference distance, is calculated using the Friis formula. The effect of Path Loss Exponent and Log-normal Shadowing is then incorporated. A table for guiding the user to choose the right values for the parameters according to any given environment is included. The implementation needs IT++ library [17] to be installed on the system. The simulator uses the library both at compilation time and at run-time. Small-scale Fading model has also been designed and carefully integrated into the simulator structure. The model is for slow flat fading channels supporting both Rayleigh and Rician cases. Like the Shadowing model, it needs IT++ library for both compilation and run-time. Extensive parameters are at user's disposal to tweak the model to their satisfaction. The user can also choose BER formulas according to the desired channel type (Different fading cases and AWGN case). Desired error distribution type could be indicated as well.

3.6.3 License

The YANS code has been released under GPLv2 license, binding users to contribute back their code and modifications to the project.

4 Overview of the YANS PHY Implementation

The implementation of IEEE 802.11 physical layer may is a tricky part in network simulators. This is due the fact that there are host of phenomena which should be taken into account if we are to accurately model the IEEE 802.11. As mentioned before, the propagation modeling is not part of the IEEE 802.11 physical layer standard, yet, it affects significantly the physical layer performance, therefore, affects any other type of protocol evaluation in any upper layer. In this section, we have opted for introducing the physical layer implementation in YANS network simulator. The intention is to shed light on how physical layer implementation could be approached and communicate to the community the experiences that we have had while implementing this design approach. The overall structure of IEEE 802.11 implementation is depicted in Figure 1.

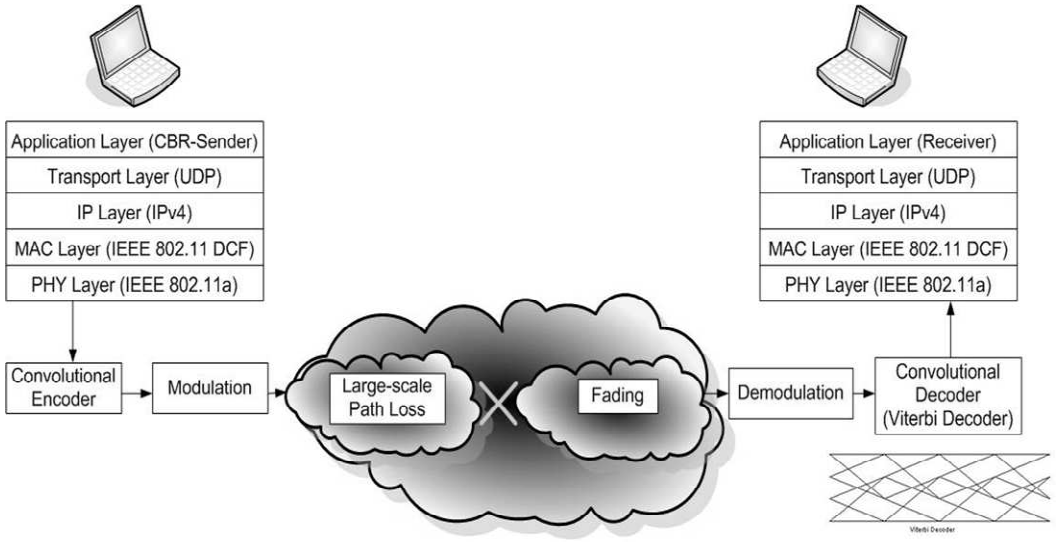

Figure 1. Overall View of IEEE 802.11 Modeling

While still not all the features mentioned in standard are implemented, the most important parts, in terms of their effects on the overall performance, have been studied and implemented. In the following sections, each of the implemented parts, as depicted in Figure 1, have been explained.

4.1 Packet Reception Method

As YANS is an event-based simulator, for receiving each packet we have the following two events:

- An event at the start of reception (first bit of a packet)
- An event at the end of reception (last bit of a packet)

The *SNIR(t)* function is evaluated twice for each packet: 1/ For the first bit, for deciding whether or not the packet could be received, considering the current state of PHY and the *SNIR(t)* level. 2/ For the last bit, for calculating the final *SNIR(t)*, considering what has happened during the packet reception, and for calculating the PER.

