Improving the ns-3 *TraceBasedPropagationLossModel* to Support Multiple Access Wireless Scenarios

Helder Fontes
INESC TEC and Faculdade de Engenharia, Universidade do Porto
Porto, Portugal
hfontes@inesctec.pt

Rui Campos
INESC TEC and Faculdade de Engenharia, Universidade do Porto
Porto, Portugal
rcampos@inesctec.pt

Manuel Ricardo
INESC TEC and Faculdade de Engenharia, Universidade do Porto
Porto, Portugal
mricardo@inesctec.pt

ABSTRACT

In wireless networking R&D we typically depend on experimentation to further evaluate a solution, as simulation is inherently a simplification of the real-world. However, experimentation is limited in aspects where simulation excels, such as repeatability and reproducibility. Real wireless experiments are hardly repeatable. Given the same input they can produce very different output results, since wireless communications are influenced by external random phenomena such as noise, interference, and multipath. Real experiments are also difficult to reproduce due to testbed operational constraints and availability.

We have previously proposed the Trace-based Simulation (TS) approach, which uses the *TraceBasedPropagationLossModel* to successfully reproduce past experiments. Yet, in its current version, the *TraceBasedPropagationLossModel* only supports point-to-point scenarios. In this paper, we introduce a new version of the model that supports Multiple Access wireless scenarios. To validate the new version of the model, the network throughput was measured in a laboratory testbed. The experimental results were then compared to the network throughput achieved using the ns-3 trace-based simulation and a pure ns-3 simulation, confirming the TS approach is valid for multiple access scenarios too.

CCS CONCEPTS

• **Networks** → **Network simulations**; **Network experimentation**; **Network measurement**; **Mobile networks**; **Mobile ad hoc networks**; *Protocol testing and verification*; Network protocol design;

KEYWORDS

ns-3, Mobile Network Simulation, Trace Based Simulations, Reproducibility of Experimental Conditions, Perpetuation of Real-World Mobile Testbeds, Offline Experimentation, Concurrent Testbed User Access

WNS3 2018, June 2018, Surathkal, India

ACM ISBN 978-1-4503-6413-3/18/06…$15.00
https://doi.org/10.1145/3199902.3199912

ACM Reference Format:
Helder Fontes, Rui Campos, and Manuel Ricardo. 2018. Improving the ns-3 *TraceBasedPropagationLossModel* to Support Multiple Access Wireless Scenarios. In *Proceedings of the 2018 Workshop on ns-3 (WNS3 2018), June 2018, Surathkal, India.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3199902.3199912

1 INTRODUCTION

Wireless networking R&D depends on experimentation for the evaluation of networking solutions in real environment, as simulation is inherently a simplification of the real-world. However, despite more realistic, experimentation is limited in aspects where simulation excels, such as repeatability and reproducibility.

Real wireless experiments may be difficult to repeat. For the same input they can produce very different output results, since wireless communications are influenced by external phenomena such as noise, interference, and multipath. Even if repeatable, experiments may still be difficult to reproduce. Namely, other researchers may be unable to reproduce an experiment and confirm previously obtained experimental results right away, because either the testbed is unavailable – offline or running other experiments when using a community testbed –, or inaccessible at all when a custom testbed was originally used.

Our experience in past and current research projects led us to realize that mobile testbeds are increasingly complex and costly to maintain. For instance, emerging scenarios such as aerial [10] and maritime [2] usually involve difficult logistic operations and reduced duration of the experiments. This has to do with the characteristics of the communication nodes such as limited battery or fuel autonomy, and the high costs related to renting boats and reserving airbase time slots. On top of this, the radio link quality – herein defined as the Signal to Noise Ratio (SNR) at the receiver – is highly unstable and the mobility of the nodes is difficult to reproduce in real-world testbeds; this makes experiments non-repeatable. Furthermore, in these complex scenarios 1) simulations usually provide very optimistic results, forcing Experimentation to more accurately validate the solution under evaluation, and 2) experiments are more difficult to repeat and reproduce.

What if we could make any wireless experiment repeatable and reproducible under the same exact conditions? What if we could share the same testbed execution conditions among an "infinite" number of users? What if we could run wireless experiments in faster than real time?

INESC TEC has been developing the Trace-based Simulation (TS) approach [5] that combines the best of simulation and experimentation to enable exactly that. By relying on Network Simulator

3 (ns-3) and its good simulation capabilities from the MAC to the application layer, we have been exploring how ns-3 can be used to replicate real-world wireless experiments. The TS approach introduces new mechanisms to 1) capture the execution conditions of an experiment and 2) enable its repetition and reproduction using ns-3.

