Cross Fertilization Between Wireless Testbeds and NS-3 Simulation Models

Guillaume Kremer, Philippe Owezarski
CNRS, LAAS,
Université de Toulouse, LAAS

Pascal Berthou
CNRS, LAAS,
Université de Toulouse, LAAS, UPS

ABSTRACT

Network simulators are often used for their simplicity and cost regarding wireless networks. However, their realism is often criticized and their results challenged. The main concern comes from the modeling of the PHY and MAC layers. To assess the performances of these simulators and their models, the results of simulations are often compared with experimental results. However, the comparison methodologies used in these studies may introduce biases. This work focuses on accurately discovering and analyzing the reasons for the calibration problems or implementation bugs in simulators and experimental devices. For this purpose, we leverage the famous Root Cause Analysis (RCA) technique for comparing traces issued from different simulations and real experiments, that includes the study of the root causes of dissimilarities. Throughout the paper, our RCA-based method has been applied to detect and analyze a performance anomaly between NS-3 simulation and our lab wireless testbed when transmitting data over a WIFI 802.11 link. It especially details how low level traffic traces have been generated in both environments for similar scenarios, and how they can accurately be compared and their differences analyzed.

1 INTRODUCTION

Wireless networks are of essential importance nowadays. Users are more and more mobile and access the Internet thanks to mobile devices as laptops, smartphones or tablets. Even when staying at home, users want to get rid of wires. The importance of wireless communication is also aimed at rapidly growing with the emergence of promising upcoming applications involving many kinds of devices constituting the Internet of Things (IoT). However, the wireless networks and their physical layers for media access are technically very complex, and can appear as very fluctuant in terms of behavior, performance, and quality. The wireless networks are more often prone to errors and performance drops than wired networks. Designing wireless networks, compared to wired ones, then requires to accurately and deeply study all communication layers from physical to application, especially focusing on Physical (PHY) and Media Access Control (MAC) ones. It especially has to focus on signal propagation issues as interferences, collisions or distortions in signal propagation.

MSWiM'17, November 21–25, 2017, Miami, FL, USA
 ACM ISBN 978-1-4503-5162-1/17/11…$15.00.
DOI: http://dx.doi.org/10.1145/3127540.3127550

Simulators are very useful tools for first designing and evaluating networks, because they are very simple to use and require less investment than emulation or experimental platforms, in terms of time, and cost. Nevertheless, the results of wireless networks simulators are constantly criticized for their lack of realism, their PHY and MAC layers implementation being largely challenged by simulator users [6]. Therefore, the results of simulations are often compared with experimental results to calibrate simulators, and their PHY and MAC layer behaviors. For instance, OMNET++ performances in terms of throughput and latency estimation are very optimistic due to extreme simplification of MAC algorithms implementation [3]. Issues also appear with the NS-2 simulators that do not consider the operating systems delays [4] as well as at the PHY level with inaccurate signal propagation models in simulators, for instance for the loss models [11]. Tan et al. [5] pointed out anomalies on the measured signal power, due to simulators that are not considering the differences between antennas.

Our objectives focus on accurately discovering and analyzing the reasons for the calibration problems or implementation bugs of both the simulators and experimental devices. For this purpose, we leverage the famous Root Cause Analysis (RCA) technique for comparing traces issued from different simulations and real experiments, that includes the study of the root causes of dissimilarities. Our comparison approach is symmetric, and it can also be used for exhibiting and analyzing deficiency of wireless protocols implementation on the wireless devices. By exhibiting their root causes, it helps network designers to correct either simulator models, or wireless device implementation.

The rest of the paper is as follows: first, the paper describes the platform for wireless communication experiments (section 2), and that is the source of all synchronized traces captured at layers 1, 2, and 3. All together, they constitute an essential database for in deep and efficient wireless network behavior analysis. It especially focuses on studying the 802.11 protocol, and comparing results with the ones of the NS-3 simulator. Section 3 details related experimentation scenarios. Section 4 shows how traces from simulators and real experiments must be paired to avoid biases. Section 5 then presents the RCA method and how it is used for calibrating NS-3 simulation models and experimental devices, as well as for detecting implementation bugs. Finally, section 6 concludes the paper

2 EXPERIMENTAL PLATFORM AND SIMULATOR DESCRIPTION

Our WIFI testbed is designed inside an anechoic room with two WIFI nodes. The nodes are controlled through a wired network to avoid interference with the wireless communication. The nodes are Avila-GW2348-4 gateway platforms and run a Linux OpenWrt

OS. The boxes have an Intel Xscale processor, 64 MB of SDRAM and 16MBytes of Flash memory. The WIFI network controllers are based on the AR5414 chipset from Atheros which uses the ath5k driver. The ath5k driver is open-source and well documented.

The configuration of the wireless interfaces is done in promiscuous mode to capture any packets sensed by their antenna. The packets are captured at the MAC layer using the PCAP library. The packets contain data from link to application layers, such as the 802.11 channel number, the type of frame at the MAC layer, or packet size at the network layer. We modified the ath5k drivers of the OpenWrt OS to permit, when possible, the propagation of packets with frame check sequence (FCS) errors to the upper layers.

