Replication of the Bursty Behavior of Indoor WLAN Channels

David Gómez, Ramón Agüero, Marta García-Arranz, Luis Muñoz
University of Cantabria
Avda. De Los Castros S/N
Santander, Cantabria (Spain)
{dgomez, ramon, marta, luis}@tlmat.unican.es

ABSTRACT

In this paper we present the design, implementation and assessment (by means of extensive simulation campaigns) of two wireless error models within the `ns-3` framework, whose main goal is to emulate the *bursty* behavior of indoor real propagation environments. The first one, called *Bursty Error model based on an Auto Regressive filter* (*BEAR*), aims to mimic, including a memory-aware contribution, the received Signal to Noise Ratio, which is afterwards used to establish the presence of errors within the frame. The second one is an extension of the widespread *Gilbert-Elliot* channel model, based on a *Hidden Markov Process*. Both of them are proved to accurately replicate the presence of long error frame bursts, as compared to the legacy approaches which are integrated in the simulator, by studying a wide range of performance figures. The paper also discusses the drawbacks exhibited by the legacy error models supported by the simulator, according to the behavior observed over real indoor wireless channels, since they lead to unrealistic results.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Simulation, Performance, Wireless channel modeling

Keywords

Indoor Wireless Channels; Channel memory; Hidden Markov Process; Bursty Error Auto-Regressive Filter

1. INTRODUCTION

Wireless technologies are constantly evolving and have become an essential part of everyday life. Namely (if we do not consider the cellular infrastructure), the birth of the IEEE 802.11-compliant technologies has lead to a remarkable increase of the popularity of these devices. Therefore, the research community has to address the multiple challenges posed by these particular sort of communications. Although the experimentation over real environments seems to be the ideal way to study such technologies, it shows clear limitations (i.e. scalability, repetitiveness...); these drawbacks bring about the need of simulation procedures, whose main goal is to fulfill those gaps which might not be covered by real measurements.

The ever-increasing computational capacity of the devices allows the development of more and more complex techniques so as to replicate the behavior of real networks, bringing about the appearance of novel and promising simulators, like `ns-3` [1], the natural successor to the popular `ns-2`.

One of the most insightful aspects of this work lies on the fact that we aim at mimicking the behavior over a real indoor channel, which were thoroughly studied by means of an empirical campaign. We will make use of the results obtained from this analysis so as to tune the performance of two novel wireless channel models, whose operation is quite different: whilst the former one relies on a *Hidden Markov Process* (*HMP*), the second one, named *Bursty Error model based on an Auto-Regressive filter* (*BEAR*), implements an auto-regressive filter so as to estimate the received signal strength and to mimic the memory effect observed over real channels. Hence, we will try to demonstrate that the bursty behavior exhibited over these indoor scenarios is appropriately reflected in our models. Last, but not least, we introduce a third option, extracted from the legacy `ns-3` simulator, whose performance, although it replicates quite well the operation of an indoor wireless channel in terms of i.e. error rate and throughput, is not able to show that memory effect which was assessed over a real scenario.

We have structured this document as follows: Section 2 outlines the main research contributions regarding IEEE 802.11 wireless channel simulation. Section 3 describes the most relevant issues of the models (real and simulated) we address to characterize. In Section 4 we present the simulation campaign we have carried out so as to compare the performance of the different channel models presented, discussing the obtained results. At last, Section 5 properly concludes the document and presents those issues we aim at tackling in the future.

WNS3 2013, March 05-07

DOI 10.4108/icst.simutools.2013.251700

2. RELATED WORK

In the latest years, a number of works [4,6,8] have analyzed the behavior of various channel models for mimicking wireless indoor environments. A common aspect is that almost none of the existing approaches is able to emulate the bursty characteristic of such environments, which causes losses to occur in bursts, rather than on an isolated, independent manner.

In order to address this shortcoming, different proposals have been made. For instance, Cardoso *et al.* [5] studied the appropriateness of *HMPs*, showing that they offer a much better performance than other alternatives. In a previous paper, we presented *BEAR* [2], whose main characteristic was that it uses the received Signal to Noise Ratio (SNR) so as to modulate the behavior of the wireless channel. This particular characteristic makes this channel model very appealing, since it can be easily used, even if nodes are mobile (this is not the case for Markov Chain based models, in which the particular configuration of the subjacent chain should be modified according to the distance between transmitter and receiver). The analysis which we carried out on BEAR performance was very promising, as it outperformed other traditional approaches like Gilbert-Elliot [8] or memory-less channels [13].

Regarding the proper modeling of IEEE 802.11 wireless channels over the `ns-3` network simulator, several pieces of work have been carried out in recent years (the first version of the simulator was released in 2008). The common denominator of most of them [11–14] is that they rely on the calculation of the *Bit Error Rates* (*BER*) as a function of the *Received Signal Strength* (*RSS*) or the *Signal to Interference Ratio* (*SIR*) in order to discard a frame or deliver it to the upper layer. Indeed, the most popular error rate models supported by `ns-3` (i.e. *NistErrorRateModel*, *YansErrorRateModel*) base their operation on the use of these curves. The main drawback of these approaches is that they do not appropriately reflect the memory effect that is observed over real wireless channels, thus leading to predictable and unrealistic results. It is worth highlighting that our work does not take into account the interference stemming from other packets, since we are studying a single transmission between two nodes over a clean scenario.