The original contributions of this work are two-fold: 1) a new version of the ns-3 *TraceBasedPropagationLossModel* for supporting Multiple Access wireless scenarios, which is able to reproduce the radio link quality in ns-3 from the real traces, including the ability to represent asymmetric radio link qualities; 2) further validation of the TS approach, which uses ns-3 as a platform for perpetuating real-world experiments based on real traces of node positions and radio link quality.

The paper is structured as follows. In Section 2 we present the State-of-the-Art on solutions exploring the concept of replaying real-world experiments. In Section 3 we present the new version of the *TraceBasedPropagationLossModel*, including its implementation details. In Section 4 we validate the new version of the *TraceBasedPropagationLossModel* using a 3-node laboratory testbed. Finally, in Section 5 we draw the conclusions and refer the future work.

2 STATE-OF-THE-ART

To address the problem of low repeatability and reproducibility of experimental results, different experimentation, emulation and simulation approaches have been proposed in the state-of-the-art introducing the concept of replaying real-world experiments.

Regarding experimentation, the CONCRETE [7] tool used in federated testbeds such as Fed4FIRE+ [3] allows to achieve repeatable experiments by analyzing the correlation of the experimental results obtained for different executions and selecting the ones that represent the system stable operation. Nevertheless, in practice the "outliers" are equally important and representative of the system operation, as they test the system operation boundaries and often reveal unpredicted phenomena that shall be evaluated too. This solution also depends on the availability of the testbed.

Regarding emulation, mininet-wifi [8], based on the mininet emulator, is a solution focused on emulation for Software-Defined Wireless Networks, which supports replaying nodes positions and Wi-Fi Received Signal Strength Indicator (RSSI). Nonetheless, it only supports Emulation Mode, symmetrical Wi-Fi links, and it is focused on Software Defined Networking (SDN). According to [6], mininet-wifi is also lacking the support for minstrel rate control algorithm, channel contention mechanisms (e.g., CSMA-CA), MAC layer retransmission and interference. Our approach, focused on replaying the physical conditions of a Wi-Fi scenario in ns-3, aims at supporting all the aforementioned missing mininet-wifi functionality, while also supporting asymmetric Wi-Fi links and simulation mode.

Regarding simulation, two main approaches can be found: 1) Packet Based Replay such as the one proposed in [9], where the authors capture traffic of real networks and try to reproduce the same experimental condition in simulation down to the per packet resolution, including the same throughput and packet rate; 2) Application Layer Replay, as the one presented in [1], where the authors try to abstract all low level variables and reproduce the traffic delays and performance bottlenecks experienced in the real network at the application layer.

The TS approach [5] is simpler than the aforementioned approaches. We combine the ns-3 TCP/IP and MAC simulation capabilities with the physical characteristics of the real-world experiment captured in traces including node positions and radio link quality. This results in reproducible experimental conditions inside the simulator, effectively perpetuating a real-world experiment. The reproducible scenario can then be used to continue improving a solution under evaluation in simulation environment, which enables: 1) concurrent user access to the same exact experimental setup; 2) running simulations in faster than real time; 3) running multiple simulation instances at the same time, exploring different variants of the solution under evaluation.

3 *TraceBasedPropagationLossModel*

3.1 Background

To implement the new version of the ns-3 *TraceBasedPropagationLossModel*, most of the details presented in [5], regarding the TS approach, remain valid and should be taken into consideration. For the sake of completeness, an overview of the most important aspects is presented in what follows.

- **Reference Scenarios.** We target emerging scenarios where experimentation is very costly and conditions are difficult to repeat and reproduce. These scenarios tend to be the ones where simulation is too optimistic and unable to model the complex and less stable reality. As such, it is not enough to properly evaluate a solution. These scenarios typically present complex mobility patterns and different noise exposures per node, which often result in rapidly changing and asymmetric radio link SNR.
- **Goal.** Our goal is to capture the conditions of the real-world experiments and enable their perpetuation in simulation environment. This will allow reproducing the experiments offline and the evaluation of an unlimited number of solutions in the exact same conditions. To attain this goal the TS approach combines all built-in ns-3 capabilities for simulating the TCP/IP protocol stack with traces characterizing relevant physical parameters, such as the variation of nodes position and the SNR of the radio links over time. We argue that in this way it is possible to reproduce real-world experiments without the complexity associated to the state of the art approaches.
- **Variables to be collected.** The TS approach assumes the physical conditions experienced by real nodes can be characterized by means of two variables: 1) the node position along the time, e.g., *GPS Coordinates*; 2) the quality of the radio links established with peer nodes, e.g., *RSSI, noise floor, and resulting SNR.* This information is used to feed the trace-based ns-3 simulation. Other characteristics also need to be collected once per experiment, such as the *Channel Frequency (MHz)*, *Channel Bandwidth (MHz)*, *TXPower (dBm)*, *Physical Rate (Fixed/Auto)*, and *Wi-Fi standard (a/b/g/n)*. In [5] we refer in detail to each of these variables and point out the way they can be collected in a real testbed.