Not to overload the WIFI node processors, UDP traffic generation and reception are made on dedicated machines which are connected to the nodes by high performance Ethernet connections. The WIFI nodes are configured as WIFI bridges and are only responsible for MAC and PHY related operations (*i.e.* 802.11 retransmission, FCS checking, ...) as well as for PCAP captures. Tests made on the test bench do not show any impact of this configuration on the accuracy of the data [7].

A WIFI sniffer device (similar to the WIFI bridge devices) is connected to the WIFI bridge 1 antenna by means of a power splitter. The sniffer is set in monitor mode and is totally passive (it does not send any frame and therefore does not perturb communications). In that configuration, the WIFI sniffer is able to capture the frames transmitted by the WIFI bridge 1. All equipment have their clock accurately synchronized by using a NTP server on a dedicated wired connection.

NS-3 is a recent network simulator commonly used to simulate wireless networks. NS-3 has been selected as it is the most recent version of the NS simulator family, a family of generic network simulator widely used in the network research and engineering community. Our configuration is set to use the YANS (Yet Another Network Simulator) models that define the PHY and MAC layer of WIFI nodes [8].

In the NS-3 simulator, nodes can be configured to capture 802.11 and IP traffic into PCAP files that can be analyzed in the same way as traces gathered on the experimental testbed.

However, NS-3 does not consider signal attenuation (cf. figure 1(b)), and thus a method is required to be able to compare NS-3 simulator with experimental testbed results. This involves noise injection capabilities. Similarly as on the experimental platform, this noise must be injected to disturb the receptor of the wireless link. To the best of our knowledge, no solution is available yet to inject noise during frame reception on the NS-3 simulator. Therefore, the YANS module has been modified to add this capability.

2.1 Frame reception in the YANS model

In our configuration, the first steps of 802.11 frames reception are carried by the PHY methods of the YANS module. These steps determine if the frame is received with or without any error. The reception of a frame p begins by the evaluation of the signal strength $S(p,t)$. This value is calculated using the Friis law from the transmission power of the frame and the traveled distance. The signal to noise plus interferences ratio for that frame p, noted $SNIR_A(p,t)$, is then obtained with equation 1. In the SNIR relation, N_f and N_i are respectively the value of the electromagnetic noise floor and the sum of all the signal powers received by the antenna at the time of the frame reception. The N_f value is constant and specific to the simulated circuit.

$$SNIR_A(p,t) = \frac{S_p t}{N_i(p,t) + N_f} \tag{1}$$

This $SNIR_A$ value will then be used by the YANS module to determine if a frame contains any error or not: a frame received with a lower SNR will have a bigger error probability.

2.2 Modifications of the YANS model to support noise injection

According to equation (1), the error probability during packet reception is affected by the strength of the received signal, i.e. the cumulative power of all the interferences and the value of the constant noise floor. To implement the experimental protocol and therefore inject an arbitrary noise power during frame reception, the reception process is modified. Another noise source, N_g, is therefore added to the denominator of the SNIR computation according to equation (2).

$$SNIR_B(p,t) = \frac{S_p t}{N_i(p,t) + N_f + N_g(t)} \tag{2}$$

To generate the N_g values, a new class specialized in generating random noise has been created. This class implements a method *Generate* which is responsible for producing the N_g values. During frame reception, this method is called by another method called *InterferenceHelper::CalculateSnr* which uses *Generate* return value to compute equation 2.

Inside the method *Generate*, N_g values are generated with the Box-Muller algorithm [9] already implemented in NS-3. The Box-Muller algorithm produces normally distributed random numbers. The mean and variance of the distribution are respectively equal to 0 and N_0. The value of N_0 is then defined for each simulation to set the value of the injected noise level.

3 SIMULATION/EXPERIMENT SCENARIOS AND GENERATED TRACES

This section aims at introducing the scenario that serves as the illustrative example in the whole paper. It especially explains how traces are generated to cover the full range of possible situations. It insists on the full set of parameters that are of significant importance for the proposed methodology.

The scenario selected for having all kinds of traffic traces with a very wide range of performance issues consists in sending traffic on an unidirectional 802.11g link, while noise perturbations are generated and injected to the receptor.

The configuration of both environments are identical and detailed in table 1. TCP is used more than UDP but TCP is also a much more complex protocol than UDP. Therefore, not to increase the complexity of these first analyses, the transport protocol used in that study is UDP. For the same reasons, most of the UDP parameters such as the packet size or the throughput are fixed. Moreover, the 802.11 rate control algorithm is disabled and the 54 Mbps mode is used exclusively for the transmission of the data frames. The

maximum number of short retries is increased and set to 14 instead of 7 which is the suggested value in the 802.11 standard. Indeed, in the preliminary measurements made to set up our test bed we tested both values and noticed that using 14 would give more interesting results in term of loss at the transmitter (metric referred as DROP).

