Wireless LAN Emulation

Răzvan Beuran, Lan Tien Nguyen, Khin Thida Latt,
Junya Nakata, Yoichi Shinoda

Internet Research Center
Japan Advanced Institute of Science and Technology (JAIST)

October 6, 2006
IS-RR-2006-015

Table of Contents

1 Introduction

Using radio frequency (RF) technology, wireless LANs (WLANs) transmit and receive data through the air. Related standards, such as 802.11a, b, and g define an over-the-air interface between a wireless client (notebook, PDA[1], wireless phone) and a base station (also called Access Point, AP) that is connected to the network in a wired or wireless manner. Communication is also possible between two wireless clients in what is called "ad hoc mode". This makes it possible to automatically create a peer network (the ad hoc network) between a large number of wireless clients.

Due to the advantages they provide in comparison with wired networks, such as ease of deployment and mobility, WLANs are being used on an ever increasing scale. However, according to a survey we did in [Beu-06], deploying on WLAN applications that have different Quality of Service (QoS) requirements, such as Voice over IP (VoIP) and TCP/IP-based applications, is difficult at the moment for several reasons:

i. WLAN QoS parameters (bandwidth, packet loss, delay & jitter) have a high variability in real-world environments, and this has a significant effect on application performance;

ii. Existing WLAN QoS mechanisms are only of limited use for managing contention when applications with different QoS requirements, such as VoIP calls and TCP/IP-based data traffic, share the same communication channel;

iii. Multimedia applications, such as VoIP or video streaming, require timely servicing of the voice traffic; this is a challenging task in WLANs, even when using QoS enforcement, since most currently-implemented QoS mechanisms focus on bandwidth provisioning;

iv. Roaming between access points, a typical WLAN event, introduces communication gaps that may even be of the order of seconds, an unacceptable situation for real-time applications.

As a consequence, a thorough analysis is required in order to investigate application performance issues on WLAN, especially when considering their use in environments with specific requirements, such as safety/mission-critical systems or disaster situations.

The starting point of such an analysis are real-world tests, in which we use network applications in real WLAN environments in order to understand the basic properties of WLAN systems and their interaction with those network applications. In parallel with real-world tests, a key element of our approach is WLAN emulation. Wireless LAN emulation enables us to study network applications on WLAN in a wide range of controllable network conditions. Therefore WLAN emulation makes it possible to have an insight into application performance issues that is not available using only real-world tests.

This document presents first an overview on our approach to WLAN emulation (Chapter 2). Then we describe in detail the steps that need to be taken into account when modelling wireless LANs, at physical layer, as well as at data-link and at network layer (Chapters 3, 4, and 5, respectively). This is followed by a description of the model we use for WLAN emulation, and the algorithm implemented in our prototype WLAN emulator (Chapter 6). The report ends with a section of conclusions and future work (Chapter 7), a list of acronyms, a list of references, as well as an appendix that presents the objects used in our

---

1 Personal Digital Assistant.

prototype implementation.

We would like to acknowledge here the contributions through numerous discussions to the ideas presented in this document of Toshiyuki Miyachi, Takashi Okada, Saber Zrelli, Assist. Prof. Satoshi Uda, Assist. Prof. Ken-ichi Chinen, and Assoc. Prof. Yasuo Tan.

2 Approach overview

This chapter introduces our approach to WLAN emulation. Wireless LAN emulation is a key element in our analysis of network application performance because it makes it possible for us to assess application performance on WLAN in a wide range of controllable network conditions.

2.1 Experimental setup

The setup we use for our experiments with an emulated WLAN is presented in Figure 1.
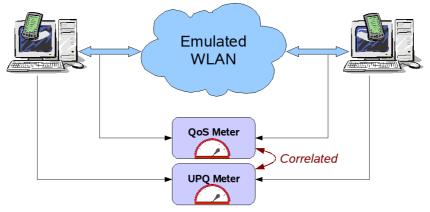

Figure 1: WLAN emulation for application performance assessment.

The setup in Figure 1 is adapted from the system we previously used to study VoIP performance on wired networks [Beu-04]. On the two end PCs we run applications similar to those available on mobile nodes, such as PDAs. Therefore the end PCs play the role of mobile nodes in our experiment. The emulated WLAN through which they communicate offers full control over the network scenario that we investigate.

Using this setup we carry out a two-level analysis. At one level, the emulated network reproduces the performance issues of WLANs, such as the dependency on signal strength and number of access points, quality degradation management techniques, etc. These are quantified objectively through the use of the "QoS Meter" block that computes the network QoS parameters based on monitored real-traffic traces.

Simultaneously, at application level, we measure the User-Perceived Quality (UPQ) for the application under study. In the case of VoIP we use methods such as the ITU-T recommendations G.107 [G.107] and P.862 [P.862]. Other classes of applications require different specific metrics, such as the goodput and transfer-time performance for TCP/IP based applications [Beu-03]. The function of assessing user satisfaction is conceptually performed by the "UPQ Meter" block.

Correlating the network and application performance in this setup permits us to establish objectively what are the requirements of an application in order to ensure user satisfaction, as well as determine what type of mechanisms are needed to meet these requirements on WLANs.

The use of the "QoS Meter" in Figure 1 is optional when the WLAN is emulated, since the emulation process itself controls the quality degradation in the network[2]. However by using a "QoS Meter" the assessment accuracy is increased through the continuos monitoring of the real traffic. This process is performed by means of network traffic taps, that mirror all the traffic on a link without affecting it, thus allowing the monitoring (sniffing) process.

The setup in Figure 1 enables us to capture the dynamic behaviour of the tested systems. Running averages and global assessments are not meaningful for short-term performance issues, which are nevertheless critical in disaster situations, for example. Moreover WLANs are dynamic environments by definition: signal reception conditions fluctuate, the number of nodes and their position vary, the access points with which the nodes communicate, or their peers in ad hoc networks change. Quality degradation in the real WLAN will vary accordingly. Through the use of the approach described above we are able to reproduce this quality degradation in the emulated WLAN, and follow application performance fluctuations over time.

2.2 Emulation process

As mentioned in the previous section, our goal is to correlate the QoS and the UPQ measurements in experiments with real applications on an emulated WLAN, as depicted in Figure 1. This makes it possible to determine objectively the relationship that exists between application performance and network quality degradation under varying network conditions.

There are several key requirements for such a methodology to be effective, and our approach addresses them. Realism of the emulation is important because it allows drawing conclusions that are thereafter useful in real-life deployments. Most application performance studies focus on bandwidth-oriented applications, typically based on TCP/IP. Existing emulators, such as W-NINE [Per-04], for instance, use models that are too simple, hence do not adequately reflect real network conditions as experienced by users, and only reproduce a trimmed down behaviour of WLANs. Models used are oversimplified in a bandwidth-oriented perspective, and the edge effects that do occur in reality, including delay variation and packet loss, are generally overlooked. Such neglected edge effects occur, for instance, during auto-rate fallback (ARF) events, when the WLAN adapters try to automatically adapt operation rate to signal reception conditions.

The transition from physical layer network effects, which are consequences of events in the physical world (signal strength variation, node movement, etc.), to data link layer and network layer effects is a task that requires the use of several techniques: modelling 802.11 protocols and understanding their behaviour, analysing real implementations of 802.11 protocols, and finally reproducing the interaction with the application traffic.

---

2   Note that when carrying out tests in a real WLAN, and the end nodes are genuine wireless systems, the "QoS Meter" block becomes mandatory.

Our approach to WLAN emulation is a two-stage scenario-driven process, as depicted in Figure 2. The architecture we propose here is based on the systems presented in [Bat-03] and [Per-04]. In the first stage, from a real-world scenario representation, we create a network quality degradation (ΔQ) description which corresponds to the real-world events. Subsequently this description is converted into an emulator configuration which is used during the effective emulation process to replicate the user scenario, allowing to study its effects on the application.
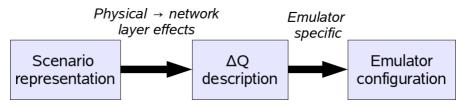

Figure 2: Two-stage scenario-driven approach to emulation.

Assume that the scenario representation describes how, from an initial condition, a mobile node moves with respect to an access point. As a consequence of the motion, the radio signal strength will change: this is the WLAN physical layer effect. In its turn this causes quality degradation at the data-link layer, as follows: the weaker signal first induces higher packet loss, then there is a delay as the WLAN adapters change the channel encoding and rate, and at the end the available bandwidth diminishes. Hence, the first stage of our approach will transform a sequence of real-world events at moments $t_0, t_1, ..., t_n$, into a sequence of ΔQ descriptors $\Delta Q_0, \Delta Q_1, ..., \Delta Q_n$, where ΔQ_i represents the state of the network (available bandwidth, packet loss rate, delay and jitter) at moment t_i, with $i = \overline{0, n}$.

In the second stage, the generic ΔQ description created previously is converted into a wired-network emulator configuration, which is specific to each emulator used. Initially the conversion target is the `dummynet` network emulator [Dum-**] running on a standalone FreeBSD PC. Alternatively a PC cluster, such as StarBED [Miy-05], can be used for more complex experiments, involving a large number of emulated wireless nodes. Note that by decoupling the two conceptually-independent stages we make it possible that later on other emulators can be used as well, emulators that are more accurate and have a richer set of features.

An important step that we will undertake in order to achieve a high degree of realism is to combine observations and traffic traces of real WLANs with analytical models of WLAN environments. As a result of this calibration, the ΔQ description that we produce from the scenario representation will accurately describe the observed network behaviour.

In summary, the emulation approach we propose transforms a user-meaningful real-world representation of a WLAN environment into a network quality degradation description, termed "ΔQ description". The ΔQ description is sufficient to subsequently configure a network emulator, and effectively reproduce an environment that corresponds accurately at network level to the emulated scenario. The next chapters of this document will focus on the details of the translation of real-world effects to the successive layers of the network communication protocol stack: physical layer, data link layer, and network layer.

3 Physical layer

The physical layer is the first layer in the ISO/OSI[3] reference model. It consists of two sub-layers, called PLCP (Physical Layer Convergence Procedure/Protocol), and PMD (Physical Medium Dependent), respectively.

PLCP is the sub-layer responsible for the Collision Sense (CS) aspect of the CSMA/CA protocol of the Data Link Layer (DLL). Other PLCP sub-layer functions are: communicating with the MAC layer through a PHY service access point, preparing MPDUs (MAC Protocol Data Units) for transmission, delivering incoming frames from the wireless medium to the MAC layer, and mapping MPDUs into a suitable frame format – the PDU (Physical Data Unit) – for transmission by the PMD.

The PMD sub-layer provides the actual transmission and reception of PHY entities between two stations through the wireless medium. This sub-layer is responsible for modulation and demodulation of frame transmission, as well as the encoding of the signal. The PLCP and PMD communicate using service primitives to manage the transmission and reception functions.

Understanding the properties of the physical layer is the first step in determining how changes in a real-world environment affect the quality degradation in a WLAN. The two issues of concern to us related to the physical layer of WLANs are:

- Wireless interface properties;
- Signal propagation.

3.1 Wireless interface properties

All wireless systems that are part of a WLAN, such as the network adapters of mobile nodes, or the access points, have wireless interfaces that make it possible to communicate using radio signals. In WLANs, the IEEE 802.11 physical layer specification is used. We shall focus here only on those elements that are important for WLAN emulation, please see [802.11] for more details about the standard.