- **Reproducing Real Node Positions in ns-3.** ns-3 already implements the *WaypointMobilitiModel*, which accepts a list of *Waypoints* – each *Waypoint* composed by a *Cartesian Coordinate Vector* – that can be directly derived from GPS coordinates. We just need to adjust the necessary offset between *Simulation Time* (starts at 0 s) and the experiment *WallClockTime*.
- **Reproducing Radio Link Quality in ns-3.** ns-3 Wi-Fi nodes communicate using a *Channel* abstraction that is a shared medium between them. In order to account for propagation loss, ns-3 supports several *PropagationLossModels*. When a node is sending a frame, ns-3 calculates its Received Signal Strength (RSS) – equivalent of the RSSI in the real wireless cards – for the destination node, considering: 1) the TX Power of the source node; 2) the antenna gain of the source node; 3) the attenuation calculated by the PropagationLossModel associated to that Channel considering the distance between the nodes; 4) the antenna gain of the destination node. While in a real node the wireless card reports the RSSI – an indicator that greatly depends on the Auto-Gain Control (AGC) being applied on the RX circuit – ns-3 uses the actual (theoretically) calculated Received Signal Strength (RSS). Note that the AGC affects both the RSSI and the Noise Floor reported by the real wireless cards. As such, when reproducing the radio link quality in ns-3, **we consider the SNR of the real system**. We can reproduce the real-world SNR by setting a user defined RSS in ns-3 equal to the ns-3 noise floor plus the real SNR. In [5] we presented a proof of concept version of the *TraceBasedPropagationLossModel*, which was capable of reproducing the real radio link behavior considering 1) the peers involved in the communication and 2) the link direction. When a *PropagationLossModel* calculates the receiving power for a give frame, the function *"DoCalcRXPower"* is called. From that function programming perspective, only the *MobilityModels* (positions of the nodes) are known (received by argument), and not the nodes themselves. Due to this fact, the first version of the *TraceBasedPropagationLossModel*, implemented in ns-3, had to know the position of the fixed node as a workaround to detect which SNR trace to use. As a results, it only supports point-to-point wireless links, considering one of the nodes has a fixed position, while the other can be mobile.
- **Trace-based Simulation Settings.** Before running the trace-based ns-3 simulation, the user should consider the following settings to properly reproduce the real-world experiment: 1) **TX Power End, TX Power Start and TX Gain** – no need to alter as the RSS is set based on traces; 2) **RX Gain** – should be kept as "0" so that it is not added to the RSS based on the real traces; 3) **WiFi Standard, WiFi Mac, Frequency, Channel BW and Remote Station Manager** – should be configured with the values assumed in the real-world experiment; 4) **Propagation Delay** – the *ConstantSpeedPropagationDelayModel* should be used; 5) **Propagation Loss** – use the new *TraceBasedPropagationLossModel*; 6) **Error Rate Model** – the *NistErrorRateModel* should be used; 7) **Data Mode and Control Mode** — should be set to the corresponding rates, if constant; 8) **Mobility Model** – use the *ConstantPositionMobilityModel* for fixed nodes, and the *WayPointMobilityModel* for the mobile nodes; 9) **2dRssArray** – Events should be scheduled throughout the simulation to reflect the correct trace values over the simulation time, and can be accessed by *Config::Set*.

3.2 The New Version

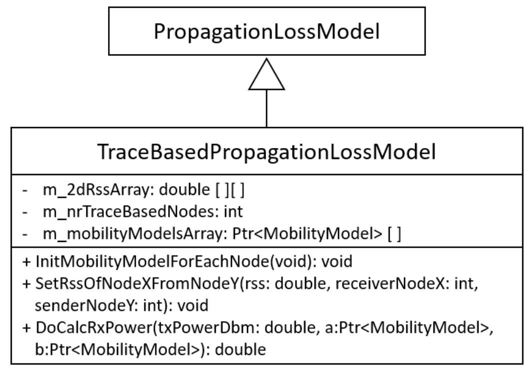

Figure 1: Class diagram for the new version of the *TraceBasedPropagationLossModel*

The new version of the *TraceBasedPropagationLossModel*, depicted in Figure 1, builds upon the first version and introduces the support for multiple access scenarios and multiple mobile nodes. It remains a subclass of the existing *PropagationLossModel*, but now it features new attributes and methods.