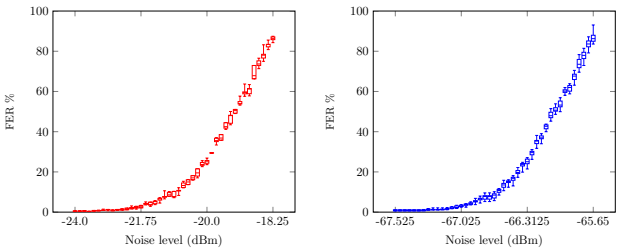
In both cases, noise power values are selected to achieve a full range of frame error ratio, *i.e.* the ratio of frames received with at least one bit-error varies from approximately 0% to 100% in the experimental and simulation datasets. The experimental and simulation injected noise ranges are linear (*i.e.* the step between two consecutive values is constant). To obtain the specific noise ranges detailed in table 1, we conducted a preliminary set of FER (Frame Error Rate) measures in experimentation and simulation using the same settings and instrumentation. However, in this preliminary set, the noise values used in simulation and experimentation respectively ranged from -75 to -40 dBm and from -75 to -15 dBm. The step between each noise level in this preliminary set was 1 dBm in experimentation and 0.1 dBm in simulation. The measured FER values allowed us to restrict the noise range in our final test set from -67.7 to -65.5 dBm in simulation and from -24.0 to -18.0 dBm in experimentation. The difference observed between these two ranges can be explained by the way noise is injected in the two environments. In the simulation case, the noise is injected directly during frame reception. In the experimentation case however, the noise is transmitted over the air to the receiver by a directive antenna, the value of the experimental noise being the peak amplitude of the noise produced by the signal generator. Therefore, in the experimental case, the noise has to suffer attenuation losses in the different mediums (cable and air) before it can reach the receiving antenna.

4 SIMULATION AND EXPERIMENTAL TRACES PAIRING FOR THE COMPARISON

The simulation and testbed traces used in our comparison methodology have been generated for the same scenario in similar conditions. To avoid biases during the comparison stage, it is essential to pair the two traces in order to make related events in the two traces correspond.

Indeed, data issued from the simulation model and the experimentations could be slightly dissimilar and must be mapped to each other to be compared. Figures 1(a) and 1(b) show the difference of frame error rate between a NS-3 simulation and the related experiment in our testbed. Curves are similar. However, they are not centered on the same noise level value. This difference has been explained in section 3.

To pair each trace of the experimental dataset with one trace selected from the simulation dataset, we propose to apply a combination of easy computable criteria w.r.t. the goals of the study. For instance, this paper focuses on wifi loss behaviour. The pairing will be made according to the frame error ratio and the loss patterns. These two methods are detailed below. While the first pairing method using the FER is well suited in the scenario used in our study, the second method using loss patterns can be more interesting in other more complex situations.



(a) Experimentations (b) Simulations

Figure 1: FER measures regarding the different noise levels in experimentations 1(a) and simulations 1(b). The values are for each trace the 1st and 3rd quartile, as well as the min and max values. The central value displayed is the median.

4.1 FER based traces pairing

Frame errors are artificially generated at the receiver side with signal perturbations to test the wireless protocol behavior. The quantity of frame errors depends on the value of the injected noise. On Figures 1(a) and 1(b) the y-axis represents the medium frame error ratio (FER) experienced for a fixed noise level value. Each point summarizes the loss rate measured in a thousand frames trace.

As shown on table 1, the noise value is different in both environments (simulation, experimentation), however the FER resulting from these values is common across both sets of traces. It ranges from 0 % to 100 %. Consequently, the FER can be used as a common pairing metric. Furthermore, the FER may impact the value of other metrics such as the performance of the link, and therefore comparing traces with different FER may be inefficient since it will result in biases during comparison. Because of this, the median of the FER value is used to pair the experimental and simulation traces. Hence, for each experimental trace, the pairing process associates a specific simulation trace. The selected simulation trace for each experimental trace is the one with the closest measured FER median value.

Therefore, given two traces x and y belonging respectively to the experimental and simulation datasets (respectively named X and Y), given $z = |Median(FER_x) - Median(FER_y)|$, traces x and y are eligible to be paired together if and only if $\forall t \in Y \setminus \{y\}$, $|Median(FER_x) - Median(FER_t)| \geq z$.

If multiple simulation traces are eligible to be paired with one experimental trace, the choice among the simulation traces is made arbitrarily. However, given the diversity measured on the FER values, this case is unlikely to happen. Furthermore, given the configuration of the simulator, two simulation traces sharing the same FER median should be quite similar and should not result in major comparison differences.

4.2 Loss pattern based traces pairing

The pairing process associates traces according to their median FER. However, although the median FER of the two associated traces are similar, their error characteristics and patterns can be different. These differences have an impact on communications. For

Table 1: Detail of the settings used for the experimentations and simulations.

