



Configuration Schemes and Assessment of NS3 Models Using a Wireless Testbed

Guillaume Kremer, Denis Carvin, Pascal Berthou, Philippe Owezarski

► To cite this version:

Guillaume Kremer, Denis Carvin, Pascal Berthou, Philippe Owezarski. Configuration Schemes and Assessment of NS3 Models Using a Wireless Testbed. 2013. hal-00817453

HAL Id: hal-00817453

<https://hal.science/hal-00817453>

Preprint submitted on 25 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Configuration Schemes and Assessment of NS3 Models Using a Wireless Testbed

Guillaume Kremer^{1,2b}, Denis Carvin^{1,2a}, Pascal Berthou^{1,2b}, Philippe Owezarski^{1,2c}

¹CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France

²Univ de Toulouse, ^bUPS, ^aINSA, ^cLAAS, F-31400 Toulouse, France

Email: {kremer, carvin, berthou, owe}@laas.fr

Abstract—In the field of Networking and Telecommunications, researchers have often the choice between several environments (real, experimental, emulated or simulated) to validate their results. Each facet of our work, from the study of network behavior, to protocol improvements, including the characterization of specific phenomena, can be related to different environments more or less realistic. We are thus confronted to a trade-off mixing the level of abstraction, bias of measurement, and implementation complexity (in term of cost and time). Nevertheless, because simulations are often regarded as quicker, easier to use and simpler environments than real testbeds, they are mostly used despite their accuracy limits. This is particularly true when simulating wireless communication, where cross-layer effects and physical medium are complex to isolate and understand. While some effort have been made to improve simulators, their comparison with real environments are still in progress and requires experimental equipments, especially for wireless networks. In this context, we built a controlled test-bed environment to calibrate wireless simulators. This paper compares the results obtained for scenarii respectively played within this environment and the Ns3 platform. Our analysis method is based on network traces comparison using classification trees. On a first basis the construction of the tree is supervised, but later, it could be unsupervised and feed the need for a comparison tool of different network environments.

I. INTRODUCTION

When dealing with communications systems, applied researches are most of the time proven empirically. Thus the environment chosen for the study is critical in term of result validation. An usual approach is to generate measurement traces that will be further studied either to understand system behaviors and their underlying mechanisms or to test new concepts. Thus, researchers are confronted to the well known environment trade-off.

With the increasing interest of wireless communications, various platforms of several kind have been designed since it is not always possible to proceed on a real case (regarding cost, reuse, flexibility or time development). The community shares experimental platforms [1], [2], and develops simulation platforms [3]. On one side the former are realistic but are still difficult to set up, can be expensive and their troubleshooting is time consuming. On the other side, the latter are flexible, cheaper, easy to troubleshoot and produce traces quickly, but their realism are still questionable.

First, among the existing experimental environments, only few of them have put an emphasis on the minimization of hazardous phenomena [4], and as far as we know, none of them are adapted to lead cross-layer analysis for two main reasons

which are (i) hazardous phenomena are not minimized like in [5] or (ii) when minimized the environment is dedicated to one OSI layer for example [4]. Second, when regarding environment comparison and simulator studies, they all follow customized methodology and only few work addressed the Ns3 open-source platform while its popularity becomes significant.

To tackle the first observation, we built an experimental testbed to study interactions between communication layers in such a way that we control the environment while keeping it realistic. By using an anechoic chamber, we can then argue that we have a real environment comparable to a simulated one in terms of hazardous wireless perturbations. This allowed us to address the second observation by comparing the traces obtained from our testbed with traces that have been generated with Ns3 for two different scenarii. Our comparison perspective is made to highlight differences between cross-layer interactions with a methodology based on decision trees.

Our walk has been the following : (1) Design an experimental testbed to highlight cross-layer interaction in an IEEE 802.11g wireless LAN. (2) Design two different scenarii that will produce the same kind of traces when played in both environments. (3) Choose from analysis pertinent criteria to compare traces. (4) Compare traces with a generic methodology using the criteria specified in (3) and finally (5) Determine for which configurations the environments are similar. This approach justifies the emphasis we will put on both the description of the experimental conditions and the methodology comparison. Thereby the remainder of the paper is structured as follows. Section II reviews related works on platform for wireless systems and their comparative studies. We then describe in section III the two considered scenarii while section IV details the two considered environments. Our comparison methodology is introduced in a fifth section. The choice of our metrics and the application of our methodology is detailed in section VI while the results obtained are given in a section VII. We finally conclude and give directives on future work in a last section.

II. WIRELESS NETWORK TESTBED AND RELATED WORK

If we have a look at the study of [6], it appears that a significant part (30 percents) of research papers in networking does not specify the validation environment, 17 percents are based on NS-2, 15 percents built their own tools and only 10 percents have used experimental testbeds, remaining are balanced over

other simulators. This observation gives an insight on the interest that have the research community for simulators, but also stir up questions on the importance of validation. Since the number of wireless platforms is constantly growing, we need comparison work to make the best trade-off when we want to choose one of them. As a result, this section reviews the state of the art regarding the existing platforms as well as the survey that evaluate them.

A. Existing Environments for Wireless Systems

When choosing between wireless network environments, one of the first choice researchers need to make is about the type of environment they need. We have classified existing environments in three types which are : (1) In production environment, (2) Experimental testbeds and (3) Simulation platforms, the emulation platforms could be classified either in the type (2) or (3) depending on their configuration. This subsection will introduce a subset of the existing platforms that we have judged pertinent regarding the current paper.