The PHY layer can be in one of four possible states:

- TX: the PHY is currently transmitting a signal. While the PHY is in this state, a received packet will be dropped regardless of its SNIR(t) level.
- SYNC: the PHY is synchronized on a signal and is waiting until it has received its last bit. While the PHY is in this state, another received packet will be dropped regardless of its SNIR(t) level. But, its signal level is recorded and taken into account in Noise Interference changes of the first packet on which the PHY was synchronized.
- BUSY: the PHY is not in the TX or SYNC, but the energy measured on the medium is higher than Energy Detection Threshold. While the PHY is in this state, a packet can be received if its SNIR(t) level is above the threshold.
- IDLE: the PHY is not in the above states. The behavior is the same as BUSY state, i.e., while the PHY is in this state, a packet can be received if its SNIR(t) level is above the threshold.

4.1.1 Steps taken when the last bit of a packet is received

When the last bit of the current packet, upon which the PHY is synchronized, is received, we again evaluate the *SNIR(t)* function and calculate the PER. Here are the details:

We remind that if any other packet was received during this time, i.e., from the first to the last bit of the current packet, all the received signal levels are recorded in the Noise Interference, *Ni*, vector and is taken into account for the current packet *SNIR(t)* calculation. If indeed, there was any other packet, i.e., the Ni vector has some elements, for each element of the vector, we calculate a Chunk Success Rate (CSR), taking into account the number of bits in that chunk, the respective *SNIR(t)* level in that chunk and the transmission mode (Modulation type, transmission rate, Convolutional code's coding rate). The CSR calculation uses the theoretical BER formulas, based on modulation type, and also takes into account the Convolutional code properties. It is in Chuck Success Rate calculation that we mention the desired type of error distribution within the packet. This process is then repeated for every *Ni* change recorded (since we have a different *SNIR(t)* value for each chunk, hence different BER and CSR). We multiply all these calculated CSRs to get the Packet Success Rate; hence the PER.

After having calculated the PER, we draw a random number from a uniform random number generator, between 0 and 1, and compare it against the PER. Whether the random number is higher than the PER or lower, we decide to mark the reception as correct, or as erroneous, respectively.

4.2 Convolutional Encoder – Viterbi Decoder

Convolutional Encoding is the standard method proposed in the IEEE 802.11a for Forward Error Correction – FEC. The Convolutional encoder used in IEEE 802.11a is depicted in Figure 2. The generator polynomials, in octal format, are g_0=133 and g_1=171 and, as evident from the figure, the base coding rate is $^1/_2$. With puncturing, however, we reach to the coding rates of $^2/_3$ and $^3/_4$.

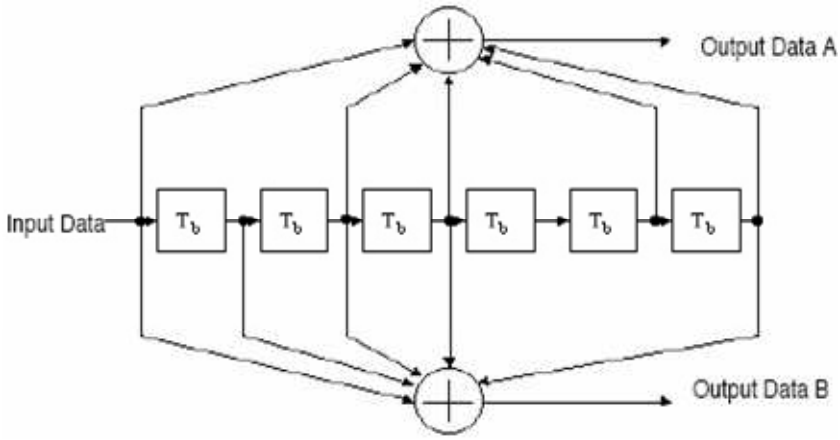

Figure 2. The Convolutional Encoder Used in IEEE 802.11a (from [18])