Setting	*Notation*	*Experimentations*	*Simulations*
Transmit power	P_{ptr}	10 dBm	
UDP Throughput	P_{DUDP}	7 Mbps	
Packets size	P_{TP}	1472 B	
Noise power range (linear)	P_{BR}	[-24.00;-18.00] dBm	[-67.7;-65.5] dBm
Corresponding generated FER range		[0%;100%]	
Data frame rate	P_{DT}	fixed to 54 Mbps	
Control frame rate	P_{DC}	fixed to 24 Mbps	
802.11 standard	P_{MAC}	802.11g-DCF-No-QoS-Long Slot (20 μSec)	
Maximum number of consecutive 802.11 retries	P_{RETR}	14	
Distance between sender and receptor	P_{DIST}	2 m (= 6.562 ft)	
Propagation environment	P_{ENV}	Anechoic room	Free space (Friis model)

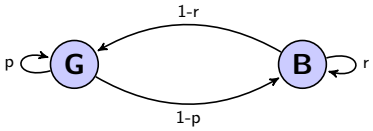


Figure 2: The Gilbert-Eliott model.

example, the BEB (Binary Exponential Backoff) algorithm, which exponentially increases the contention window size between each successive retries, can have significant consequences on a link capacity if this link experiences long bursts.

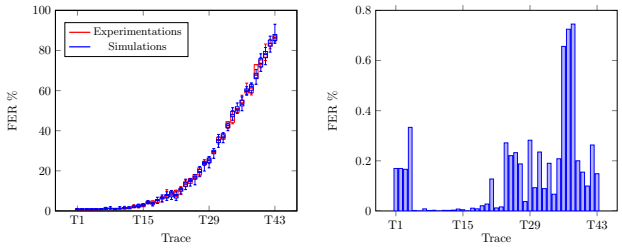
The Gilbert-Eliott loss model displayed on figure 2 is used to model error patterns over data transmission channels. This model is based on a 2-states Hidden Markov Model. The state labeled G (good) corresponds to the successive reception of error-free packets (also called an *interval*) whereas the state labeled B (bad) corresponds to the successive reception of erroneous packets (also called a *burst*).

p and r are the respective transitions associated with the transition from state B to state G and vice versa. The stationary probabilities associated with state G and B are respectively noted π_G and π_B. The channel memory is defined as the μ parameter.

Hence, additionally to the pairing process which associates traces according to their FER median values, the comparison of the μ, π_B and π_G parameters on the paired traces guarantees that these traces are similar with regards to their error patterns. These two methods limit biases during the following of the comparison process. This is demonstrated in the following section.

4.3 Traces pairing and validation

Figure 3 shows the resulting pairing. The accuracy of the method allows the association of traces that have a median FER with less than 1% difference between each other. The Gilbert-Eliott coefficients π_B, π_G and μ have been calculated for each of these traces.



(a) Experimentations and simulations FER after trace pairing

(b) Differences between paired FER

Figure 3: FER values after trace pearing and absolute difference between FER.

There was no difference between the error patterns measured in experimentation and simulations. Moreover, the μ coefficient evolution computed on the traces are similar in both environments and their value is close to 0. This demonstrates the independence of the generated errors.

For a better comprehension, in the rest of the text, the paired traces will now be noted according to table 2 and prefixed by a T character.

5 BEHAVIOR DISSIMILARITIES DETECTION AND ANALYSIS BETWEEN SIMULATORS AND EXPERIMENTAL TESTBEDS

Our method for detecting behavior dissimilarities between simulators and/or testbeds, and analyzing their causes takes advantage of the RCA model. RCA is a diagnosis method that identifies root causes of problems and symptoms detected on a monitored system. It specifically relies on the expertise of network administrators and architects. RCA models have been successfully used in [12, 13].

Table 2: Notation of the traces after the pairing process. For commodity reasons, only a small part of the associations are shown.

New notations		$T1$	...	$T15$	...	$T29$	...	$T43$
Respective	experimentations	-24.0	...	-21.75	...	-20.0	...	-18.25
noise level (dBm)	simulations	-67.525	...	-67.025	...	-66.3125	...	-65.65

5.1 The RCA model and its related deduction tree

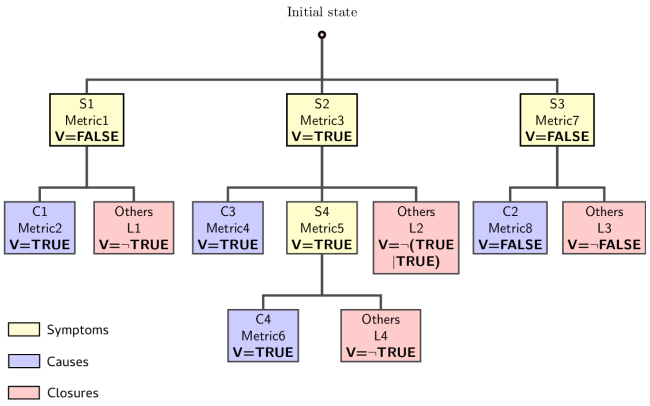


Figure 4: Example of a RCA tree for comparison between traces. V values displayed on the nodes are sample values set arbitrarily and used to present how the analysis process propagates down the RCA tree.

Any RCA model requires a knowledge base which describes the relation between symptoms and causes. A deduction tree [10, 14] can be used for that purpose. Figure 4 presents an example of such tree.

