Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 380410, 17 pages
doi:10.1155/2010/380410

Research Article

Accurate Simulation of 802.11 Indoor Links: A "Bursty" Channel Model Based on Real Measurements

Ramón Agüero, Marta García-Arranz, and Luis Muñoz

Department of Communications Engineering, University of Cantabria, 39005 Santander, Spain

Correspondence should be addressed to Ramón Agüero, ramon@tlmat.unican.es

Received 10 June 2009; Revised 19 October 2009; Accepted 27 December 2009

Academic Editor: Arnd-Ragnar Rhiemeier

We propose a novel channel model to be used for simulating indoor wireless propagation environments. An extensive measurement campaign was carried out to assess the performance of different transport protocols over 802.11 links. This enabled us to better adjust our approach, which is based on an autoregressive filter. One of the main advantages of this proposal lies in its ability to reflect the "bursty" behavior which characterizes indoor wireless scenarios, having a great impact on the behavior of upper layer protocols. We compare this channel model, integrated within the Network Simulator (ns-2) platform, with other traditional approaches, showing that it is able to better reflect the real behavior which was empirically assessed.

1. Introduction

The advent and relevant growth of wireless LANs is behind the tremendous research activity which has recently been seen in this type of technologies. One of the main aspects of this research is the use of different simulation platforms. Their use is sensible, considering the difficulties that real testbeds usually impose, as well as the intrinsic advantages they offer (scalability, mobility, ability to perform multiple repetitions, etc.).

However, in order to ensure trustworthy results and to obtain performances similar to the ones which would be expected over real environments, it is really important to mimic, as accurately as possible, the conditions of real networks. In this sense, one of the most crucial aspects is the use of appropriate and reliable channel models. Many of the network simulation frameworks are criticized because of their lack of proper wireless channel models, since the results they offer about the performances of different protocols or algorithms are usually extrapolated to real life.

In this paper, we followed a different approach; first we carried out an extensive measurement campaign, over a real 802.11b testbed, to acquire a deep knowledge about the performance of UDP and TCP protocols over an error-prone indoor wireless channel. Later, this empirically assessed information was used to design and fine-tune a novel channel model, able to reflect the observed performance. In this sense, we ensure that the results which would be achieved when using it are reliable, and that it would be sensible when used to analyze the behavior of other protocols or algorithms.

It is also well known that one of the most widespread simulation tools is Network Simulator (`ns-2`). A great number of papers and works are based on this framework. We integrate our channel model in this platform, and we compare its performance with those of more traditional approaches, demonstrating that our proposal outperforms them. Considering the characteristics of any network simulator, in general, and `ns-2` in particular, one key requirement of the novel channel model must be its simplicity, in order not to jeopardize the main goal of these platforms, namely, the analysis of upper layer entities.

In order to cover the aforementioned points, this paper has been structured as follows: Section 2 discusses the performance of UDP and TCP protocols over a real error-prone wireless indoor environment. Based on these empirical results, a novel error channel model (based on an AR filter) is proposed in Section 3. Section 4 compares its behavior with those from the approaches which are traditionally employed by the simulator. Section 5 discusses related work, while the paper is concluded in Section 6, which advocates some items for future work.

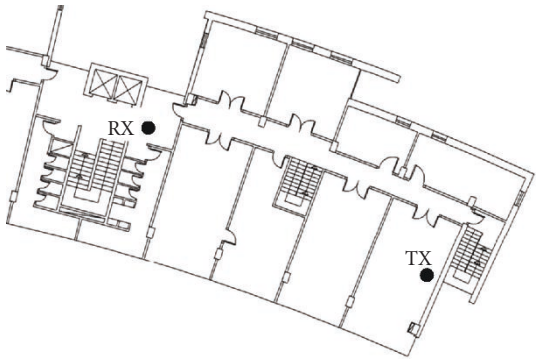


Figure 1: Measurement scenario.

2. Empirical Characterization of Indoor 802.11b Error-Prone Channels

This section presents some of the main characteristics of a real indoor wireless channel, which will be used afterwards so as to tune the parameters of the proposed model. For this, an extensive measurement campaign was carried out over a real testbed. The experimental set-up consists of two Linux boxes, running Linux kernel v2.4.9, one of them being the transmitter (TX) and the other one taking the receiver (RX) role. Both the TX and the RX incorporate IEEE 802.11 b Lucent/Orinoco cards, in which the use of management frames was disabled, thanks to their proprietary ad hoc mode, as was done for example in [1]. The RTS/CTS mechanism was also disabled during the experiments. We performed a set of measurements over a typical office environment, the transmitter and the receiver being separated by around 18 meters, without line of sight, and with both metallic obstacles and people moving within the channel, as shown in Figure 1.