The following new attributes are includes in the new *TraceBasedPropagationLossModel*:

- **m_2dRssArray** – a two-dimensional array to store the current RSS (in dBm) of all nodes with respect to all other nodes. Table 1 represents an example snapshot of this two-dimensional array for a trace-based ns-3 simulation containing three nodes experiencing asymmetric SNR.
- **m_mobilityModelsArray** – an array to store the pointer of the *Mobility Model Object* belonging to every node. This helps identifying the sender and receiver of each frame to be sent, whenever the *DoCalcRxPower* method is called.

The following new methods are featured in the new *TraceBasedPropagationLossModel*:

- **InitMobilityModelForEachNode** – method that must be called before the simulation starts, in order to initialize the *m_mobilityModelsArray*;
- **SetRssOfNodeXFromNodeY** – method that updates a given position of the *m_2dRssArray*, allowing the dynamic change of the RSS of the link between Node X and Node Y according to the real experiment traces;
- **DoCalcRxPower** – method that calculates RSS of the node receiving a given frame. Each time the method is called, the *Mobility Model* pointers "a" – the sender node – and "b" – the receiver node – are matched with the ones present in

Table 1: Example of the 2D RSS (in dBm) array for a trace-based ns-3 simulation containing three nodes, where the lines represent the receiver nodes and the columns represent the sender nodes

	TxNode0	TxNode1	TxNode2
RxNode0	-	-60	-70
RxNode1	-65	-	-40
RxNode2	-71	-45	-

m_mobilityModelsArray to find the ids of the corresponding nodes. These ids are then used to fetch the respective RSS value from the *m_2dRssArray* two-dimensional array.

As in the first version [5], the SNR is used in the *ErrorRateModel* to calculate the probability of receiving a frame with errors and dropping it. The *ErrorRateModel* used is the *NistErrorRateModel*, which is considered in the ns-3 documentation to be a more realistic one for OFDM modulations, compared to the *YansErrorRateModel*.

The code of the new version of the *TraceBasedPropagationLossModel* is available for download in [4]. Instructions on how to add and use this model in ns-3 version 3.27 are also included.

4 EVALUATION OF THE NEW VERSION

In order to validate the new version of the *TraceBasedPropagationLossModel*, two independent tests were performed: 1) a functional test, to assure the correct operation of the model; 2) a Multiple Access Trace-based ns-3 simulation, reproducing the network conditions recorded in a laboratory experiment using three Wi-Fi nodes.

4.1 Functional Testing

In order to assess whether the new module is capable of reproducing the link quality configured in the two-dimensional array of RSSs – *m_2dRssArray* –, a simple simulation scenario was created in ns-3. The simulation scenario consisted of three IEEE 802.11a nodes configured in channel 36 (center frequency of 5180 MHz) with 20 MHz of bandwidth. The three nodes were set fixed throughout the simulation using the *ConstantPositionMobilityModel*. Node1 ran an *UdpEchoServerApplication*, while Node0 and Node2 ran an *UdpEchoClientApplication*. The clients sent UDP packets with 1400 bytes to Node1 at a rate of 1 packet per second. The packet capture option was enabled in the three nodes. The IP addresses configured for Node0, Node1, and Node2 correspond, sequentially, to the range of 10.0.0.1–10.0.0.3. The RSS for each node was configured based on the example values presented in Table 1. The simulation scenario is defined in file “first_scenario_2018.cc” available for download in [4].

After running the simulation, we obtained three output *.pcap* files generated by ns-3 with *RadioTap* header included. By opening those files in *Wireshark* we got the data represented in Figure 2, Figure 3, and Figure 4. Each node was able to capture the network traffic, even for unicast flows between its neighbors, and the RSSI values reported by *Wireshark* correspond to the values defined by the trace. This proves the new version of the *TraceBasedPropagationLossModel* is working as expected, now with support for Multiple Access wireless scenarios.