Last, but not least, the work carried out by Al-Bado *et al.* [3] performed an extensive empirical campaign over a real indoor scenario in order to model a brand new `ns-3` wireless channel model, tailored from the *Frame Detection Rate* (*FDR*), *Frame Error Rate* (*FER*), as well as the capture and interference patterns observed over the real measurements, for different physical rates (i.e. IEEE 801.11g at 6, 24 and 54 Mbps). Although their solution reflect quite adequately the frame reception events (i.e. by means of the perceived *RSS*), they focused most of their work on the interferences arisen from simultaneous transmissions, since their test scenario consisted in several stations, disregarding the dependency between consecutive frame loses, thus ignoring the memory effect observed over real indoor wireless channels. In our work, we does not consider any interference or capture effect source, aiming at mimicking the actual channel behavior as much as possible.

3. FROM THE REAL WORLD TO NS-3: BUILDING A CHANNEL MODEL

In this section we will briefly introduce the performance observed over a real indoor wireless scenario, whose results were further used to tailor the operation of two different novel channel models on `ns-3`. At last, we present another alternative, extracted from the legacy simulator's code, so as to prove that it is clearly outperformed by our proposed channels.

3.1 Empirical characterization

As introduced above, we have carried out an empirical campaign over a real indoor wireless channel in order to characterize the behavior of the IEEE 802.11b recommendation[1]. The experimental set-up consisted of two Linux boxes, one of them being the transmitter (TX) and the other one taking the receiver (RX) role. Both the TX and the RX incorporated *WaveLAN 11 Mb Lucent/Orinoco PCMCIA cards*, configured in a proprietary *Ad Hoc* (pseudo-IBSS) mode which did not use management frames. The RTS/CTS mechanism was disabled during the experiments. The transmitter and the receiver were separated by around 15 meters, without line of sight, and with both metallic obstacles and people moving within the channel (typical office environment). Both cards were configured at 11 Mbps. 10000 UDP/IP unicast datagrams, with 1472 bytes of payload, were sent from the TX to the RX in each of the independent measurements, saturating the wireless link. Furthermore, we ensured that the presence of 802.11 traffic from other networks was negligible during the whole campaign. The corresponding wireless card driver was modified so as to be able to track whether incoming frames were corrupted (CRC failed) as well as the received SNR. Last, but not least, the maximum number of transmissions for an IEEE 802.11 frame was fixed to 4. The main results and statistics for each of the measurements are gathered in Table 1.

In addition, we will manage the following statistics in order to analyze the behavior of each channel model:

- *FER.* Ratio between the erroneous and the overall frames arrived at the received node.
- *Packet Error Rate (PER).* This statistic represent the packets that could not been recovered by the IEEE 802.11 retransmission scheme, yielding the definitive loss of the corresponding information.

The most relevant issue to be highlighted is the extreme observed variability, abridged in terms of throughput, *FER* and *PER*; it covers a range from almost ideal situations (i.e. measurements #14 and #15), to rather opposite cases in which the channel has a harmful effect over the transmissions (i.e. in #1 half of the packets could not be retrieved at the receiver node). The channel memory effect is clearly reflected on the presence of *Erroneous Frame Bursts* (*EFB*) [2], where we can appreciate the existence of long corrupted consecutive frames even over good channel conditions, showcasing that there actually exists a clear dependence between consecutive receptions. This issue might severely jeopardize the transmission performance, since if

[1]Namely, in this work we have carried out the characterization of IEEE 802.11b, working at a binary rate of 11 Mbps.

Table 1: UDP performance between a single source and single receiver over a saturated indoor channel (15 independent trials, sorted by throughput in descending order)

#	***Thput*** *[Mbps]*	***FER***	***PER***	***EFB*** *Avg.*	*Max.*	*Var.*
1	0.82	0.814	0.5	15.975	2759	9586.490
2	1.34	0.709	0.314	5.929	1035	740.949
3	1.49	0.676	0.297	7.502	1229	1675.485
4	2.32	0.53	0.146	3.644	1927	1478.843
5	2.33	0.517	0.179	6.217	821	983.664
6	2.72	0.465	0.105	3.014	383	166.981
7	3.58	0.331	0.058	2.6	258	79.528
8	3.76	0.301	0.059	3.408	259	138.689
9	3.8	0.298	0.127	4.836	219	301.490
10	4	0.268	0.044	2.767	320	134.182
11	4.04	0.261	0.05	3.065	321	221.041
12	4.79	0.163	0.025	2.633	144	57.627
13	5.5	0.069	0.012	3.136	75	76.007
14	5.96	0.014	0.002	2.84	16	12.854
15	5.99	0.013	0.001	1.361	7	0.932

the erroneous burst is longer than the number of IEEE 802.11 transmission attempts, the datagram will be definitively lost.

At this point, it is worth remarking that the whole measurement campaign was carried out under the same conditions (i.e. distance between the nodes, external interferences, people moving). We can therefore argue that the high variability, which is reflected in Table 1 is due to the random fading effects (slow and fast) of the wireless channel, which exhibited a non-predictable performance, not appropriately replicated by most of the legacy and available emulated channel models for network simulators.