Viterbi Decoding is the recommended way of decoding Convolutional codes in the standard. The Bit Error Rate is not equal before and after the Viterbi decoder, due to error correction mechanisms provided by Convolutional codes. The procedure to derive the BER after the decoder is as follows. As the first step, we calculate the probability of selecting an incorrect path by the Viterbi decoder which is in distance k from the all-zero path (due to linear characteristics of the encoder, without loss of generality, we consider that the sent data were a train of zero bits). The probability P_k is derived as in Equation Set 1. [19]

$$P(k) = \sum_{n=\frac{k+1}{2}}^{k} \binom{k}{n} p^n (1-p)^{k-n}$$

$$k : odd \qquad (1)$$

$$P(k) = \sum_{n=1+k/2}^{k} \binom{k}{n} p^n (1-p)^{k-n} + \frac{1}{2}\binom{k}{\frac{1}{2}k} p^{k/2}(1-p)^{k/2}$$

k : even

Where *p* is the BER before the decoder.

However, computation of this formula takes a lot of processing power, especially if it is done for several k values in each run. To improve the performance, according to [19], we utilize the Chernoff upper bound for calculating P_k which gives nearly the same result with significantly less computation overhead.

$$P(k) < [4p(1-p)]^{k/2} \qquad (2)$$

k: even or odd

For calculating BER for each chunk of bits in the packet (Note that chuck was the set of bits over which *SNIR* value is constant, i.e., if there is no interference in the reception of the packet, each packet is comprised of two chunks; one for Physical layer header, or *PLCP* header, and one for the Physical layer payload), we calculate the first 10 elements of P_k, multiply each by the corresponding C_k[1] value and sum over the result of multiplications. This sum is the BER after decoder for the bits in the given chuck. BER is calculated from C_k and P_k values according to the Equation 3 [20, 21].

$$BER < \frac{1}{Punc} \sum_{k=d_{free}}^{\infty} C_k P_k \qquad (3)$$

Punc, in Equation 3, is the puncturing period of the Convolutional code. Typical values of free distance (d_{free}) and $C_{k=d}$ for various Convolutional codes are mentioned in a study documented in [21].

As evident in the preceding paragraphs, the implementation of Convolutional encoder and Viterbi decoder in the simulator is not direct, i.e., these components have not been actually implemented. Instead, for the sake of both lessening the implementation burden and decreasing simulation run-time, using the mentioned concepts and equations, the effects of these two components have been taken into account in the simulator core calculations.

4.3 Modulator –Demodulator

IEEE 802.11a uses OFDM on the Physical Layer. From the 52 OFDM sub-carriers, 48 ones carry data bits. In each sub-carrier, data bits are sent with BPSK, QPSK, or M-QAM modulation. Table 1 summarizes all the information regarding the modulation schemes and Convolutional codes details that are standardized in IEEE 802.11a air interface.

[1] Ck is the bit error number associated with each error event of distance k

Data rate (Mbits/s)	Modulation	Coding rate (R)	Coded bits per subcarrier (N_{BPSC})	Coded bits per OFDM symbol (N_{CBPS})	Data bits per OFDM symbol (N_{DBPS})
6	BPSK	1/2	1	48	24
9	BPSK	3/4	1	48	36
12	QPSK	1/2	2	96	48
18	QPSK	3/4	2	96	72
24	16-QAM	1/2	4	192	96
36	16-QAM	3/4	4	192	144
48	64-QAM	2/3	6	288	192
54	64-QAM	3/4	6	288	216

Table 1. Modulation and Coding Schemes, from [18].

For each sending bit rate, it mentions the modulation scheme used in each data sub-carrier, the Convolutional coding rate, coded bits per sub-carrier, the total of coded bits per each sent OFDM symbol and the total number of the original data bits, i.e., before the encoder, in each OFDM symbol sent over the air interface.

