

# Validation of the IEEE 802.11 MAC Model in the ns3 Simulator using the EXTREME Testbed

Nicola Baldo, Manuel Requena-Esteso, José Núñez-Martínez, Marc Portolès-Comeras,  
Jaume Nin-Guerrero, Paolo Dini, Josep Mangués-Bafalluy  
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC)  
Av. Carl Friedrich Gauss 7, 08860 Castelldefels (Barcelona), Spain  
{nbaldo, mrequena, jnunez, mportoles, jnin, pdini, jmangues}@cttc.es

## ABSTRACT

We validate the IEEE 802.11 MAC layer model in ns3 by means of measurements on the EXTREME testbed. We consider different scenarios: communications within a single pair of nodes, multi-user communications using either VoIP or saturated traffic, and communications in the presence of hidden nodes. For each scenario we describe in detail our testbed and simulation setup, and compare the results provided by the ns3 simulator with the performance measured on the testbed.

## Categories and Subject Descriptors

I.6 [Simulation and Modeling]: General, Model Validation and Analysis, Model Development; C.4 [Performance of Systems]: Measurements Techniques

## General Terms

Experimentation, Measurement, Performance, Verification

## Keywords

Validation, Simulator, Testbed, IEEE 802.11, Medium Access Control

## 1. INTRODUCTION

The research in the field of wireless communications and networking has always relied rather heavily on network simulation tools for the performance evaluation. The benefits of using simulators instead of real testbeds are well known: simulation offers faster prototyping times, controllable environment, repeatable results, and in many cases better scalability. However the main problem with simulators is that, though they are designed to reproduce the behavior of real systems, it is often questioned whether they do it with satisfactory accuracy or not. Clearly, researchers should use simulation tools that are known to provide an accurate representation of the behavior of the system that they want to

study. With this respect, the process of verifying simulator accuracy, which is commonly referred to as validation, is of primary importance.

ns3 is a very recent network simulator which incorporates a remarkable number of interesting features [1]. Due to this fact, and furthermore due to its open source nature, it is becoming more and more popular in the networking research community. However, because of its young age, only a few of its models have been subject to validation studies [2], and the vast majority of the models provided within the simulator have not been validated.

To partially fill this gap, in this paper we present the study that we carried out to validate the ns3 model of the IEEE 802.11 MAC layer. Our validation approach is to compare the results obtained with ns3 to the performance obtained from a real communication system, which in our case is provided by our EXTREME Testbed <sup>®</sup> [3]. We note that this validation strategy is commonly regarded as the most desirable. For example, in [4] the problem of simulator validation is discussed in a broad view. In particular, the authors argue that the direct comparison of simulation output with measurement obtained from a real network is, when possible, the best solution. In [5], in the context of discussing good simulation practices, the authors discuss also validation with respect to data obtained from real measurements, arguing that “it is certainly the most direct and scientifically rigorous means of ascertaining simulation credibility”, of course with the caveat of verifying the measurement instrumentation.

While examining the literature in the field of validation for wireless network simulators, we realized that the vast majority of the studies that are carried out concern the validation of the physical layer and channel model [2, 6–10], whereas little validation is done for the MAC layer. One possible reason is the assumption that, since the MAC model of a simulator is normally expected to be a verbatim implementation of the protocol as described by the standard, most people take it for granted that the simulator accurately reproduces the behavior of real devices.

However, while there are institutions in charge of certifying the compliance with the standard of devices produced by different vendors, no such institution exists for network simulators, and therefore the validation of network simulators needs to be done by other means. For open source simulators like ns3, it is commonly expected that the presence of a big community of users and developers will ensure such standard compliance, according to the principle that “given enough eyeballs, all bugs are shallow” [11]. However, at the time of this writing, the source code of the Wifi model in

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*SIMUTools 2010* March 15–19, Torremolinos, Malaga, Spain.  
Copyright 2010 ICST, ISBN 78-963-9799-87-5.

ns3 amounts to more than 27k lines, whereas the number of people involved in the active development of this code is still rather low; as a consequence, some bugs related to the incorrect behavior of the ns3 Wifi model have been reported, and probably more could be found. While we are confident that the ns3 user base will be expanding in the near future, and so this problem will eventually be solved, we still think that at this time a validation study of the Wifi MAC model in ns3 would be of great interest, and would possibly help in enhancing the credibility of the simulator.

Furthermore, even supposing that some “standard compliance” can be assured for network simulators, we still have to face the problem that in many cases people are interested in reproducing the performance of real devices, rather than the performance of a perfect standard-compliant device. With this respect, Bianchi et al. [12] experimentally assessed the backoff behavior of different commercial cards, showing that they might feature non-standard behavior opportunistically aimed at performance enhancement. Similarly, in our previous work [13] we analyzed the throughput performance of different commercial wireless cards, concluding that the differences in the way in which the standard is implemented translate into differences in the obtained performance. As a consequence, we argue that the performance of real (i.e., commercial) devices should be taken into consideration for a proper validation of simulator performance. This is another reason why we chose to perform a validation study of the ns3 simulator using commercial 802.11 devices.

In the remainder of this paper, we describe our validation setup, and discuss the chosen validation scenarios; for each scenario, after outlining the scenario itself, we will describe how we tuned the ns3 simulator and the testbed for the validation, and we will discuss the obtained results.

## 2. VALIDATION SETUP

### 2.1 Testbed setup

All the experiments with commercial wireless devices were carried out using the EXTREME Testbed <sup>®</sup> [3] deployed at CTTC. This is a multi-purpose networking experimental platform featuring high automation capabilities that support automatic execution of the experiments, data collection and data processing.