1) *In Production Environments:* We call "in production environment", a wireless network environment that has real users, and from which some measurements and tests results have emerged. In production environment are usually rare to obtain. Indeed, performing tests without disturbing the end user is tricky, and having the legal access rights to perform them or just to retrieve measures is even more complex. This is probably one of the reason why [7] intends to merge production network and testbed for wired networks. Since it is nearly impossible to characterize with precision the physical medium in these environments, most of the 'in production' evaluation takes higher level measures. This is the case for [5] where only the transport layer and above is studied and [8] from which a social analysis of user can even be drawn with analysis of the traces.

2) *Experimental Testbeds:* We call experimental testbed, a real wireless network environments that has been built exclusively to perform tests and evaluations. Experimental testbeds for wireless systems have lighted up this last decade since simulation platforms have been judged less realistic. Nevertheless experimental testbeds are still technically heavy to set up [9] and so are often shared between institutions to reduce their cost [10]. If they offer a better realism, it is still difficult to control the parameters of their environment. The main reasons are (i) they are sometimes remote or (ii) their medium are not always well characterized. We can find wireless testbeds from very different nature: number of nodes, communication standards, physical environment and configuration. Among them exist EmuLab [1], ORBIT [2], CREW project, ASSERT [11]. It is possible to distinguish these environments depending on the level of control over the physical medium. Controlling the physical medium is expensive, in this sense those testbeds are usually smaller and limited to very specific studies [12], [13].

3) *Simulation Platforms:* We call simulation platform, a wireless network environment entirely modeled in a computer program. This kind of environment is often preferred for its cost and the repeatability of the experiences. Despite the fact that they have been discussed for their physical layer model, they are still exhaustively used as a first step to prototype protocols and algorithms. Depending on the type of experiments, some simulation platforms are preferred over others. A complete review of the existing simulators would not be pertinent since it has been the main subject of previous work [6], [12], [14], [15]. It is still important to point out the most commonly used : Ns2, OPNet, OMNET++, QualNet, Ns3. The latter is the youngest which gives us an additional reason to study it.

B. Previous Comparative Studies of Network Simulators

Most of the presented environments have been compared each others adopting several point of view and approaches. We will mostly consider simulator evaluations. These evaluations can cover simulator performances [14] or their different behaviors considering several settings [12], [13], [15]. We will focus our interest on previous work that deal with a comparative study between a real environment and a simulator and those which considers the assessment of Ns3.

Considering the most recent network simulators [14] compares the performances in terms of run-time and memory usage for similar scenarios. It appears that Ns3 offers the best overall performances compared to Ns2, OMNet++, simPhy and JiST/SWANS. This result does not consider realism of the simulation, but takes part in the final trade-off. The relevance of simulators are evaluated respectively in [12], [13], [15] where authors compared the results obtained from simulators and real or experimental testbeds.

In [13], Rachedi & al. evaluate Ns2, QualNet and OPNET and more specifically the physical and mac layer. It is clear from the study that the tuning of simulator is a crucial point and depends on the scenario itself. Tan & al. also covered behavioral differences of Ns2 and QualNet when confronted to a real testbed in [12]. They show interesting differences on PHY, MAC and IP layers, but confirmed that configuration of simulators could lead to acceptable results on the path loss model. However, their real testbed was not a controlled environment and the physical layer could not be well characterized.

On its side, [15] only considered wired scenario within OPNET, Ns2 and a Network Testbed. If the previous study were not performed in a controlled environment, [4] did experiment in an anechoic chamber similar to ours, but the equipments served to study ray tracing models and was not examining other layer than the physical one. Finally Ns3 channel propagation models has been studied in [16] where Stoffer and Riley showed significant differences in terms of performances and results.

After having observed the state of the art we were able to identify three interesting points which are: (1) The use

of a controlled environment when comparing simulator and a testbed has not been deeply investigated. (2) All the previous studies were specific to a point of view and used various analysis approaches: histogram superposition [16], time series comparison [15] or scatter diagram [13]. (3) Despite the fact that Ns3 has been designed to feed the need for wireless network simulators, only few assessments has been conducted so far. Thus, we suggest a scheme to assess and compare two environments and specifically apply it to Ns3 with a real characterized testbed.

III. SCENARI DESCRIPTIONS

For the purpose of our investigation, we designed two scenarii that we will introduce in this section. The details of their implementation will be drawn in the next section since they are platform dependent. These scenarii are played under our two environments to generate traces. Traces are instantiated by pcap file that have been captured in promiscuous mode on nodes in their respective environment. The files obtained are post-processed and compared for each scenario. Both of the scenarii consider an IEEE 802.11g infrastructure Basic Service Set in the simplest possible cases. These basic scenarii should not play on the same protocol mechanisms. Thus, we should obtain complementary results that would give insight on other situations. The table I summarizes the parameters that we have fixed.

TABLE I: Global Settings of Scenarii's Parameters

Experimental Parameter	Value Setting
Wifi Standard	802.11g
Qos Enabled	False
Mac control rate algorithm	Constant Rate
Data frame bit rate	54 Mbps
Control frame bit rate	24 Mbps
Data and Control frame modulation	ERP-OFDM
Transmission Power	10 dBm
UDP data length	1470 Bytes

A. Scenario 1 : Point to Point CBR Communication With an Additive Gaussian Source Noise

This scenario is designed purposely to highlight the effects of the signal noise plus interference ratio (SNIR) on the service offered to the layers three and above. More specifically we will examine the Round Time Trip (RTT) and the Frame Error Ratio (FER) at the MAC layer. In this context we set up two IEEE 802.11 nodes. An access point (AP) and a station (STA). They are distant from 2 meters. The access point generates a constant bit rate UDP traffic toward the station. We set up a Gaussian source noise that only perturbs the station (UDP destination), and thus creates an asymmetric link between the two nodes. The scenario can be split in time steps, for each time step, the UDP source generates 10000 packets at periodic intervals and under a constant power spectral density (PSD) of white noise. Since the data length is fixed for our experiment, the interval is determined by the UDP bandwidth which are respectively 7 Mbps, 10 Mbps and 15 Mbps. Regarding the PSD, we first established the significant range of values which

were depending on the environment (see next section) and tested them every 0.25 dBm. The scenario 1 leads to a number of 60 combinations of parameter values, it is illustrated in figure 1.

