Generation of Realistic 802.11 Interferences in the Omnet++ INET Framework Based on Real Traffic Measurements

Juan-Carlos Maureira - Olivier Dalle
Join project team MASCOTTE
INRIA, I3S, CNRS, Univ. Nice Sophia, France.
B.P. 93, F-06902 Sophia Antipolis Cedex, FRANCE.
{jcmaurei|odalle}@sophia.inria.fr

Diego Dujovne
Project team PLANETE
INRIA Sophia Antipolis Méediterrannée
ddujovne@sophia.inria.fr

ABSTRACT

Realistic simulation of 802.11 traffic subject to high interference, for example in dense urban areas, is still an open issue. Many studies do not address the interference problem properly. In this paper, we present our preliminary work on a method to recreate interference traffic from real measurements. The method consists in capturing real traffic traces and generating interference patterns based on the recorded information. Furthermore, we assume that the coordinates of the sources of interference in the real scene are not known a priori. We introduce an extension to Omnet++ INET-Framework to replay the recreated interference in a transparent way into a simulation. We validate our proposed method by comparing it against the real measurements taken from the scene. Furthermore we present an evaluation of how the injected interference affects the simulated results on three arbitrary simulated scenarios.

Categories and Subject Descriptors

C.2.1 [**Network Protocols**]: Wireless Communications; C.4 [**Performance of Systems**]: Measurement techniques—*Reception power measurements*; I.6.5 [**Model Development**]: Modeling methodologies—*Wireless Interference model*

1. INTRODUCTION

Although simulations of wireless environments use different techniques to take into consideration the nature of interference, the use of direct observation of the reality, while very relevant to increase the accuracy of the results, is not a common practice.

In this paper, we propose a method to incorporate the traffic and the effect of real observations in order to improve the realism of wireless simulations. Our method is based on 802.11 traffic sampling and the generation of interfering background traffic into the simulation from the captured packet traces. Since interference generation is based on 802.11 packet captures, we limit our approach to interference related to 802.11-compliant devices.

The interference scenarios, provided by our method, are represented in two dimensions: spatial and temporal. On the spatial dimension, the traffic injected is received by the simulated wireless nodes with a calculated reception power obtained from the signal loss between the receiver and a virtual position of the interfering transmitter. This *virtual position* is calculated from the traces captured during a measurement campaign. On the temporal dimension, the packet timing is reproduced from the captured traces, generating two kind of interference scenarios: the first one, where the simulated system reacts to the interfering traffic, but has no interaction with the interfering sources; and the second one, where the simulated system and the interfering traffic have mutual interactions.

Within the method, we provide a sampling technique and the means to generate the interfering traffic in the OMNeT++ simulator. We base our method on the OMNeT++ INET-Framework, since the current version (v3.3) includes significant support for studying 802.11 systems. We validate our method by comparing real measurements of reception power with simulated measurements produced by our method. Additionally, we present an evaluation of how the injected interference affects an artificial simulated scenario, in which we evaluate the impact of the hidden station problem on the communication of two wireless hosts associated to an Access Point. The evaluation is conducted with and without external interferences, showing that the results are different enough as to deliver misled or incomplete conclusions. The contributions of this paper are (1) a method to generate interferences from real measurements and, (2) the integration of the method into the OMNeT++ INET-Framework.

The rest of the paper is organized as follows: in Section 2 we present related works on interference models based on real measurements for 802.11 simulated systems; in Section 3 we explain our method. In Section 4 we describe the validation methodology and we show the validation results. In Section 5 we introduce an example of how the generated interference affects the results obtained from a wireless artificial simulated scenario, and finally, in Section 6 we draw our conclusions along with proposals to continue this work.

OMNeT++ 2009, Rome, Italy.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5602
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5602

2. RELATED WORK

The study of wireless networks through simulations have often been questioned [6][2]. The common complain is the lack of realism on several aspects, such as interference representation, radio propagation models and PHY modeling. Normally, the realism delivered by simulations is based on a complex theoretical modeling. But improvements based on real measurements have not been explored with the same emphasis. Reis et al. [9] have examined how a measurement-based approach could improve the precision of wireless models: creates a channel model which uses packet delivery probability ratio to decide on the packet decoding success. Additionally, Kashyap et al.[4] models three parameters to include in the channel model: the reception power value, the deferral probability and packet delivery probability ratio. Our approach differs from these two former works on the method to improve the simulation process: We include 802.11 interference sources themselves to participate within the simulation while avoiding any change on the physical channel model.