The key properties of a wireless interface in the context of emulation are:

- Transmitted signal power;
- Receiver minimum input level sensitivity;
- Interference issues.

The combination of these properties with signal propagation and its effects determines how well communication takes place between wireless devices. Therefore all these elements affect communication robustness and need to be studied in detail.

---

3   International Standards Organization/Open Systems Interconnections.

3.1.1 Transmitted signal power

Absolute power is expressed in Watt [W]. In connection with WLANs, power is usually expressed as relative power measured in dBm, that is the power gain (or loss) relative to 1 milliwatt (1 mW):

$$P = 10 \log_{10} \frac{P\,[mW]}{1\,[mW]} \; [dBm].$$

(3.1)

The higher the transmitted signal power, the larger the range over which communication can be achieved. Given that wireless systems are used in close proximity to humans, the Effective Isotropic Radiated Power (EiRP) is limited by law. The legal limit for radiated power of WLAN adapters is generally set to 100 mW (equivalent to +20 dBm). The EiRP for current adapters is in the range 10dBm to 20 dBm. For access points the EiRP can be higher, especially if they are intended for outdoor use and they have high-gain antennas. For example the InPath Access Point 2480AA has a radio-conducted output power of 23 dB (200 mW), and a maximum antenna gain of 14.5 dBi[4], resulting in a maximum EiRP of 36 dBm (4 W). See Table 1 for a list of values of the typical transmitted signal power (EiRP) values and antenna gains for several adapters and access points (both indoor and outdoor) that are in use at present.

3.1.2 Receiver sensitivity

Receiver sensitivity represents the degree to which the reception system responds to incoming electromagnetic waves. The higher the sensitivity, the better the reception of weak signals is. The required receiver minimum input level sensitivity is defined by the IEEE 802.11 standard [802.11] as follows:

> The frame error ratio (FER) shall be less than 8×10^{-2} at an MPDU length of 1024 bytes for an input level of −80 dBm measured at the antenna connector. This FER shall be specified for 2 Mbit/s DQPSK modulation. The test for the minimum input level sensitivity shall be conducted with the energy detection threshold set ≤ −80 dBm.

In practice adapters usually have higher sensitivities than the required minimum, and in addition sensitivity levels differ between encodings (hence, between operating rates). For each WLAN adapter and access point its specific receive sensitivities for the various encodings are provided by the manufacturer. Table 1 summarizes these values for several manufacturers, for the encodings associated to the 802.11b WLAN standard. Receiver sensitivity is defined in practice as the received power level at which the frame error rate for 1024-byte frames reaches 8% [Int-00]. This is equivalent to a Bit-Error Rate (BER) of approximately 10^{-5}. Note that the fact that a frame is errored doesn't mean the associated data is also considered lost. A certain number of retransmissions will occur at the data link layer before the data is considered lost at layer 2. See Chapter 4 for a more detailed discussion of these aspects.

From Table 1 we see that sensitivity values of wireless transceivers are between −80 and −90 dBm at 11 Mb/s, but can even go down to −95 dBm for the lowest operating rate, 1 Mb/s. The lower the value for a certain operating rate, the more robust the receiver is.

---

4 Decibels isotropic, a relative gain measurement with respect to an isotropic radiator in free space. An isotropic radiator is a theoretical antenna with equal gain in all points on the isotropic sphere. The greater the dBi value for an antenna, the higher the gain and the area of coverage.

Generally the more recent (and more expensive) cards have better receiver sensitivities (e.g. Cisco Aironet 802.11a/b/g).

Type	Adapter/AP name	Tx power [dBm]	Antenna gain [dBi]	Rx sensitivity [dBm]			
				11 Mb/s	5.5 Mb/s	2 Mb/s	1 Mb/s
WLAN Adapter	D-Link DWL-G650	15±2	N.A.	−82	−85	−86	−89
	EDIMAX ED-7108PCg	16−18	N.A.	−85	N.A.	N.A.	−93
	EDIMAX EW-7608PG	18±2	N.A.	−90	N.A.	N.A.	−95
	3COM 3CRXJK10075	N.A.	N.A.	−86	−88	−91	−93
	Encore ENPWI-G	N.A.	N.A.	−85	N.A.	N.A.	N.A.
	Cisco Aironet 802.11a/b/g	10−20	1	−90	−92	−93	−94
	Cisco Aironet 350	20	N.A.	−85	−89	−91	−94
	ORiNOCO 11b	15	N.A.	−82	−87	−91	−94
Indoor AP	3COM 3CRWE725075A	17	N.A.	−83	−87	−89	−91
	D-Link DWL-7200AP	18	5	−83	−88	−89	−92
	NETGEAR ProSafe Dual Band WAG302	20	5	−89	−92	−92	−93
	SMC SMC2552W-G	16	2	−87	−90	−90	−93
	Intel PRO/Wireless 2011 LAN Access Point	18	N.A.	−81	−84	−85	−87
Outdoor AP	Cisco Aironet 1500 Lightweight Outdoor Mesh Access Point	24	N.A.	−88	−90	−94	−94
	D-Link AirPremier DWL-2700AP Wireless Outdoor Access Point	20	5	−83	−88	−89	−92
	InPath Outdoor Access Point 2480AA	23	7.5	−89	−91	−93	−95

Table 1: Typical transmitted power, antenna gains and receive sensitivity (measured at 8% FER) for several WLAN adapters and access points (indoor & outdoor), at 802.11b operating rates.

In order to take into account the receiver sensitivity threshold (denoted here by *T*) when modelling reception there are two possibilities:

- Use the threshold to determine whether reception is good or bad (e.g., frames are received correctly if the received power $P_r > T$, and are considered to be in error for $P_r < T$);

- Use the threshold value and the information about how it is computed to fit a model on the dependency between P_r and FER.

Given that the second approach is more accurate, and also more suited for the architecture we propose, we have taken the second approach. For this purpose we took into account the fact that, following the specification of the required receiver minimum input level

13

sensitivity, the receiver sensitivity is determined as that value of the received signal power at which the frame error rate reaches 0.08 when receiving 1024-byte frames [Int-00]. The model we propose is described by the equation below:

$$FER_1 = FER_T \cdot e^{\alpha(T - P_r)},$$
(3.2)

where:

- FER_1 is the frame error rate calculated using the Pr-threshold-based model;

- T is the receive sensitivity threshold for the current operating rate;

- FER_T is the frame error rate value when $P_r = T$ (FER_T equals 0.08 for 1024-byte frames);

- α is a constant to be determined by calibration, with the default a value of 1.

An equivalent equation is:

$$BER_1 = BER_T \cdot e^{\alpha(T - P_r)},$$
(3.3)

where:

- BER_1 is the bit error rate calculated using the Pr-threshold-based model;

- BER_T is the bit error rate value when $P_r = T$ (BER_T equals 10^{-5} for 1024-byte frames).

In Figure 3 we plot the model given by equation (3.2) for the values of T equal to the receive sensitivity thresholds of the ORiNOCO 11b adapter.
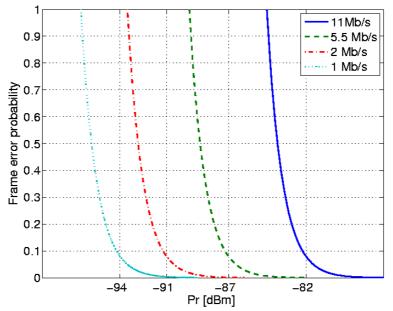

Figure 3: Frame error probability, FER_1, versus received power, P_r,
using an exponential-dependency model based on the P_r threshold and
the associated frame error rate value.

3.1.3 Interference issues

Another factor that influences error rates at the receiver is the interference with other electromagnetic signals – regarded as noise. This interference is measured by means of the Signal to Noise power Ratio (SNR). The interference (noise) in RF communication systems is of two main types: thermal noise, and industrial noise. Industrial noise is due to microwave ovens or cellular/cordless phones, interfering noise of other WLAN systems using the same frequency band, including Bluetooth, etc. SNR is defined as follows:

$$SNR = 10 \log_{10} \frac{S[W]}{N[W]} \ [dB],$$ (3.4)

where:

- S is the (useful) signal power (expressed in Watt);
- N is the noise power (expressed in Watt).

If the signal is more powerful than the noise, SNR will be positive. If the signal is buried in noise, the ratio will be negative. Larger values of SNR are better, and each system has specific inferior limits on SNR that must not be exceeded for operation at a certain data rate. In Table 2 we show the minimum accepted SNR values for each operating rate of the ORiNOCO 11b adapter [Pro-04].

Operating rate [Mb/s]	11	5.5	2	1
Minimum SNR [dB]	16	11	7	4

Table 2: Minimum SNR values for each operating rate of the ORiNOCO 11b adapter.

In cases when the noise level is very low, the WLAN system performance will be limited more by the receiver sensitivity than by the SNR. However if the noise level is high, then it will be the SNR that will count in achieving a given data rate. In such a case the received power must be increased, either by using a higher transmission power or by using antennas. Note that omnidirectional antennas will also amplify the noise, but directional antennas can be used to achieve the desired effect.

In normal conditions, without any other WLAN on the same operating frequency, and without industrial noise, the environmental noise level will be around −100 dBm. In order to sustain an 11 Mb/s operating rate with an ORiNOCO 802.11b card, for example, we would need a received power that is 16 dB higher (given the minimum acceptable SNR for this card at 11 Mb/s operating rate, see Table 2). Therefore the minimum received power, from the point of view of interference, must be at least −100 + 16 = −84 dBm. However since the minimum receiver sensitivity level for this card is at −82 dBm (see Table 1), in this case the minimum receiver sensitivity is the limiting factor for the system.

Although the discussion above is correct, it is not of a very accurate nature, since it doesn't take into account bit-error-rate variation in the vicinity of thresholds, which does occur in reality. A more comprehensive view on the relationship between SNR and BER for the operating rates of 802.11b is given in Table 3 below, based on the specifications of the Intersil HFA3861B Direct Sequence Spread Spectrum Baseband Processor [Int-02]. As mentioned before, a BER of 10^{-5} corresponds to a FER of approximately $8 \cdot 10^{-2}$ for 1024-byte frames.

	Operating rate			
SNR [dB]	**1 Mb/s**	**2 Mb/s**	**5.5 Mb/s**	**11 Mb/s**
5	$5 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$
6	$5 \cdot 10^{-2}$	$6 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$6 \cdot 10^{-3}$
7	$1.2 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$	$4.1 \cdot 10^{-3}$	$2 \cdot 10^{-3}$
8	$4.1 \cdot 10^{-3}$	$6 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	$7 \cdot 10^{-4}$
9	$1.1 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$	$3.3 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$
10	$2.2 \cdot 10^{-4}$	$4 \cdot 10^{-4}$	$8 \cdot 10^{-5}$	$8 \cdot 10^{-5}$
11	$4 \cdot 10^{-5}$	$6.3 \cdot 10^{-5}$	$1.5 \cdot 10^{-5}$	$2.7 \cdot 10^{-5}$
12	$2.9 \cdot 10^{-6}$	$8.9 \cdot 10^{-6}$	$2.7 \cdot 10^{-6}$	$8 \cdot 10^{-6}$
13	$3.6 \cdot 10^{-7}$	$1.3 \cdot 10^{-6}$	$5 \cdot 10^{-7}$	$1.9 \cdot 10^{-6}$
14	$4 \cdot 10^{-8}$	$2.7 \cdot 10^{-7}$	$5 \cdot 10{-8}$	$3.9 \cdot 10^{-7}$
15	$3 \cdot 10^{-9}$	$4 \cdot 10^{-8}$	10^{-8}	$1.02 \cdot 10^{-7}$
16	$1.8 \cdot 10^{-10}$	$4 \cdot 10^{-9}$	$1.1 \cdot 10^{-9}$	$3 \cdot 10^{-8}$
17	$1.8 \cdot 10^{-10}$	$4 \cdot 10^{-9}$	$1.1 \cdot 10^{-9}$	$4 \cdot 10^{-9}$

Table 3: BER versus SNR for 802.11b operating rates (Intersil HFA3861B).