In the demodulator side, to calculate the *BER*, or *p* in Equation Set 1, we go through the following process:

$$P_r \rightarrow SNIR \rightarrow E_{bit}/N_0 \rightarrow BER$$

Where P_r is the received signal power, *SNIR* is the signal to noise plus interference ratio, E_b is energy per bit and N_0 is the noise power density. In what follows, we provide the details of this process.

In every chuck in the packet, where N_i and signal level are constants, we calculate *SNIR* from received power (P_r). At the same time, E_b/N_0 can be calculated from *SNIR* by Equation 4.

$$\frac{E_b}{N_0}(k,t) = SNIR(k,t)\frac{B_t}{R_b(k,t)} \qquad (4)$$

Where E_b is energy per bit, N_0 is the noise power density, B_t is the bandwidth of the signal (20 MHz in 802.11a) and $R_b(k,t)$ is the bit rate of transmission for packet *k* at time *t*.

Derivation of *BER* from E_b/N_0, however, depends on the modulation type and transmission channel conditions. In what follows, we provide the relevant information for accurately utilizing the available theoretical formulas to derive the *BER*.

4.3.1 Base Formulas

BER formulas are mostly written based on the Q-function. For reference, we mention the relationship between the Q-function and *erfc* function in Equation 5 [22] (the *erfc* function exists in math library of C language).

$$Q(x) = 0.5 \times erfc(\frac{x}{\sqrt{2}}) \qquad (5)$$

The relationships between bit-based and symbol-based expression of formulas are given in Equation Set 6 [23].

$$\gamma_b = SNR_b = \frac{E_b}{N_0} \qquad \gamma_s = SNR_s = \frac{E_s}{N_0}$$
$$SNR_s = \log_2^M \times SNR_b \qquad P_s = \log_2^M \times P_b \qquad (6)$$

Where P_s and P_b are Symbol Error Probability/Rate and Bit Error Probability/Rate, respectively. The above approximate conversions typically assume that the symbol energy is divided equally among all bits, and that Gray encoding is used so that at reasonable *SNRs*, one symbol error corresponds to exactly one bit error.

4.3.2 Fading Related Definitions

Definitions:

- T_s: Symbol Transmission Duration
- T_c: Signal Fade Duration
- *Average Error Probability* (P_s): Averaged over the distribution of SNRs

- *Outage Probability (P_{out})*: Defined as the probability that SNR falls below a given value corresponding to the maximum allowable Ps

4.3.3 Correspondence between type of error probability and type of fading channel

- *Slow Fading: $T_s << T_c$*
  Better to use: Outage Probability

A deep fade will affect many simultaneous symbols. Thus, fading may lead to large error bursts, which cannot be corrected for with coding of reasonable complexity. Therefore, these error bursts can seriously degrade end-to-end performance. In this case acceptable performance cannot be guaranteed over all time or, equivalently, throughout a cell, without drastically increasing transmission power. Under these circumstances, an outage probability is specified so that the channel is deemed unusable for some fraction of time or space. This type of Fading Channel is more relevant to Indoor 802.11 Networks.

- *Normal Fading: $T_s \sim T_c$*
  Better to use: Average Probability of Symbol Error

Since many error correction coding techniques can recover from a few bit errors, and end-to-end performance is typically not seriously degraded by a few simultaneous bit errors, the average error probability is a reasonably good figure of merit for the channel quality under this condition.

- *Fast Fading: $T_c << T_s$*
  Better to use: BER for AWGN channel

Fading will be averaged out by the matched filter in the demodulator. Thus, performance is the same as in AWGN.
BER formulas as functions of *SNR*, modulation type and channel type are presented in Table 2.