The two common approaches to simulate interference and radio propagation are either to use a complex and computational expensive models[7] or a simple one with the risk of obtaining misleading conclusions[6]. In [3], Iyer et al. point out that a complete model should include accurate descriptions for signal-to-interference-and-noise-ratio (SINR) calculation to determine a packet reception event. We use this statement to focus our approach on how to improve a simple propagation model, such as Pathloss free space propagation, with real measurements in order to improve the SNIR calculation by means of the inclusion of external interfering traffic, sampled from the reality. In this direction, the Omnet++ INET-Framework propagation model uses the additive interference model[1] to calculate the SNIR and packet reception. Improvements based on a theoretical approach have been presented in [1] and [10], adding Non-line of sight (NLOS) effects into the Pathloss calculation. Our work explores a measurement-based approach in order to improve the realism of the results delivered by the OMNeT++ INET-Framework.

3. METHOD DESCRIPTION

Before we present a description of our method, we introduce the concepts that we use along this paper. We call **scene** to the collection of events within a limited space during a certain period of time. In our context, **sampling** the scene corresponds to describe how the ongoing traffic is observed, in terms of reception power and airtime usage, from several locations into the scene. We call **interference** to all the unwanted signal at the receiver that degrades the SNR to a level where the packet cannot be decoded correctly. We denote **uncoordinated interference** (UI) to those signals coming from sources which share the physical channel but do not participate on the 802.11 DCF, such as Bluetooth, Zigbee devices, DECT wireless telephones and microwave ovens. On the other side, we define **coordinated interference** (CI) to that generated from sources sharing the same channel and participating on the 802.11 DCF. Finally, we define the **Collision Domain** as the set of wireless nodes, under mutual coverage, that suffer coordinated interference.

[1]From a node point of view, all the rest of ongoing transmissions are considered as noise when calculating noise level

The proposed method is to collect several traffic traces from a scene by means of a measurement campaign, use these traces to discover the position of all the detectable sources, and then, inject the traces from the calculated positions into the simulator to produce interfering background traffic.

In the following, we describe each step in detail: the traffic sampling, the localization of the sources, and how to use this information to generate interfering traffic into the Omnet++ INET-Framework simulation model.

3.1 Sampling

Assuming the scene is larger than a coverage area of a 802.11 station, the traffic needs to be sampled from multiple locations. At this point, two methods of sampling must be considered. Simultaneous sampling from multiple locations or sequential sampling. The main difference is that simultaneous sampling allows the detection of collisions, but traces must be *synchronized* to remove ties.

We have prepared a sampling tool to simplify the measurement campaign. It consists in a set of probes and a sink server. The probes are based on a modified version of the Kismet sniffing server[5]. It runs in a laptop or in a OpenWRT capable device (such as Linksys WRT54G wireless router). The sink server must to be wired to each probe in order to record the required information from the captured traffic. The tool records the reception power, source mac address, packet size and transmission rate of each captured packet. After capturing, the sink server performs statistical calculations on each detected source, estimating the distance between the probe that is seeing it and the source itself.

Finally, the tool will report all the sources detected, the probes that detect them and a probe-source distance estimation based on Pathloss propagation model.

3.2 Discovering the Location of the Sources

The localization of the sources is obtained using the well-known triangulation technique, based on the distance estimation given by the theoretical propagation model. Additionally, we use cluster analysis to discard false positions reported from wrong-behaved measurements or not congruent distance estimations. This cluster analysis is done by using a sparse matrix with interval arithmetic. In a 2-D sparse matrix, the X-Y coordinates are calculated by using intervals. Figure 1 shows the process. When two probes detect the same source, the triangulation is executed to estimate two source positions, (x_0, y_0) and (x'_0, y'_0). These positions are inserted into the sparse matrix. When another probe detects the former source, the same procedure is performed once for each possible pairing between the probes. Now, the new source coordinates (x_1, y_1) and (x'_1, y'_1) are compared to the previous one by using interval arithmetic. This results in: $(x_0, y_0) = (x_1, y_1) + (dx, dy)$, or by coordinate: $x_0 = x_1 + dx$ and $y_0 = y_1 + dy$. Then, the algorithm of insertion in the sparse matrix uses the same location if the coordinates are equal according to the previous boolean expressions. After the position of the source has been approximated by all the available probes who detect that source, the system defines its position as a **virtual position**.