Table 3 shows the measured dependency between BER and SNR. In order to be able to easily compute the BER from SNR, we decided to fit an analytical equation on the measured dependency. The following equations show the fit we obtained using the Excel trend-line tool for all rates of 802.11b. More advanced fitting mechanisms can be used in the future, and these equations are given just for illustration purposes. Another possibility is to use local linear interpolation by taking into account the known values to the left and right of the point of interest.

$$BER_{11\,Mb/s} = a_{11\,Mb/s} \cdot e^{b_{11Mb/s} \cdot SNR} = 12.44204 \cdot e^{-1.234009 \cdot SNR} \; ;$$
$$BER_{5.5\,Mb/s} = a_{5.5\,Mb/s} \cdot e^{b_{5.5Mb/s} \cdot SNR} = 243.0763 \cdot e^{-1.562894 \cdot SNR} \; ;$$
$$BER_{2\,Mb/s} = a_{2\,Mb/s} \cdot e^{b_{2Mb/s} \cdot SNR} = 787.4195 \cdot e^{-1.548256 \cdot SNR} \; ;$$
$$BER_{1\,Mb/s} = a_{1\,Mb/s} \cdot e^{b_{1Mb/s} \cdot SNR} = 4255.180 \cdot e^{-1.811341 \cdot SNR} .$$

(3.5)

The bit error rate computed using the SNR-based model will be denoted in this document by BER_2 and it can be used to compute the corresponding frame error rate, FER_2, by taking into account frame size, as it will be detailed in the next chapter.

Besides using measured dependencies, as indicated above, it is possible to determine the theoretical relationship between BER and SNR as well. We provide here only a few elements about this approach, since we prefer using the measured dependency for real equipment in order to increase the realism of our model.

The noise in WLAN environments is usually modelled as additive white Gaussian noise (AWGN). In this case one can use the Shannon capacity formula for an AWGN channel [Kum-04] to determine that maximum number of bits per symbol that can be used under given channel conditions. This quantity is denoted by C, and is also called *channel capacity*:

$$C = \frac{1}{2} \log_2 \left(1 + \frac{P_r}{2WN_0} \right) \ [bits/symbol] ,$$ (3.6)

where:

- W is the signal bandwidth;
- N_0 is the power of the AWGN process (expressed in W/Hz).

Note that $\dfrac{P_r}{2WN_0}$ in formula (3.6) is the received SNR.

A related result of even more importance for this topic is the probability of error of a binary modulation scheme in the presence of AWGN. Note that this is equal to the BER of such a scheme, since each symbol encodes only one bit:

$$P_{error-AWGN} = Q \left(\sqrt{\frac{2E_s}{N_0}} \right) ,$$ (3.7)

where:

- Q is the following function, $Q(\tau) = \int\limits_{\tau}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} \ dx$;

- E_s is the energy of the signal (expressed in W/Hz).

In Figure 4 we show the dependency between BER and P_r. For this plot we fixed the the noise power, N, at different values, and varied P_r. The resulting SNR was used to compute the BER by using the SNR-based model given by equation (3.5). On the same graph we plotted the P_r-threshold-based model described by equation (3.3). Note that the plot is drawn on a logarithmic scale.

From Figure 4 we can observe that for low noise levels (the noise power, N, less than approximately −94.5 dBm) the SNR-based model has an ignorable effect, and the P_r-threshold-based model dominates the source of errors. However for noise levels larger than -94.5 dBm, the influence of the noise starts becoming more significant, and the P_r-threshold-based model doesn't provide anymore accurate results by itself, since it doesn't take noise into account.

The P_r-threshold-based model refers to how well a signal can be decoded, whereas the SNR-based model refers to how well the signal can be differentiated from noise. Depending on the noise level one of the causes of errors may be more significant, however since they are independent we believe that the global model should take into account both causes simultaneously. See next chapter for more details regarding the modelling of this aspect to compute the frame error rate.
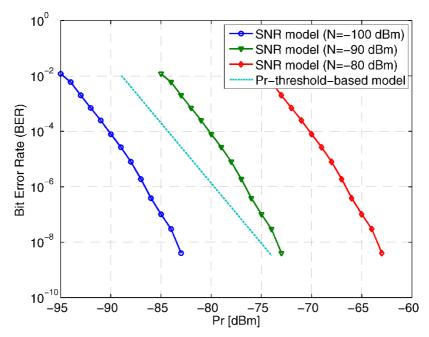
*Figure 4: Bit error rate, BER, versus received power, Pr, according to the SNR model
and several values of the noise power, N, as well as according to the
Pr-threshold-based model for the 11 Mb/s operating rate.*

3.2 Signal propagation

To arrive from transmitter to receiver, signals propagate through the surrounding environment. The side effect is attenuation, that is the loss of signal strength due to RF propagation. In general, the further a receiver is from the transmitter, the weaker the signal. Obstacles, such as walls or objects in the case of indoor operation, or buildings in the case of outdoor operation, cause additional attenuation by blocking or weakening radio signals. Floors, walls and ceilings (depending on what they are made of) can either strengthen or weaken WLAN signals [DIS-05].

Therefore, in addition to transmission and reception, studying wireless communication requires understanding the propagation of the transmitted signal until it reaches the receiver. There are two aspects concerning signal propagation that need to be investigated:

- Propagation in ideal conditions;
- Propagation in real environments.

3.2.1 Ideal conditions

To model the propagation in ideal conditions, the Friis transmission equation is used:

$$\frac{P_r}{P_t} = \frac{G_t G_r}{L} \left(\frac{\lambda}{4 \pi R} \right)^2 , \tag{3.8}$$

where:

- P_t and P_r are the power of the transmitted and received signals, respectively;

- G_t and G_r are the gains of the transmitting and receiving antennas, respectively;
- L is the system loss factor;
- λ is the signal wavelength;
- R is the distance between transmitter and receiver.

Considering $G_t = G_r = 1$, equivalent to 0 dBi antenna gain, and $L = 1$ (i.e., both no antenna gain and no system loss), we obtain the following simplified formula for the power loss, P_{loss}, for propagation in ideal conditions (free space), expressed in dB:

$$P_{loss} = 10 \cdot \log_{10} P_t - 10 \cdot \log_{10} P_r = 20 \cdot \log_{10} \frac{4 \pi R}{\lambda} \ [dB]. \tag{3.9}$$

Formula (3.9) can be further simplified by considering the particular case of 802.11b/g networks that transmit in the range 2.41−2.48 GHz (which is the same as that of Bluetooth, and Zigbee networks, defined by the IEEE 802.15 Working Group). Thus for an average frequency f of 2.45 GHz, we have the wavelength, λ, given by:

$$\lambda = \frac{c}{f} = \frac{3 \cdot 10^8 [m/s]}{2.45 \cdot 10^9 [Hz]} = 0.1224 \ [m], \tag{3.10}$$

where c is the signal propagation velocity. Using numerical values, equation (3.9) becomes:

$$P_{loss} = 20 \cdot \log_{10} 102.67 R \ [dB]. \tag{3.11}$$

As an example, considering $R = 1$ m in equation (3.11), we obtain $P_{loss} = 40.23$ dB. Using this calculation, if powers are expressed in dBm, then for an adapter transmitting at $P_t = 20$ dBm, the received power at a distance of 1 m will be of approximately $P_r = -20$ dBm. The received power at a distance of 1 meter in free space is used as a reference power in the log-distance path loss model of propagation in real environments (see next section).

3.2.2 Real environments

In real environments the communication channel is not only contaminated by additive white Gaussian noise, and attenuated through propagation in free space. In practical situations signals undergo additional attenuation, for example due to obstructions and reflections from the surrounding objects, and delay. Therefore Friis's equation cannot be used in real environments.

One of the specific propagation effects that occur in real environments is related to delay [Kum-04]. Radio waves are absorbed by the media they traverse, and scattered by the objects they impinge on. Unless a very narrow antenna beam is used, the received waves will arrive along several paths. Therefore the received signal is the sum of attenuated and delayed versions of the original signal. Superposition of delayed signals from various paths can cause a symbol from one path to overlap a neighbouring symbol from another path. The delay spread represents the time difference between the arrival of the same symbol on different paths, and the maximum acceptable value is part of the specifications of each WLAN adapter. Table 4 shows the delay spread values for the Cisco Aironet 802.11a/b/g adapter:

Operating rate [Mb/s]	11	5.5	2	1
Delay spread [ns]	130	200	300	350

Table 4: Maximum acceptable delay spread for Cisco Aironet 802.11a/b/g adapter (802.11b rates).

The other aspect that needs to be taken into account related to signal propagation in practical situations is signal attenuation. Table 5 shows examples of attenuation values of common office construction [DIS-05]:

Office construction	Attenuation [dB]
Plasterboard wall	3
Glass wall with metal frame	6
Cinder block wall	4
Office window	3
Metal door	6
Metal door in brick wall	12.4

Table 5: Typical attenuation values of common office construction (source: [DIS-05]).

In order to study in more detail signal attenuation, it is possible to identify the following three components:

- Path loss;
- Shadowing;
- Multi-path fading.

Path loss refers to signal strength loss due to propagation through an environment other than free space. However if attenuation is measured at various points at distance *d* from the transmitter, the attenuation values will have a certain randomness, due to the variations in the terrain and the media through which the signal may have passed. This is the *shadowing* component of the attenuation (also known under the names of *slow fading* or *large-scale fading*). *Multi-path fading* is the third component of attenuation which occurs because of local node mobility, and it is also known as *fast fading*, or *small-scale fading*. The strength of this effect depends on the carrier frequency, the speed of movement and the speed of light.

A widely used model for signal attenuation in real environments is the "log-distance path loss model" [Rap-02], which takes into account path loss and shadowing. The model name derives from the logarithmic dependency of P_r on the distance between receiver and transmitter. According to this model, the received power $P_r(d)$ (in dBm) at a distance d (in meters) from the transmitter is given by:

$$P_r(d) = P_{r0} - 10 \cdot \alpha \cdot \log_{10}(d) + X_\sigma,$$
(3.12)

where:

- P_{r0} is the reference signal strength at 1 meter in free space from the transmitter (in dBm);
- α is the "path-loss exponent";
- X_σ is a Gaussian random variable with mean 0 and standard deviation of σ dB.

The log-distance path-loss model takes into account the shadowing effect due to different obstacles present in multiple transmitter-receiver paths with the same separation. Within this model this phenomenon is referred to as "log-normal shadowing", since it is modelled by the Gaussian process X_σ. Considering the case of $\sigma = 0$ dB (i.e., with no shadowing), the influence of α on P_r is shown in Figure 5 below.