			Ref. -Notes
AWGN	BPSK	$P_b = Q(\sqrt{2\gamma_b})$	[23]
	QPSK	$P_s(E) = 2Q(\sqrt{\frac{E_s}{N_0}}) - Q^2(\sqrt{\frac{E_s}{N_0}})$	[24]
	M-QAM	$P_s = 1 - \left(1 - \frac{2(\sqrt{M}-1)}{\sqrt{M}} \times Q(\sqrt{\frac{3\overline{\gamma_s}}{M-1}})\right)^2$	[23] [1]
Slow Fading	All Mod.	$P_{out} = 1 - e^{-\gamma_0/\overline{\gamma_s}}$	[23] [2]
Normal Fading	BPSK	$\overline{P_b} = \frac{1}{2}[1 - \sqrt{\frac{\overline{\gamma_b}}{1+\overline{\gamma_b}}}]$	[23]
	QPSK	$\overline{P_{s,Ray}} = 1 - \frac{1}{M} - \frac{1}{\sqrt{1+\alpha}} + \frac{1}{\pi\sqrt{1+\alpha}} \tan^{-1}[\sqrt{1+\alpha}\tan(\pi/M)]$ where $\alpha = 1/[\frac{E_s}{N_0}\sin^2(\pi/M)]$	[22] [3]
	M-QAM	$\overline{P_s} = \frac{\alpha_M}{2}[1 - \sqrt{\frac{0.5\beta_M\overline{\gamma_s}}{1+0.5\beta_M\overline{\gamma_s}}}]$ where $\alpha_M = \frac{4(\sqrt{M}-1)}{\sqrt{M}}$ and $\beta_M = \frac{3}{M-1}$ (Rectangular M-QAM)	[23]
Fast Fading	All Mod.	Like the AWGN case	[23]

1: $\overline{\gamma_s}$ is Average Energy per Symbol and we assume that we have Rectangular Signal Constellation.
2: P_{out} is independent of modulation type.
3: $\overline{P_{s,Ray}}$ is average symbol error probability for Rayleigh fading and M is 4 for QPSK

Table 2. BER Formulas

4.4 Propagation Models

All the three classical large-scale path loss models, i.e. Free-Space, Two-Ray and Shadowing, are implemented in the simulator. By selecting one of these models, simulator calculates the received power using the respective formula (Formulas are mentioned in Table 4).

In the implementation of the Shadowing model, at the start of execution and during the initialization of the classes, we generate a vector of random numbers, used as shadowing parameter, with specified shadowing variance and mean. IT++ [17] library, a widely-used C++ library, has been integrated into the simulator to aid with the random number generations. We loop through this vector and read its elements during the execution of the program. The vector elements are taken as Shadowing and used at the power calculation of the corresponding symbol.

Fading, as mentioned before, is the phenomenon responsible for rapid fluctuations of signal over a short period of time or distance. In reality, we can have only one channel type, be it Large-scale Path Loss Channel, or Fading Channel. However, due to modeling constraints, it was chosen to separate what each of these two models represents, i.e., when we have only Large-scale Path Loss, then the channel can be chosen to act so, however, when we want to have Fading channel in the simulator, we need to use both models in cascade. The first part of the channel would be one of three Large-scale Path Loss Models and the second part of the channel would be the Fading channel. In this type of approach, Fading channel will not have effect on the power of signal on average; it only introduces power fluctuations to the received signals. It is the Large-scale Path Loss model who accounts for the general attenuation of signal power based on distance.

The current implementation in YANS, models a slow flat fading channel, i.e., the channel is neither frequency-selective, nor of fast fading type. According to the results reported in [25], each Wi-Fi channel bandwidth is not larger than the coherence bandwidth, so, considering the channel frequency non-selective, seems to be a safe assumption. Also, the channel does not experience any changes during the transmission of each symbol, i.e., channel's coherence time is bigger than transmission time of each symbol. This latter assumption is again logical, especially in the context of indoor 802.11, where we do not have extremely fast movements in the environment.