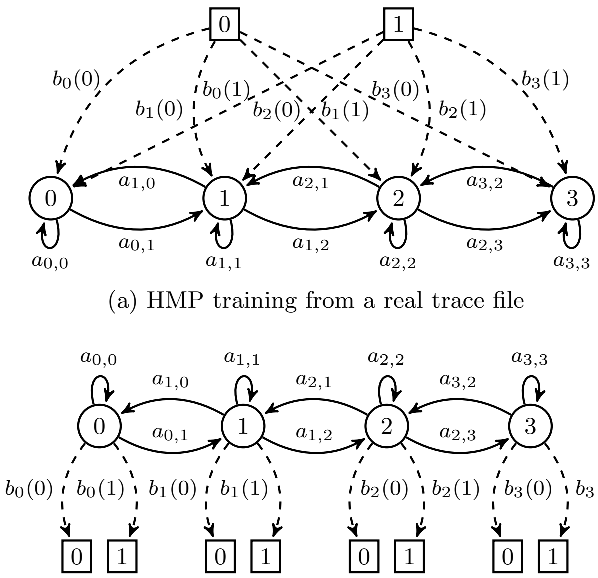
These results will be used to tailor the operation of both *BEAR* and *HMP* models in order to mimic the behavior observed herein.

3.2 Channel model based on an HMP

In order to address the description of this concrete channel model, we will describe how we have adapted the principles of a Hidden Markov Process [7] to replicate the bursty behavior of an indoor wireless channel. We define a finite number of independent states (N) that compose the Markov chain, as shown in Figure 1, where each of them represent a particular response against a frame reception. The transition probabilities between these states are gathered into a $N \times N$ matrix (A), whose coefficients are represented as $a_{i,j}$. Each of these states will have two output values associated, namely the probability of having a corrupted frame ($b_i(0)$, being i the state at which "decision" belongs to) or triggering a successful reception ($b_i(1)$). All these combinations form the decision matrix (B).

We have carried out the following process in order to determine which are the optimum coefficients:

- The first step to be addressed consists in parsing the trace files gathered during the empirical campaign. For each of the 15 measurements (Table 1), the output file presents, for every received frame, a 0 value if it was corrupted and later discarded by the receiver node or, otherwise, an 1 if the frame was correctly received. With the resulting binary vector, as Figure 1a[2] shows, we will "train" the corresponding Markov chain by means of the **hmmtrain** *Matlab's* function, passing as arguments the following ones: the traced vector, the number of states of the corresponding Markov chain, additional constraints we want to apply (i.e. birth-and-death process) and finally, either the maximum number of iterations or the allowed convergence error. Regarding the Markov process, the chain itself is "hidden", whilst the "observables" are the reception events, which act herein as the system input. This function will return both transition and decision matrices, which will be used within the ns-3 environment.
- However, the *HiddenMarkovErrorModel* (this is how we have named this model within the ns-3) operation, which is outlined in Figure 1b, presents a rather different composition. In this case, the visible part of the process lies on the Markov chain, whose set of transition probabilities defines the behavior of the model. Unlike the previous description, the hidden part of the process corresponds to the inherent frame reception decisions: error (0) or success (1). In other words, during the execution flow the channel will be changing its current state, according to their subjacent transition probabilities (A) and, when a node receives a frame, the error model will decide whether the frame is correct or not, comparing a random value with the corresponding $b_i(0)$ coefficient, being i the current state at the Markov chain.



(a) HMP training from a real trace file

(b) HMP simulation performance

Figure 1: HMP chain description

The model can be addressed from two different insights: the

[2]The Markov chain shown in the figure represents a 4-state birth-and-death process, thus limiting the transitions between neighboring states.

first one, used by Cardoso *et al.* [5], employs a *frame-based* operation, where after each reception event, the Markov chain is prone to change its current state, with a probability $a_{i,j}$. This makes the model extremely dependent of the training traces and their inherent traffic conditions (i.e. packet lengths, data rates...). Although it mimics quite well the actual performance under such conditions, if we modified any of these parameters, the obtained results would be different, thus showcasing that this sort of simulation is not able to replicate the behavior of a real wireless channel over any circumstances, but only those ones which were used to train the process.

In order to overcome this limitation, we proposed in [10] a new approach to configure an *HMP* to model the behavior of a bursty channel. Therein, we describe a *time-based* analysis which aimed at calculating the average time spent at an arbitrary state, $\overline{t_i}$, as shown in (1), where ψ represents the average inter-frame duration (≈ 2 *msec* in a saturated IEEE 802.11b transmission).

$$\overline{t_i} = \frac{\psi}{1 - a_{i,i}} \tag{1}$$

After this parameter is obtained for every state, we just need to schedule a timer, t_i, whose expiration will trigger a state transition. Thanks to this modification, we override the limitations exhibited by the *frame-based* solution, thus allowing an elastic/dynamic traffic configuration without jeopardizing the expected behavior.

It goes without saying that that *HMP* does not support any kind of dependency with the link quality, hence it totally depends on its configuration matrices.