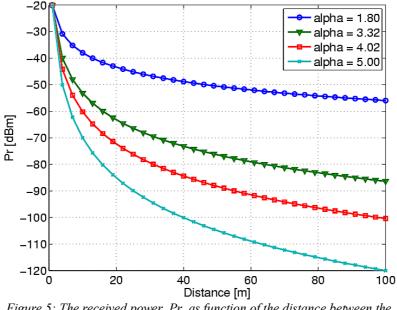*Figure 5: The received power, Pr, as function of the distance between the transmitter and the sender for various values of α ($\sigma = 0$ dB).*

The values of the parameters (α, σ) are heavily dependent on the environment. Empirical values for α lie in the range 1.8 (lightly obstructed environments with corridors) to 5 (multi-floored buildings). It is considered that α is in the range [3,4] for office environments [Far-05], while ITU-T recommends a value of 3 for modelling 802.11b networks, and 3.1 for 802.11a [Yee-02]. Values for σ usually fall in the interval [4, 12] dB [Rap-02].

In conclusion, out of the three components of signal attenuation, the log-distance path-loss model aggregates the path-loss attenuation induced by obstacles (walls, doors, windows, etc.) into the parameter α, while σ is used to take into account the shadowing component. The assumption of very slow motion allows ignoring the multi-path fading component of attenuation.

Formula (3.12) above is mainly intended for use in indoor environments. For outdoor environments the additional parameter W is used (external wall attenuation in dB), as follows:

$$P_r(d) = P_{r0} - 10 \cdot \alpha \cdot \log_{10}(d) - W + X_\sigma,$$
(3.13)

[Far-05] reports obtaining the following results by fitting the log-distance path-loss model on experiment data obtained in a real-world environment:

- $\alpha = 4.02$ and $\sigma = 7.36$ dB for an indoor environment;
- $\alpha = 3.32$ and $\sigma = 3.1$ dB for an outdoor environment, with W for one wall being 4.8 dB.

Note that both α and σ are smaller in outdoor environments due to the more significant

presence of free space. In Figure 6 one can see the variation of P_r with distance in the conditions specified above. The graphic for outdoor conditions starts from a lower received power value because of the wall attenuation. However the decrease of P_r with distance is slower than in the case of indoor environments.

Figure 6: The received power, P_r, as function of the distance between the transmitter and the sender for indoor and outdoor environments (using parameter values from [Far-04]).

In the previous discussion we have ignored the effects of local node mobility on signal reception. However when modelling nodes that move faster than walking speed (approximately 0.5 m/s), this effect must be taken into account. Multi-path fading models with Rayleigh or Ricean distributions are commonly used to describe the WLAN environments that include mobile nodes (sometimes called MANET, Mobile Ad Hoc Networks). The fading models using the Rayleigh distribution are suited for highly mobile conditions with no line-of-sight between nodes, while the Ricean distribution accounts for the case of line-of-sight path between nodes. Raiyleigh distribution is a particular case of the Ricean one with the "Ricean K factor" equal to 0 [Pun-00].

3.3 Operation range

Operation range can be defined as the maximum distance at which the equipment can operate with the desired performance (for example, concerning data rate, packet loss levels, delay bounds). The range over which a wireless equipment can operate is an important element for WLAN planning. By taking into account the radio properties of the wireless communication system, as well as the signal propagation characteristics of the environment, one can determine the operation range of the equipment.

Although both the AP and the wireless client adapter card (wireless Network Interface Card, NIC) are transceivers (they both receive and transmit), the location of the access point affects the operating range more that that of the NIC. One reason is that APs continuously send beacons to wireless nodes, that must be properly received in order to maintain connectivity. In addition, an AP must be "hearable" by multiple wireless nodes, whereas each node is only required to "hear" the AP.

4 Data link layer

The Data Link Layer (DLL) consists of two sub-layers: Logical Link Control (LLC) and Medium Access Control (MAC). 802.11 uses the same LLC specified by the IEEE 802.2 Ethernet standard, and the same 48-bit addressing as other 802 LANs, allowing for very simple bridging from wireless to IEEE wired networks.

However the MAC sub-layer in 802.11 is unique to WLANs. The main function of the MAC protocol is to regulate the usage of the medium through a channel access mechanism. This mechanism is a way to divide the main resource between nodes, i.e., regulate the use of the radio channel – which is the shared wireless medium. The channel access mechanism is the core of the MAC protocol. In addition to medium access, this layer is also responsible for calculating the CRC checksum, encryption, packet fragmentation (when transmitting), reassembling fragments (when receiving), and rendering these processes transparent to higher level protocols. MAC layer management is also responsible for synchronization, power management, roaming, etc.

In order to determine the effects the physical layer has on the data link layer, it is important to understand the interaction between signal strength – the physical layer element –, and operating rate, frame error rate, delay, etc. – the data link layer parameters. Therefore we shall look next in more detail at the 802.11 MAC protocol, which is specific to the 802.11 data link layer.

4.1 Basic 802.11 MAC

This section describes the common Media Access Control layer used by the 802.11 family of standards. There are 3 main classes of channel access mechanisms for radio access:

- TDMA (Time Division Multiple Access);
- CSMA/CA[5]-based access, termed Distributed Coordination Function (DCF);
- polling-based access, termed Point Coordination Function (PCF).

The CSMA/CA protocol is by far the most used in 802.11 networks, therefore we shall investigate it in more detail. CSMA/CA is related to the CSMA/CD[6] protocol used in 802.2 Ethernet. In CSMA/CA, a station that has something to send listens to the medium, and when there is no station active (and that is the difference with respect to CSMA/CD), waits a random time before sending its data, while still monitoring the medium. This way the chance that two stations will send at the same time is small. The loss in speed due to the random waiting time is compensated by the lower number of retransmissions. In principle, the more stations there are in a network, the higher the benefit of this technique.

CSMA/CA is, like all Ethernet protocols, peer-to-peer, hence there is no requirement for a master station. Below we detail the steps a wireless node that wants to transmit must perform:

1. Listen on the desired channel;
2. If the channel is idle (i.e., there are no active transmitters) send the packet;
3. If the channel is busy (i.e., there is an active transmitter), wait until transmission

---

5 Carrier Sense Multiple Access with Collision Avoidance.
6 Carrier Sense Multiple Access with Collision Detection.

stops, then a further contention period. The contention period is a random waiting period after every transmit operation, and statistically allows nodes to have equal access to the media[7].

4. If the channel is still idle at the end of the contention period, transmit the packet; if the channel became busy during the waiting, the waiting is stopped and the process is repeated from step 3 above until a free channel is present. Note that each time this happens, only the remaining contention period is taken into account while waiting. In the event that the waiting time becomes 0 without having transmitted the packet, a new random contention period is generated at step 3 with an exponentially increasing range[8].

The contention period is however not the only waiting period in 802.11 MAC operation. In between the end of a frame and the contention period there is an additional time called Inter Frame Space (IFS). The minimum IFS value is called Short IFS (SIFS), equal to 10 µs; this short waiting time can only used for high-priority frames, such as control and management frames. Data frames should use either the Distributed coordinated function IFS (DIFS), equal to SIFS + 2 slots = 50 µs, in the case of DCF operation, or the Point coordinated function IFS (PIFS), equal to SIFS + 1 slot = 30 µs.

The CSMA/CA mechanism described above has as side effect the introduction of variable jitter in packet transmission. This must be taken into account when modelling MAC operation, especially since it can become important in high-load situations.

In order to improve communication efficiency, additional features are employed in the 802.11 MAC:

- Positive-acknowledgement scheme;

- MAC-level retransmission;

- Frame fragmentation.

4.1.1 Positive-acknowledgement scheme

An error recovery mechanism is required to compensate for the errors that occur during wireless communication. The two main classes of error recovery mechanisms are:

- Automatic Repeat reQuest (ARQ);

- Forward Error Correction (FEC).

The current MAC mechanisms uses an ARQ-type scheme, since it is considered more efficient and simpler than FEC [Ni-04]. The particular version of ARQ used is "Stop and Wait Automatic Repeat reQuest" (SW-ARQ). In this scheme, at the end of every packet, the receiver – if it has successfully received the packet – will return an acknowledgement (ACK) packet. If a packet is received with errors (or not received at all), the receiver will not respond, i.e., in SW-ARQ there is no negative ACK (NACK). The transmit window allows for ACK transmission through the use of IFS, i.e., the contention period starts after the ACK should have been sent.

While waiting for the ACK from the receiver, the sender will not transmit any packet. As

---

7   For busy/idle detection purposes, the contention time is divided in slots of 20 µs.
8   The initial range of the contention period is [0, 31] slots. This range increases as more attempts to transmit are made, as follows: [0, 63] for the second attempt, [0, 127] for the third attempt, [0, 255] for the forth attempt, [0, 511] for the fifth attempt, and [0, 1023] from the sixth attempt on.

soon as the sender correctly receives the ACK, it will start the CSMA/CA procedure in order to transmit the following frame. However if there is no response after a time-out, the sender will retransmit the frame and wait for another ACK. Given the above description, the ACK time-out must be at least equal to the SIFS time, plus ACK transmission time. The retransmission mechanism itself is described in the next subsection. Note that the contention window increases between retransmissions, according to the rule presented in the previous section.

The disadvantage of SW-ARQ is that a lot of idle time is wasted for waiting the sequenced ACKs. Some other mechanisms are proposed to solve this problem, such as Selective Repeat ARQ (SR-ARQ) and Go-Back-N ARQ (GBN-ARQ). These schemes tried to improve MAC layer performance by removing the waiting time for ACKs. The next frame could be sent even if ACKs of former frames are not received or a time out event occurs. Nevertheless they are not used in practice because of their complexity.

4.1.2 MAC-level retransmission

As detailed in the previous section, the 802.11 MAC employs an ACK-based error correction mechanism. If no ACK is received during the time-out period, the sender will retry the transmit (using the normal CSMA/CA procedure) until either transmission is successful (i.e., confirmed by a successfully received ACK), or the operation is abandoned after several retries. The total number of transmissions can be either 4 or 7, depending whether the RTS/CTS protocol is enabled or not[9]. A packet is considered lost only if the maximum number of retransmissions is reached without getting an ACK from the receiver end.

Note that even though through the use of this mechanism frame loss may be hidden from higher layers, and there is no packet loss at network layer, the retransmission scheme introduces additional delay and jitter for packets. Frame errors and retransmissions also interact with a rate adaptation mechanism such as ARF (see section 4.2.2).

4.1.3 Frame fragmentation

Bit error rates on wireless systems (normally around 10^{-5}–10^{-6}) are substantially higher than for wire-line systems (10^{-12}). Large blocks may approach the number of bits where the probability of an error occurring is close to 1, which means that every sent block could fail, including the retransmissions. To reduce the possibility of this happening large blocks may be fragmented by the transmitter and reassembled by the receiver node, for example, a 1500-byte block (12,000 bits) may be fragmented into 5 blocks of 300 bytes (2,400 bits).