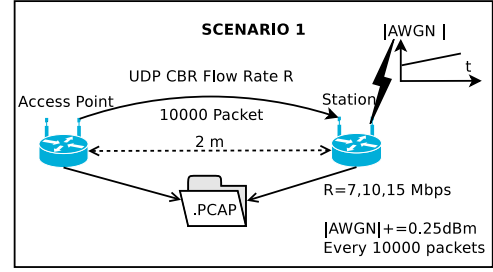


Fig. 1: Scenario 1

B. Scenario 2 : Concurrent Access of CBR Sources in Free Space Propagation

This scenario has been chosen to be out of any hidden terminal considerations and other particular cases of Wifi experiments. We will examine the same effects than for the scenario 1 except that the SNIR is mainly produced by the interference of one station on the others. In this configuration, the access point is an UDP sink, which receives two CBR UDP flows, one from each station. Each experiment lasts for 60s. During the first 20 seconds the first source is transmitting alone, then the second source is started and finishes alone for the 20 last seconds. The spatial configuration of the scenario is illustrated in figure 2, since there is no geometrical symmetry, we derived two sub-cases. In the first one, the flow of the station 1 starts before the flow of the second one, the second case is the contrary. We have tested these two sub-cases for different UDP bandwidths for each station with the values 7 Mbps, 10 Mbps and 15 Mbps. The scenario 2 leads to a number of 18 combinations of parameter values, it is illustrated in figure 2.

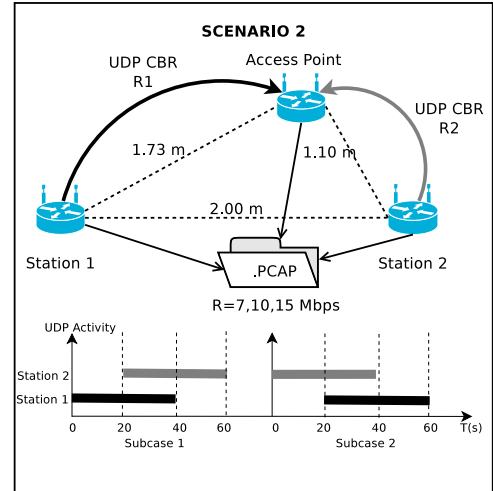


Fig. 2: Scenario 2

IV. ENVIRONMENTS DESCRIPTION AND CHARACTERIZATION

This section puts an emphasis on the environment that we used for our comparison. Indeed, it is crucial to know the condition of our experimentations. Therefore, we will first introduce the wireless communication systems metrology platform that we have built to study cross-layers mechanisms in IEEE 802.11 networks. The second subsection gives an overview of Ns3 simulator and the model we used for the comparison. The last subsection details the translation that we make to navigate between the simulation and the real worlds regarding the noise generation.

A. Wireless Communication Systems Metrology Platform

As pointed out by Rehman & al in [9], building an experimental platform is not a pleasure cruise, we judge important to detail the steps (design, configuration, usages and pitfalls) that was necessary to finally obtain measures.

1) *Experimental conditions:* Our wireless test-bench was designed to have a level of control as close as possible from the one of a simulation environment. Therefore, we made the choice to use an anechoic chamber. An anechoic chamber is a protected RF room which simulates free space conditions. Our model of chamber is 4,10 meters long for 2,50 meters large (161.5in x 98.5in). It is commonly used to measure antenna patterns for frequency up to 40 GHz. Inside walls are covered of microwave absorbers materials that break and scatter any wireless signal that would come from an inside source. The chamber is then free of any multi-path propagation. There are different types of absorber, each of them is defined for a specific frequency range which allow us to use the anechoic chamber for different purposes and frequencies. The absorbers protect also the inner environment of the room from outside perturbations, therefore it is a very RF protective room. This protected context minimizes the uncontrolled parameters of our communication.

Inside the anechoic chamber we placed several Wifi nodes that will be used in the scenari. The nodes are controlled through a wired network to not interfere with 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. The Wifi network controllers are based on the AR5414 chip-set from Atheros which uses the ath5k driver and are attached to an omni-directional antenna. The choice of the wifi chipset and its driver was crucial because they define the amount of metrics and the accuracy that will be possible to obtain. This choice conditions the possibilities in terms of scenari. The Ath5k drivers is open-source and well documented thanks to an active online community support. It has also a good integration within the OpenWrt OS. The OpenWRT OS is flexible enough to allow the implementation of new functionalities so that it accelerates the upgrade of the bench.

2) *Synchronization of The Nodes:* As we use several equipments to get measures, they needed to have their clock synchronized. This was done with NTP by using a wired

connection to a remote NTP server. The NTP protocol provides a correct synchronization for most of our analyses. In complement and for the most accurate measurements, we use a more precise time-stamp called 'Timer Synchronisation Function Timestamp' (TSFT) which is a value returned by the Atheros radio and which provides a microsecond accuracy. However as we don't know precisely the accuracy of the TSFT synchronization between each node, the tsft value is only used to compute relative and local metrics such as the inter-arrival time between packets.

3) *Capture Configuration:* The command and control of the nodes is done using distributed software deployed on the boxes and written in RUBY. The application allows the configuration of wireless networks settings and permits to launch command to control the measurement process. The system is based on Remote Method Invocation which gives access to a set of operational methods on each boxes and then permits the network to be controlled and configured transparently in live measurements from a single node in the network.