3.3 BEAR

In this case, from the work initiated by *Agüero et al.* [2], we have ported the implementation of this model to the `ns-3` network simulator (namely, `ns-3.13 version`). Figure 2 depicts the operation followed by *BEAR*, from the transmission of a packet to the point which the receiver entity has to decide whether that particular frame is correct or not.

The cornerstone of this model consists in estimating the received link quality, splitting the original received signal into three different contributions, all of them encapsulated into a single *ArModel* instance, which, in turn, derives from the *PropagationLossModel* class. We abridge below their main features:

- The first one depends on the distance between the transmitter and the receiver nodes; it is normally characterized by a factor $d^{-\nu}$, where d represents the separation between the two edges and ν is tuned according to the propagation loss model (we could name this parameter "pathloss exponent"). For this work we rely on the "LogDistancePropagationLossModel" mechanisms provided by the legacy simulator.
- The second component reflects the slow variations on the received signal (*Slow Fading - SF*) which could be ascribed to the presence of physical obstacles within the path. In order to mimic the behavior of such effect, BEAR uses an *auto-regressive* filter[3] to estimate and shape the aforementioned slow variations, as shown in (2). As can be seen, the next value of the *SF* contribution, $SV[i]$, is "predicted" from the previous stored samples, $SV[i-j]$, limited by the *AR* filter order, T; the $a[j]$ correspond to the filter coefficients. Finally, ϵ is a white noise contribution with average power P_ϵ.

$$SV[i] = \sum_{j=0}^{T} a[j] \cdot SV[i-j] + \varepsilon[i] \tag{2}$$

- The latter one reflects the multi-path channel nature, leading to fast signal variations. The literature refers to this phenomena as *Fast Fading* (*FF*) or *shadowing* effect. In this work, it will be modeled as a random (i.e. Gaussian) variable with a mean zero and a variance of σ^2 dB^2, through the legacy *RandomPropagationLossModel* object.

Afterwards, the sum of all the propagation contributions are combined with an equivalent noise power so as to calculate the *SNR*, which will be the input of a decision entity, of the received frame. In order to deal with this operation, we followed the steps which are described below:

- Since the physical layer (i.e. *YansWifiPhy*) is tightly linked to the *ErrorRateModel* operation, we had to modify the legacy frame reception scheme (i.e. the *YansWifiPhy::EndReceive* method) since we have implemented the *BurstyErrorModel* as a derived entity of the base *ErrorModel* class. Therefore, if the received signal strength is actually higher than the reception threshold (i.e. *EnergyDetectionThreshold*), the delivery to the upper layer will rely on the operation of our error model, silently passing through the legacy physical reception[4].
- Instead of using the legacy *BER* curves supported by the simulator, we have tailored a logistic function, as shown in (3), which calculates the *FER* as a function of the SNR, establishing three different operation zones. It is worth mentioning that all these parameters (i.e. a, b, c and the two thresholds: LT and HT) were tuned from the results obtained over the real scenario.

$$FER = \begin{cases} 1, & SNR < LT \\ \frac{a}{1+e^{b\cdot(SNR-c)}}, & SNR \,\epsilon\, [LT, HT] \\ 0, & SNR > HT \end{cases} \tag{3}$$

- For the sake of simplicity, every IEEE 802.11 acknowledge or broadcast frame will be correctly received (during the experimental measurements we assessed that the probability of losing an ACK is much lower than the one seen for a data packet.

3.4 Default channel model

The last wireless channel model analyzed in this work corresponds with one of the mainstream WiFi models supported by the legacy `ns-3` simulator. In this particular case, as shown in Figure 3, we have configured the lower layer as follows: on the first hand, we have two propagation elements, the former is based on a *LogDistancePropagationLossModel*

[3]Whose coefficients were tuned from the results we got over the empirical campaign.

[4]It is worth highlighting that this code snippet does not affect the default simulation, since the default execution flow will remain unaltered in case we disable the *ArModel* solution.