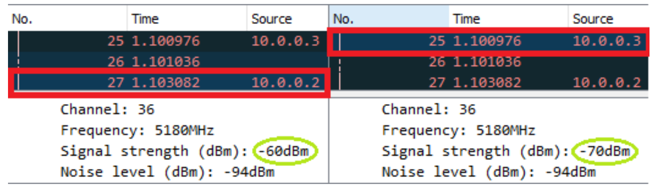

Figure 2: Wireshark screenshots showing Node0 RSSI perspective of the network

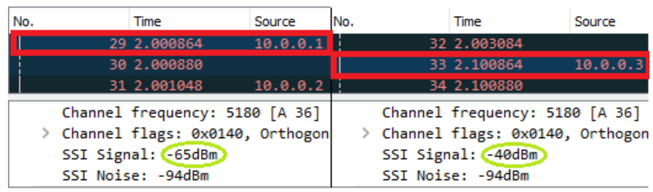

Figure 3: Wireshark screenshots showing Node1 RSSI perspective of the network

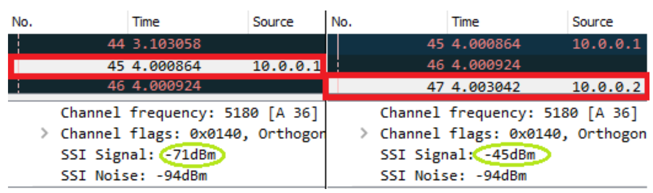

Figure 4: Wireshark screenshots showing Node2 RSSI perspective of the network

4.2 Setup and Experimental Results

In order to obtain the necessary real-world experimental traces for feeding the *TraceBasedPropagationLossModel*, we setup a laboratory testbed composed of three static Alix 3D3 nodes (see Figure 5) with MiktoTik R52 IEEE 802.11a/b/g cards configured to the IEEE 802.11a standard. These nodes were placed on the floor of a storage room, at a distance of 2 m between each other, as illustrated in Figure 5. There were no obstacles between the nodes, and the antenna of each node was oriented vertically. The LEDE operating system was running in each node and the clocks were synchronized using Network Time Protocol (NTP), so that the traces generated by the nodes represented a correct collective snapshot of the network link quality at each moment. The Wi-Fi cards operated with auto-rate in channel 36 (center frequency of 5180 MHz) with 20 MHz bandwidth, TX Power set to 10 dBm (to lower the SNR and incite auto-rate adaptation), and a 3 dBi dipole antenna (Wi-Fi diversity disabled). There were no other concurrent Wi-Fi networks operating in overlapping frequency channels.

Based on our hands-on experience with testbeds and different Wi-Fi cards, we know that there are differences of effective TX Power and RX sensitivity between them, even if the same model is used. This becomes even more apparent when running experiments with extensively used hardware. An interesting point to note is that the hardware components wear can affect more one of the TX/RX functions than the other. This aspect, by itself, represents a source of

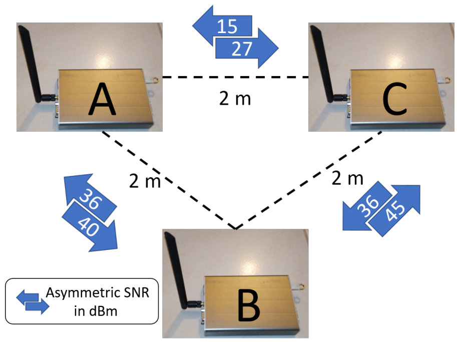

Figure 5: Diagram of the real testbed used for the wireless experiments, with the average SNR measured per link direction

possible link asymmetry alongside with different noise and interference exposure for each node. In practice, this is one further reason why experimental results are difficult to reproduce in simulation. Recognizing this fact we selected, on purpose, Wi-Fi cards from the same model (MikroTik R52) but subjected to different wear levels. Figure 5 presents the average SNR values measured throughout all the experiments, clearly showing that: 1) NodeA has less sensitivity on its RX circuit; 2) NodeC has TX Power lower than expected; 3) NodeB is the card with better TX/RX performance. In conclusion, although at first sight this testbed scenario would be creating optimal network conditions, the aforementioned characteristics create a richer scenario to test the *TraceBasedPropagationLossModel* in a Multiple Access scenario with link heterogeneity and asymmetry. Also, knowing these characteristics beforehand, leads us to expect the worst network performance in flows from C to A.

In all experiments we used *iperf2* to generate UDP flows with an application rate of 54 Mbit/s, which exceeds the maximum possible effective rate of an 802.11a link, usually between 28-32 Mbit/s. The objective was to perform all the experiments using the full channel capacity, so that we could latter compare the same limits in simulation. In each experiment, we generated the necessary UDP flows during 60 seconds and recorded the obtained average UDP throughput per second, alongside with traces containing the average values of SNR per second for each possible bidirectional link. The closer the UDP throughput obtained using the TS approach is to the one measured in the real experiment, the better the *TraceBasedPropagationLossModel* and ns-3 are reproducing the real system.