Fragmentation is in practice controlled by the fragmentation threshold. Frames larger than the fragmentation threshold will be fragmented. Threshold values are usually in the range 256 to 2048 bytes. Setting the threshold to the maximum frame value in 802.11 LANs, 2048 bytes, effectively disables fragmentation. Note that fragmentation only takes place for unicast frames, as for efficiency reasons 802.11 doesn't fragment multicast or broadcast frames.

While there is some overhead in doing this, both the probability of an error occurring is reduced and, in the event of an error, the re-transmission time is also reduced, since only

---

9   RTS/CTS stands for Request To Send/Clear To Send, and is an optional feature of 802.11 addressing the hidden node problem (see section 4.3 for details).

the errored fragment needs to be retransmitted. It is generally considered that fragmentation becomes efficient for collision rates higher than 5% [Gei-02]. However if collision occurs because of the existence of hidden nodes, using RTS/CTS is a better alternative (see section 4.2.1).

4.2 Rate adaptation

Current wireless network systems are able to communicate at different rates, lower rates being more robust than higher ones. Since channel conditions in WLANs vary, the rate adaptation mechanism was introduced, which represents the automatic switching between operating rates to match channel conditions.

The Auto Rate Fallback (ARF) [Kam-97] scheme is the most widely spread mechanism mechanism used to adapt the WLAN operating rate to channel conditions. Channel quality estimation based on signal strength is a proactive approach, and not always accurate because of fast changes in the wireless environment. ARF on the other hand is a simple reactive channel quality sensing, which gauges the changing of channel conditions based on the success or failure of previous frame transmissions.

The ARF mechanism is a timer-driven mechanism that keeps track of missed acknowledgements (ACK). When an ACK is missed for the first time after an earlier successful transmission, the first retransmission is carried out at the same data rate. After a second failure, the transmission rate is decreased to the next lower data rate, and a timer is started to track received and missed ACKs[10]. The data rate will be increased to the next higher data rate if the timer expires, or if 10 consecutive frames are transmitted successfully (i.e., 10 ACKs are successfully received). When the rate is increased, the first packet transmission (termed *probing transmission*) must succeed (i.e., the corresponding ACK must be received). Otherwise the rate is immediately decreased to the next lower data rate, and the timer is restarted, rather than trying the higher rate a second time. There are several studies of ARF performance, such as [Lac-04], that identifies two main problems with this scheme:

- Under rapidly changing channel conditions, ARF cannot adapt fast enough to sub-packet optimal rates given the large number of packets required for adaptation (10 to increase the rate, and 2 to decrease it);

- Under non-varying or slowly-varying conditions, ARF still tries to use a higher rate every 10 successfully transmitted packets. This leads to an increase in the number of retransmission attempts, and consequently a decrease in application throughput.

Another rate adaptation technique is Receiver Based Auto Rate (RBAR) [Hol-01]. However RBAR is not employed in practice since it redefines the standard use of the RTS/CTS control frames.

A simple model that we propose for ARF is the following probabilistic model. We compare the probability to lose two consecutive frames with a threshold, T_{down}, in order to decide whether rate decrease should take place. For rate increase we compare the probability to successfully send 10 consecutive frames with another specific threshold, T_{up}. In order to take into account the probing transmission we also compare the frame

---

10 Although information about the initial value of this timer was not available at the time of writing this document, it is reasonable to assume this value is of the same order of magnitude with the time necessary to successfully transmit 10 consecutive frames.

error rate at the higher rate, FER_{higher}, with the threshold T_{keep}. This model is summarized by the following equations:

$$FER^2 > T_{down} \quad \Rightarrow \quad rate \ decrease;$$
$$(1 - FER)^{10} > T_{up} \ \wedge \ FER_{higher} < T_{keep} \quad \Rightarrow \quad rate \ increase. \tag{4.1}$$

We propose to use initially $T_{down} = T_{up} = T_{keep} = 0.5$, the usual value for taking decisions based on event probability. Values for thresholds that lead to more accurate results compared to reality may be determined through calibration. Note that this simple model does take into account both the timer-driven and the frame-driven aspects of ARF, since we use probability thresholds and not real frames to decide on rate changes. However it doesn't account by itself for the time penalty incurred by the probing transmission if the rate increase is aborted.

4.3 Other 802.11 MAC issues

There are several other problems that the 802.11 MAC addresses, and they will be described next. Note however that they are presented only for informative purposes, and are not take into account by our models.

The *hidden node* problem occurs in a point to multi-point network, and it is defined as being a case in which three (or more) nodes are present. Consider a scenario where nodes A, B and C are located as depicted in Figure 7.
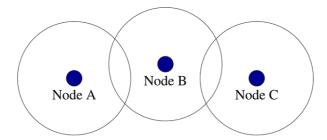

Figure 7: Example of the hidden node problem in a WLAN.

In such a case it is possible for node B to "hear" node A (and vice versa), and for node B to "hear" node C (and vice versa), but node C cannot hear node A. In a CSMA/CA environment, nodes A and C would both properly transmit: since they cannot hear each other during the "listen" phase, nodes A and C can both simultaneously and correctly transmit a packet. However node B would always get corrupted data due to the collisions between the packets transmitted by nodes A and C. In such a case nodes A and C are said to be *hidden* from each other, i.e., nodes that are not within each other's range, but communicate with the same access point or another peer node.

The hidden node problem is solved by the use of a RTS/CTS (Request To Send/Clear To Send) protocol prior to packet transmission. The RTS/CTS handshaking provides positive control over the access to the shared media, and reduces the collision probability between hidden nodes. In the three-node network example above, node A sends an RTS packet, asking for permission to transmit, which is heard by node B. Node B in its turn sends a CTS packet to node A, giving it permission to transmit. Since the CTS packet is heard both by node A and C, node C will not transmit data in this case. A similar RTS/CTS handshake must be performed by node C when it wants to transmit.

Note that in practice RTS/CTS is not enabled by default for all frame sizes because of the overhead incurred by its use. The key parameter related to RTS/CTS mechanisms is the RTS/CTS threshold, i.e., the frame size over which the RTS/CTS protocol will be used by the node.

A related issue to the hidden node problem is the *exposed node* problem, which affects nodes that are prevented to transmit because they sense other transmissions, even though the positioning makes it so that their transmission would have been possible without negative effects.

4.4 FER calculation

As mentioned in chapter 3, we use two models to compute the bit/frame error probability. One uses an exponential dependency with parameters indicating the received power sensitivity threshold, and is termed P_r-threshold-based model. This model can be used to compute the frame error rate, denoted by FER_1, using formula (3.2). The other model, termed SNR-based model, uses the dependency between SNR and BER to compute the bit error rate denoted by BER_2, as given by equation (3.5). In its turn BER_2 can be used to compute the corresponding frame error rate, FER_2, as it will be discussed below. To combine the predictions of the two independent models we propose to use the following equation[11] to compute the frame error rate, FER:

$$FER = FER_1 + FER_{2,} \qquad\qquad (4.2)$$

where:

- FER_1 is the frame error rate calculated using the P_r-threshold-based model;

- FER_2 is the frame error rate calculated from the bit error rate obtained using the SNR-based model.

The probability FER_2 must be determined from BER_2. One can use the results in [Pra-03] for this purpose (in what follows we kept the notation in that paper). With the assumption that ACK frames (14 bytes payload) have a very low error probability compared to data frames, the probability of error for a data frame, P_{e-data}^m, is given by:

$$P_{e-data}^m(L) = 1 - (1 - P_e^1(24)) \cdot (1 - P_e^m(28+L)), \qquad\qquad (4.3)$$

where:

- m denotes the operating rate, m = 1,2,3, and 4 for rates 1, 2, 5.5, and 11 Mb/s, respectively;

- L is the length of the frame payload;

- $P_e^1(24)$ is the probability of error of the PLCP preamble (18 bytes) and header (6 bytes), which is transmitted at 1 Mb;

- $P_e^m(28+L)$ is the probability of error of the MPDU, including the overhead of the MAC header (24 bytes) and the Frame Check Sum (4 bytes).

The error probability for a sequence of x bytes, $P_e^m(x)$, that is used in the above equation can be expressed as function of the BER for the corresponding operating rate, m, by the following formula:

---

11 Note that care must be taken in the implementation of this equation so that the result doesn't exceed 1.

$$P_e^m(x) = 1 - (1 - P_b^m)^{8x},$$

(4.4)

where P_b^m is the BER for the operating rate m.

For large enough frames we can also ignore the headers, especially since they are sent at lower rates, for which the error probability should also be lower. As a consequence, using our notation we can express FER_2 as:

$$FER_2 = 1 - (1 - BER_2)^{8 \cdot P_{Frame}},$$

(4.5)

where P_{Frame} is the size of the frame payload in bits.

4.5 Delay calculation

Delay is another parameter that we must calculate, in addition to frame error rate. At the data link layer we are concerned with that component of the total delay that is a function of the frame size, as well as the properties of the 802.11 MAC protocol. This is the delay component that we will model in detail in this section, and we shall denote this variable delay by $D_{variable}$.

4.5.1 Variable delay

In what follows we assume that the WLAN operates in DCF mode; other operation modes may be taken into account in the future. Considering the frame error rate, FER, defined in the previous section, one can compute the variable delay as the weighted average of the delays, D_i, of a frame that was retransmitted a number of i times, with $i = \overline{0, r}$. Accoridng to [802.11], the defualt maximum number of retransmissions is $r = 3$ if RTS/CTS is enabled, and $r = 6$ if RTS/CTS is disabled. The weights in the weighted average are the probabilities to have a successfully received frame after 0, 1, ..., and r retransmissions, namely, (1-FER), FER(1-FER), ..., and FER^r(1-FER), respectively. Using these weights the variable delay can be written as:

$$D_{variable} = \frac{\sum_{i=0}^{r} (1 - FER) FER^i D_i}{\sum_{i=0}^{r} (1 - FER) FER^i}, \quad FER \neq 0 \wedge FER \neq 1.$$

(4.6)

Equation (4.6) can be further simplified to obtain:

$$D_{variable} = \frac{1 - FER}{1 - FER^{r+1}} \sum_{i=0}^{r} FER^i D_i, \quad FER \neq 0 \wedge FER \neq 1.$$

(4.7)

In the particular cases when $FER = 0$ or $FER = 1$, the delay is given by:

$$D_{variable}|_{FER=0} = D_0; \quad D_{variable}|_{FER=1} = \infty.$$

(4.8)

I order to determine the variable delay one has to compute first the transmission time for a frame, $F_{transmission}$, which includes the headers and depends on the current operating rate, R [bits/s].

$$F_{transmission} = (H_{PHY} + H_{MAC} + P_{Frame})/R,$$

(4.9)

where:

- P_{Frame} is the size of the frame payload in bits;

- H_{PHY} is the size of the physical layer header (192 bits);

- H_{MAC} is the size of the MAC header (224 bits);

Given the functioning way of the 802.11 MAC protocol, it is necessary to include the acknowledgment reception time in the delay of one frame, as well as the contention period. Hence, the delay for a frame that was retransmitted 0 times (i.e., the delay of the initial transmission of a frame), D_0, can be computed using the following formula:

$$D_0 = SIFS + ACK_{transmission} + DIFS + T_{backoff\,0} + F_{transmission} + T_{propagation}, \qquad (4.10)$$

where:

- $SIFS$, the Short Inter Frame Space of 802.11 MAC, is equal to 10 μs;

- $ACK_{transmission}$, the time to transmit an ACK frame, is equal to 304 μs, computed using formula (4.9), given that an ACK frame has a payload of 112 bits, no MAC header and is transmitted at $R = 1$ Mb/s;

- $DIFS$, the DCF Inter Frame Space of 802.11 MAC, is equal to $SIFS + 2 \cdot D_{slot}$, where D_{slot} is the duration of a slot (20 μs);

- $T_{backoff\,0}$ is the waiting time due to the contention-window mechanism for a first transmission (retransmission index = 0);

- $T_{propagation} = 1$ μs, considering a maximum open space range of 300 m and the propagation speed of 300.000 km/s; this value will be ignored in our subsequent calculations.