A deduction tree is composed of symptoms and causes nodes which are respectively associated with one or several metrics of the monitored systems. These associations take the form of a logical expression constituted with these metrics. A hierarchical relationship between two nodes represents a causal relationship between the metrics which are respectively associated to the nodes in the modeled system. The symptoms are non terminal nodes since they must be tracked to its or their possible cause(s). At the opposite, cause nodes are terminal nodes. However, a cause can have sibling nodes (symptoms or causes) which are issued from the same parent node.

When the tree is applied on the data of the monitored system, the logical expression associated with each node is evaluated to a boolean value. The application of the tree starts at the initial node and propagates towards terminal nodes. A node is evaluated if and only if its parent is evaluated to *true*. For that reason, high level metrics (e.g. performance metrics or outputs of the monitored system) should be associated with nodes located at the upper levels of the tree since they capture more symptoms than metrics with small radius (e.g indicator metrics or parameters of the system). The goal of such arrangement is to maximize the visibility of the system at the upper levels of the tree and reduce the spectrum of causes while going down the tree levels.

The model of a specific system can be incomplete (either for a lack of knowledge, modeling or measurement capabilities). For that reason, a third type of node is used: closures. Closures allow the inference of the possible cause of a symptom even if all the values of its sibling nodes are evaluated to $false$. The value associated to a closure is therefore automatically set to the following value: if $V = \{V_1, ..., V_n\}$ is the set of logical values of the S symptom siblings, then the value of the associated closure L equals $\neg(V_1 \mid ... \mid V_n)$, where $\mid$ is the exclusive disjunction. At the end of the tree application, if a closure is evaluated to *true*, further analysis may be needed to identify the exact cause of a detected symptom.

For illustration purposes, the propagation of the analysis process in the tree described on figure 4 is the following:

(1) Level 1 nodes $S1$, $S2$ and $S3$ are first evaluated. Only, $S2$ is evaluated to $true$.
(2) Node $S2$ has 3 possible causes: $C3$, $S4$ and $L2$. $L2$ being the closure of $S2$. $C3$ and $S4$ are evaluated to $true$, therefore $L2$ is evaluated to $false$.
(3) The sub-tree issued from $S2$ is evaluated. The value of $S4$ and $C3$ are $true$.
(4) Node $C4$ is evaluated to $true$.

As a conclusion, anomalies have been detected on nodes $S2$, $C3$, $S4$ and $C4$. It follows that the possible root causes are $C3$ and $C4$.

5.2 The RCA deduction tree for comparing the behaviors of our NS-3 and testbed example

The specific RCA model defined for our environments (simulators, testbed) comparison is shown in figure 5. This model has been defined according to our expertise in the wireless domain and our measurement capabilities. The performance metric is the IP throughput of the wireless link. Other metrics are related to timing, errors patterns, or configurations. The 1st node of the tree compares the Bw values computed on the simulation and experimental datasets.

The second level of the tree uses metrics whose variations are known to directly impact the throughput on the link. These metrics are FER and $DROP$ which are then respectively associated with nodes $S2$ and $S5$. IPERF (the tool used to generate traffic) and packet size parameters have also a direct impact on the measured throughput. Therefore, they need to be checked at that level of the tree.

To find the cause of reception errors dissimilarities, node $S5$ is linked to node $C2$ which compares the values of the Gilbert-Eliott (GE) coefficients in the two datasets. As explained in the

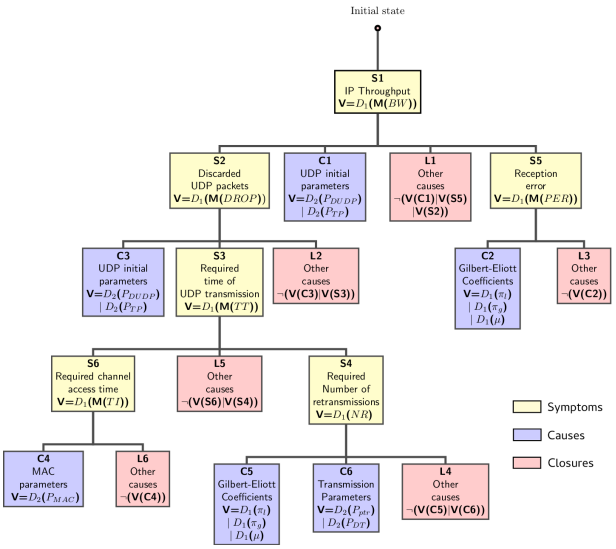


Figure 5: Deduction tree for trace comparison. The M(a) notation used on the nodes is a short notation to express the median of the metric a. For example, M(BW) is the median of the throughput values measured every second on each trace.

previous sections, the GE coefficients can be directly linked with the reception errors.

A large part of the tree is responsible for finding the cause of congestion dissimilarities in the experimental and simulation datasets. This subtree is issued from node $S2$. The causes of congestion are related to limited resources on the wireless link. This resource limitation may be due to longer transmission delays (node $S3$) which may then be caused by longer medium access delays (node $S6$) or harsher medium conditions (node $S4$). For that last node, the number of frame retransmissions may be subordinated to the transmission parameters of the 802.11 frame (e.g. transmission power) and to the GE coefficients. Finally, in our single-link configuration, the channel access time associated with node $S6$ is essentially impacted by the MAC parameters, and hence by the link between nodes $S6$ and $C4$.