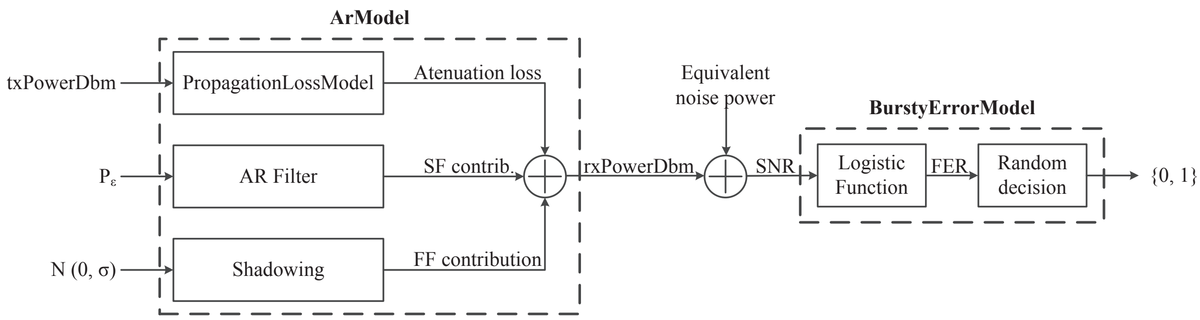


Figure 2: BEAR operation diagram

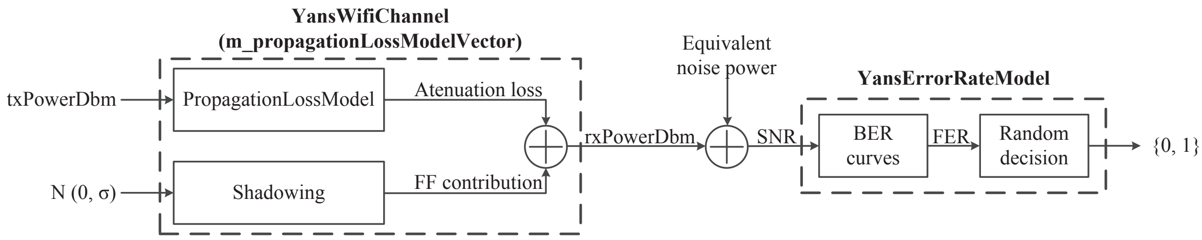


Figure 3: Default model operation diagram

and returns a deterministic value as a result of the distance between the nodes; besides, the second contribution mimics a shadowing (fading) effect, which is modeled through a normal random process N (0, σ^2).

Like *BEAR*, we obtain the *SNR* combining the received signal strength with the equivalent noise power (modeled as well by means of the *RandomPropagationLossModel* instance, which is chained to the deterministic propagation loss model). This value will be used to find the corresponding *FER*[5], which will be compared with a random value to decide whether a frame is correct or not. Hereinafter, we will refer this channel model configuration as *Default model*.

4. SIMULATION AND RESULTS

In order to showcase the benefits brought about by the models we propose in this work, we have carried out a thorough simulation campaign in which we aim at demonstrating that both *HMP* and *BEAR* lead to a much more realistic behavior than the ones offered by the legacy models of the simulator.

4.1 Simulation setup

The common denominator of every simulation is outlined as follows: a source node sends 10000 UDP datagrams (with a payload length of 1472 bytes each) to a receiver entity. We also assume that there is always traffic to be sent at the transmitter, thus saturating the wireless channel. It is worth highlighting that we have modified the transmission power (i.e. *txPowerDbm*), because with the legacy value (i.e. 16.0206 dBm), errors began to appear[6] at a distance between nodes of around 70 meters, far from the values which were observed during the empirical campaign. Hence, we reduced the power transmission to 0 dBm.

We have two sets of simulations: the first one aims at illustrating the memory effect observed over real indoor wireless channels. We have performed 500 independent measurements with both nodes at fixed positions. On the other hand, the second batch of experiments deals with the characterization of the distance influence over the error rate of a transmission. For this purpose, we have covered distances from 5 to 39 meters, with 20 simulations per position[7].

As mentioned in Section 3, we analyzed the following performance metrics so as to characterize the channel behavior: *FER* and *PER*, measured at the physical layer (namely, by means of the *YansWifiPhy::EndReceive* trace callback).

Besides, it is worth highlighting the most relevant issues regarding the configuration we have carried out for each channel model so as to compare their performance, in terms of memory (or bursty behavior) and link quality dependency:

- The *BEAR* model will make use of a third order autoregressive filter ($T = 3$) biased with a white noise power $P_\epsilon = 5 \cdot 10^{-3} W/Hz$, while the *FF* contribution will be modeled as a normal random variable N(0, 2.8 dB^2).

[5] In this case, *YansErrorRate* uses *BER* curves in order to estimate the *FER* for a particular *SNR* value.

[6] With a log distance propagation loss model.

[7] Due to the high variability of *BEAR*, we used a higher number of simulations per point (50) for this model.

- Regarding the *HMP* model, we will illustrate the behavior of a 4-state hidden Markov chain[8]. We have arbitrarily chosen three measurements from the ones shown in Table 1 in order to represent the wide range of behaviors: *Good* channel conditions (measurement #12), an *Average* transmission (measurement #9) and, last, a link characterized for its *Bad* performance, where more than half of the frames were corrupted (measurement #5).
- Finally, the so-called *Default model* will replicate the same shadowing contribution as *BEAR* ($\mathrm{N}(0, 2.8\, dB^2)$).

4.2 Main results

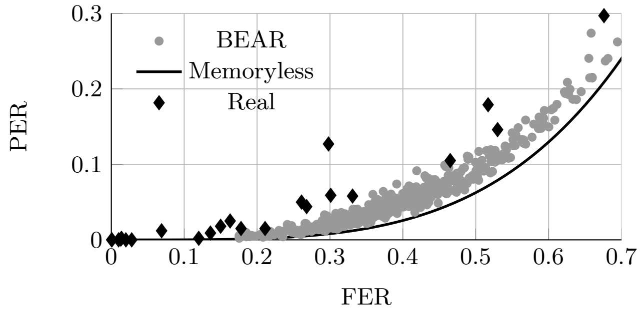
It is well-known that one of the most relevant characteristics of indoor wireless propagation environments is that erroneous frames tend to be grouped in bursts, since wireless channels showcase a certain level of memory. An insightful indicator of this behavior is illustrated with the relationship between the *FER* and the *PER*, as shown in Figure 4, which compares the results obtained via simulation to those seen over a real indoor channel and, finally, a memoryless channel[9]. The first aspect to be highlighted corresponds with the high variability that *BEAR* is able to reflect, almost covering the broad range of results which were observed over the real channel; moreover, despite the *HMP* channel does not offer such a disperse set of results, it reaches a closer level (in terms of memory) to the real measurements in all its configurations than *BEAR*. Meanwhile, the *Default* model totally lacks from any memory, showcasing almost predictable outputs.