3.3 Virtual Position of the Sources

When using the Pathloss propagation model to estimate the distance between a transmitter and receiver, there are

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5602
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5602

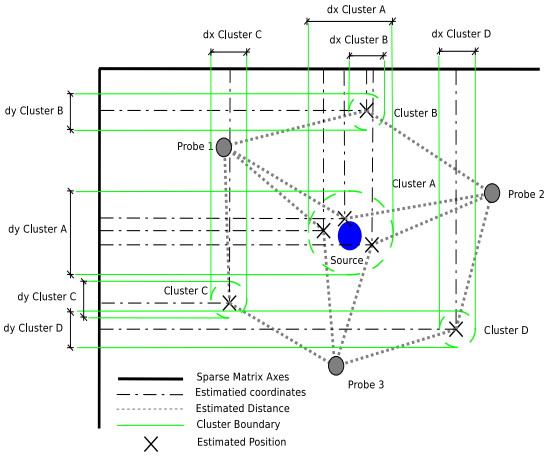

Figure 1: Triangulation with Cluster Analysis

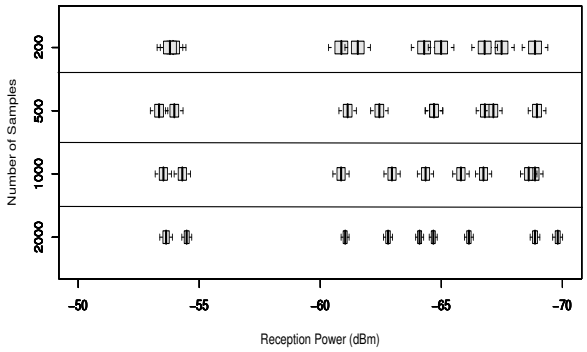

Figure 2: The received power means with a 99% confidence interval computed with (a) 200, (b) 500, (c) 1000 and (d) 2000 recorded samples.

several factors to be also taken into account, for example, non-line-of-sight paths, obstacles, reflections and dispersion. OMNeT++ INET Framework assumes a fixed Pathloss exponent in all the simulation playground. This means, no obstacles, no reflections and always line-of-sight, or, in other words, a circular coverage radio, which is not realistic. Additionally, the transmission power is also assumed to be fixed for all the wireless stations in the simulation, which also lacks realism. It is true that each wireless device uses a fixed transmission power[2]. But, this does not mean that all the stations must use the same one. Nevertheless, we explore the possibility **to compensate** the induced error by means of misplacing the position of the source. The objective of misplacing the sources is to measure, inside the simulation, a reception power similar to the measurements taken from the traffic traces. This change of position is expected to compensate attenuations due to obstacles, but it will certainly not deal with reflections. Reflections normally are shown as a reception signal variation in time. Thus, if the recorded reception power variation is small enough, an approximate distance estimation can be achieved.

3.3.1 *Experimental Validation*

We analyze how steady must be the average of the reception power and how large must be the standard deviation to have a reasonable distance estimation by means of experimental measurements. We have recorded, with a single probe, the reception power from an already-known source, in line of sight, outdoor, at increasing steps, starting at 16m, with steps of 8m, ending at 80m.

Using a factorial analysis of variance (ANOVA), we found statistical differences between two consecutive sampling blocks, i.e. 24 and 32 meters. Figure 2 illustrate the ANOVA results.

The graph shows the reception power mean for each distance block, based on the sampled data. The graph (a) shows overlapping mean intervals at successive blocks, which does not allow a statistical difference between them. When the number of samples is increased to 500 on (b), there are still three groups which are overlapped. Furthermore, on (c), when 1000 samples are used, there remains only one undifferentiated block; while an increase to 2000 samples as in (d) finally reaches the objective of mean estimation with non overlapped blocks, which allows us to state that the reception power mean is statistically different by each sampled block.

In conclusion, the distance estimation based on the mean value of the received power can be calculated, and the position can be obtained with a precision of 8m when sample size is bigger than 2000 values.

3.3.2 *The Receiver point of View*

As we are able to estimate a reasonable reception power interval for packets received in a sampling location, in terms of sampling variance, now we discuss about how to deal with obstacles and the scene topology by means of misplacing the position of the sources.

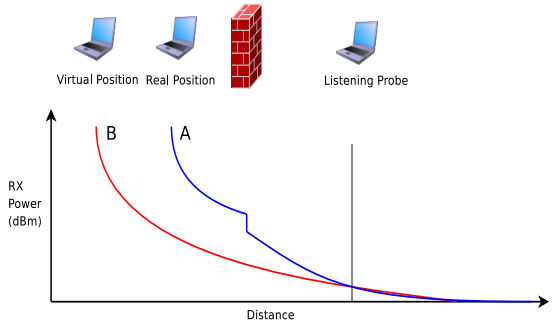

Figure 3: Signal attenuation in the Pathloss Model.