The configuration of the network 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 and tools when arrived in the kernel interface. The packets contains data from link to applications layer such as the 802.11 channel number, the type of frame at the MAC layer or packet size at the Network layer. Additionally, a packet also contains a RADIOTAP header which gives radio level information such as the received signal strength (RSSI) reported by the ath5k driver. This metric is complex and serves as an indicator for the upper layers, but its value is defined by the constructor.

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 while on the original kernel they were discarded. The propagation is only possible if the error corrupted the data but not the header fields. Following this modification the RADIOTAP header now contains a flag specifying whether a FCS error was detected when decoding the packet.

4) *Usage of The Testbench:* The use of an anechoic chamber is a good trade-off in terms of realism and precision of measures. From these point of views, we think that our environment can have its place in between a simulator and an "in production environment". While the simulation provides a full control of many parameters, a real environment provides very low control over the parameters, and particularly over the ones which affect the wireless medium (move, interferences, etc.). One of the objectives of our environment is to minimize the presence of these uncontrolled parameters on the communication. Another objective is to generate and control selected parameters that will impact our communications.

The noise and the interferences significantly impact the communication. In figure the set-up described in 3 we inject noise in the environment using a signal generator to perturb the communication. The signal generator is a device which emits RF signals. It can be configured to generate more realistic

noise. Among the parameters of the generated noise, two important elements will have a crucial impact, on a first hand the modulation used will determine the main characteristics of the noise signal in the time and frequency domains, i.e. it will determine the spectral occupancy of the generated signal, its fading or narrowness. On a second hand, the amplitude of the signal will also affect the measured level of noise on the receiver side.

We found that the AWGN noise modulation was a good choice because it is a common model of noise. The level of amplitude was determined empirically by testing the effects of the noise on the communication. The impact of the noise level in term of FCS errors on the link can be found in the subsection IV-C.

Finally, a major element that affects the noise generated in the anechoic chamber is the antenna. It will determine the waveform, the direction and the amplitude of the noise wave. In scenario 1, in order to perturb only one side of the communication we used a very directional antenna pointed to the receiving station.

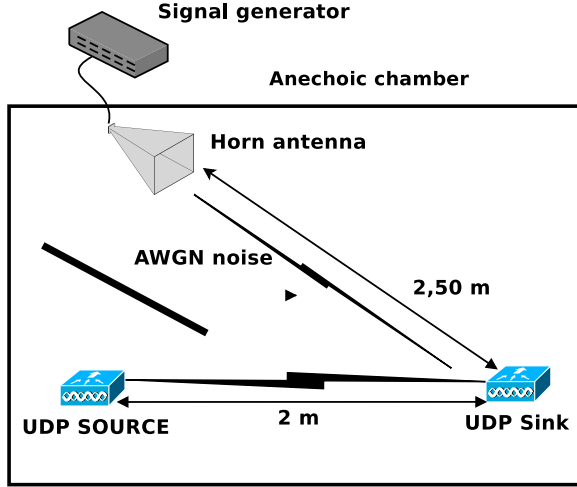


Fig. 3: Experimental setup for scenario 1

In the scenario 2, the controlled parameter was the level of interference. This was achieved by starting and stopping UDP flows sequentially from two different stations to the access point. We placed the AP at the center of the anechoic chamber. The two stations were disposed as specified in figure 2.

B. An Overview of NS3 for Wifi Simulation

Ns3 simulator is an open source project which has been launched in 2006. It intends to replace its old brother Ns2, but at the moment researches are balanced between the better performance and models of Ns3 for wireless network and the myriad of models that Ns2 offers. Nevertheless the Ns3 community is active and a significant effort has been made to the design of the simulator and its documentation. Thus, implementing a new model for Ns3 requires to declare and specify a set of classes by following a predefined skeleton. At this point, the best way to get into Ns3 is still to follow

the official tutorial, so that a full description of Ns3 simulator is out of this paper scope. The main point that we would like to highlight is that roughly every description of the real world is an object in Ns3 (channel, physical and mac layer, network device, node, Internet stack, application and so on). In this sense, describing an Ns3 simulation is equivalent to create links between objects and specify the property of these objects (behavior or parameters) using a high-level API (helpers). Because of the huge domain possibilities of Ns3, it is quickly complex to finely tune parameters, most of the times Ns3 users consider the official Wifi implementation of [3] with the default parameters. We then use this model since it is the most popular.

It is necessary to keep in mind section 8 of [3]. In few words: The Physical layer state will depend on the value of the energy of the first bit received. The energy received is computed following the Friis equations. The way to decide whether a frame is dropped or not is based on the SNIR value at different instants of the frame and on the modulation scheme. We have two modulation schemes for our scenario because we use two constant rates (for data frame and control frame). While we have not modify Ns3 simulator for scenario 2, scenario 1 required to introduce a noise generator, the easiest way to proceed was to tune the N_f term in the equation (3) of [3]. Because N_f has a specific meaning, we have preferred to introduce an additional term N_g produced by a noise generator object at each computation, so that the equation is now :

$$SNIR(k, t) = \frac{S_k(t)}{N_i + N_f + N_g(t)} \quad (1)$$

To be able to compare the results obtained in this environment with a real testbed, we used the tracing system of Ns3 that allows the generation of .pcap traces, imitating a promiscuous mode with a RADIOTAP header. The main difference with the testbed is that corrupted frames in Ns3 are not traced. We then applied callback on the drop functions in objects modeling the physical and mac layers to trace them.

C. Set-Up Verification

We described in the previous section, differences between the two implementations, specifically for the generation of the noise signal. In the experimental testbed, the signal generator is distant from the station and sends a signal in its direction. Considering the antenna directionality and the absorption of the chamber, it is correct to say that the noise received by the access point is negligible. In the simulation environment, the noise signal is directly introduced on the receiver side. Thus the level of noise introduced in both environments must be of different amplitude. We can verify a strong equivalence in terms of impact on the frame error ratio.