The EXTREME testbed is composed of a cluster of computer nodes. All these nodes are Pentium 4 PCs with a 3GHz processor, 512MB of RAM memory, and running Linux with kernel 2.6.27. Every node can be equipped with up to two wireless Network Interface Cards (NICs). Three types of wireless NICs have been used: LevelOne WNC-0300, D-Link DWA-556 and Z-COM ZDC XI-626. The LevelOne and the D-Link models are based on the Atheros 11b/g chipset and the Z-Com model carries the popular Prism chipset. The automated experiment setup in EXTREME makes an extensive use of the wireless extensions API [14] to configure and control wireless devices. The `madwifi` driver [15] supports this API and controls LevelOne and D-Link cards. The `hostap` driver [16] also supports this API and controls Z-COM devices.

Since the main objective of this study is to examine the MAC layer behavior of 802.11 devices, our testbed setup was designed as to minimize the effects of channel propagation on the measurements. To this aim, all the communications between wireless devices are done through coaxial wires.

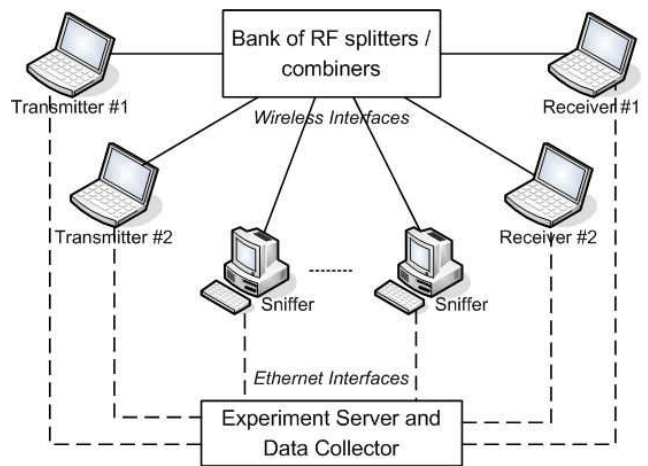


Figure 1: Setup of the EXTREME Testbed <sup>®</sup>

The setup is represented in Figure 1: all wireless devices are connected to a central bank of splitters and combiners. This bank of splitters replicates with very low attenuation (in comparison to open-air propagation) all signal inputs in each of its ports to the rest of ports. The bank of splitters and combiners is composed of minicircuit ZX10-4-27 splitters (with 4 ports) and minicircuit ZFSC-2-10G splitters (with 2 ports).

Depending on the scenario, one or more transmitters send a data stream to one or more receivers. Data streams are sent at different packet rates using packets of variable size. The bank of splitters/combiners replicates the data stream to the intended receivers and to a group of wireless sniffers. Both the intended receivers and the sniffers capture all received 802.11 frames and store them for later analysis.

The application used to generate data streams is the Multi-GENERator toolset (MGEN) [17]. The reason for choosing this application was manifold. First, MGEN lets the user specify the characteristics of the traffic to be generated, such as the distribution of the inter-packet generation times and the size of the generated packets. Furthermore, by using MGEN, the traffic source can activate an option called “precise on” that increases the timing accuracy in the generation of packets in order to provide better guarantees that the specified traffic profile will be respected. Finally, MGEN allows to store the received packets for later processing.

### 2.2 Simulator setup

We used a snapshot (revision number 4825) of the development version of the ns3 simulator [18]. Depending on the particular scenario considered, we had to apply some patches in order to obtain from the simulator a behavior identical to the one that was seen for commercial devices. In particular, the patches we used fulfill the following purposes:

1. Use the PLCP short preamble instead of the long preamble. In the version of the simulator that we used the preamble was not user-configurable but rather it was hard-coded. This patch changes the hard-coded value of the preamble to short preamble.
2. Force the PHY rate of the ACK frames to the rate of the preceding DATA frames. The problem is that when beacons are not exchanged then the Basic Rate

set, i.e., the set of PHY rates which a mobile station (STA) is allowed to use, is empty, and so ACKs get sent always at the basic (lower) rate. This patch modifies the rate selection process for ACK packets so that the rate of the preceding DATA packet is always used.

3. Modify the minimum and maximum values of the Contention Window for the IEEE 802.11 DCF procedure. These values were supposed to be user-configurable, but actually, due to a bug, it was not possible to configure them using the ns3 attribute system. The patch that we developed hard-codes the desired values.

These patches are publicly available and can be downloaded from [19].

### 2.3 Validation methodology

For the validation procedure, we adopted the practice suggested in [4], which consists of using aggregate performance statistics specifically chosen for each particular scenario being considered. The particular metric considered will be discussed in the description of every scenario in the next section.

To compare the results obtained with the ns3 simulator with the performance measured on the testbed, we exploit the fact that the ns3 simulator supports, among its output formats, traces in the PCAP format, which are the same that can be obtained from EXTREME. In particular, to get a more detailed insight on the MAC layer performance, we retrieve the information stored in the so called PHY headers, such as the radiotap header and the prism header. Many drivers for commercial Wifi cards, such as the `madwifi` and `hostap` drivers, which we use in EXTREME, can optionally add this information to all traced packets. Unfortunately, when we started this study, neither radiotap nor prism were supported in ns3; as a consequence, we contributed those to the simulator (available since release 3.5 of ns3)

Another convenient tool for MAC and PHY layer performance analysis which is available for `madwifi` is `athstats`. This tool allows for the retrieval of several interesting link layer statistics from the wireless device, such as the number of failed transmission attempts and the number of reception errors due to various reasons (bad CRC, PHY error, etc.). Again, we were interested in comparing this data with ns3, so we implemented an `athstats`-like tool for ns3 (included in the official ns3 since release 3.6).