2.1. Characterization Based on UDP Traffic. Both cards were configured to work at their highest bit rate (i.e., 11 Mbps). 10000 UDP/IP datagrams, with 1472 bytes of payload, were sent from the TX to the RX in each of the independent measurements, saturating the wireless link (i.e., we assume that there was always traffic to be sent at the transmitter). Furthermore, we ensured that the presence of 802.11 traffic from other networks was negligible during the whole campaign. Different performance metrics were recorded for each of the measurements and, in addition, the corresponding wireless card drivers were modified so as to be able to track whether incoming frames were corrupted (CRC failed) as well as the received SNR.

Table 1 shows the performance figures that were recorded for the 10 independent repetitions of the experiment. They include the Frame Error Rate (FER), the Packet Error Rate (PER), the throughput, the duration of the transfer, as well as the average, variance, and maximum Erroneous Frame Burst (EFB). Regarding the SNR, we have obtained both the average and the variance for the correct and erroneous (and overall) received frames. Below, a brief explanation of some of these parameters is given.

(i) Throughput. It provides the performance which was achieved for each of the tests; over an ideal IEEE 802.11b channel the raw performance offered by the UDP layer is around 6 Mbps.

(ii) Frame Error Rate (FER). Ratio between the erroneous and the overall frames arriving at the receiver.

(iii) Packet Error Rate (PER). IEEE 802.11b MAC layer uses a retry scheme, by means of which a frame is transmitted up to four times (in our particular configuration) before discarding it; in this sense, for a datagram to be lost, there must be, at least, four consecutive erroneous frames, and thus the PER does not match the FER. This PER could also be referred to as IP loss rate.

(iv) Erroneous Frame Burst. We have considered that the minimum length of a burst is one erroneous frame. The overall number of bursts per test is also indicated in the table

As can be seen, the most prominent channel's characteristic is its large variability. In the same position, we observe a wide range of behaviors, which vary from almost ideal performances (tests no.9 and no.10) to very poor ones (test no.1) for all the metrics which are presented. According to [2], most of the tests are long enough to provide accurate average data for the analysis of the wireless performance.

In addition, Figure 2 shows the behavior in terms of the link quality that was observed at the receiver. Measurements were obtained for 139098 received frames, which correspond to all the processed frames, that is, considering the overall individual measurements. As can be seen, the probability density function of the received SNR resembles a Gaussian random variable (with a standard deviation of $\sigma = 2.64$ dB), while the FER follows a smoothly decreasing trend. Although all the results presented are obtained for 11 Mbps, the right hand figure also shows the relationship between SNR and FER for 2 Mbps, in order to assess the different behavior which might be expected in this case (we also verified that the SNR pdf was valid for this binary rate). Furthermore, Figure 3 shows the distribution of the erroneous bursts' lengths which was observed for all the measurements. For short bursts, it resembles the decay of a geometrical random variable, but the probability of having relatively long bursts is much higher than the one which would have been obtained with a legacy geometrical. Note that this complementary cumulative distribution function is obtained for all the measurements in Table 1, comprising 10787 bursts.

In order to better understand the particular behavior which was assessed, Figure 4 shows the instantaneous variation of both the SNR as well as the presence of errors for four different tests (no.1, no.4, no.5 and no.8). The x-axis represents the order of arrival of the different frames, while the y-axis shows both the SNR of each incoming frame as well as the information regarding the presence of errors (CRC check). "1" indicates a correct frame, while "0" implies that a particular frame was corrupted. Figures 4(a) and 4(b) correspond to measurements no.4 and no.5, respectively. In both cases, the frame error rate was around 30%, being a bit

Table 1: Real measurements statistics, obtained over a real indoor wireless channel using UDP traffic, working at 11 Mbps.