We will assume that in our experimentation the frame error ratio (FER) only depends on the SNIR because we have a fixed modulation. Since the received signal strength is nearly constant, the SNIR should only depend on the received noise. We can then deduce a bijection between the FER and the level of introduced noise. While figure 4 illustrate the

frame error ratio against the two levels of introduced noise in their respective environment, figure 5 is their scatter diagram relatively to their FER.

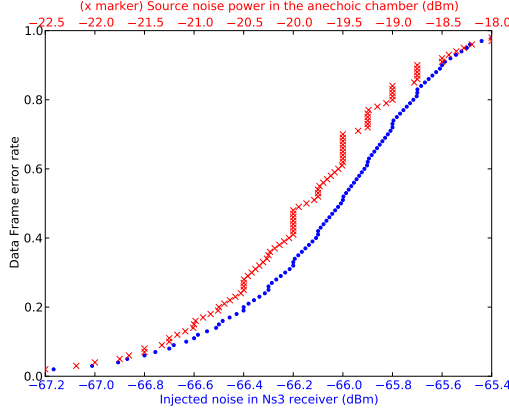


Fig. 4: Evolution of The Data Frame Error Ratio Vs Level of Noise

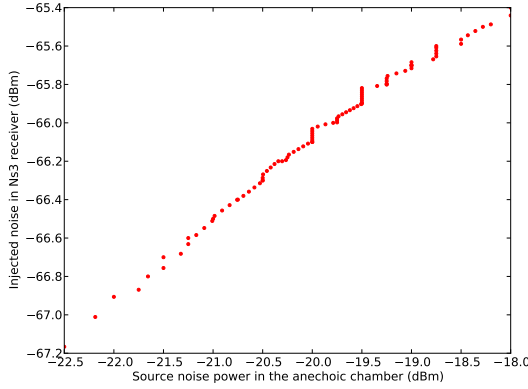


Fig. 5: Scatter Diagram of The Level of Noise in The Two Environment

V. COMPARISON METHODOLOGY

When it comes to compare environments, researchers have to consider an infinite number of parameters. As a result, it is near to be impossible to make an intrinsic comparison of platform and consider a white-box methodology. Most of the time, results are drawn from traces analysis, then, it is not important to have realistic environments as far as we have realistic traces. Under this assumption, comparing traces produced by different environments under the same scenario turns out to be more pertinent than comparing environment themselves.

Trace comparison is always specific to a point of view and implies the notion of distance which is far to be trivial for complex mathematical objects. At first, one can think that if distances between traces is less than an absolute threshold, then traces are similar. We believe that fixing a threshold is

even more complicated and does not make sense, so that it is then truer to state : "From point of view X , environments A and B are closer for scenario 1 than for scenario 2".

Then, given two environments under several scenarii, we will be able to compare distances between the produced traces and conclude that for some scenarii, environments are closer than for others under a particular perspective. This section provides a framework to establish distances between traces given a specific point of view. We will describe its general concepts and assumptions, detail the model we used and finally its implementation for our comparative study (ie: point of view) of the interaction between OSI model layers 1,2 and 4.

A. General Concepts and Assumptions

We assume that the traces we want to compare are both defined as sets of N -dimensional points. Thus, in the remainder of the paper we will interchange the terms trace and set, sub-trace and subset. As we pointed out before, we wanted to establish distances between traces. Our problem is now to establish distance between sets of points.

Our approach to compare traces is to summarize them in a descriptive and synthetic way and then to compare their summary. The way we have chosen to summarize those sets is the following. (1) We partition traces into subsets, following identical rules for both traces. (2) We construct the normalized repartition of a property over subsets. (3) We evaluate the distance between the repartition using existing metrics. The rules to construct the subspaces depends on the point of view of the user. The possibilities to compare traces in an unsupervised way (ie: without having to choose any expert point of view) is left as future work. The next subsection details the formalism that we used to summarize traces using decision trees and then define distance on the set of trees.

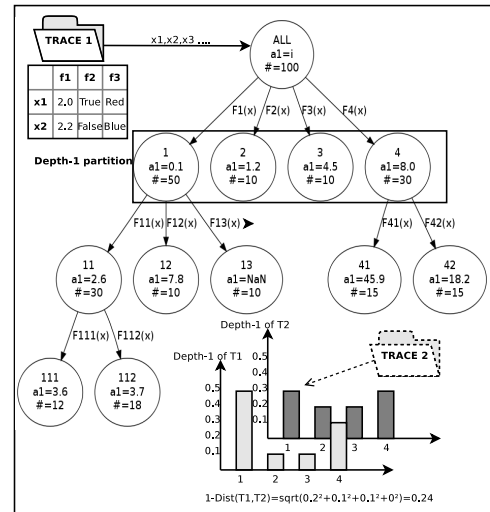


Fig. 6: Summarize a Trace using a Tree

B. Decision Tree Model and Trace Comparison

The model used to describe traces follows the structure of a decision tree. A node in the tree identifies a subset of points. For a given node N , the set of its successors forms a partition of N . The edge between a node and its successors can be seen as a splitting function defining the requirements to be in the child subset. The set of functions (or edge) leaving a node are then mutually exclusives. The root node identifies the whole set. We can define the partition of depth N , P_n , as the union of the set of the N depth nodes and the set of the leaves with a depth of N or less. As shown in figure 6, we keep in each node, some attributes on the subset (i.e. : an average over a dimension or any other descriptive value). We choose to limit our interest to the cardinality attribute to summarize a set of points. We argue that tree structures are adapted to network traces for several reasons:

- Points can be easily instantiated by packets, flows or metrics.
- Each depth of a tree can represent an OSI layer.
- Adapted to multiple scale time series analysis (when first functions are temporal based).
- Their construction complexity is $O(n)$ (when functions are defined for n points)
- They are visual and helpful for a human eye.