5.2.1 Nodes evaluation. The nodes evaluations are made using the boolean functions D_1 and D_2 described below. These functions compare values measured both on the associated traces obtained in simulations and on the testbed. If $Sim(a)$ is the value of a measure a in simulation and $XP(a)$, the value of the same metric obtained experimentally, functions D_1 and D_2 are respectively defined by equations i and ii.

$$\text{(i)} \quad D_1(a) = \begin{cases} true & \text{if } \mid Xp(a) - Sim(a) \mid > t_a \\ false & \text{otherwise} \end{cases}$$

with t_a the threshold associated with metric a

$$\text{(ii)} \quad D_2(a) = \begin{cases} true & \text{if } (Xp(a) \neq Sim(b)) \\ false & \text{otherwise} \end{cases}$$

The $D1$ function requires the definition of a different threshold value for each metric of the tree. The definition of the threshold values have a great importance on the comparisons since they will affect their output and the efficiency of the tree. If the threshold values are too high, all the comparisons will be evaluated to $false$ and no dissimilarities will be found. At the opposite, if the threshold values are to low, the comparison will always be evaluated to $true$. Setting the right values can be difficult and requires some knowledge and experience on the measured environments, the data and the scenario under test. A solution would be to automatically find the best values by doing several consecutive comparisons with different threshold values and select the best configuration, using a dichotomy based method. Because of space limit, the demonstration of such process will not be detailed here. In our configuration and for the example application, we set the thresholds to the following values:

- For node $S1$, the threshold is noted τ_{bw}. It corresponds to a difference of 500 kbps observed on the median of the throughput values measured on the traces. This value has been set to avoid false-positive due to imperfections during the measurement process.
- The threshold τ_{DROP} used by node $S2$ is set to 42 packets/s. Given the packets size, this value corresponds to the 500 kbps limit set for node $S1$.
- The threshold τ_{FER} threshold specifies that the maximum difference allowed for FER is fixed to 1%. This value corresponds to the maximal accuracy available with the pairing algorithm (see part 4.3).
- The threshold value τ_{tt} is an approximation of the time required to send 42 1470-bytes frames in the 54 Mbps PHY datarate. This time corresponds to the sum of the medium access time, the acknowledgement reception and the flying time such as: $42 * DIFS + 42 * SIFS + \frac{42*1470*8}{54*10^6}$. The backoff time is ignored here. With the standard DIFS (DCF InterFrame Space) and SIFS (Short InterFrame Space) values set to 28 μs and 10 μs [1], τ_{tt} value is equal to 0.0091s.
- τ_{TI} is the threshold fixed for the interarrival time of packets. It corresponds to the theoretical difference of interarrival packets between flows of throughput respectively equal to 7 Mbps and 6.5 Mbps (according to the 500 kbps threshold), *i.e.* $\tau_{TI} = 0.0002$ s.
- τ_{NR} is the threshold difference used for the number of retransmissions. This value is set to 1, *i.e.* the values are considered different if their median number of retransmissions is greater than 1.

5.3 Application of our RCA model on gathered experimental and simulation data

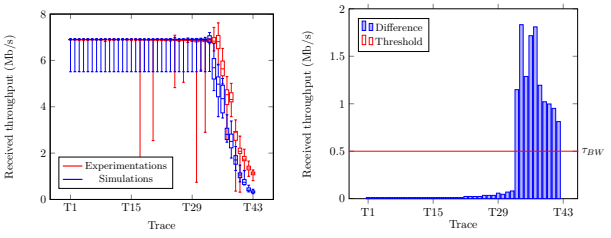
The tree model shown on figure 5 is applied on gathered traces. The nodes evaluation during this application is presented on table 3.

A first statement concerns the value of the nodes associated with the parameters comparison (UDP parameters, MAC parameters, ...). As described in section 2, the initial experiment and simulation parameters are the same. Consequently, the evaluation of nodes $C1$, $C3$, $C4$ and $C6$ which make the comparisons of these parameters are $false$. Similarly, if nodes $C2$ and $C5$ which compare the GE

Table 3: Nodes evaluation during tree traversal.

Node	*Logical expression*	*Evaluation Result*
S1	D_1(M(BW))	$true$
S2	D_1(M($DROP$)	$true$
C1	$D_2(P_{DUDP})\mid D_2(P_{TP})$	$false$
L1	¬(V(C1)|V(S5) |V(S2))	$false$
S5	D_1(M(FER))	$false$
C2	¬(V(C1)|V(S5)|V(S2))	(not evaluated)
L3	¬(V(C2))	(not evaluated)
C3	$D_2(P_{DUDP}) \mid D_2(P_{TP})$	$false$
L2	¬(V(C3)|V(S3))	$false$
S3	D_1(M(TT))	$true$
S6	D_1(M(TI))	$true$
L5	¬(V(S6)|V(S4))	$false$
S4	$D_1(NR)$	$false$
C4	¬(V(C1)|V(S5)|V(S2))	(not evaluated)
L4	¬(V(C5)|V(C6))	(not evaluated)
C6	$D_1(\pi_l)\mid D_1(\pi_g)\mid D_1(\mu)$	(not evaluated)
C4	$D_2(P_{MAC})$	$false$
L6	¬(V(C4))	$true$

coefficients are computed, their value is false given the results illustrated in part 4.3. Finally, the value of $S5$ is also $false$ given the accuracy of the pairing process which is more important than the τ_{DROP} threshold set to 1%.