Moreover, it has a key relevance the characterization of the relationship between the perceived link quality and the resulting performance of a wireless channel model. Therefore, we have represented the error rate[10] at different levels (i.e. frame and packet) as the distances between the nodes increases, showing both *FER* and *PER* average values as well the 95% confidence interval for each of the points. We have split the results into two main groups: the former one, called *Deterministic*, disables all the random contributions, relying the performance on the *LogDistancePropagationLossModel* and the particular decision criteria of each model; whilst the second group (named *Random*) gathers the complete operation of the various models. On the first hand, Figure 5 illustrates the *FER* (Figure 5a) of the *Default model*, which covers an area of approximately 15 meters ($\sim 19 - 34\ m.$)). It is worth mentioning the lack of variability exhibited by the *Deterministic* configuration, showcasing a poor range of less than 7 meters. Regarding the *PER* output (Figure 5b), we can see the errors start growing from a *FER* of ~ 0.4, asserting the memoryless behavior of this model (we need a high *FER* to bring about a burst of errors long enough to bypass the IEEE 802.11 recovery scheme and trigger a packet loss), where an error event is completely independent from the previous ones.

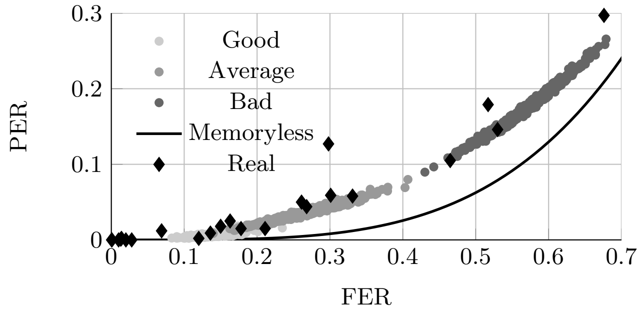
[8]We also studied configurations with higher number of states (i.e. 8 and 16) and the resulting performance was alike.

[9]In this case, we assume that $PER = FER^4$, being this 4 the total number of IEEE 802.11 transmissions, since it is considered that the random processes that model the reception between frames are completely independent.

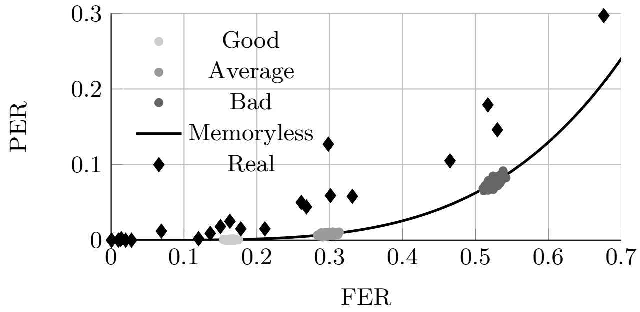
[10]Since it keeps a strong correlation with the received signal strength.



(a) BEAR



(b) HMP model

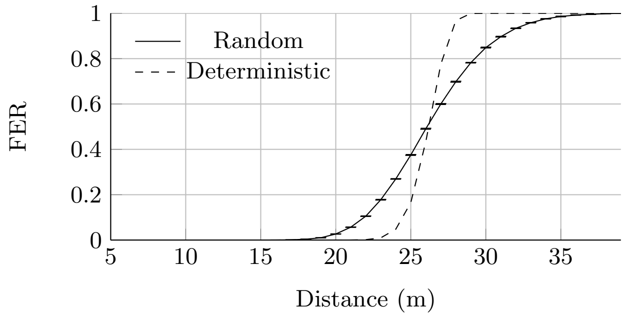


(c) Default model

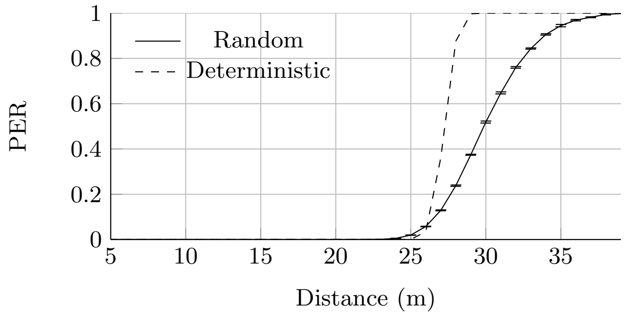
Figure 4: Relationship between FER and PER as a channel memory indicator

On the other hand, in Figure 6 we can observe the *BEAR* performance. It exhibits a significative range of distances in which a frame error might be triggered, thus covering values throughout the represented distances (and even more). Furthermore, the variability shown by the 95% confidence interval reflects the unpredictable behavior that we observed during the experimental campaign, effect that we are not able to appreciate with the *Default model*.