No. test	Thput (Mbps)	Transfer duration (s)	Received frames	FER (%)	PER (%)	No. EFB	SNR correct Avg/Variance (dB/dB^2)		EFB Avg/Max/Variance (frames/frames^2)			SNR erroneous Avg/Variance (dB/dB^2)		SNR total Avg/Variance (dB/dB^2)	
1	1.49	55.57	21731	67.59	29.69	1958	10.47	6.88	7.50	1229	1675.48	8.84	6.57	9.37	7.25
2	2.32	43.36	18199	52.98	14.57	2646	9.77	3.88	3.64	1927	1478.84	8.97	3.78	9.35	3.98
3	2.33	41.49	17024	51.71	17.91	1416	9.56	4.87	6.21	821	983.66	7.40	5.94	8.45	6.59
4	3.58	30.98	14096	33.11	5.83	1795	10.38	4.34	2.60	258	79.53	9.24	6.26	10.00	5.27
5	3.80	27.04	12610	29.76	12.72	776	9.70	4.34	4.84	219	301.49	6.69	3.87	8.81	6.11
6	4.04	27.69	12868	26.06	5.00	1094	10.93	6.95	3.06	321	221.04	8.38	6.59	10.26	8.10
7	4.79	23.98	11664	16.26	2.45	720	11.44	4.73	2.63	144	57.63	9.22	3.79	11.08	5.24
8	5.50	21.15	10632	6.93	1.18	235	11.38	3.07	3.13	75	76.01	7.92	6.56	11.14	4.09
9	5.96	19.72	10136	1.40	0.19	50	12.84	3.61	2.80	16	12.85	8.49	6.57	12.78	3.91
10	5.99	19.65	10138	1.30	0.05	97	10.97	3.89	1.35	7	0.93	10.11	3.02	10.95	3.89

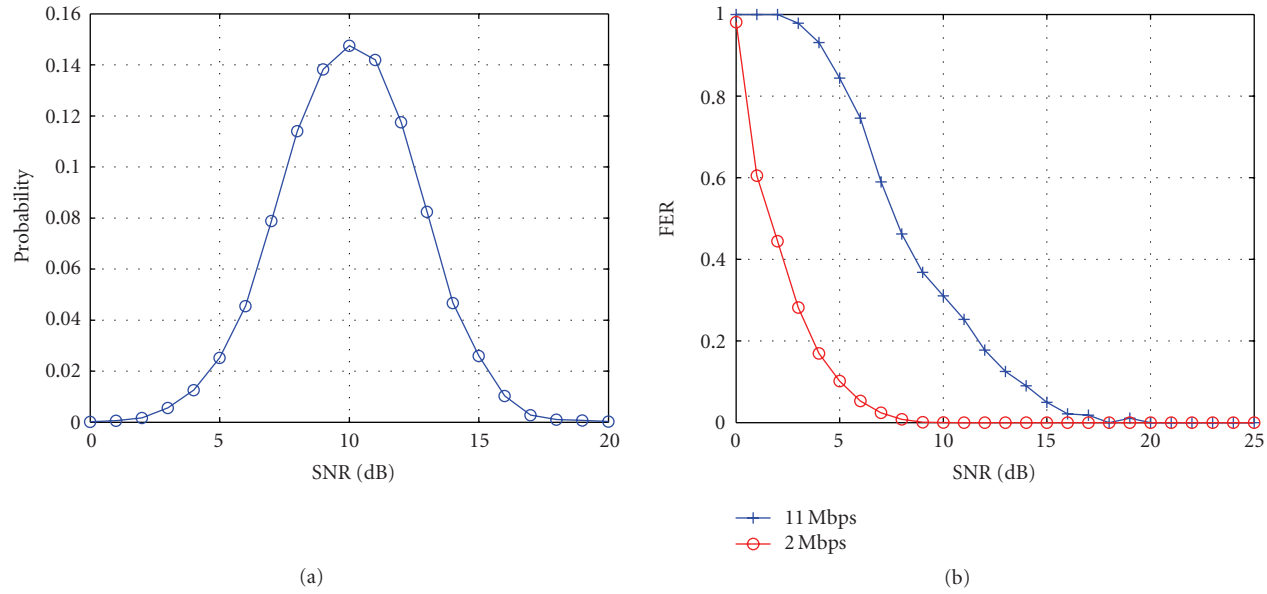


Figure 2: Probability density function of the SNR (a) and relationship between SNR and FER for 11 Mbps and 2 Mbps transmissions (b).

higher for the first one, although the average SNR is almost 1 dB better than in test no.5 (it also shows a lower variance); in this latter case the SNR falls below 10 dB for a relevant percentage of the incoming frames, which is mainly reflected in the fact that the average SNR for the erroneous frames is 6.7 dB, compared to 9.2 dB in test no.4. This different behavior is also reflected in how the erroneous frames have been distributed: in test no.4 there were almost twice the number of bursts as in no.5, but with a smaller length (the average EFB is below three frames, while it almost reaches five for no.5). This implies a higher PER (twice as much) for the latter, although it shows a slightly better throughput.