Like in the implementation of Shadowing model, *IT++* library has been used in the implementation of the fading channel. This implementation is very flexible and puts all the power of *IT++* library at the user's disposal. The user may select a Rayleigh channel or a Rician one for simulating a slow flat fading channel. After setting the necessary parameters, we generate the fading process and use it during the simulation. During the execution of the program, we loop through the fading process matrix and upon reception of every symbol, and we take an element as the fading factor and increase the position marker in the fading process.

4.4.1 PER Calculation Methods – Error Distribution Models

In YANS, there are two implemented PER Calculation Methods. The first method is the simple Uniform Error Distribution, and the second one, is a new method presented in [26]. The first method of PER calculation makes the assumption that bit errors are uniformly distributed within the packet. In the second method, the authors in that study argue that uniform error distribution leads to over-estimation of PER. They have carried out a theoretical work leading to new PER calculation formulas which have been studied and implemented in the simulator.

5 Summary

In this report, we analyzed the state of IEEE 802.11 implementations in the widely-used open-source network simulators. As mentioned before, for evaluating IEEE 802.11 network mechanisms, one needs to have proper modeling and realistic implementation of IEEE 802.11 MAC and physical layers along with all the necessary propagation models. The choice of propagation model depends on the environment in which we assume our network has been setup. We gave a quick introduction to the well-known propagation models and went on to inspect their implementations in the simulators. In Section 4, we presented how we approached the design and the implementation of different aspects of IEEE 802.11 Physical layer and propagation models.

The contribution of this work is three-fold: Firstly, in Section 2 of this paper and in a very concise format, reader is familiarized with the terminology and the involved concepts about IEEE 802.11 MAC and Physical layer and the Propagation Models. Secondly, Section 3 inspects the existence and implementation state of the very features presented in Section 2, in the widely-used open-source network simulators. This one-of-a-kind survey on the IEEE 802.11 implementations could help the research community with the selection of a simulator with the right features considering their current project needs. Thirdly, Section 4 presented our design and implementation approaches in developing a capable feature-rich IEEE 802.11 simulator. In this section, major building blocks of IEEE 802.11 Physical layer along with the propagation models have been inspected.

Finally, the content of Section 3 of the paper have been reformatted and presented in three informative tables: Table 3 which at one glance determines whether or not a major feature (MAC functionalities, PHY standard and propagation models) is present, Table 4 and Table 5 that present the information regarding the propagation models in a greater detail. The information in the latter two tables is divided between Large-scale Path Loss models and Fading models and both give slightly more implementation-oriented information compared to what was presented in Section 3. We hope that this report would be a positive step forward in clearing up the confusion in our overly-fragmented simulator development community.

Features / Simulators	License	IEEE 802.11 MAC				IEEE 802.11 PHY-Propagation Model				
		Ad-hoc		Infrastructure		PHY Spec.	Free-Space	2-Ray	Shadowing	Rayleigh/ Rician
		DCF	EDCA	PCF	HCCA					
NS-2	GPLv.2 [1]	☼	☼	☼	☼	802.11a [2]	☼	☼	☼	☼
OMNET++	GPL-Co.[3]	☼	×	☼	×	802.11b	☼	×	×	×
GloMoSim	X-Co. [4]	☼	×	×	×	802.11-97	☼	☼	×	☼
J-Sim	BSD	☼	×	×	×	×	☼	☼	×	×
JiST-SWANS	X [5]	☼	×	×	×	802.11b	☼	☼	×	☼
YANS (NS-3)	GPLv.2	☼	☼	×	☼	802.11a	☼	☼	☼	☼

1: With specific exception. There are also other licenses: “Modified BSD”, “Apache 2.0” and “Original BSD”
2: The module has gone through bug-fixing and significant improvements in the YANS project.
3: GPL for academic use – Commercial License from SimulCraft for commercial use
4: Free for educational use – Commercial License from UCLA
5: Cornell Research Foundation is the copyright holder. Free for non-commercial academic use.