(a) Calculated values

(b) Differences between simulations and experimentations data

Figure 6: Data for BW metric. Graph 6(a) shows the median, the 1st and 3rd quartile, as well as the min and max values for the BW metric measured on the simulation and experimentation traces. Graph 6(b) shows the differences measured for $M(BW)$ between the paired traces of the simulation and the experimental datasets.

Node $S1$, which is the first visited node, compares the measured throughput in both environments. These values and the differences between the paired traces from the experimental and the simulation dataset are respectively shown on figure 6(a) and 6(b). On figure 6(a), a slight difference is observed between the experimental and the simulation values. These values stay stable for the lower noise values and correspond to less than 2,30% of FER (traces $T1$ to $T17$). From trace $T18$, the noise level is high enough to affect the throughput, a slight difference is measured between the experimentation and simulation traces. This difference increases significantly after trace $T33$ (25% of FER) and reaches its maximum value (2 Mbps) for trace $T35$. On figure 6(b), the difference between the traces is greater than the τ_{bw} threshold which is the threshold associated to metric Bw. Therefore node $S1$ is evaluated to $true$.

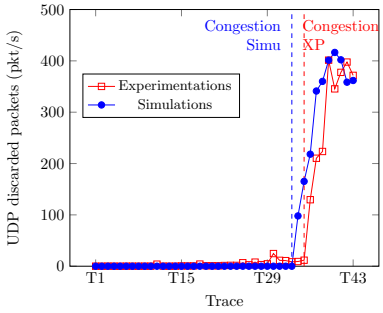


Figure 7: Values of $M(DROP)$ on the paired traces.

At the 2nd comparison level, nodes $S2$, $C2$, $L1$ and $S5$ are visited. Node $S2$ is evaluated to $true$. The values for the associated median metric $M(DROP)$ are presented on figure 7. The loss differences follow the same pattern as the one observed on the throughput values. Data on that figure demonstrates that the congestion caused by the medium saturation happens for lower perturbation in simulation. On both environments, the number of losses per seconds reaches a plateau at 400 packets per second in both environments. When this plateau is reached, the differences calculated between the experimental and simulation traces correspond to 150 lost packets per second; this is similar to the throughput difference of 2 Mbps observed during $S1$ evaluation.

To determine the causes of loss differences observed between the two environments, node $S3$ is evaluated. This node compares the differences between transfer delays in experimentations and simulations. This node is evaluated to $true$. These transport times differences may be caused by the medium access times. This hypothesis is confirmed by the respective evaluation of nodes $S6$ and $S4$ set to true and false. To conclude this root cause detection, the closure node $L6$ is evaluated to $true$ since node $C4$ is evaluated to $false$.

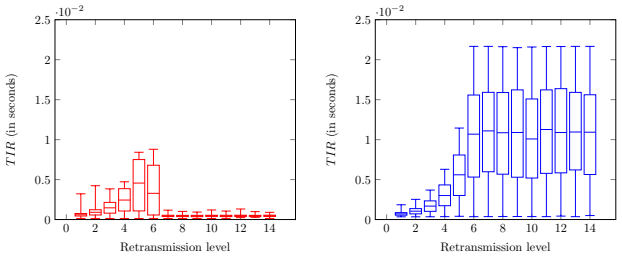
Hence, the root cause of these performance differences is traced to channel access times. Since the MAC parameters are the same in both environments, the implementation of the MAC access methods may be responsible of these differences. To accurately demonstrate the implication of these methods further analysis is needed.

5.4 Results analysis

The Binary Exponential Backoff algorithm (BEB) is one of the main factor acting on the channel access time. Before each 802.11 frame transmission, nodes have to randomly pick a transmission slot. The number of slots available to a specific node is limited according to the current number of retransmissions of the frame and managed

by the BEB algorithm. The progression of this value, called Cw is given by equation 3. From this equation, Cw follow a geometric progression between the 1st and the 6^{th} level of retransmission of the frame. At the 6^{th} level, Cw has reached its maximum value (1023 slots), and therefore will not be increased during the next level of retransmission. When the frame is successfully transmitted, the Cw values is reset to its initial value Cw_0.

$$Cw_{n+1} = max((CW_n + 1) * 2 - 1, 1023) \quad \text{with } Cw_0 = 15 \tag{3}$$



(a) Experimentations (b) Simulations

Figure 8: *TIR* values measured on the experimentations 8(a) and simulation data 8(b). Values are presented with regards to the level of retransmission. For each trace, the figures depict the min, median and max values as well as the 1st and 3rd quartiles.