By neglecting $T_{propagation}$ we can write the formula for the delay, D_i, of a frame received after i retransmissions, in a recursive manner, as follows:

$$D_i = D_{i-1} + SIFS + ACK_{transmission} + DIFS + T_{backoff\,i} + F_{transmission}, i = \overline{1,r} \qquad (4.11)$$

where: $T_{backoff\,i}$ is the waiting time due to the contention-window mechanism for the retransmission i, $i = \overline{1,r}$.

For computation purposes one can use the following particular values:

$$
\begin{aligned}
T_{backoff\,i}|_{min} &= 0; \\
T_{backoff\,i}|_{avg} &= D_{slot} \cdot CW_i/2; \\
T_{backoff\,i}|_{max} &= D_{slot} \cdot CW_i,
\end{aligned}
\qquad (4.12)
$$

where CW_i is the maximum size of the congestion window after i retransmissions, with the following values: $CW_0 = 31$, $CW_1 = 63$, $CW_2 = 127$, $CW_3 = 255$, $CW_4 = 511$, $CW_5 = CW_6 = 1023$.

Note that in the case the RTS/CTS mechanism is enabled, other delay terms have to be added to those given in equation (4.11), as follows:

$$D_{i\,RTS/CTS} = D_i + RTS_{transmission} + CTS_{transmission} + 2 \cdot SIFS, i = \overline{1,r}, \qquad (4.13)$$

where:

- $RTS_{transmission}$, the time to transmit an RTS frame, is equal to 352 μs, computed

30

using formula (4.9), given that an RTS frame has a payload of 160 bits, no MAC header and is transmitted at $R = 1$ Mb/s;

- $CTS_{transmission}$, the time to transmit a CTS frame, is equal to 304 μs, computed using formula (4.9), given that a CTS frame has a payload of 112 bits, no MAC header and is transmitted at $R = 1$ Mb/s;

- $2 \cdot SIFS$ corresponds to the wait time before the transmission of the RTS and the CTS frames.

Coming back to the case when RTS/CTS is not used, the best-case delay will be obtained when the back-off time is minimum for all retransmissions, while the worst-case delay will be obtained when the back-off time is maximum for all retransmissions. The average-case delay is obtained by taking into account an average back-off time. Considering operation at 11 Mb/s and the frame payload $P_{Frame} = 1024 \cdot 8 = 8192$ bits, using equations (4.10) and (4.11) we obtain the following values for $D_i,\ i = \overline{0, r}$.

Delay [ms]	D_0	D_1	D_2	D_3	D_4	D_5	D_6
Best case	1.147	2.293	3.439	4.586	5.732	6.879	8.026
Average case	1.457	3.233	5.650	9.346	15.603	26.980	38.356
Worst case	1.767	4.173	7.860	14.106	25.473	47.079	68.686

Table 6: The best case, average case, and worst case delay for 1024-byte-payload frames at 11 Mb/s operating rate in function of the number of retransmissions required for successful reception.

The following graph shows the dependency of the variable component of the mean delay, $D_{variable}$, as given by equation (4.7), on the frame error rate in the average case, as well as in the best and worst cases. The frames have a payload of 1024 bytes, operation rate is 11Mb/s, and the case of up to 6 retransmissions was considered.
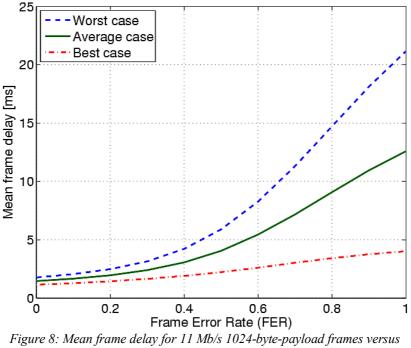

Figure 8: Mean frame delay for 11 Mb/s 1024-byte-payload frames versus frame error rate (number of retransmissions, r = 6) in the average, as well as best and worst cases.

31

4.5.2 Jitter

In order to compute the jitter, which is a measure of delay variation, we employ a definition similar to that given by ITU-T in [I.380] for IP packets. In this document ITU-T recommends to compute the jitter as the average absolute deviation with respect to the mean delay.

Consider we want to compute the jitter corresponding to the number of retransmissions i, J_i, with respect to the average delay, $D_{variable}$. A simple formula to compute J_i is the following:

$$J_i = |D_i - D_{variable}|. \tag{4.14}$$

Note that given the properties of the 802.11 MAC, and more specifically the uniform distribution of delays within an interval equal to the size of the contention window for each retransmission, the delay after i retransmissions will have a non-uniform distribution in the interval $[D_i - CW'_i/2, D_i + CW'_i/2]$, where CW'_i is defined by the formula below:

$$CW'_i = \sum_{j=0}^{i} CW_j, \quad i = \overline{0, r}. \tag{4.15}$$

As a consequence formula (4.14) is only an approximation of the jitter value after i retransmissions, giving accurate results only if $D_{variable} \notin [D_i - CW'_i/2, D_i + CW'_i/2]$. In the case in which $D_{variable} \in [D_i - CW'_i/2, D_i + CW'_i/2]$, then the non-uniform distributions of delay must be taken into account. In the future we shall consider this aspect in more detail and provide error bounds if necessary.

In order to compute the global jitter, we take into account the fact that the components of the mean delay have different weights. Therefore we shall compute the overall jitter, J, as the weighted mean of the jitter values previously calculated using (4.14). The formula is the following:

$$J = \frac{\sum_{i=0}^{r} (1 - FER) FER^i J_i}{\sum_{i=0}^{r} (1 - FER) FER^i}, \quad FER \neq 0 \wedge FER \neq 1. \tag{4.16}$$

Formula (4.16) can be further simplified to obtain:

$$J = \frac{1 - FER}{1 - FER^{r+1}} \sum_{i=0}^{r} FER^i J_i, \quad FER \neq 0 \wedge FER \neq 1. \tag{4.17}$$

In the particular cases when $FER = 0$ or $FER = 1$, the jitter is given by:

$$J|_{FER=0} = D_{slot} \cdot \frac{CW_0 + 1}{4}; \quad J|_{FER=1} = \infty. \tag{4.18}$$

In order to illustrate the variation of delay we plotted in the figure below the average-case mean frame delay, with the jitter values computed using equation (4.17) as error bars. The frames have a payload of 1024 bytes, operation rate is 11Mb/s, and the case of up to 6 retransmissions was considered.

Figure 9: Average-case mean frame delay with jitter as error bars (11 Mb/s, 1024-byte-payload frames, r = 6 retransmissions).

4.6 Number of retransmissions

Although it is not a ΔQ parameter, the average number of retransmissions is important for other aspects of the interaction between the 802.11 MAC protocol and applications, in case these applications need to adapt to MAC layer conditions. The Received Signal Strength Indication (RSSI) from the physical layer is another example of such parameter, that could be used by WLAN-aware applications for handover or mobile IP protocols.

We model the average number of retransmissions, $N_{retransmissions}$, using the following formula:

$$N_{retransmissions} = (1-FER) \sum_{i=1}^{r} FER^i \, i \, , FER \neq 0 \wedge FER \neq 1, \qquad (4.19)$$

where $r = 4$ if RTS/CTS is enabled, and $r = 7$ if RTS/CTS is disabled.

Note that in the particular case when FER = 0, the average number of retransmissions $N_{retransmissions} = 1$, and if $FER = 1$, $N_{retransmissions} = r$.

4.7 Remarks

The mechanism of retransmission as well as that of ARF depend on frame loss, more particularly on the number of consecutive frames being lost. It is important to underline that having in mind our implementation (in which time is considered discrete, being divided in slots of equal size), in modelling both delay & jitter, and the number of retransmissions, we did not consider the possibility of an operating rate change within such a time slot.

For delay & jitter, if rate changes would be allowed during retransmissions, then the operating rate at which the frame itself is sent would potentially change for each two retransmissions, which would lead to an increase of delay compared to that modelled by us. However taking into account such an effect is not in agreement with the constraint regarding rate changes during a time slot that was mentioned above.

Moreover, even if we were to model such aspects, at the moment we are not aware of the exact interaction that exists between ARF and the retransmission mechanisms. This means that we do not know what the effects of the change of a node's operating rate are upon the internal state of the MAC. More specifically we have not found explicit information so far on whether the counter for the number of retransmission is reset upon operating rate changes triggered by ARF or not. In the absence of such information we assumed that there is no reset. This assumption affects the calculation of delay & jitter, and that of the average number of retransmissions.

5 Network layer

The network layer is the level three of the seven-level ISO/OSI model. It responds to service requests from the transport layer, and issues service requests to the data link layer. The network layer addresses messages, and translates logical addresses and names into physical addresses. It also determines the route from the source to the destination computer and manages traffic problems, such as switching, routing, and controlling the congestion of data packets, creating logical paths (known as virtual circuits) for transmitting data from node to node, etc. Routing and forwarding are functions of this layer, as well as addressing, inter-networking, error handling, congestion control and packet sequencing.

In essence, the network layer is responsible for end to end (source to destination) packet delivery. The network layer provides the functional and procedural means of transferring variable length data sequences from a source to a destination via one or more networks, while maintaining the quality of service requested by the transport layer. The network layer performs network routing, flow control, network segmentation/de-segmentation, and error control functions. The best known example of a layer 3 protocol is the Internet Protocol (IP).

When frames are received by a PC, the data link layer headers are usually not available to applications, which generally work with data from the network layer and above. Our WLAN emulator will effectively be implemented by the use of a wired-network emulator, such as `dummynet`. As a consequence we only have to model the quality degradation up to the network (IP) layer, since this is the level at which we can enact this degradation.

5.1 ΔQ parameters

We need to determine the influences of the data link layer on the network layer, so that we can reproduce these effects using a wired-network emulator. These influences, as quantified by means of ΔQ (network quality variation) parameters are the following:

- Frame error rates in the MAC will cause packet loss rates, according to the retransmission mechanism, and this packet loss rates must be enforced at network layer; another mechanisms triggered by frame errors is rate adaptation;

- The CSMA/CA, retransmission and rate adaptation mechanisms will introduce delay and delay variation (jitter);

- The operating rate used by the MAC, as well as the delay (as a measure of the overall overhead) will affect the effectively available bandwidth as "seen" by IP-level applications, a constraint achieved in the wired-network emulator using rate-limitation mechanisms;

Models for the above mechanisms are needed in order to accurately describe at network layer the data link layer effects (which in their turn describe the physical layer effects caused by the real-world scenario). The 802.11 MAC details needed for these models have already been provided in the previous chapters, such as the retransmission mechanism. We will see next how those models influence the network layer parameters. The overall model and the algorithm we implemented are presented in the next chapter.