The most obvious limitation of the *HMP* model is that its performance does not show any kind of dependency with the received signal strength (that is to say that there is no relationship between the distance between the nodes and the erroneous performance of the transmission), hence we actually rely the model performance on the configuration matrices (transition and emission) which were obtained from the empirical campaign described in Section 3.1. However, we can exploit the results of Table 1, thanks to the high

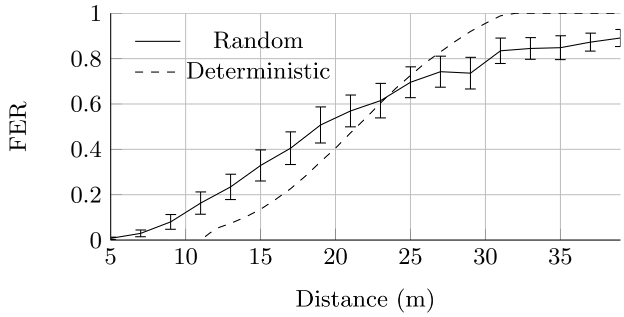


(a) Distance vs FER

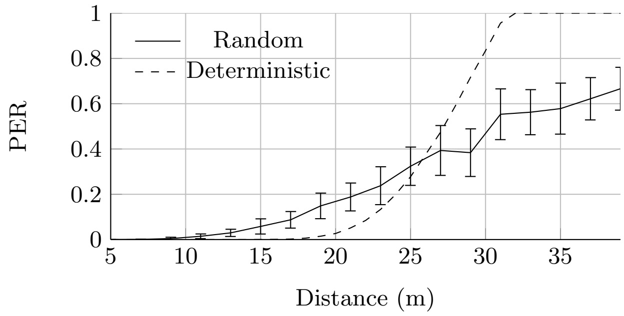


(b) Distance vs PER

Figure 5: Relationship between error rates and distance (Default model)



(a) Distance vs FER



(b) Distance vs PER

Figure 6: Relationship between error rates and distance (BEAR)

variability the channel exhibited by the real channel and "discretize" the operation of the *HMP* model along the distance, by means of a finite number of points, trained from the traces files which composed the element of the aforementioned table (namely, we will take the measurements #1, #3, #5, #6, #10, #12, #13 and #15 to address this new performance[11]). Figure 7 shows, in terms of *FER* and PER, the possibility of including some indirect distance dependency with the *HMP* model. We have chosen, as a reference, the *FER/distance* curve we got with the *BEAR* model (namely, its *Deterministic* configuration) to carry out the distance-matrices bindings (through a curve-fitting tool, in order to set the optimum "switching points"). As can be seen in Figure 7a, all *FER* values are reliably covered. Furthermore, we can see in Figure 7b that the resulting PER configuration shows an appropriate behavior, compared to the *BEAR* deterministic performance.

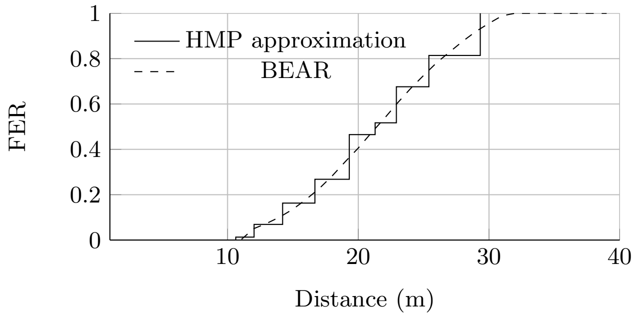
5. CONCLUSIONS

In this work we have developed two wireless channel models for the `ns-3` network simulator, tailored from the results obtained during a real empirical campaign (carried out over an IEEE 802.11b channel), showing two opposed concepts: the first one (*BEAR*) estimated the received signal strength by means of an auto-regressive filter, whilst the second one (*HMP*) "discretized" the error response of the wireless channel in a finite number of states, establishing a hidden Markov chain. The cornerstone of both models is that they aim at reflecting the bursty nature that characterizes real wireless channels, whose memory behavior is usually disregarded by the vast majority of mainstream simulators, which usually yields a rather predictable behavior.

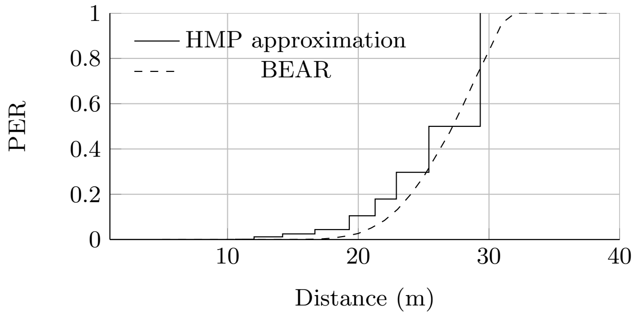
Furthermore, we have characterized the relationship between the distance between the nodes (tightly linked to the link quality, quantified by the received signal strength) and the error rates (at both MAC and IP levels). We have seen again that the legacy `ns-3` model (namely, the tuple *LogDistancePropagationLossModel* + *RandomPropagationLossModel* + *YansErrorRateModel*) showed a limited range of results, reflecting a rather deterministic behavior, while *BEAR* covered a wider span of distances with a higher variability. On the other hand, we have proposed a way to enforce *HMP* to dynamically operate according to the separation between the nodes, varying the configuration matrices as a function of the distance.