Our objective was to test the network throughput achievable in each direction of every possible wireless link in the 3-node testbed. Since the three nodes are in wireless range, the first six experiments (see Table 2) consisted in each node generating one-hop traffic flows to each of its two neighbors. Table 2 presents the details of each flow tested per experiment, together with the obtained average UDP throughput. Because of the lower link SNR, the worst performance was measured in Exp.#2 and Exp.#5 – communication between NodeA and NodeB – with Exp.#5 getting particularly low throughput (5.9 Mbit/s), when compared to the 21.7 Mbit/s measured in Exp.#1, for example.

Then, we ran experiments for testing the multiple access scenario, where each node acted as a sink for simultaneous flows originating from its two neighbors. The three resulting experiments are presented in Table 3. As expected by the lower SNR between NodeA and NodeC, Exp.#7 and Exp.#9 were the ones exhibiting the most asymmetric UDP average throughput per flow, achieving respectively 4.1 Mbit/s and 9 Mbit/s.

4.3 Trace-Based ns-3 Simulation

To reproduce such scenarios via Trace-Based ns-3 simulation, the ns-3 scenarios "expX_scenario_2018_trace.cc" were coded to replicate the experimental setup, with "X" representing the number of the experiment. The average real SNR experimentally collected each second was used as a basis; these ".cc" files, as well as the ".csv" files with SNR traces are also available in [4]. In ns-3, we generated the UDP traffic using the ns-3 *OnOffApplication* traffic generator and measured the average UDP throughput for the same exact set of experiments. The same was considered for pure simulation in order to compare what would be the results assuming the *FriisPropagationLossModel* and stable and symmetric link qualities; the Pure Simulation scenarios "expX_scenario_2018_PURE_SIM.cc" are also available in [4]. Table 2 and Table 3 show the average UDP throughput results obtained as well as the relative error with respect to the experimental results.

The maximum throughput achieved in simulation is higher than in the real experiments. This could be related to limitations on the performance of the real hardware that are not accounted in ns-3, such as the time of packet buffer copy and processing operations that, in reality, take time and reduce the efficiency of the channel. This is more evident for IEEE 802.11a as it does not perform frame aggregation, so every processing inefficiency in the Wi-Fi card is "amplified" by that fact, especially in high SNR scenarios where every small amount of time not sending packets at the highest rate possible – 54 Mbit/s – increased the gap between simulation and experimental results. Because of this limitation, every experiment for which the flows are transmitted through high SNR links do not bring added value to this comparison. The TS approach is only able to lower the error from simulation results when the real SNR decreases to a point that it triggers the auto-rate adaptation mechanism caused by packets being lost and re-transmitted. Due to this fact, we now focus our analysis on the results for the links that do not max out the Wi-Fi throughput performance.

Considering the less ideal links, the true benefit of the TS approach becomes apparent, as depicted by the results highlighted in bold in Table 2 and Table 3. For instance, in Exp.#2 the error drops from 47.9% to 17.7%, while in Exp.#5 it drops from 379.7% to 67.8%. In every case, considering the individual flows of all the nine experiments, the TS approach lowered the error. On average, from 91.2% to 34.2%, for the single flow scenarios, and from 63.7% to 21.8%, for the multiple flows scenarios, when compared to Pure Simulation.

4.4 Trace-Based ns-3 Simulation based on High SNR Sampling Rate

There were cases, such as Exp.#5, where the relative error was still considerably high, although much lower than in pure simulation.

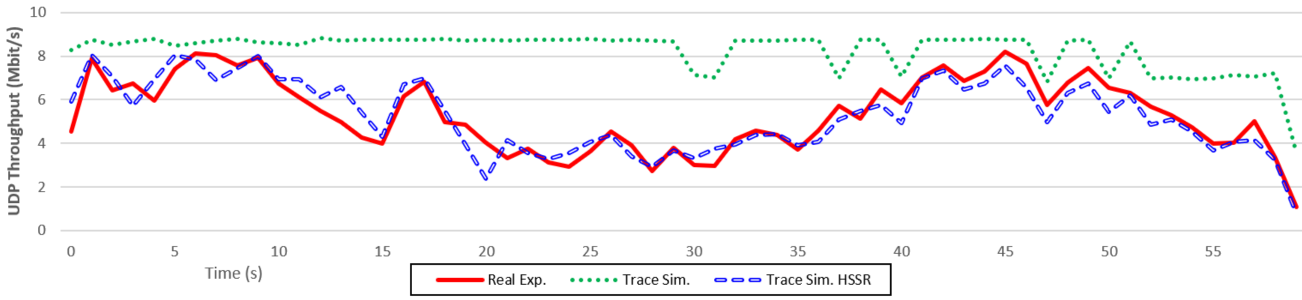

Figure 6: Comparison of UDP Throughput per second measured during 1) the repetition of real Exp.#5; 2) the corresponding Trace-Based Simulation based on traces containing the real SNR average per second; and 3) the corresponding Trace-Based Simulation "High Definition" based on traces containing the real SNR with a per-packet resolution