The Figure 3 illustrates how the signal is attenuated between a transmitter and receiver (listening probe). The line A depict the signal strength as a discontinuity going through a obstacle, while the line B shows the Pathloss model signal attenuation. The obstacle attenuates the signal in a unusual way (discontinuity), making the receiver to perceive a weaker signal than the perceived one if the obstacle was not there. We obtain a similar attenuation if we pull away the transmitter. Therefore, misplacing the source and latter using the misplaced position is equivalent to receive the same attenuated signal at the receiver. Furthermore, when trian-

[2]Stations do not change its transmission power along the time as a normal behavior

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5602
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5602

gulating the source position by using these "corrected distances" between probe-source, the cluster analysis will group all the estimations that suggest a convergent position, discarding the estimations that do not contribute to enforcing it.

In summary, we denote **virtual position** to the required position to produce a reasonable approximation, in the simulator, of the receiver power for all the probes that have seen the source.

3.4 Replaying Interference into OMNeT++

While exploring the possible scenarios that the OMNeT++ INET-Framework can represent, we realize that the hidden station scenario and the simultaneous equal discrete backoff choices are, both, possible situations; It is not the case of interferences caused by multipath fading, that is not supported yet into the simulator. Hence, our proposed solution is to include *Shadow Sources* (SS), based on the triangulated positions, replaying the recorded traffic in order to regenerate the mentioned interfering scenarios. The definition of each SS must be similar to a regular simulated wireless host, but we need to introduce the feature to enable or disable the Carrier Sense (CS) mechanism in order to replay the recorded trace exactly in the same way that it was recorded. Enabling or disabling CS mechanism will change the airtime distribution on each Collision Domain. Hence, CS disabled shadow sources will access the medium no matter who is transmitting, producing collisions in the studied system only. If we analyze this fact, we realize that it is not a realistic scenario, since the studied system should change also the SS injected traffic. Nevertheless, if they did so, the recorded scene would no longer exist, changing our referenced context. So, we define two types of interference interactions between the studied system and the SS. The first one, where the studied system reacts in front of external traffic, but the external traffic do not react in front of the studied system traffic; and the second one, where the studied system also changes the interfering traffic. The first one has the advantage to evaluate the studied system with a fixed airtime distribution on each Collision Domain. The second one is the most realistic one, but we can not preserve the locality and time awareness of the studied system, since the mutual interaction produces a different scene.

Based on this analysis, we define all the necessary elements to be implemented into the simulator as an extension of the OMNeT++ INET Framework (v3.3):

- **Shadow Source (SS)** : wireless host containing a Trace Player, IEEE80211Mac module (implementing CSMA/CS mechanism with an pass-through switch to enable or disable it), and a IEEE80211Radio module (PHY).
- **Trace Player** : simple module that reads a single trace file, one recorded packet at once, and generates the resulting simulated packet datagram (packet size mostly), and sends it to the Link Layer to be transmitted.
- **IEEE80211Mac** : the same already implemented in the INET Framework, but with the addition of a new flag that allows to bind the *uppergateIn* port with the *lowergateOut* port in order the bypass the CSMA/CA mechanism.
- **WifiWorld** : compound module that contains the shadow sources. This compound is placed into the simulated playground in order to separate concerns between the studied system and the SS that will produce the interfering traffic. This compound must have at least the same size as the main simulation playground.
- **Channel Controller** : the same already implemented in INET Framework, but now implementing initialization methods to load the traces and the position of each SS, placing them into the WifiWorld container. Also it must consider WifiWorld coordinates as playground coordinates.

3.5 Integration and Execution

The main issue to address when thinking about integrating the proposed method into the simulator is to preserve strictly the **backwards compatibility** with the INET Framework models, and also to minimize intrusion in the code of the already existing simulations. Additionally, to include our proposed method into an already existent simulation it is sufficient to include the *WifiWorld* container with a pointer to the recorded trace to be used. Automatically the *Channel Controller* will use this information to generate all the Shadow Sources and prepare the traffic to be injected in the simulation.

Regarding the execution of simulated models, it is clear that there is an additional overhead to be handled by the simulator (the injected traffic and SS events), causing higher execution times. Nevertheless, as this overhead is strictly proportional to the amount of injected traffic, the resultant overhead will depend on the ratio between the amount of traffic handled by the simulation with and without the SS. Our experiments have shown an overhead in execution time of 30%, when the injected traffic is about 1/3 of the generated traffic by the simulation itself.