Which of the proposed models performs better? There is not a clear answer to this question. If we needed a dependency with the distance between nodes (e.g. if they move), we would obviously take *BEAR* (*HMP* does not deal with this feature); alike, if we wanted to simulate an unpredictable channel, covering a broad range of outputs. On the other hand, if we wanted to mimic the memory effect of the channel or if the goal was to replicate a similar performance (e.g. during simulation campaigns), *HMP* would perform better. Besides, if we compare both of them according to their runtime performance, *HMP* outperforms *BEAR* again, since its computational complexity is much lower.

[11]We have discarded the measurements #2, #4, #7, #8, #9, #11 and #14 because they did not bring about any insightful information (they were really close to the chosen ones).



(a) FER



(b) PER

Figure 7: Proposed FER-distance mapping for *HMP* (tailored from the deterministic *BEAR* simulation campaign)

The breakthrough of these models shall be extended to new physical extensions (i.e. IEEE 802.11g/n/ac) with different data rates and modulations. In order to tackle these challenges, we need to empirically characterize the new channels, to train the hidden Markov processes or tuning the autoregressive filter coefficients and the *BEAR's* logistic functions, used to decide whether a received frame is corrupted or not.

Besides, it is within the scope of this work to extend the empirical characterization with different workouts (for example study the actual dependency with the distance between nodes, modify the traffic conditions, analyze the performance over non-saturated channels, etc.). We will use the corresponding results so as to train our models, e.g. improving the *HMP* operation in order to replicate the distance dependency of a real wireless channel, without requiring the reference provided by *BEAR*.

At last, it is worth mentioning that the models implemented along this work (both *HMP* and *BEAR*) have been made available to the scientific community [9]. We encourage the interested researchers to use the code, testing the suitability of the models, thus helping us to improve their performance by means of an active feedback.

Acknowledgements

The authors would like to express their gratitude to the Spanish government for its funding in the project C3SEM (TEC2009-14598-C02-01). Besides, we would like to thank as well Mr. Juan Ramón Santana for his help in this work.

6. REFERENCES

[1] The ns-3 network simulator. http://www.nsnam.org/.

[2] R. Agüero, M. García-Arranz, and L. Muñoz. Accurate simulation of 802.11 indoor links: a bursty channel model based on real measurements. *EURASIP J. Wirel. Commun. Netw.*, 2010:16:1–16:12, April 2010.

[3] M. Al-Bado, C. Sengul, and R. Merz. What details are needed for wireless simulations? - A study of a site-specific indoor wireless model. In *INFOCOM*, pages 289–297. IEEE, 2012.

[4] P. Barsocchi, G. Oligeri, and F. Potorti. Measurement-based frame error model for simulating outdoor Wi-Fi networks. *IEEE Transactions on Wireless Communications*, 8(3):1154 –1158, March 2009.

[5] K. Cardoso and J. De Rezende. Accurate hidden markov modeling of packet losses in indoor 802.11 networks. *IEEE Communications Letters*, 13(6):417 –419, June 2009.

[6] G. Convertino, S. Oliva, F. Sigona, and L. Anchora. MMC05-1: An adaptive FEC scheme to reduce bursty losses in a 802.11 network. In *IEEE Global Telecommunications Conference, 2006. GLOBECOM '06*, pages 1 –6, 27 2006-Dec. 1 2006.

[7] Y. Ephraim and N. Merhav. Hidden Markov processes. *IEEE Transactions on Information Theory*, 48(6):1518 –1569, June 2002.

[8] V. R. Gandikota, B. R. Tamma, and C. S. R. Murthy. Adaptive FEC-based packet loss resilience scheme for supporting voice communication over Ad hoc Wireless Networks. *IEEE Transactions on Mobile Computing*, 7(10):1184–1199, Oct. 2008.

[9] D. Gómez and R. Agüero. Documentation and code for HMP and BEAR IEEE 802.11b channel models. http://dgomezphd.wordpress.com/2012/03/15/searching-a-realistic-wireless-model-in-network-simulator-environments/.

[10] D. Gómez, R. Agüero, M. García-Arranz, and L. Muñoz. On the Modeling of a Realistic Wireless Channel by means of a Hidden Markov Process. In *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, October 2012.

[11] M. Khosroshahy. *Study and Implementation of IEEE 802.11 Physical Channel Model in YANS (NS3 prototype) Network Simulator*, 2006.

[12] M. Lacage and T. R. Henderson. Yet another network simulator. In *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, WNS2 '06, New York, NY, USA, 2006. ACM.

[13] G. Pei and T. Henderson. Validation of ns-3 802.11b PHY model, May 2009. http://www.nsnam.org/p̃ei/80211b.pdf.

[14] A. Wang and Z. Liu. Analysis and utilizing of the error models in teaching wireless network. *JNW*, 6(2):222–229, 2011.