Table 2: Average UDP throughput results obtained for each individual link, one flow direction at each time, in a Real Experiment, a Trace-Based Simulation, and a Pure Simulation, including the relative error with respect to the Real Experiment results

		Average UDP Throughput (Mbit/s)			*Relative Error*	
Exp.#	**Flow**	**Real Exp.**	**Trace Sim.**	**Pure Sim.**	**Trace Sim.**	**Pure Sim.**
1	A->B	21.7	28.5	28.5	31.3%	31.3%
2	A->C	19.2	22.6	28.4	**17.7%**	**47.9%**
3	B->A	21.8	28.4	28.4	30.3%	30.3%
4	B->C	21.1	28.3	28.3	34.1%	34.1%
5	C->A	5.9	9.9	28.3	**67.8%**	**379.7%**
6	C->B	22.8	28.2	28.2	23.7%	23.7%
					Avg. all flows	
					34.2%	**91.2%**

Table 3: Average UDP throughput results obtained for two simultaneous flows from different senders to each sink node, in a Real Experiment, a Trace-Based Simulation and a Pure Simulation, including the relative error with respect to the Real Experiment results

		Average UDP Throughput (Mbit/s)			*Relative Error*	
Exp.#	**Flow**	**Real Exp.**	**Trace Sim.**	**Pure Sim.**	**Trace Sim.**	**Pure Sim.**
7 Sink A	B->A	11.3	12.2	14.3	**8.0%**	**26.5%**
	C->A	4.1	5.7	14.1	**39.0%**	**243.9%**
	Total	15.4	17.9	28.4	16.2%	84.4%
8 Sink B	A->B	11.9	14.3	14.3	20.2%	20.2%
	C->B	14.1	14.0	14.0	0.7%	0.7%
	Total	26.0	28.3	28.3	8.8%	8.8%
9 Sink C	A->C	9.0	4.0	14.2	**55.6%**	**57.8%**
	B->C	20.8	19.3	13.9	**7.2%**	**33.2%**
	Total	29.8	23.3	28.1	21.8%	5.7%
					Avg. all individual flows	
					21.8%	**63.7%**

That could be related to the fact that we were using real SNR averages per second, maintaining that value constant during each simulated second. We could easily add a random component to make the SNR less stable, such as in reality, by using, for example, a Normal, Rayleigh, or Rician distribution. Still, we would be simulating again, and not using the actual traces to reproduce reality.

As such, we decided to repeat the experiment with the biggest relative error we got using the TS approach – Exp.#5 –, and assessed whether the error could be reduced by increasing the real SNR sampling rate.

The results after rerunning Exp.#5 with a higher real SNR sampling rate (once per packet received) are presented in Table 4. The table shows the average UDP throughput and the relative error of

Table 4: **Average UDP throughput results obtained when rerunning *Exp.#5* and considering a Trace-Based Simulation - High SNR Sampling Rate (HSSR), a Trace-Based Simulation, and a Pure Simulation, including the relative error with respect to the Real Experiment results**

		Average UDP Throughput (Mbit/s)				*Relative Error*		
Exp.#	**Flow**	**Real Exp.**	**Trace Sim. HSSR**	**Trace Sim.**	**Pure Sim.**	**Trace Sim. HSSR**	**Trace Sim.**	**Pure Sim.**
5 (second run)	C->A	5.4	5.3	8.3	28.2	**1.4%**	**53.9%**	**426.2%**

all simulation-based alternatives to reproduce the real experiment. In this case, we can see the relative error dropped from 426.2%, for pure simulation, to 53.9%, for Trace-Based simulation. But, using the much higher number of SNR samples we were able to reduce even further the relative error to 1.4% only. Figure 6 shows a plot comparing the instantaneous throughput per second for the trace-based simulation, the trace-based simulation with high SNR sampling rate, and the real experiment. The pure simulation throughput was not plotted as it remained almost constant at around 28 Mbit/s, and would reduce the throughput resolution needed to better compare the other results. We can see that trace-based simulation results using per packet SNR samples was significantly more accurate, almost overlapping with the real UDP throughput results. This proves the potential of the TS approach, but comes at a price: 1) we need more storage space to save per packet SNR samples from real-experiments; 2) the higher number of events to update the SNR throughout the simulation can increase the simulation time. Nevertheless, we argue that it is computationally lighter to just read and update SNR values that are already known, than using simulation models to calculate new values. Thus, we believe that even with the higher rate of SNR updates the running time should remain lower than in pure simulation.