Distance Between Trees: As specified above, nodes keep some descriptive values on their subsets and more specifically their cardinality. We have then defined several distances for tree which have the same structure :

- Let T_i be a Tree.
- Let \mathcal{P}_i^d be the d -depth partition of T_i .
- Let $t_d^i(n)$ be the n -th element of \mathcal{P}_i^d
- Let $p_d^i(n) = \frac{|t_d^i(n)|}{|T_i|}$ be the normalized cardinality of $t_d^i(n)$

Using this notation, we define two types of distance which are based on the euclidean distance.

A d -Depth distance is defined by:

$$Dist_d(T_i, T_j) = \sqrt{\sum_{n=1}^N (p_d^i(n) - p_d^j(n))^2} \quad (2)$$

A leaves distance is defined by:

$$lDist(T_i, T_j) = Dist_d(T_i, T_j) \quad (3)$$

s.a : $depth(T_i) = depth(T_j) = d$

The d -Depth distance will be used to evaluate trace distances with a specific level of grain. The d -Depth distance tends to increase with d as we consider a higher level of details (so called curse of dimensionality). By construction the 0-Depth distance is always zero. It is still possible to define hierarchical distances and mix several level of details. Note that the recursive structure of trees also gives us the possibility to evaluate distances subtree by subtree. Thus we are able to compare patterns for some subsets of the trace only. In the following section we introduce the splitting functions we have chosen to build a tree that highlights cross-layer interactions.

VI. OBSERVING CROSS-LAYER INTERACTIONS WITH A SPECIFIC TREE

In wireless communications, cross-layer interactions are complex behaviors to analyze. It requires a multi-layer and multi-scale analysis of metrics. Our goal here is to evaluate how comparable are the environments regarding the behavior of several metrics which are : (1) The RTT at the MAC layer for Data Packets, (2) The Number of Mac retries before a transmission, (3) The Data Frame Error Ratio at the MAC layer (FER), (4) The Received Strength Signal Indicator (RSSI). Figure 7 details the association between the RTT and the injected level of noise. When observing the evolution of the average RTT with the average number of retry aggregated by seconds, figure 8 clearly shows a linear relation for the experimental test-bench (correlation coefficient $r = 0.90$). It is even more obvious that it exists a direct relation between the second metric and the FER. Finally, figure 9 denotes also a more complex link between the FER and the RSSI (note also the proximity with figure 4) in the anechoic chamber. Thus, these four metrics are for sure linked. The next subsection explains the way we retrieved them to obtain figure below and design a summarizing tree.

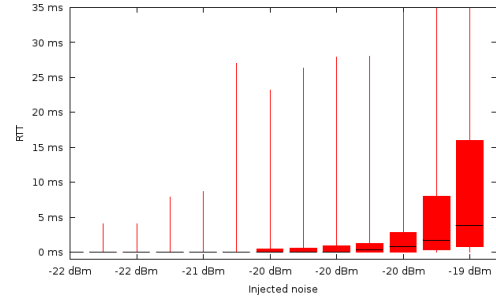


Fig. 7: Experimental relation between the average RTT and the injected noise

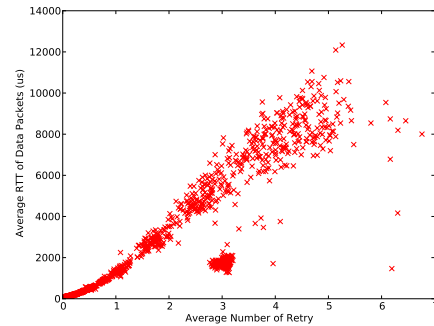


Fig. 8: Experimental relation between the average RTT and the average number of retry over a second

A. Metrics retrieval from the pcap files

All the traces derived from the pcap files can be viewed as set of points. Each point is relative to a time period of 1 second

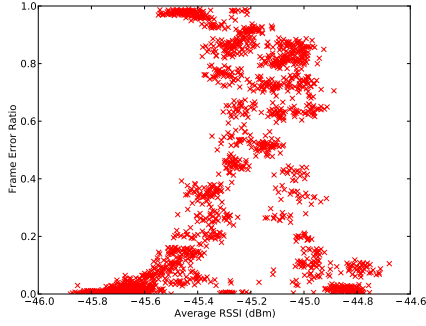


Fig. 9: Experimental relation between FER and average RSSI over a second

for a flow. This aggregation is sufficient to smooth the curve without missing details of interest. The mandatory features of the points are : the average RTT, the average number of retry, the FER, and the average RSSI. The traces will merge all the experiments that have been done for each scenario in each environment. Thus we defined optional features to keep a trace of their effect on the metrics. For scenario 1 additional features of a point are the bandwidth of the UDP flow and the level of injected noise. While for the scenario 2 we kept the bandwidth of each station and the source of the flow.

RTT of Data Packets: The RTT is computed at the sender side, it is the time between the first sent (in the case of retransmission) of an UDP packet and the reception of the associated acknowledgement (note that we set the UDP data length less than the Maximum Segment Size to avoid fragmentation).

Number of Retry: The number of retry for a frame is estimated on the receiver side. It counts the number of consecutive FCS error before having each correct frame. This is an under estimation since some frames could be missed and not recovered at all if the header is corrupted, this is not the case for the simulation.

Data Frame Error Ratio: The FER is computed on the receiver side. It is the ratio between the number of received data frame containing an FCS error and the number of received data frame. Once again this is an under estimation that we have not in the simulator.