## 3. VALIDATION SCENARIOS

The core of our validation study is discussed in this section. Each of the following subsections refers to a particular scenario. Within each subsection, we will first describe the scenario setup, then we will provide some information on the particular testbed and simulator tuning that was required for that scenario, and finally we present and discuss the obtained results.

### 3.1 Scenario 1: single pair

#### 3.1.1 Scenario description

We first analyze the performance of the simulator in a simple scenario in which one AP sends a UDP data flow to a single client station at a given application-level packet rate. This is the same scenario that we considered in our

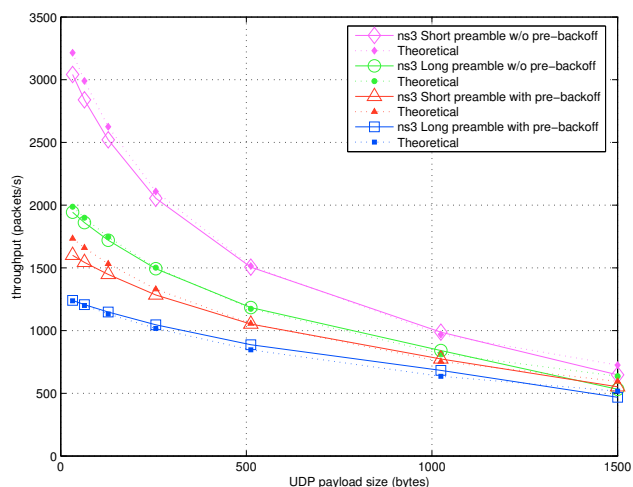


Figure 2: Single pair scenario: maximum achievable throughput at 11 Mbps

prior work [13], where we compared the behavior of commercial WLAN devices from different manufacturers. The performance achievable in these conditions has been extensively studied in the literature. In particular, the authors in [20] showed the existence of a theoretical upper limit to the maximum throughput achievable by a single WLAN station; this limit is imposed by the protocol functionality itself and is independent of the physical modulation used. Therefore, by choosing this scenario, we have an interesting opportunity to compare the results obtained with the simulator not only with the performance obtained by real devices, but also with the theoretical performance.

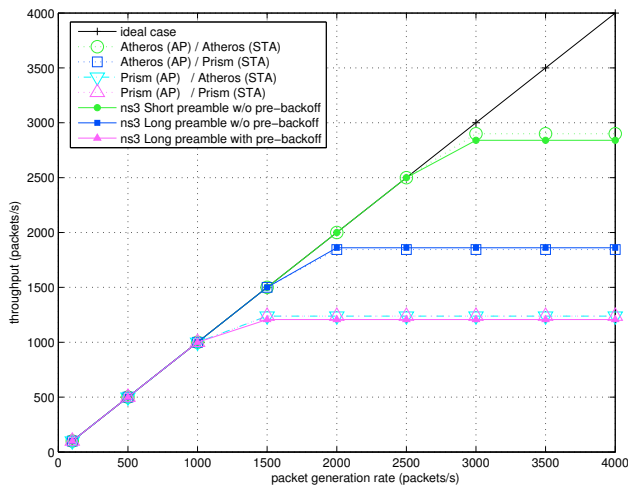
#### 3.1.2 Simulator and testbed tuning

Different WLAN devices and/or device drivers can use different default parameters of the IEEE 802.11 standard. For example, all devices implement the long PLCP preamble, which is mandatory according to the DSSS PHY specification (802.11b), but they can also optionally use the short PLCP preamble to increase throughput performance; in fact, some devices actually use the short preamble as the default configuration. Furthermore, some commercial devices also exhibit non-standard behavior. For instance, according to the IEEE 802.11 standard, a STA should enter backoff after each successful transmission of a packet; this procedure is called post-backoff. However, some commercial Wifi devices do not perform post-backoff.

To be able to simulate the types of behavior that we just described, we applied to the ns3 simulator the patches 1. and 3. that we introduced in Section 2.2.

#### 3.1.3 Validation results

For each of the configurations that we described in Section 3.1.2, we ran different experiments varying the packet rate generated by the application. In Figure 2 we compare the throughput performance achieved using ns3 with the maximum theoretical throughput calculated as per [20] and [13]. We note that there is a rather good match between the two models: in the worst case, there is an error of less than 5% between the theoretical calculations and the results obtained in the simulator.



**Figure 3: Single pair scenario: relation between the packet generation rate and the obtained throughput**

In Figure 3 we compare the performance obtained with ns3 with the one measured on the testbed. In this case, it can be observed that with a proper configuration of the simulator it is possible to model the performance of different commercial WLAN devices. For example, the Atheros device can be simulated by using short preamble and not doing post-backoff. The Prism device can be simulated by using long preamble and respecting the post-backoff.

## 3.2 Scenario 2: VoIP

### 3.2.1 Scenario description

In this scenario, we consider one AP node and a variable number of STAs performing VoIP communications. For each STA, two voice flows are generated, one from the STA to the AP and the other from the AP to the STA. An additional STA is passively monitoring the wireless channel for the purpose of measuring different metrics of interest. In EXTREME, noise-related channel errors are minimized by using RF cables, whereas in ns3 the same effect is achieved by locating wireless nodes close to the AP. In each experiment, all the nodes use the same fixed PHY rate and the same voice codec. We repeated several experiments varying the number of STAs, the voice codec (G.711, G.723, G.729) and using the 1 and 2 Mbps PHY rates of the DSSS PHY specifications. This is the type of scenario which is considered in the most of prior works dealing with VoIP over WLAN [21, 22]. Note that the chosen maximum number of STAs always exceeds the VoIP capacity as defined in [21]. We used the E-Model [23] as method to evaluate the perceived voice quality by the user. For every particular setting, 25 independent repetitions of the same experiment were run.