Table 3. Simulators vs Features Reference

Models / Simulators	Free Space			Two Ray	Shadowing
	$P_r = \frac{P_t G_t G_r \lambda^2}{(4 \times \pi \times d)^2 \times L}$			$P_r = \frac{P_t G_t G_r (h_t h_r)^2}{d^4 L}$	Received Power (in dBW) = Calculated Reference Power (in dBW) - Path Loss Exponent × 10.0 × log(current distance) + Shadowing
	Free Space	Two Ray	Shadow-ing	Specific Implementation Notes	
NS-2	☼	☼	☼	- Two Ray: A Cross-over distance (d_c) is calculated. P_r for distances smaller than d_c is calculated using the Free-Space model.	
OMNET++	☼	×	×	- Free Space: Model not explicitly mentioned. Parameters can not be set all at once in one place.	
GloMoSim[1]	☼	☼	☼	- Shadowing: A model named “Generic” is mentioned which resembles Shadowing model in terms of the used parameters; hence noted here for completeness	
J-Sim[2]	☼	☼	×	- Free Space: The model only calculates the Path Gain (function of λ and d), as other parameters are taken into account elsewhere in the receiving side's PHY. - Two Ray: A Cross-over distance is calculated as well. See NS-2's note.	
JiST-SWANS	☼	☼	×	- Free Space/Two Ray: The code is based on the implementation of Glo-MoSim	
YANS (NS-3)	☼	☼	☼	- Free Space/Two Ray: Classical formulas have been implemented -Shadowing: A reference power is calculated using the Free Space model. A table is at user's disposal to choose the Path Loss Exponent and Shadow-ing Variance according to the simulated environment. Simulator needs IT++ library to generate Log-normal Shadowing parameters to be used in final reception power calculation.	

- In this table, a "☼" sign, without any notes or footnotes, indicates that all of the parameters mentioned in the respective formula are taken into account and are available to be set. A "×" sign, however, indicates that the model has not been implemented at all in the simulator.
- 1: "Path loss trace files" and "SIRCIM-Simulation of Indoor Radio Channel Impulse-Response Models (Topography-Building Type)" have also been mentioned as other available path loss models
- 2: "Irregular Terrain Model" is another implemented propagation model (See explanation earlier in the paper; in J-Sim section). The necessary inputs to use this model (Most have default values however): Directory that contains Globe data files (obligatory input), number of points between sender and receiver in the terrain profile, antenna polarity, climate (desert, etc.), surface refractivity, ground dielectric, ground conductivity and signal frequency.

Table 1. Simulators vs. Large Scale Propagation Models Reference

Features / Simulators	Fading Model [Rayleigh / Rician]			
	Fading Channel Implemented	Rayleigh/ Rician	Fading Channel Class[1]	Specific Implementation Notes
NS-2	☼	☼	Not Clear	- The model is used to modulate the output of the Two-Ray model. - Inputs: MaxVelocity (to calculate Doppler freq.) and Rician K factor. - Vague points in implementation: The manner with which fading factors are applied to packets/bits. The purpose of interpolating fading elements before application.
OMNET++	×	–	–	–
GloMoSim	☼	☼	Flat	Considering time-dispersiveness of fading channel (Flat fading as opposed to freq-selective). However, no info on time-varying nature of fading channel (No Doppler freq. , etc.)
J-Sim	×	–	–	–
JiST-SWANS	☼	☼	Not Clear	- The code is based on the implementation of GloMoSim. - The only settable parameters are: Rayleigh distribution variance constant Rician K factor (Standard deviation is calculated using K factor and implemented zero- and first-order Bessel functions)
YANS (NS-3)	☼	☼	Slow Flat	- IT++ library is used to generate fading factors which are used to add fluctuation to the power level calculated by large-scale path loss models. The user can set Rician K factor, number of generated fading factors, signal max baud rate and Doppler frequency. There are also other parameters to further customize BER/PER calculation.

1: Fading Channel Class refers to any of four possible combinations of fading types: [Flat or Frequency-Selective Fading] × [Slow or Fast Fading].