5 CONCLUSIONS AND FUTURE WORK

Wireless networking R&D is increasingly dependent on experimentation to further evaluate and validate a solution in real environment. Experiments are also increasingly complex and difficult to repeat and reproduce, even in controlled federated testbed scenarios such as the ones from Fed4FIRE+ and GENI. INESC TEC has already proposed, in the past, an approach for replicating real experiments by feeding traces such as SNR and node positions into ns-3. This approach combines the ns-3 TCP/IP and MAC simulation capabilities with the real world physical characteristics captured during real experiments. Although being a preliminary validation, that proof-of-concept implementation showed great potential. In this paper we presented a new version of the *TraceBasedSimulationLossModel*, now adding support to Multiple Access wireless scenarios, which greatly broadens the scope for its application.

We implemented the new *TraceBasedPropagationLossModel* in ns-3.27. We also configured a laboratory testbed and executed nine experiments which were later reproduced in ns-3 using the new model to assess its correct operation and to further validate our approach. Our evaluation confirmed its capability to produce results much closer to the ones from real experiments than using pure simulation. Reproducing the real SNR with a per-packet resolution, we were able to demonstrate an example where the Trace-Based simulation reproduction relative error came down to 1.4%.

As future work, we will continue to further improve the *Trace-BasedPropagationLossModel* and develop a framework to assist the related processes of traces capturing, managing, reusing, and sharing. We expect to extensively test this approach in a broader set of scenarios, in the near future, throughout a recently approved H2020 Fed4FIRE+ project, where we will focus on leveraging ns-3 simulation as a complementary Offline Experimentation tool for such wireless testbeds.

ACKNOWLEDGMENTS

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT – Fundacao para a Ciencia e a Tecnologia within project POCI-01-0145-FEDER-016744.

The first author would like to thank the support from the Portuguese Foundation for Science and Technology (FCT) under the fellowship SFRH/BD/69051/2010.

REFERENCES

[1] P. Agrawal and M. Vutukuru. 2016. Trace Based Application Layer Modeling in ns-3. In *2016 Twenty Second National Conference on Communication (NCC)*. Guwahati, India, 1–6. https://doi.org/10.1109/NCC.2016.7561126

[2] R. Campos, T. Oliveira, N. Cruz, A. Matos, and J. M. Almeida. 2016. BLUECOM+: Cost-effective Broadband Communications at Remote Ocean Areas. In *OCEANS 2016 - Shanghai*. Shanghai, China, 1–6. https://doi.org/10.1109/OCEANSAP.2016.7485532

[3] Fed4FIRE+. 2018. Fed4FIRE+ – The Largest Federation of Testbeds in Europe. (Feb. 2018). https://www.fed4fire.eu/

[4] H. Fontes. 2018. Source Code for the TraceBasedPropagationModel – Use in ns-3.27. (Feb. 2018). http://telecom.inesctec.pt/~hfontes/trace_based_propagation_loss_model2018.zip

[5] H. Fontes, R. Campos, and M. Ricardo. 2017. A Trace-based ns-3 Simulation Approach for Perpetuating Real-World Experiments. In *Proceedings of the Workshop on ns-3 (WNS3 17)*. ACM, Porto, Portugal, 118–124. https://doi.org/10.1145/3067665.3067681

[6] R. Fontes, M. Mahfoudi, W. Dabbous, T. Turletti, and C. Rothenberg. 2017. How Far Can We Go? Towards Realistic Software-Defined Wireless Networking Experiments. *Comput. J.* 60, 10 (2017), 1458–1471.

[7] S. Keranidis, W. Liu, M. Mehari, P. Becue, S. Bouckaert, I. Moerman, T. Korakis, I. Koutsopoulos, and L. Tassiulas. 2012. Concrete: A Benchmarking Framework to Control and Classify Repeatable Testbed Experiments. In *FIRE Engineering Workshop*. Ghent, Belgium.

[8] mininet wifi. 2018. mininet-wifi – Emulator for Software-Defined Wireless Networks. (Feb. 2018). https://github.com/intrig-unicamp/mininet-wifi

[9] P. Owezarski and N. Larrieu. 2004. A Trace Based Method for Realistic Simulation. In *2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577)*, Vol. 4. Paris, France, 2236–2239. https://doi.org/10.1109/ICC.2004.1312915

[10] SUNNY. 2018. SUNNY - Smart Unattended Airborne Sensor Network for Detection of Vessels Used for Cross Border Crime and Irregular Entry. (Feb. 2018). http://www.sunnyproject.eu/