### 3.2.2 Simulator and testbed tuning

Since we already had a working setup for the VoIP scenario in the testbed for other research activities, we tuned the ns3 simulator so that it matched the configuration of the testbed. For this purpose, we applied to the ns3 simulator the patches 1., 2. and 3., which are described in Section 2.2. Furthermore, we used a custom ns3 application that accurately models VoIP traffic characteristics and that addition-

ally provides some application-layer performance metrics of interest.

### 3.2.3 Validation results

The results obtained for this scenario are reported in Figures 4-8; for each figure, a particular metric is plotted as a function of the number of VoIP flows for different VoIP codecs and PHY rates.

Figure 4 shows the total number of frames that are successfully decoded by the monitoring STA; this number is equivalent to the network throughput. We observe that the throughput increases linearly with the number of VoIP flows, up to some point at which saturation is reached. We note that for some scenarios (e.g., G.711 at 1Mbps) there is a good match between the simulator and the testbed, whereas for other scenarios (e.g., G.723 and G.729 at 1Mbps) the saturation point observed in the testbed occurs for a lower number of VoIP flows with respect to that observed in the simulator. A similar behavior is observed in Figure 5, where we plot the busyness ratio of the channel, defined as the average fraction of time in which the monitoring STA perceives the channel as busy due to the transmission of an 802.11 frame<sup>1</sup>. To understand the reason for this, we plot in Figure 6 the frame error probability measured by the monitoring node: a mismatch in the collision probability is evident between the simulator and the testbed. This is confirmed by the results of Figure 7, which show a similar mismatch for the number of retransmission attempts per second measured by every VoIP STA. One candidate explanation for this behavior is that in real devices, due to the fact that several functionalities are implemented in software, it is possible that in some situations the ACK frame which is to follow a DATA frame is sent with a delay higher than the SIFS value. This triggers an ACK timeout at the sender, thus causing an unneeded retransmission, which consumes additional channel resources. To assess the validity of this explanation, we analyzed the number of duplicate frames that are detected by the monitor node; this is reported in Figure 8. From this figure, we see that also in the ns3 simulations duplicate packets are seen; this is very surprising, since duplicate frames can be only due to late or lost ACK frames, which should not occur in our particular scenario in which there are no hidden nodes and frame errors are always due to collisions and should involve DATA frames only. We suggest that this phenomenon might be due to a bug in the state machine of the ns3 802.11 MAC model.

In Figure 9 we report the application layer performance measured using the E-model[23], in particular its  $R$  factor metric. We report the performance in downlink only, as it is well known that in this type of scenario the downlink is the bottleneck [21]. We remark that values of  $R \geq 70$  can be interpreted as acceptable voice quality, while values of  $R < 70$  indicate bad voice quality. Now, looking at Figure 9, we notice that the general trend for all curves is that for low number of voice flows the quality is acceptable, and remains constant as the number of flows is increased, until a point is reached at which quality degrades sharply. The value of

<sup>1</sup>We note that the busyness ratio is calculated from those frames that appear in the PCAP trace; since only successfully decoded frames appear in the PCAP, this busyness ratio does not account for the time in which the channel is busy due to collided frames.

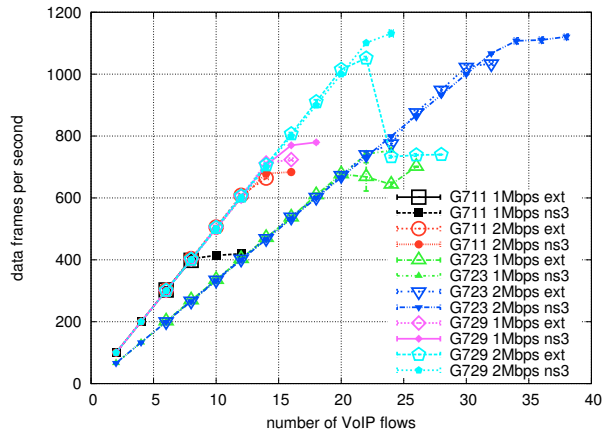


Figure 4: VoIP scenario: number of successfully decoded transmission attempts per second

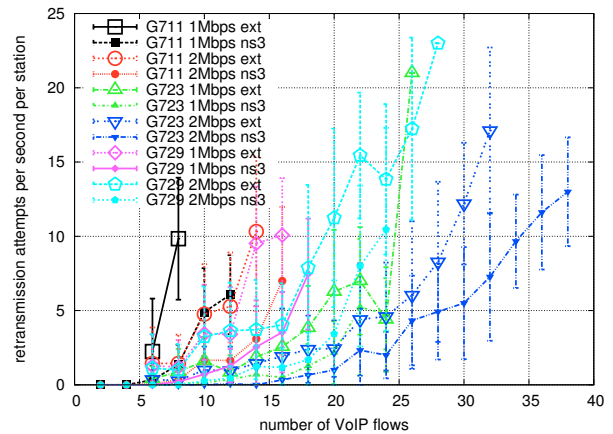


Figure 7: VoIP scenario: number of retransmission attempts per second

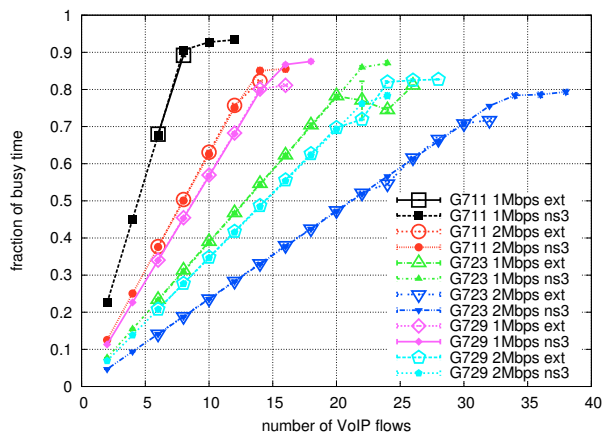


Figure 5: VoIP scenario: fraction of channel time spent for successfully decoded transmissions