The *TIR* (Retries Inter-arrival Time) metric corresponds to the channel access time measured in experiments and simulations according to the retransmission level of the frames. The experimental and simulation data for this metric are respectively presented on figures 8(a) and 8(b). In the simulation case, the values follow a geometric progression from the 1^{st} to the 5^{th} level of retransmission. From levels 6 to 14, a plateau is reached and the values stay the same, and close to 1.1 ms. At the opposite, in the experimental case, the *TIR* values increase from the 1st to the 5^{th} retransmission. However between levels 5 and 6, the median of the values decreases. From levels 7 to level 14, the values stay steady and lower than 0.1 ms.

From these statements, only data obtained in simulations seem close to the theoretical results. At the opposite, the experimental implementation of the BEB algorithm in the Atheros chipset seems unexpected for the retry values tested. These results concord with those obtained in [2] which exhibits the unexpected implementations of the backoff mechanism in several WIFI cards. This phenomenon that appears at high levels of retransmission seems to concord with throughput differences observed on figure 6(a). This phenomenon starts at trace $T30$ which corresponds to nearly 20 % of FER.

6 CONCLUSION

This paper presents a full contribution for WIFI network engineering. It includes the evaluation and assessment of WIFI (protocol design, implementation, ...) thanks to the classical NS-3 network simulator, and a lab wireless network testbed. The paper then presents a methodology for comparing the behaviors of NS-3 and a WIFI testbed. This methodology allows the detection of dissimilarities, but also the analysis of their root causes. For that purpose, it takes advantage of the famous RCA method, especially showing how designing and using the RCA comparison tree. To illustrate and validate our detection and analysis methodology, the wireless experimental testbed has been set-up in the framework of an anechoic room. An essential dataset has been built for accurately analyzing the behavior of WIFI networks. This dataset is publicly available.

The paper demonstrates the efficiency of this methodology by analyzing the behavior of our testbed compared to its implementation under the NS-3 simulator.

REFERENCES

[1] *802.11-2012 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* Technical Report IEEE Std 802.11™-2012. IEEE-Inst.

[2] G Bianchi, A Di Stefano, C Giaconia, L Scalia, G Terrazzino, and I Tinnirello. 2007. Experimental Assessment of the Backoff Behavior of Commercial IEEE 802.11b Network Cards. In *26th IEEE International Conference on Computer Communications (INFOCOM).*

[3] UM Colesanti, C Crociani, and A Vitaletti. 2007. On the Accuracy of Omnet++ in the Wireless Sensornetworks Domain: Simulation vs. Testbed. In *PE-WASUN'07: Proceedings of the 4th ACM Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor,and Ubiquitous Networks.* ACM, New York, NY, USA, 25–31. http://doi.acm.org/10.1145/1298197.1298203

[4] S Ivanov, A Herms, and G Lukas. 2007. Experimental Validation of the ns-2 Wireless Model using Simulation, Emulation, and Real Network. In *Communication in Distributed Systems (KiVS), 2007 ITG-GI Conference.* 1–12.

[5] T Kefeng, D Wu, A Chan, and P Mohapatra. 2010. Comparing simulation tools and experimental testbeds for wireless mesh networks. In *World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a.* 1–9.

[6] S Khan, B Aziz, S Najeeb, A Ahmed, M Usman, and S Ullah. 2013. Reliability of network simulators and simulation based research. In *IEEE 24th International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC).*

[7] G Kremer, P Owezarski, P Berthou, and G Capdehourat. 2014. Predictive Estimation of Wireless Link Performance from Medium Physical Parameters Using Support Vector Regression and k-Nearest Neighbors. In *TMA'14: Proceedings of the 6th international workshop on Traffic Monitoring and Analysis.*

[8] M Lacage and TR Henderson. 2006. Yet Another Network Simulator. In *WNS2'06: Proceeding from the 2006 Workshop on Ns-2: The IP Network Simulator.*

[9] AM Law and WD Kelton. 2000. *Simulation modeling and analysis.* McGraw-Hill. http://books.google.fr/books?id=QqkZAQAAIAAJ

[10] L Li, M Li, R Fan, and L Li. 2010. A fault diagnosis method based on decision tree for wireless mesh network. In *Communication Technology (ICCT), 2010 12th IEEE International Conference on.* 231–234.

[11] C Phillips, D Sicker, and D Grunwald. 2013. A Survey of Wireless Path Loss Prediction and Coverage Mapping Methods. *Communications Surveys Tutorials, IEEE* 15, 1 (First 2013), 255–270.

[12] M Siekkinen, G Urvoy-Keller, EW Biersack, and D Collange. 2008. A Root Cause Analysis Toolkit for TCP. *Comput. Netw.* 52, 9 (June 2008).

[13] M Siekkinen, G Urvoy-Keller, EW Biersack, and T En-Najjary. 2005. Root Cause Analysis for Long-lived TCP Connections. In *CoNEXT'05: Proceedings of the 2005 ACM Conference on Emerging Network Experiment and Technology.* ACM, New York, NY, USA, 200–210.

[14] X Xiang-Hua, Z Biao, and W Jian. 2009. Tree Topology Based Fault Diagnosis in Wireless Sensor Networks. In *Wireless Networks and Information Systems, 2009. (WNIS). International Conference on.* 65–69.