RSSI: The RSSI is given by the RADIOTAP header. This metric is specific to the constructor. By looking at the ath5k driver sources, we know that our RSSI is a value computed from the RX antenna gain and from a noise floor value updated periodically. The nearest indicator of the RSSI in the simulator is the given level of noise for packet.

B. Traces Elaboration and Tree Construction

Since we denote relations between the metrics in figures above, we elaborated a tree based on the set of point already described. The structure of the tree is illustrated in figure 10. Each depth deals with a metric. We have organized the order of the metric by OSI layer. Thus the tree could be seen as a causal tree where we could examine the reasons of the high layer

metrics value. At each node we call a splitting function that compares the value of the metrics to a threshold. We limited the depth of the tree to 4 and its breadth to 2, it thus avoids the curse of dimensionality when computing the distance between the histogram.

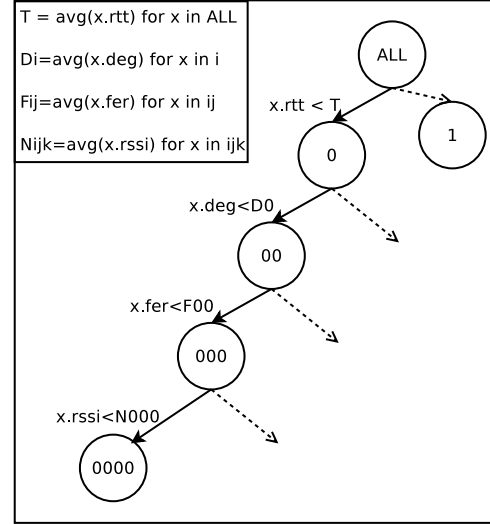


Fig. 10: Design of The Cross-Layer Metrics Tree

As illustrated in figure above, at each node we compute the average value of the next metric to fix its threshold. By doing so, we allow us to have a rough idea of the distribution of the metric in the considered subset of points. Fixing a threshold relatively to its subset is more descriptive than fixing it on the whole set. For example, if we consider the average number of retry before a transmission over the whole set of the scenario 1, it will be quite high since we fixed a level of noise explicitly to observe different behaviors. Thus, all the points would be below the threshold for point with a small average RTT and above for highest ones. Using this structure, we are able to easily keep in mind the scheme of the tree. Indeed, each path to a node has a binary representation which is meaningful in term of position in its distribution (one is over the average). This allows us to associate a number to a leaf.

VII. RESULTS COMPARISON

As we explained in previous sections, we have processed pcap files to obtain a set of points for each environment and for each scenario. Then the resulting sets have been summarized using the 4-depth tree of figure 10.

A. Identification of environment specificities

As explained above, splitting a set will give information on the distribution. By observing the value of the threshold, we are capable to identify a cluster in figure 8. The table II gives the threshold that have been used in the experimental trace for the scenario 1. Applying the two first split on figure 8, we found a border of a cluster which corresponds to a specific behavior of our network controllers.

TABLE II: Comparison of split values and cluster coordinates.

Feature	Split Value	min cluster	max cluster
RTT	1,86 ms	1,27 ms	2,2 ms
#Retry when RTT > 1,86	3.26	2,76	3,2

B. Distance Levels Between Environments

We computed for each scenario the d-Depth distance between two traces issued from both environments. As we expected, distance increases with depth. The table III summarized the computed values. We can state that Ns3 is closer to a real environment in the case of scenario 1 than in scenario 2.

TABLE III: d-Depth distance between the two environments for each scenario

Depth	Scenario 1	Scenario 2
0	0.0	0.0
1	0.13	0.24
2	0.14	0.25
3	0.11	0.44
Leaves	0.28	0.58

One can observe that the 3-depth distance for the scenario one is the smallest one. Keeping in mind that the 3-Depth distance considers the distribution of the FER, this result seems coherent since we calibrated the level of injected noise with the value of Frame Error Rate (see IV-C). However the distance significantly increases for depth 4 when splitting on the RSSI value.

In order to highlight the reasons of this growth, we plot the bar graph corresponding to depth 3 and 4 (see 13). It clearly appears that the RSSI differs strongly depending on the environment. For example in scenario 2, Ns3 models a level of received noise with a constant value since the three stations are fixed just like their power transmission and their noise figure. Thus the last metric has a null information gain for the scenario 2 in this environment. This is why the last split on this metric failed since every points has an RSSI strictly equals to the threshold. This is not the case in a real environment. The same phenomena appeared in scenario 1 where the RSSI distribution is reverted between environments.