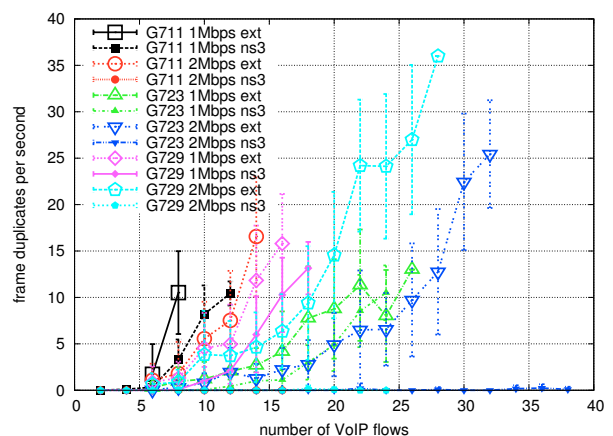


Figure 8: VoIP scenario: number of duplicate frames

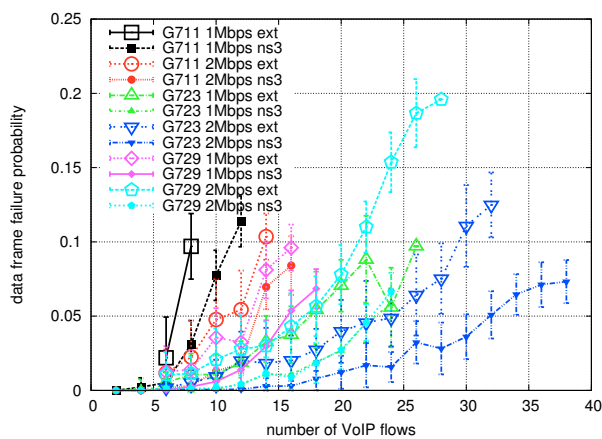


Figure 6: VoIP scenario: data frame failure probability

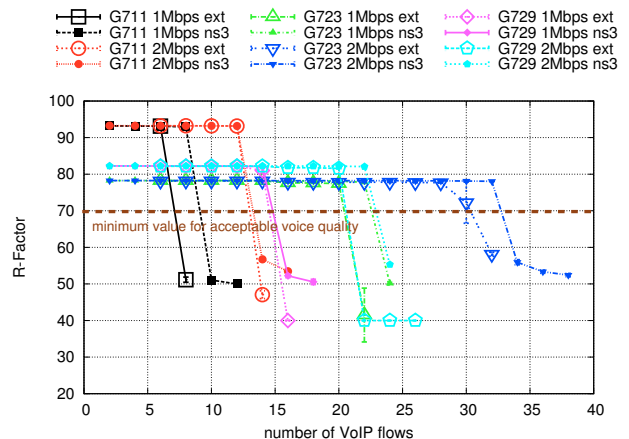


Figure 9: VoIP scenario: application layer performance

the number of flows at this point is referred to as VoIP capacity, and is known to vary with respect to the PHY rate and the chosen voice codec [21]. We observed that only in two of the cases considered (G.711 at 2 Mbps and G.729 at 1 Mbps) the VoIP capacity obtained with EXTREME is the same obtained with ns3; in the other 4 cases, the capacity results to be always two flows less (i.e., one less STA) for EXTREME. We argue that this is due to the higher frame transmission failure probability which was observed in EXTREME, which in turn translates in higher MAC overhead and lower VoIP capacity.

### 3.3 Scenario 3: saturation

#### 3.3.1 Scenario description

In this scenario, we have a given number of nodes, all within transmission range. An additional node acts as an Access Point. All nodes communicate with the AP using the same fixed PHY rate. The objective of this scenario is to model a situation in which every wireless node always has a pending data packet to transmit on the wireless channel. We achieved this both in the simulator and in the testbed by using an application rate for every node such that the summation of the application rates (without considering any protocol overhead) was higher than the PHY rate being used.

#### 3.3.2 Simulator and testbed tuning

For this scenario, the tuning procedure that we adopted is the same that was discussed in Section 3.2.2 for the VoIP scenario.

#### 3.3.3 Results

The results are reported in Figures 10, 11 and 12, where we plot respectively the frame failure probability, the channel busyness ratio, and the number of transmission per seconds as per the definitions that we gave in Section 3.2.3. We note that there is good agreement between the simulator and the testbed as far as the channel busyness ratio and the number of transmitted frames per seconds; however, a significant difference can be noted with respect to the frame loss probability, in accordance to what we have observed for the VoIP scenario.

### 3.4 Scenario 4: hidden node

The third case studied is a hidden node scenario. This happens when two sender stations are distant enough from each other so that they cannot sense their respective transmissions. In this case, the CSMA/CA protocol fails in regulating their transmissions to other nodes which are within communication range of the first two nodes and will therefore see a collisions when the first two nodes transmit simultaneously. This scenario is used here to illustrate how ns3 can be used, in some cases, to assess the performance of a testbed and tune its behavior.