Another important issue to address is the way the trace is loaded into memory and scheduled. In order to minimize the impact on the memory used by the event set, the trace is loaded one packet at a time.

4. VALIDATION

We propose to validate our method by comparing the real measurements against the same scene into the simulator. In other words, we build into the simulator the same scene conditions, placing simulated probes into the same sampling locations, and introducing the detected sources into their calculated virtual positions. Then, we replay the real traces from each shadow source, recording the reception power of the captured traffic on each probe. This methodology will allow us to compare each real trace against its simulated one in order to quantify the difference between them. This quantification will give us a hint about how accurate is our method in terms of how the propagation model and virtual positions describe the recorded scene.

4.1 Real Experiment

We have chosen the ground floor level of our laboratory building as our experimental scene. We captured traffic samples for 980 seconds during office hours, in order to have enough samples to check statistically the results. The sampling locations are illustrated on the Figure 4. We choose these sampling locations (noted P1,P2,P3 and P4) based on

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5602
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5602

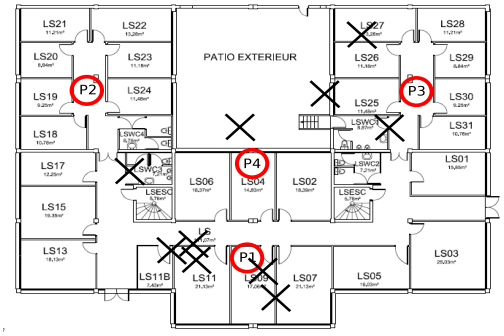

Figure 4: Sampling locations (circles) and detected traffic sources (crosses).

the building geometry and having in mind a minimal effective coverage range of 50 meters. We record traces in the scene in a sequential sampling. After the analysis and triangulation, we were able to detect 10 traffic sources as the Figure 4 depicts.

Remember that the position of each source is **virtual**.

4.2 Simulated Experiment

Following the previously stated methodology of validation, we introduce into the simulation the detected sources as Shadow Sources and we place simulated probes in the same locations P1, P2, P3 and P4. We replay the captured traffic within the simulation, one trace at the time, since the sampling was taken sequentially, and we record the traces resulting for the four simulation runs (one for each trace recorded).

It must be noticed that, among all the real traces recorded, only the traffic transmitted from the detected sources is injected. The rest of the traffic is ignored, since we were unable to detect the location of their sources. Additionally, the Pathloss setting into the simulator was exactly the same as the Pathloss setting used to triangulate positions.

An interesting remark is to implement a probe into the OMNeT++ INET-Framework, some modifications must be introduced into the PHY Radio module in order to implement a *Monitor Mode* to capture the packets in a *promiscuous capture*, adding to each packet a *Radiotap* header [8] in the *ControlInfo* field of the message.

4.3 Results

To evaluate our results, we quantify the difference between the real traces and the simulated ones in two ways: we evaluate how the reception power mean of the traces compares to the simulated ones; and we feed our triangulation algorithm with the simulated traces in order to quantify how different the estimated position of the sources is in comparison with the position calculated from the real traces.

Discussing about the traces, the difference between the samples number on the real/simulated traces, and the detected number of sources in real and simulated experiments are evident when thinking that only the traffic of the detected sources were injected. The interesting number is the detected sources versus significant sources detected. We observe that in the real case, we have to discover several sources by listening packets from them. But, only a small number

Probe	*Real Traces*			*Simulated Traces*		
	N.Samples	*DS*	*SSD*	*N.Samples*	*DS*	*SSD*
P1	72421	141	13	33661	10	9
P2	125521	886	29	39746	8	6
P3	141907	888	29	49357	10	10
P4	126389	148	16	56944	10	10

Table 1: Total number of collisions in the 5 evaluated scenarios. DS: Detected Sources. SSD: Significant Sources Detected.

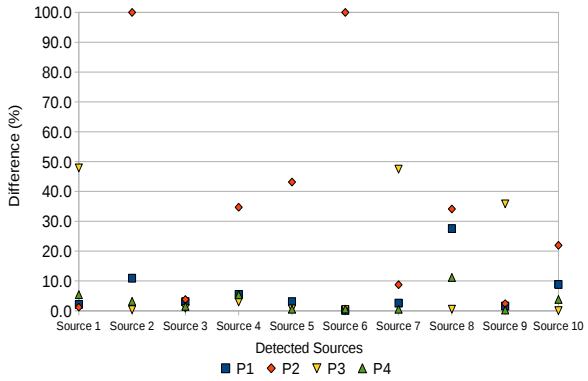

Figure 5: Reception power differences between real traces and simulated traces

were significant (recorded samples number > 2000). Thus, when using all the simulated information to triangulate the positions of the shadow sources, we found 9 of the 10 sources that where considered significant (we miss one source because it was at the playground's border).