Table 2. Simulators vs. Fading Models Reference

6 Bibliography

[1] NS-2 Network Simulator: http://www.isi.edu/nsnam/ns/, Version: ns-2.30 released on Sept 26, 2006.

[2] NS-2 Network Simulator – Contributed Module: Carnegie Mellon University-CMU Monarch project in their adhockey extension to NS-2 Version: 1.1.2, 11 August, 1999.

[3] NS-2 Network Simulator – Contributed Module Anders Lindgren of Lulea University of Technology Version: 0.8b 2001.

[4] NS-2 Network Simulator – Contributed Module INRIA-Planète Group, Version: 14.2 Sep 7, 2005.

[5] NS-2 Network Simulator – Contributed Module Computer Netwoking Group at the University of Pisa Version: 2006-08-23.

[6] NS-2 Network Simulator – Contributed Module Telecommunication Networks Group of Technische Universität Berlin. Version: 1.0 beta Feb. 14, 2006.

[7] NS-2 Network Simulator – Contributed Module Antenna and Radio Communications Group of Carnegie Mellon University Version: Sep.2000.

[8] OMNET++ Network Simulator: http://www.omnetpp.org/ .

[9] OMNET++ Network Simulator – INET Framework: http://www.omnetpp.org/staticpages/index.php?page=20041019113420757, Version: 2006/10/20.

[10] OMNET++ Network Simulator – Ipv6SuiteWithINET: http://ctieware.eng.monash.edu.au/twiki/bin/view/Simulation/IPv6Suite Version: 2006/08/09.

[11] OMNET++ Network Simulator – Mobility Framework: http://mobility-fw.sourceforge.net/ Version: August 13, 2006.

[12] GolMoSim Network Simulator: http://pcl.cs.ucla.edu/projects/glomosim/ Version: Last release, 2.03-Dec 2000; before switching to the commercial product QualNet.

[13] J-Sim Network Simulator: http://www.j-sim.org/ Version: 1.3 released on 2004/02/2; latest patch: 2006/05/07, patch 4.

[14] JiST-SWANS Network Simulator: http://jist.ece.cornell.edu/ Version: 1.0.6 – March 2005.

[15] YANS Network Simulator: http://yans.inria.fr/ Version: Release 0.9.0 (2006-05-20) with modifications up to December 2006.

[16] NS-3 Network Simulator: http://www.nsnam.org/ Version: No Official Release as of this Writing.

[17] IT++ C++ Library: http://itpp.sourceforge.net/ Version: 3.10.5 (15 August 2006).

[18] "ISO/IEC 8802-11:1999/Amd 1:2000(E); IEEE Std 802.11a-1999 "[Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications – Amendment 1: High-speed Physical Layer in the 5GHz band".

[19] J.G. Proakis, "Digital Communications" 4th ed., McGraw-Hill, 2001.

[20] A.J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems", University of California, Los Angeles, CA, IEEE Transactions on Communications, 1971.

[21] P.Frenger, P. Orten, T. Ottosson, "Multi-Rate Convolutional Codes", Technical Report, April 1998, Communication System Group, Chalmers University of Technology, Sweden.

[22] R. E. Ziemer and R.L. Peterson, "Introduction to Digital Communication", 2nd ed., Prentice Hall, 2001.

[23] A. Goldsmith, "Wireless Communications", Cambridge University Press, 2005.

[24] M.K.Simon, M.-S. Alouini, "Digital Communication over Fading Channels", 2nd ed., John Wiley & Sons, 2005.

[25] G.F.S. Moya, J.L.Z. Flores, “Parameters Of A 2.4GHz Wide Band Radio Channel For WLAN applications”, Univ. Autonoma Metropolitana, Mexico City, Mexico, 14th International Conference on Electronics, Communications and Computers, 2004. CONIELECOMP 2004. 16-18 Feb. 2004.

[26] R. Khalili, K. Salamatian, “An Analytical Model for Residual Errors in Convolutional Codes”, LIP6-CNRS, Université Pierre et Marie Curie, France, Technical Report, 2006