#### 3.4.1 Scenario setup

We consider the hidden node scenario depicted in Figure 13. In this scenario, stations STA1 and STA3 send traffic to the same destination (STA2). STA1 and STA3 are close enough to STA2 so that the packets sent can be correctly received if no collision occurs; however, STA1 and STA3 are also far enough from each other so that when one initiates

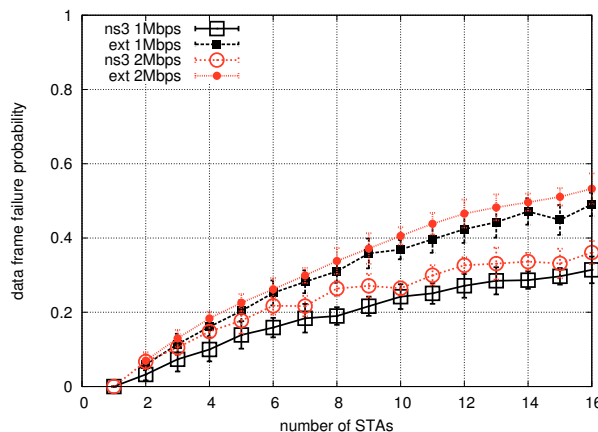


Figure 10: Saturation scenario: average frame transmission failure probability

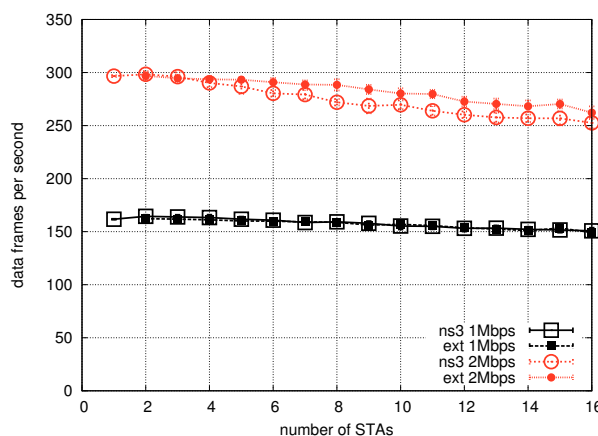


Figure 11: Saturation scenario: average number of successfully decoded transmission attempts per second

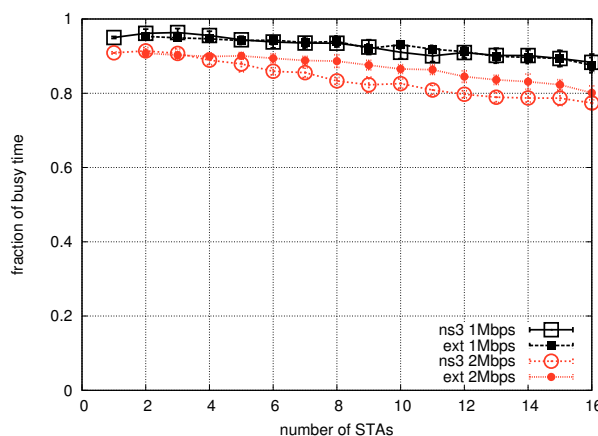


Figure 12: Saturation scenario: average channel busyness ratio

a transmission the other one will always detect the channel as idle. In other words, STA2 is within reception range of

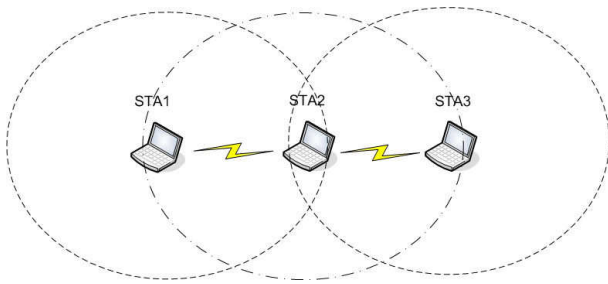


Figure 13: Hidden node scenario

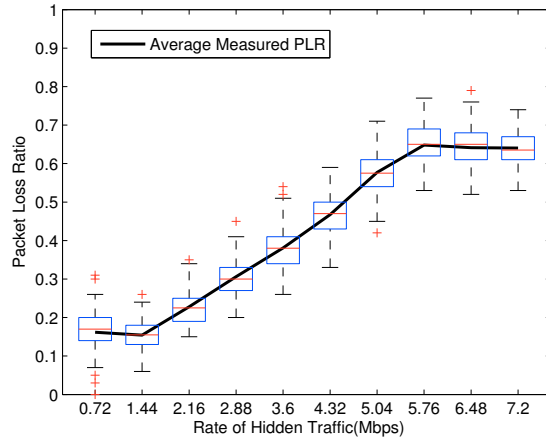


Figure 14: Hidden node scenario: Packet Loss Ratio using the first testbed configuration (STA3 with two virtual interfaces)

STA1 and STA3, but STA1 and STA3 are beyond Carrier Sense range of each other.

The three nodes in the scenario (STA1, STA2 and STA3) are each one equipped with one wireless card working in ad-hoc mode. All the nodes operate in the 2.412GHz band (i.e., channel 1). The DSSS PHY specification (802.11b) is used, and the PHY rate is fixed to 11Mbps. STA1 sends broadcast packets according to a Poisson process whose mean inter-packet generation time is 0.1 s. The payload length of the broadcast packets is 64 bytes. This is our reference flow and will be used to measure packet loss. STA3 sends a UDP flow at a constant bit rate (CBR). This CBR flow (sent by STA3) is not sensed by STA1, and is therefore called “hidden traffic” in the rest of this section. The data payload size for the hidden traffic flow is 1440 bytes. We ran different experiments varying the rate of the hidden traffic from 0.72 Mbps to 7.2 Mbps.

### 3.4.2 Simulator and testbed calibration

The hidden node scenario prepared using the ns3 simulator uses the Friis free space propagation model. In order to simulate a hidden node scenario, we set up the Energy Detection (ED) threshold to -90 dBm, we positioned STA1 and STA3 at a distance of 400 m from each other, and we placed STA2 in the middle, thus at a distance of 200 m from the other STAs. We use a transmission power of  $P_{tx} = 0$  dBm and a carrier frequency  $f = 2.412$  GHz.