In order to analyse the estimations of the reception power, the Figure 5 depicts the differences between the real and the simulated measurements. For each source, we have 4 reception power estimations, one for each probe. In total, we have 10 sources * 4 estimations = 40 estimations. From them, 25 (62.5%) are under 10% error, 13 (32.5%) are between 20% and 50% and 2 (5%) are 100% wrong. All of this considering a sequential sampling of the scene.

In order to analyse the estimation of the position of the sources, we use the same triangulation algorithm for both traces, real and simulated ones. The metric we used to quantify the difference is the euclidean distance between the real and estimated locations. Thus, analyzing these differences, we see that the largest distance is 2.87 meters, the smallest distance is 0.1 meter, and the standard deviation of all distances was 0.96 meters.

In conclusion, this preliminary evaluation suggests that the simulated traces describe the reception power within the scene with a reasonable accuracy. This statement is enforced by the fact that we are able to retrieve similar locations with simulated traces. Nevertheless, this same cross checking could be used as a feedback to recalculate the sources position in order to minimize the perceived reception power error in the simulator. The exploration of this improvement and others is planed as further work.

5. EVALUATION

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5602
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5602

In order to evaluate how the interfering traffic affects a simulated system, we build an artificial scenario. The idea is to evaluate how the hidden station problem affects the communication between two wireless stations. Thus, we introduce our simulation tested as the Figure 6 depicts.

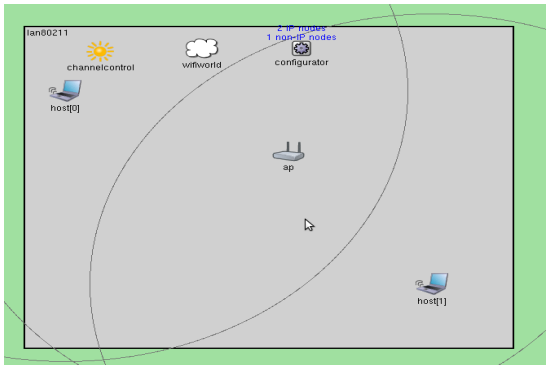

Figure 6: 3 nodes simulated scenario

The station *host[0]* and *host[1]* are already associated to the Access Point (AP) and also both stations are placed according to set an hidden station scenario[11] with the AP. We select three traffic profiles to evaluate how the depicted scenario affects the communications between the stations. We define the first traffic profile as the ICMP Ping profile, with a rate of 10 packets/sec; the second is called TCP FTP profile, representing a file transfer of 30 Mbytes; and finally, the UDP Video Streaming profile, in which we send an unidirectional UDP stream of 80 Kbps in constant bit rate (CBR). Each traffic profile was evaluated, first, without the interference generated by the shadow sources; and then, including the interference patterns generated by the shadow sources. Four replicas of the simulation were configured, each one using a different interfering traffic trace, coming from each of the sampling locations. In summary, we evaluate 5 simulated scenarios.

The simulation parameters are:

- Wireless data rate link was set to 2Mbps.
- The radio transmission power was set to 100mW.
- Sensitivity to -85dBm.
- The signal to noise interference ratio threshold at 4dB.
- The Pathloss coefficient alpha to 4.

5.1 ICMP: Ping profile

The ICMP Ping traffic is usually characterized by the round trip time (RTT). Fig.7 shows the recorded round trip time from *host[1]* to *host[0]* for each scenario (without interference and with the four previously defined interference scenarios). We can see that the RTT value oscillates bellow 4ms with a very low variance (0.466ms) on the non-interference scenario. While we observe the perceived value is higher in average (4.4ms) and also a higher variance (std.dev 2ms) in the four interference scenarios. Confidence intervals calculated on the raw RTT data show statistical difference between the scenario without interference and the four scenarios with interference.