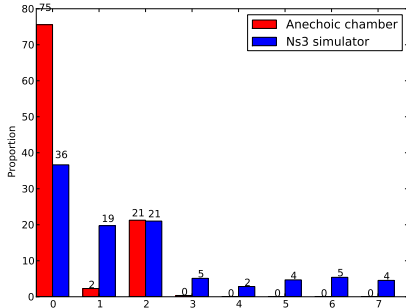


Fig. 11: 3-Depth partition in two environments for scenario 2

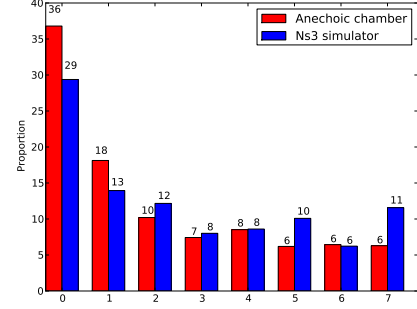


Fig. 12: 3-Depth partition in two environments for scenario 1

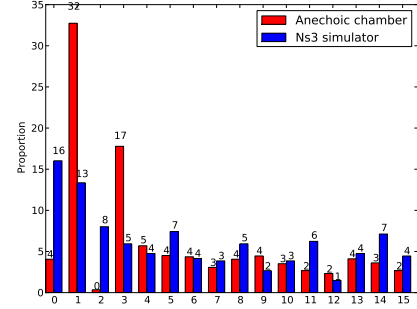


Fig. 13: Leaves partition in two environments for scenario 1

VIII. CONCLUSIONS AND FUTURE WORK

A. Main Contributions

In this paper, we tried to tackle the problem of environment comparison for wireless networks. Considering the state of the art we noticed that working under a controlled environment was a benefit for wireless network. This is specifically true when researchers want to identify interactions between multiple metrics at different layers of the OSI model. We analyzed comparative studies and conclude that there were a need for (1) a comparison methodology and (2) Ns3 evaluation. Thus, our contribution were the following: first we established an highly characterized test-bed for Wireless Network, then we suggest a comparison methodology which is based on the generated traces. Finally we applied this methodology to discover not only association between metrics but also differences between environments. As a reminder, we established a linear relation between the average number of retry and the round time trip offered to the network layer, and pointed out network controller specificity cluster regarding the retry scheme policy of our network card at the mac layer. This has been confirmed by the multi-layer analysis provided by our binary tree. The tree also confirmed the precision of Ns3 regarding its physical layer model for a noisy environment. Indeed the impact of the noise on the frame error rate and higher level metric is similar to an experimental testbed. However when the level of noise is low, high level metrics are impacted by medium access control mechanisms. It appears that these mechanisms are different

from our test-bed. This observation reminds researchers that constructor choice of implementations can introduce unknown behavior.

B. Future Work

The presented work only considered two basic cases. Obviously, other situations can be tested, like the hidden terminal situation, a greater number of node or the use of TCP as a transport layer. Also it is important to know that the choice of metrics is crucial. Metrics have been used to build the decision tree. Metrics are chosen to analyze a specific situation. However they are hidden in traces and finding them is a new research lead. Our tree has been built in a supervised way, future work on the way to choose splitting feature and their values could be done base on gini criterion. We believe that in the future, tree could be also used to summarize network traces and serve as a pattern identifier. Other distances on tree could then be applied, specially to validate the repeatability of an experimental environment by verifying a small distances between produced traces.

ACKNOWLEDGMENT

This work is partially funded by the French National Research Agency (ANR) under the project ANR VERSO RESCUE (ANR-10-VERS-003)

REFERENCES

- [1] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar, "An integrated experimental environment for distributed systems and networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 255–270, Dec. 2002. [Online]. Available: <http://doi.acm.org/10.1145/844128.844152>
- [2] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *IN PROCEEDINGS OF THE IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE (WCNC)*, 2005, pp. 1664–1669.
- [3] M. Lacage and T. R. Henderson, "Yet another network simulator," in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, ser. WNS2 '06. New York, NY, USA: ACM, 2006. [Online]. Available: <http://doi.acm.org/10.1145/1190455.1190467>
- [4] Y. Zheng and D. M. Nicol, "Validation of radio channel models using an anechoic chamber," in *Proceedings of the 2010 IEEE Workshop on Principles of Advanced and Distributed Simulation*, ser. PADS '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 114–121. [Online]. Available: <http://dx.doi.org/10.1109/PADS.2010.5471660>
- [5] R. Hutchins and E. Zegura, "Measurements from a campus wireless network," in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 5, 2002, pp. 3161–3167 vol.5.
- [6] S. Mehta, N. Ullah, M. Kabir, M. Sultana, and K. S. Kwak, "A case study of networks simulation tools for wireless networks," in *Modelling Simulation, 2009. AMS '09. Third Asia International Conference on*, 2009, pp. 661–666.
- [7] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, and G. Parulkar, "Can the production network be the testbed?" in *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, ser. OSDI'10, 2010, pp. 1–6.
- [8] D. Kotz and K. Essien, "Analysis of a campus-wide wireless network," *Wirel. Netw.*, vol. 11, no. 1–2, pp. 115–133, Jan. 2005. [Online]. Available: <http://dx.doi.org/10.1007/s11276-004-4750-0>
- [9] S. U. Rehman, T. Turletti, and W. Dabbous, "Benchmarking in Wireless Networks," *Interne*, Oct. 2010. [Online]. Available: <http://hal.inria.fr/inria-00530329>
- [10] in *NETWORKING 2011*, ser. Lecture Notes in Computer Science, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds., 2011, vol. 6640.
- [11] E. Nourbakhsh, J. Dix, P. Johnson, R. Burchfield, S. Venkatesan, N. Mittal, and R. Prakash, "Assert: A wireless networking testbed."
- [12] K. Tan, D. Wu, A. J. Chan, and P. Mohapatra, "Comparing simulation tools and experimental testbeds for wireless mesh networks," *Pervasive and Mobile Computing*, vol. 7, no. 4, pp. 434 – 448, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S157411921100040X>
- [13] S. Lohier, A. Rachedi, E. Livolant, and I. Salhi, "Wireless sensor network simulators relevance compared to a real ieee 802.15.4 testbed," in *Wireless Communications and Mobile Computing Conference (IWCMC)*, 2011 7th International, 2011, pp. 1347–1352.
- [14] E. Weingartner, H. vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," in *Communications, 2009. ICC '09. IEEE International Conference on*, 2009, pp. 1–5.
- [15] G. F. Lucio, M. Paredes-farrera, E. Jammeh, M. Fleury, and M. J. Reed, "Opnet modeler and ns-2: Comparing the accuracy of network simulators for packet-level analysis using a network testbed," in *In 3rd WEAS International Conference on Simulation, Modelling and Optimization (ICOSMO)*, 2003, pp. 700–707.
- [16] M. Stoffers and G. Riley, "Comparing the ns-3 propagation models," in *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2012 IEEE 20th International Symposium on, 2012, pp. 61–67.