5.1.1 Packet loss

Packet loss is computed from frame error rate by taking into account the retransmission mechanism in 802.11 MAC. According to this, a packet is considered lost only if the maximum number of retransmissions is reached without getting an ACK from the receiver end.

Figure 10 shows the dependency between frame error rate and packet loss rate, *PLR*, through the use of the retransmission mechanism.
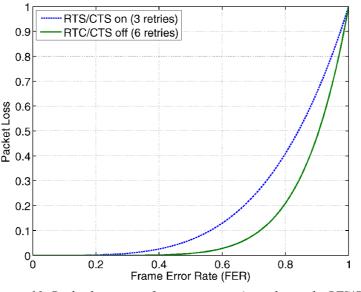

Figure 10: Packet loss versus frame error rate (considering the RTS/CTS protocol "on" or "off").

This dependency was plotted using the following equation:

$$PLR = FER^{r+1},$$ (5.1)

where *r* is the IEEE recommended maximum number of retransmissions. The value of *r* is equal to 3 or 6, depending whether RTS/CTS is enabled or not.

5.1.2 Delay & jitter

The other ΔQ parameter to be computed are the pair delay & jitter. We consider delay to have two components:

- fixed delay, that is determined by properties of the communication equipment, and is independent on the wireless technology itself;

- variable delay, that depends on the wireless operation, and is a function of the frame size, as well as the properties of the 802.11 MAC protocol; this is the delay component which we presented in the previous chapter.

The formula below is used to compute the delay, *D*, associated to a successfully received packet (the lost packets can be considered to have an infinite delay):

$$D = D_{fixed} + D_{variable},$$ (5.2)

where:

- D_{fixed} is the fixed component of the delay, independent on WLAN operation;

- $D_{variable}$ is the variable component of the delay, due to WLAN operation, as given by equation (4.7).

No additional calculations are needed to determine the jitter at network layer, which is considered to be the same with that given by equation (4.17).

5.1.3 Bandwidth

We propose a simple model that makes use of the delay computation to estimate the average throughput capacity, limited because of the properties of 802.11 MAC operation. This model takes into account the current operating rate, and the overhead of the MAC 802.11 protocol (including its variation with frame error rate). We will use the term average bandwidth, $B_{average}$, to refer to this estimated average throughput capacity.

Packets flowing through a WLAN connection may suffer from packet loss, as an effect of frame errors that were not hidden by the retransmission mechanism. However the available bandwidth too, achievable by the packets that are not lost, depends on the frame error rate. A higher frame error rate leads to an increased transmission delay, and consequently a proportional decrease of available capacity. Accordingly, the average throughput capacity, $B_{average}$, is defined as follows:

$$B_{average} = \frac{P_{tranmission}}{D_{variable}} \cdot R \, , \qquad (5.3)$$

where:

- $P_{transmission}$ is the time needed to send only the frame payload, i.e., the network layer packet; it is computed as payload size divided by operating rate;

- $D_{variable}$ is the mean variable delay computed by equation (4.7), which represents how long it actually takes, on the average, to transmit a packet;

- R is the current operating rate.

For example, using the data plotted in Figure 8, for frames with a payload of 1024 bytes, when frame error rate is zero, the available bandwidth will be approximately $B_{average} = 5.6$ Mb/s, i.e., the maximum efficiency is around 0.51. For frame error rates approaching 1, the available bandwidth drops to approximately $B_{average} = 0.65$ Mb/s, i.e., a utilization efficiency of about 0.06.

Note that although the payload (e.g., a network layer packet, such as IP) is never sent by itself, we considered the use of $P_{transmission}$ because wired-network emulators will always limit bandwidth at layer three. A variation of this is to also add to the fraction numerator of equation (5.3) both the time needed to transmit the physical and the MAC headers, or only that necessary for the MAC header, depending on the level at which the bandwidth is to be computed.

The discussion above referred to effects perceived by a single user due to communication environment issues. However, in a real case, the total throughput of a wireless network also depends on the number of users. When used by more users, given the way in which they share the wireless communication medium by using the 802.11 MAC protocol, the attainable throughput is even lower than that predicted by equation (5.3). This is mainly due to the effects of the congestion avoidance mechanism.

As a rule of thumb, it is generally considered that throughput efficiency is around 37% of the operating data rate for DCF operation, and around 75% for PCF operation [IEC-05]. A model of WLAN capacity that assumes there are *n* users, each capable of transmitting *W* bits/s predicts the throughput capacity (i.e., the maximum achievable throughput), $C_{Throughput}$, of the network assuming optimal conditions as follows [Gup-99]:

$$C_{Throughput} = \frac{W}{\sqrt{n \cdot \log(n)}}.$$

(5.4)

Assuming an ideal case of no frame errors due to unfavourable communication conditions, i.e., a maximum peak rate of 5.6 Mb/s (according to our computation above), the dependency between the number of users and the total achievable capacity is shown in Figure 11. We will address this issue in more detail in the future.
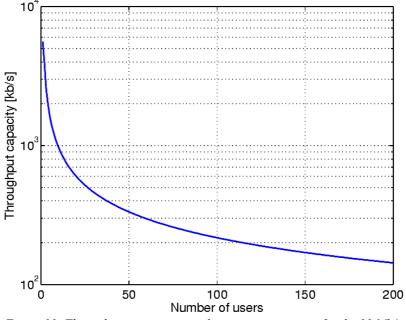

Figure 11: Throughput capacity in multi-user environments for the 11 Mb/s operating rate of 802.11.

5.2 Remarks

Due to the fact that network parameters are computed in advance, we have no knowledge about the size of the packets that will actually flow through the emulated WLAN. However the computation of all the ΔQ parameters, bandwidth, packet loss, and delay & jitter depends on packet size (i.e., frame payload). At the moment in our implementation it is possible to provide the average packet size for the connection. The default value we considered, 1024 bytes, should be changed according to user scenarios (e.g, 200 bytes when experimenting with VoIP packets). The longer term solution we envisage is to modify an existing open source wired-network emulator so that it can receive packet-size-independent values for bandwidth, packet loss and delay & jitter, and then compute on the fly the parameters for each packet, depending on its size. However in this case more of the WLAN functionality needs to be integrated in the wired-network emulator, which might prove troublesome.

6 WLAN model

In the previous chapters we have discussed the details concerning successively the WLAN physical layer and how it is influence by the real-world scenario, the WLAN data link layer and the effects the WLAN physical layer has on it, as well as the consequences of the data link layer operation on the network layer – the layer at which we use a wired-network emulator to reproduce the network layer behaviour of a WLAN environment.

In this chapter we put together the models previously proposed as an overall model for WLAN emulation. After an overview of the overall model, we provide the description of the algorithm that was implemented in practice. Please refer to the Appendix for some additional implementation details.

6.1 Model overview

Our aim is to model WLANs so that we can reproduce their behaviour in a controlled environment. As discussed before WLAN performance depends on numerous factors related to communication conditions, adapter properties (such as receive sensitivity), etc. Depending on all these circumstances the transmitter-receiver pair can operate with higher or lower performance. When signal strength at the receiver diminishes more than a certain threshold corresponding to the current operating rate, for example, the frame error rate will increase, and eventually there will be packet loss. Other effects such as delay and jitter, as well as rate adaptation also occur. Since rate adaptation through ARF is implemented in most existing WLAN adapters, we must take it into account in order to achieve a realistic emulation. All these elements must be taken into account and accurately modelled in order to obtain a realistic reproduction of the WLAN behaviour.

In this section we propose a WLAN model that we use for WLAN emulation. This model takes the real-world scenario to WLAN physical layer, then to WLAN data link layer, and finally to network layer. The overall WLAN model is a combination of the following models that were previously described:

1. Real-world scenario $\rightarrow$ Physical layer

   a) Log-distance path-loss model for determining the received power, P_r, as function of the transmitted power, P_t, and the distance, d, between receiver and transmitter. The communication environment is described by parameters such as α (the path-loss coefficient), σ (shadowing parameter), or W (wall attenuation). The equations of interest are (3.12) and (3.13);

   b) Frame error rate (FER) P_r-threshold-based model to determine FER_1 as function of the received power, P_r. Equation (3.2) is used;

   c) Bit error rate (BER) SNR-based model to determine BER_2 as function of the received power, P_r, and the noise power, N (additive white Gaussian noise). Equation (3.5) or an equivalent one is employed. In this case we also need a frame error rate model to determine FER_2 as function of BER_2 and frame payload, P_{Frame}. Equation (4.5) can be used for this purpose;

2. Physical layer → Data link layer

   a) ARF probability-threshold model to determine the operating rate of the WLAN adapters. This model is described by equation (4.1);

   b) Total frame error rate calculation based on the two bit/frame error rate models used at physical layer, and described by equation (4.2);

   c) Delay model for determining the variable delay and jitter introduced at data link layer, using equations (4.7) and (4.17), respectively;

3. Data link layer → Network layer

   a) Packet loss rate (PLR) model to determine PLR from FER by taking into account the 802.11 MAC retransmission mechanism. Equation (5.1) is used;

   b) Delay model to determine the total delay, by taking into account both the fixed (WLAN-operation independent) and the variable (WLAN-operation dependent) components. This is given by equation (5.2) for delay, while jitter is given still by (4.17);

   c) Bandwidth model to determine the effectively available bandwidth as "perceived" at network layer, according to equation (5.3).

6.2 Algorithm description

The previous section showed an overview of the overall WLAN model that we use in emulation. We give here the algorithm we proposed and implemented to convert a real-world scenario to physical layer, data link layer, and finally network layer effects in WLAN environments. The steps of the algorithm are the following:

1. Determine the current real-world scenario description (absolute position of nodes, particular conditions, etc.). The scenario description may change at different moments of time due to node movement, transmission power adaptation, etc;

2. For each transmitter-receiver pair compute the signal strength at the receiver, P_r, knowing the transmitted power, P_t, and the distance between them, d; the indoor or outdoor log-distance path-loss model will be used for this purpose, with an appropriate selection of (α, σ, W) corresponding to the emulated environment – *equations (3.12), (3.13)*;

3. Compute the frame error rate, *FER*, for each node

   a) Use the P_r-threshold-based model to compute FER_1 as function of the signal strength at the receiver, P_r, and the node's receive sensitivity threshold, T, that is associated to the current operating rate (T depends on the emulated wireless interface properties) – *equation (3.2)*;

   b) Use the SNR-based model to compute BER_2 as function of the signal strength at the receiver, P_r, and the noise power, N. Compute FER_2 from BER_2 for a specific frame size, at the moment considered as the average frame size on the modelled connection – *equations (3.5), (4.5)*;

   c) Combine FER_1 and FER_2 – *equation (4.2)*;

4. Compute the packet loss rate, *PLR*, from *FER*, by taking into account the MAC retransmission protocol – *equation (5.1)*;

5. Use the ARF mechanism model to determine whether a rate change will be triggered – *equation (4.1)*;

   a) If $FER^2 > T_{down}$ (i.e., the probability of losing two consecutive frames is high), and the node is not already operating at the minimum rate, change the rate to the next lower operating rate;

   b) Else, if $(1 - FER)^{10} > T_{up}$ (i.e., the probability of successfully transmitting 10 frames is high), the node is not already operating at the maximum rate, and $FER_{higher} < T_{keep}$ (i.e., the probability to successfully send a packet at a higher rate), change the rate to the next higher operating rate;

6. Compute the delay and its variation that are induced by the above-mentioned mechanisms – *equations (5.2), (4.17)*;

7. Compute the available bandwidth knowing the delay and the current operating rate – *equation (5.3)*.

7 Conclusions

In this document we presented our approach to modelling WLAN behaviour. Our approach is layer oriented, as it computes first the physical layer effects that the changes in the communication channel have on the received power. This is followed by an estimation of the bit error rate (BER) based on the received power and the noise power. BER is further converted into frame error rate (FER) by taking into account the specificity of the 802.11 MAC layer. Finally the network layer effects are determined: packet loss, delay & jitter, and available bandwidth.