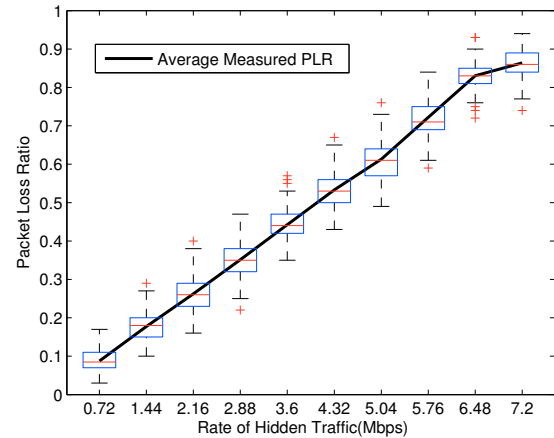


Figure 15: Hidden node scenario: Packet Loss Ratio using the ns3 simulator

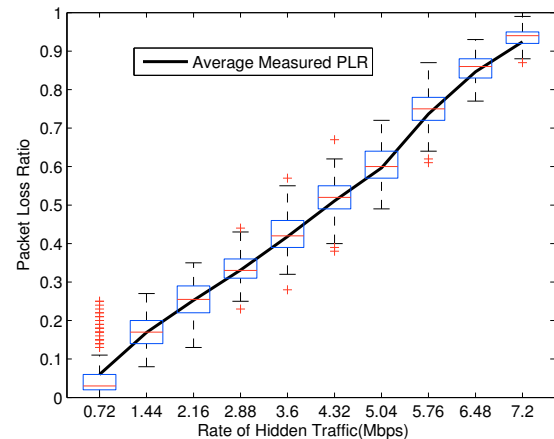
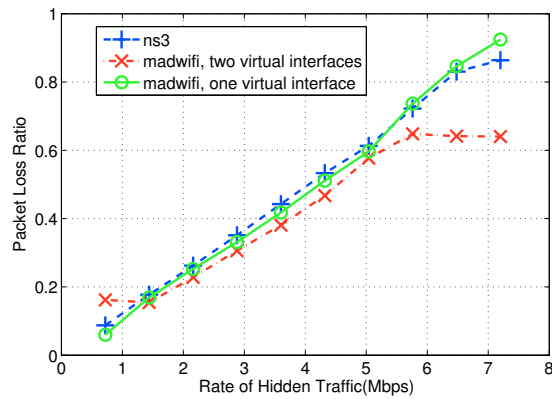


Figure 16: Hidden node scenario: Packet Loss Ratio using the first testbed configuration (STA3 with one virtual interface)

According to the Friis model, the received power in dBm is calculated as  $P_{rx} = P_{tx} + 10 \log_{10}(c/(4\pi fd))^2$ , where  $d$  is the distance in meters between the transmitter and the receiver, and  $c$  is the speed of light. This results in a value of  $P_{rx} = -86.110$  dBm for the communication from both STA1 and STA3 to STA2, which is above the ED threshold, and a value of  $P_{rx} = -92.131$  dBm from STA1 to STA3 and from STA3 to STA1, which is below the ED threshold.

In the testbed, we have forced a hidden node scenario by employing attenuators. Specifically, we place 60dB of attenuation between STA1 and STA2, and from STA2 to STA3. As for the other scenarios, RF cables were utilized to interconnect the three nodes in order to reduce problems derived from external interference and noise. In order to make sure that STA1 and STA3 were hidden from each other, we empirically tested that STA1 and STA3 were not able to communicate at a PHY rate of 1Mbps. CCA configuration of the wireless cards guarantees that this test is enough to guarantee hidden transmissions between STA1 and STA3.



**Figure 17: Hidden node scenario: comparison of the average Packet Loss Ratio obtained with different configurations**

To match the configuration and behavior of the hardware used in the testbed, we applied to the ns3 simulator the patches 1. and 2. which are described in Section 2.2. Moreover, we set the length of the Wifi MAC queue in ns3 to 200 packets, in order to match the default value used by the madwifi driver. Finally, RTS/CTS and packet fragmentation are disabled in both environments.

Additionally, an ns3 application was added in order to select the distribution at which data packets are sent. In this case, broadcast probe packets (our reference flow from STA1 to STA2) are sent following a Poisson distribution. In the EXTREME testbed, the MGEN application is configured to generate packets following a Poisson distribution.

### 3.4.3 Results

Figures 14–16 report the Packet Loss Ratio (PLR) that STA2 observes with respect to those packets that STA1 sends. In particular, the figures show the fraction of packets sent by STA1 that are not received in STA2. The higher the rate at which STA3 transmits (hidden traffic), the higher the value of the expected PLR.

Specifically, the figures illustrate the distribution of the PLR using boxplot representations of the measure obtained after 100 independent repetitions of each process. The PLR for each repetition is calculated after STA1 has sent a sequence of 100 packets. The figures also show (i.e. solid black line) the average value of the PLR obtained for each one of the hidden traffic rates tested.

In order to illustrate the benefit of using ns3 simulations to assess the correct functionality of a testbed deployment we have considered two different testbed configurations. In the first configuration, we set up the madwifi driver in STA3 with two virtual interfaces associated to the physical interface; one of the interfaces is used to monitor packets (in monitor mode) and the other is used to communicate (in sta mode). This causes that all frames received by the hardware at the physical interface are replicated within the driver in order to be passed to both virtual interfaces. In the second configuration, we set up the madwifi driver to use a single virtual interface to communicate (in sta mode); the traffic is then captured directly from this interface. We note that also in this case the packets are replicated, but by the operating

system rather than by the device driver.