Figure 7: Ping RTT value versus time

5.2 TCP: FTP profile

The TCP incremental sequence number was chosen to analyze this traffic profile. The graph presented in the Figure 8 illustrate the evolution of sequence numbers in time. We choose this parameter to evaluate this profile because we are looking for changes in the TCP behavior and also in how the transfer overall time is affected. The first effect that can be observed on the graphic is the difference in download time, which is 800 seconds for the non-interference scenario, and at least 90 seconds higher for the all interferences scenarios. Also statistical differences in the downloading time can be found between the non-interference scenario and the four interference scenarios when confidence intervals are built using several replications on each scenario with different random seeds.

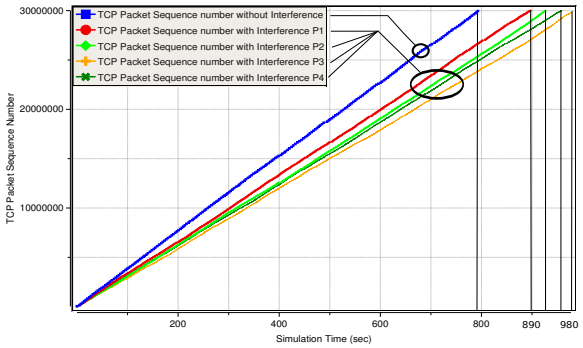

Figure 8: TCP packet sequence number versus time

Examining the other TCP metrics, such as window size and throughput, there are no statistical differences. Nevertheless, small increase in the number of retransmission is detected when interference is introduced.

5.3 UDP: Video streaming

The 80Kbps traffic is generated as follows: UDP packets of 1000 bytes size delivered from *host[1]* at 0.1s intervals. The profile of streaming packet delay can be observed in Fig. 9. It shows stronger variability in all the cases where the interfering traffic is included than in the non-interfering case. Additionally, a zoom of the non-interference scenario between 0s and 25s is given on the lower left side of the plot. Furthermore, the minimal delay with interference traffic is

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5602
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5602

evidently higher than the delay without interference traffic, and it shows a greater variance than the non-interference case.

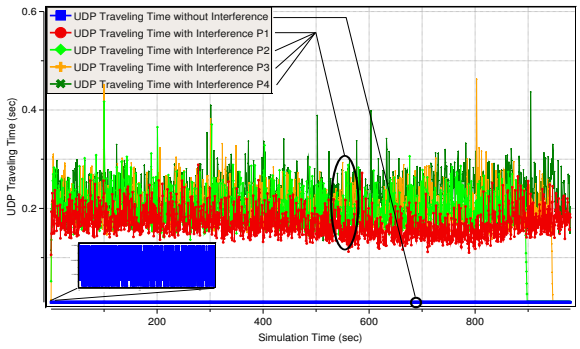

Figure 9: Streaming packet delay versus time

5.4 Layer 2 evaluation

Focusing now on the MAC behavior, we explore the effect of collision occurrences on each simulated scenario by exploring the Backoff and Deferral patterns. Also the Airtime (the channel usage measured in time) is used to explain the previously given results on the layer 3 protocols.

In Table 2, we can observe the influence of the injected traffic on the channel. The table shows a significant increase in the number of collisions on the external enabled interference scenarios. Thus, in the FTP case, we can observe between 30% and 50% more collisions than the interference enabled scenarios, while in streaming case, the number of collisions is between 23% and 41% more. There is also a noticeable increase on the Ping traffic, that is influenced by the number of collisions that delay retransmission.

Scenario	*Collisions Number*				
	Without	*P1*	*P2*	*P3*	*P4*
Ping	0	1734	3518	4327	4122
FTP	205406	262468	285987	303306	302458
Streaming	1808	56235	66746	72023	83304

Table 2: Total number of collisions in the 5 evaluated scenarios

In Table 3 we can see the number of collision events during the simulation from which we have subtracted the collided packets produced by the shadow sources. We observe the packet collisions produced by the system itself. We observe changes from the non-interference scenario due the change of the behavior in back-off occurrence mainly. Now analyzing each traffic profile, for the FTP case, with an interference environment we can see between 10% and 20% more collisions than in the external interference-free case; while for the streaming, the collisions are ten times higher. Furthermore, even though the ping traffic is low, we can observe a significantly high number of collisions with the interfering traffic.

The successive retransmissions (due to the packet loss at the MAC level) help to explain the increment in packet collisions. When a packet is lost at MAC level (no ACK is received), the MAC retransmits the packet a certain number of

Scenario	*Collisions Number*				
	Without	*P1*	*P2*	*P3*	*P4*
Ping	0	32	17	26	27
FTP	205406	228133	234395	246551	236741
Streaming	1808	18243	15997	15761	16514

Table 3: Total number of collisions without external interference in the 5 evaluated scenarios

times until its ACK is received, or the retransmission counter is reached. These retransmissions increment the amount of total traffic being sent to the channel. Thus, the available Airtime will be less, incrementing the probability of collision with another ongoing packet. This situation is evident in the ICMP Ping and Streaming profiles, since these protocols do not implement traffic congestion control algorithm. The situation is not the same for the FTP profile, where the TCP congestion control avoid to transmit more packets when saturation is detected. This explain the differences in the number collisions with and without interference.