Through the use of the WLAN model that is summarized above, the real-world WLAN scenario is converted into a sequence of quality degradation (ΔQ) states of the network, termed "ΔQ description". The ΔQ description accurately reproduces the wireless network behaviour corresponding to the emulated scenario, and can be subsequently used to configure a wired-network emulator to recreate this behaviour in a network experiment.

In Chapter 6 of this document we have shown how the models presented in previous sections can be put together to create an overall model for the real-world scenario to physical layer, then to data link layer, and finally to network layer effects. This chapter also provides the algorithm that we used to implement the first version of the WLAN emulator, QOMET[12] v1.0, with references to all the formulas used.

The main direction for future work is the calibration of the WLAN model that we proposed. This will be achieved through equivalent tests using simple real WLAN environments. In this stage we will make use of traffic captures performed using WLAN monitoring and analysis tools, such as AiroPeek from WildPackets. After calibration the model accuracy will be validated by comparison with a more complex WLAN scenario.

Also part of our future work is to extend the functionality of the WLAN emulator by modelling more advanced aspects, such as power adaptation of mobile nodes, AP-specific issues such as buffer size and uplink/downlink asymmetry, or aspects related to AP-based versus ad hoc mode operation, such as ad hoc routing algorithms.

---

12 Quality Of applications in transforMing Environments Testbed.

List of acronyms

ACK	ACKnowledgement (frame)
AP	Access Point
ARF	Auto-Rate Fallback
ARQ	Automatic Repeat reQuest
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
CRC	Cyclic Redundancy Check
CS	Collision Sense
CSMA/CA	Collision Sense Multiple Access with Collision Avoidance
CSMA/CD	Collision Sense Multiple Access with Collision Detection
CTS	Clear To Send
CW	Congestion Window
DCF	Distributed Coordination Function
DIFS	Distributed coordinated function Inter Frame Space
DLL	Data Link Layer
DQPSK	Differential Quadrature Phase Shift Keying
EiRP	Effective Isotropic Radiated Power
FCS	Frame Check Sum
FEC	Forward Error Correction
FER	Frame Error Rate
GBN-ARQ	Go-Back-N ARQ
IEEE	Institute of Electrical and Electronics Engineering
IFS	Inter Frame Space
IP	Internet Protocol
ISO/OSI	International Standards Organization/Open Systems Interconnections
ITU-T	International Telecommunication Union, the Telecommunication division
LAN	Local Area Network
LLC	Logical Link Control
MAC	Medium Access Control
MOS	Mean Opinion Score
MPDU	MAC Protocol Data Units
NACK	Negative ACKnowledgement (frame)
NIC	Network Interface Card
PCF	Point Coordination Function
PDA	Personal Digital Assistant
PDU	Physical Data Unit
PESQ	Perceptual Evaluation of Speech Quality
PHY	PHYsical Layer
PIFS	Point coordinated function Inter Frame Space

PLCP	Physical Layer Convergence Procedure/Protocol
PLR	Packet Loss Rate
PMD	Physical Medium Dependent
QOMET	Quality Of applications in transforMing Environments Testbed
QoS	Quality of Service
RBAR	Receiver Based Auto Rate
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
RTS	Request To Send
SIFS	Short Inter Frame Space
SNR	Signal-to-Noise Ratio
SR-ARQ	Selective Repeat ARQ
SW-ARQ	Stop and Wait Automatic Repeat reQuest
TCP/IP	Transmission Control Protocol / Internet Protocol
TDMA	Time-Division Multiple Access
UPQ	User-Perceived Quality
VoIP	Voice over IP
WLAN	Wireless Local Area Network

References

[802.11]   ANSI/IEEE Standard 802.11, 1999 Edition (Reaffirmed 2003),
            http://standards.ieee.org/getieee802/download/802.11-1999.pdf

[Bat-03]   M. Bateman, C. Allison, A. Ruddle, "A Scenario Driven Emulator for
            Wireless, Fixed and Ad Hoc networks", *Proc. of PGNet2003*, Liverpool,
            2003, pp. 273-278.

[Beu-03]   R. Beuran, M. Ivanovici, B. Dobinson, N. Davies, P. Thompson, "Network
            Quality of Service Measurement System for Application Requirements
            Evaluation", *International Symposium on Performance Evaluation of
            Computer and Telecommunication Systems, SPECTS'03*, Montreal,
            Canada, July 20-24, 2003, pp. 380-387.

[Beu-04]   R. Beuran, M. Ivanovici, "User-Perceived Quality Assessment for VoIP
            Applications", *technical report (delivered to U4EA Technologies), CERN-
            OPEN-2004-007*, CERN, Geneva, Switzerland, January 2004.

[Beu-06]   R. Beuran, "VoIP over Wireless LAN Survey", research report, *IS-RR-
            2006-005, Japan Advanced Institute of Science and Technology (JAIST)*,
            Ishikawa, Japan, April 2006.

[DIS-05]   Defence Information Systems Agency (DISA), "WLAN Site Survey
            Addendum to the Wireless Security Technical Implementation Guide",
            U.S.A. Department of Defense, October 2005.

[Dum-**]   Dummynet FreeBSD network emulator,
            http://info.iet.unipi.it/~luigi/ip_dummynet.

[Far-05]   D. B. Faria, "Modelling Signal Attenuation in IEEE 802.11 Wireless
            LANs – Vol. 1", *Technical Report TR-KP06-0118*, Kiwi Project, Stanford
            University, July 2005.

[G.107]   ITU-T Recommendation G.107, "The E-model, a computational model for
            use in transmission planning", *ITU-T*, March 2005.

[Gei-02]   J. Geier, "Improving WLAN Performance with Fragmentation", *Wi-Fi
            Planet tutorial*, September 2002.

[Gup-99]   P. Gupta, P.R. Kumar, "Capacity of wireless networks", *technical report*,
            University of Illinois, Urbana-Champaign, 1999.

[Hol-01]   G. Holland, N. Vaidya, P. Bahl, "A rate-adaptive MAC protocol for multi-
            hop wireless networks", *Proc. of 7th Conf. On Mobile Computing and
            Networking*, 2001, pp. 236-251.

[I.380]   ITU-T Recommendation I.380, "Internet Protocol (IP) Data
            Communication Service - IP Packet Transfer and Availability Performance
            Parameters", *ITU-T*, February 1999.

[IEC-05]   The International Engineering Consortium, "Including VoIP over WLAN
            in a Seamless Next-Generation Wireless Environment", *white paper*, 2005.

[Int-00]   Intersil, "Measurement of WLAN Receiver Sensitivity", *technical brief*,
            February 2000.

[Int-02]   Intersil, "HFA3861B: Direct Sequence Spread Spectrum Baseband

Processor", *data sheet*, February 2002.

[Kam-97]   A. Kamerman, L. Monteban, "WaveLAN-II: A high-performance wireless LAN for the unlicensed band", *Bell Lab Technical Journal*, 1997, pp. 118-133.

[Kum-04]   A. Kumar, D. Manjunath, J. Kuri, "Communication networking: an analytical approach", *Morgan Kaufmann Publishers*, 2004.

[Lac-04]   M. Lacage, M. H. Manshaei, T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach", *Proc. of the 7th ACM International Symposium on Modeling, analysis and simulation of wireless and mobile systems*, Venice, Italy, 2004, pp. 126 – 134.

[Miy-05]   T. Miyachi, K. Chine, Y. Shinoda, "Automatic Configuration and Execution of Internet Experiments on an Actual Node-based Testbed", *TridentCom2005*, Trento, Italy, February 2005, pp. 274-282.

[Ni-04]   Q. Ni, L. Romdhani, T. Turletti, "A survey of QoS enhancements for IEEE 802.11 wireless LAN", *Wireless Communications and Mobile Computing*, vol. 4, 2004, pp. 547-566.

[P.862]   ITU-T Recommendation P.862, "Perceptual evaluation of speech quality (PESQ), an objective method for end to end speech quality assessment of narrow-band telephone networks and codecs", *ITU-T*, February 2001.

[Per-04]   T. Perennou, E. Conchon, L. Dairaine, M. Diazet, "Two-Stage Wireless Network Emulation", *Proc. of WCC2004*, 2004, pp. 57-66.

[Pra-03]   J. del Prado Pavon, S. Choi, "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement", *IEEE International Conference on Communications, ICC '03*, vol. 2 , 11-15 May 2003, pp.1108 – 1113.

[Pro-04]   Proxim, "ORiNOCO 11b Client PC Card Specification", *data sheet*, 2004.

[Pun-00]   R.J. Punnoose, P.V. Nikitin, D.D. Stancil, "Efficient Simulation of Ricean Fading within a Packet Simulator", *Proc. of IEEE Vehicular Technology Conference*, September 2000, pp. 764-767.

[Rap-02]   T. S. Rappaport, "Wireless Communications: Principles and Practice", *2nd edition*, Prentice Hall PTR, 2002.

[Yee-02]   J. Yee, H. Pezeshki-Esfahani, "Understanding Wireless LAN Performance Trade-offs", Communication Systems Design, November 2002, pp. 32-35.

Appendix

The first version of the WLAN emulator, QOMET v1.0, was implemented in C. The input scenario description is given in XML format, and the output is a file containing the ΔQ parameter variation in time, for each of the participating nodes. This file can be used to reproduce the WLAN effects of the real-world scenario in a wired-network environment, as we did in several demonstration experiments by using shell scripts to control the emulation process.

This appendix describes the objects used in our implementation of the WLAN emulator. The fields of these objects are generally initialized through the use of the XML scenario representation, but some of them are determined dynamically during the scenario-to-ΔQ computation phase:

- Scenario
  - Participating nodes, environments used for communication, motions associated to nodes, and connections between nodes;
  - Scenario properties (e.g, duration, time step);
- Node
  - Node type (e.g., mobile node, access point);
  - Connection type (e.g., ad hoc, base station/access point);
  - RF properties (e.g., Tx power, Rx sensitivity, operating rate);
  - Position in (x, y, z) coordinates;
  - ΔQ parameters;
- RF Environment
  - Environment type (e.g., indoor, outdoor);
  - Communication channel parameters (alpha, sigma, W);
- Node motion
  - Motion type (e.g., linear motion, circular motion);
  - Motion parameters (e.g., speed in (x, y, $z)$ coordinates);
  - Motion duration (start time, stop time);
- Connections
  - Connected nodes and the environment through which they connect.

Future extensions of the WLAN emulator include, but are not limited to:

- Mobile-node specific properties (e.g., power adaptation);
- Random motion, repetitive motion;
- AP-specific properties (e.g., buffer size, number of connected nodes);
- Dynamic connections.