Figures 14, 15 and 16 show the PLR measurements obtained in each one of the two cases explained. When the first configuration is used, the PLR measurements reach a maximum value of around 0.7. However, as figure 15 reveals, when the second configuration is adopted, the PLR reaches a maximum value of around 0.9. We suggest that the reason for this behavior is that packet replications done within the device driver, as implemented in the version of the madwifi driver that we used, limit the maximum packet rate at which a wireless station can transmit. In our case, this leads to a lower rate of the hidden traffic in the channel, and thus to lower values of the PLR.

Interestingly, the results obtained with the ns3 simulator (Figure 15) show a closer match to those gathered using the appropriate testbed configuration. This fact is better inferred from Figure 17, where we compare the average value obtained for the two experimental cases and the simulator. This illustrates that it is not always true that when there is a mismatch in the results between the simulator and the testbed, the simulator is the culprit. In fact, in this case the results obtained from the testbed prior to appropriate tuning were not reliable. We suggest that researchers should handle any difference observed between simulations and experiments with care, and judge them on a case-by-case basis by trying to understand the reason for the mismatch.

## 4. CONCLUSIONS

In this paper we presented a validation study of the ns3 model of the IEEE 802.11 MAC performed in a variety of scenarios by comparing the result obtained from the simulator with the measurements that we carried out on the EXTREME testbed. This study showed that, while in general there is a good qualitative agreement between simulator and testbed, in several cases there are noticeable quantitative differences. Furthermore, we found that the simulator is not always the culprit, but also the particular choice and configuration of the devices in the testbed can play an important role. We therefore conclude that a parallel tuning process, in which validation is repeated several times between the simulator and the testbed in order to refine their configuration, is advisable.

## 5. ACKNOWLEDGEMENTS

This work was supported in part by the Spanish Ministry of Science and Innovation under grant number TEC2008-06826/TEC (project ARTICO).

## 6. REFERENCES

- [1] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, “ns-3 project goals,” in *Proc. of WNS2*, Pisa, Italy, October 2006.
- [2] G. Pei and T. Henderson, “Validation of ns-3 802.11 b PHY model,” May 2009. [Online]. Available: <http://www.nsnam.org/~pei/80211b.pdf>
- [3] EXTREME Testbed ®. <http://www.cttc.es/en/projects/testbeds/project/EXTREME.jsp>.
- [4] J. Heidemann, K. Mills, and S. Kumar, “Expanding confidence in network simulations,” *IEEE Network*, vol. 15, no. 5, pp. 58–63, 2001.



- [5] P. Muessig, D. Laack, and J. Wroblewski, "An integrated approach to evaluating simulation credibility," U.S. Naval Air Warfare Center, 2001.
- [6] P. Barsocchi, G. Oligeri, and F. Potorti, "Validation for 802.11b wireless channel measurements," CNR-ISTI, Tech. Rep., 2006. [Online]. Available: <http://fly.isti.cnr.it/curriculum/papers/pdf/WiFi-measure-ISTI06.pdf>
- [7] —, "Frame error model in rural Wi-Fi networks," *Proc. of ICST WiOpt*, April 2007.
- [8] A. Di Stefano, A. Scaglione, G. Terrazzino, I. Tinnirello, V. Ammirata, L. Scalia, G. Bianchi, and C. Giaconia, "On the fidelity of IEEE 802.11 commercial cards," in *Proc. of IEEE WICON*, Budapest, Hungary, July 2005.
- [9] P. Fuxjager and F. Ricciato, "Collecting broken frames: Error statistics in IEEE 802.11 b/g links," in *Proc. of ICST WiOpt*, Berlin, Germany, April 2008.
- [10] J. Liu, Y. Yuan, D. Nicol, R. Gray, C. Newport, D. Kotz, and L. Perrone, "Empirical validation of wireless models in simulations of ad hoc routing protocols," *Simulation*, vol. 81, no. 4, p. 307, 2005.
- [11] E. Raymond, *The cathedral and the bazaar: musings on Linux and open source by an accidental revolutionary*. O'Reilly & Associates, 2001.
- [12] G. Bianchi, A. Di Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello, "Experimental assessment of the backoff behavior of commercial IEEE 802.11 b network cards," in *Proc. of IEEE INFOCOM*, Anchorage, Alaska, USA, 2007.
- [13] M. Portoles-Comeras, M. Requena-Esteso, J. Mangues-Bafalluy, and M. Cardenete-Suriol, "Monitoring wireless networks: performance assessment of sniffer architectures," in *Proc. of IEEE ICC*, Istanbul, Turkey, June 2006.
- [14] Wireless Extensions for linux. [http://www.hpl.hp.com/personal/Jean\\_Tourrilhes/Linux/](http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/).
- [15] The MadWifi project. <http://madwifi-project.org/>.
- [16] Host AP driver for Intersil Prism2/2.5/3, hostapd, and WPA Supplicant. <http://hostap.epitest.fi/>.
- [17] The Multi-Generator Toolset (mgen). <http://cs.itd.nrl.navy.mil/work/mgen/index.php>.
- [18] <http://code.nsnam.org/ns-3-dev/>.
- [19] <http://iptechwiki.cttc.es/Ns3-wifi-MAC-validation-using-EXTREME>.
- [20] Y. Xiao and J. Rosdahl, "Performance analysis and enhancement for the current and future ieee 802.11 mac protocols," *ACM SIGMOBILE Mobile Computing Communications Review*, vol. 7, no. 2, pp. 6–19, April 2003.
- [21] S. Garg and M. Kappes, "Can I add a VoIP call?" in *Proc. of IEEE ICC*, 2003.
- [22] A. G. Forte and H. Schulzrinne, "Distributed Delay Estimation and Call Admission Control in IEEE 802.11 WLANs," in *Proc. of IEEE ICC*, 2009.
- [23] "ITU-T Recommendation G.107, "The E-Model, a computational model for transmission planning"," March 2005.