In summary, we can observe that, for each case, the conclusions about how the hidden station scenario influences the communications of two stations could be biased or mislead when external interference is not considered. Certainly, our proposed method of interference is far from accurate represents the interference conditions into a scene. But, our contribution is a first step to use real measurements to improve the interference representation into OMNeT++ INET-Framework.

6. CONCLUSION AND FURTHER WORK

We have shown that the injection of recorded traffic traces in a simulation changes considerably the simulation results, from the layer 3 and layer 2 points of view. It is evident that, by choosing properly background traffic profiles and the location of interfering sources, it will produce similar effects on results that we have shown in our evaluation. Nevertheless, the realism added to the simulation will be based on the way to choose the traffic profiles and the location of the sources. Contrarily, by using our method, the level of the realism of the results is based on the fact that the interfering traffic profiles and the location of sources come from real measurements. In other words, the novelty is the realism added to the simulation lies in the generation of the background traffic based on real measurements.

Furthermore, we must notice that a probe-based technique does not fully describe what happens in reality. In particular, a single probe placed in a single location cannot capture collisions, because it only reports the packets it successfully decoded. To the contrary, capturing the traffic from multiple locations at the same time, would minimize the underestimation of collisions. Hence, adding more probes to sample at the same time, should help to minimize this underestimation. We will address this item on further work to quantify the relationships between the number and placement of the probes.

We also consider several improvements as further directions. However, the most important one is related to the assumptions taken at the begin of this paper. Minimize the error when estimating the reception power on each sampled place of the scene. This minimization could be done by adjusting the transmission power on the detected sources

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5602
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5602

or being more accurate with the location of the interfering sources.

Wireless networks are everywhere. The inner-city 2.4Ghz spectrum is crowded by ever growing number of wireless equipments. Therefore, considering the realistic effects of interference in this context has become a true challenge. Based on this fact, we provide a method to sample a scene and use that information to generate interfering traffic in the INET-Framework of the OMNeT++ simulation software. The proposed technique to sample the traffic, locate the sources, mapping them in a simulated playground, and use the recorded traffic as interference in a simulated model add realism to the results. Indeed, the validation shows tendency to converge when simulated and real measurements are compared.

7. ACKNOWLEDGMENTS

The authors would like to thank everyone who helped us write this paper. This work was partly founded by the IST-FET AEOLUS project.

8. REFERENCES

[1] A.Köpke, M.Swigulski, K.Wessel, D.Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin. Simulating wireless and mobile networks in OMNeT++ the MiXiM vision. In *Proceeding of the 1. International Workshop on OMNeT++*, Mar. 2008.

[2] T. Andel and A. Yasinsac. On the credibility of manet simulations. *Computer*, 39(7):48–54, July 2006.

[3] A. Iyer, C. Rosenberg, and A. Karnik. What is the right model for wireless channel interference? In *QShine '06: Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks*, page 2, New York, NY, USA, 2006. ACM.

[4] A. Kashyap, S. Ganguly, and S. R. Das. Measurement-based approaches for accurate simulation of 802.11-based wireless networks. In *MSWiM '08: Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 54–59, New York, NY, USA, 2008. ACM.

[5] The Kismet wireless program - http://www.kismetwireless.net/.

[6] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. Technical Report TR2004-507, Dartmouth College, Computer Science, Hanover, NH, June 2004.

[7] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 171–182, New York, NY, USA, 2007. ACM.

[8] Radiotap "de facto" standard for 802.11 frame injection and reception - http://www.radiotap.org/.

[9] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. *SIGCOMM Comput. Commun. Rev.*, 36(4):51–62, 2006.

[10] S. E. Robert Nagel. Efficient and realistic mobility and channel modeling for vanet scenarios using omnet++ and inet-framework. In *Proceeding of the 1. International Workshop on OMNeT++*, Mar. 2008.

[11] A. Tsertou and D. I. Laurenson. Revisiting the hidden terminal problem in a csma/ca wireless network. *IEEE Transactions on Mobile Computing*, 7(7):817–831, 2008.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5602
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5602