On the other hand, Figures 4(c) and 4(d) show the comparison between two completely different measurements, both in terms of the frame error rate, as well as in the perceived link quality. The first one corresponds to test no.8, in which the FER did not exceed 7% and the average SNR was around 11 dB, while the second reflects the behavior assessed in test no.1, with an FER close to 70% and an average SNR around 9.5 dB. Regarding the error distribution, in spite of the low frame error rate in measurement no.8, 235 bursts were observed, with an average length of 3 frames, and a maximum of 75. By analyzing the figure, two phases are clearly differentiated, both of them having almost the same duration. In the first half, the quality of the incoming frames heavily fluctuates, even falling below 5 dB, corresponding to those situations in which the longest bursts are perceived. Afterwards, the SNR stabilizes, and errors tend to happen in shorter bursts (2 or 3 frames), so there are almost no IP losses during this second part of the transfer. Concerning test no1, there are almost 2000 bursts, with an average length of 7.5 frames and a maximum one surpassing 1000 frames which happens, according to the figure, at the beginning of the second half of the measurement. In this, case, errors have

Table 2: Performance of TCP Reno with SACK over an error-prone real 802.11b indoor channel working at 11 Mbps.

No. test	Thput (Mbps)	FER Data (%)	PER Data (%)	FER ACK (%)	PER ACK (%)	Rtx	Max Rtx	Max idle time	Triple ACK	Dup ACK
1	0.55	41.85	7.24	0.58	0.02	620	9	28.16	114	1123
2	0.67	36.01	3.52	8.37	0.02	264	9	39.68	103	732
3	2.43	31.80	2.26	2.55	0.45	185	5	1.92	108	920
4	1.19	29.17	4.16	1.70	0.00	314	6	8.32	68	586
5	2.23	27.91	3.38	0.69	0.00	278	5	2.08	99	811
6	2.39	25.53	2.91	0.07	0.00	217	5	1.28	61	577
7	1.30	21.19	3.86	0.00	0.00	321	7	8.00	39	474
8	3.55	18.64	2.39	5.20	0.49	177	3	0.64	30	341
9	2.86	17.10	0.00	19.79	1.11	1	1	0.80	0	0
10	3.23	15.30	1.30	0.24	0.02	103	4	2.24	415	36
11	3.17	14.27	1.15	16.77	1.61	84	4	2.08	313	23
12	3.67	10.50	1.68	0.67	0.07	138	3	0.83	17	277
13	3.50	9.02	1.49	0.11	0.00	120	4	1.68	21	303
14	4.36	5.21	0.01	0.65	0.02	1	1	0.20	1	45
15	4.84	2.53	0.00	0.01	0.00	0	0	0.02	0	0

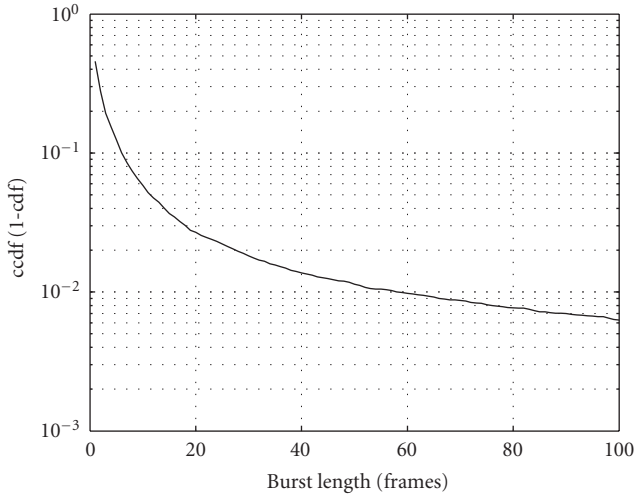


Figure 3: Complementary cumulative distribution function (ccdf) of the erroneous bursts' lengths.

been distributed according to relatively long bursts during the whole test, resulting in a high packet error rate, as can be seen in Table 1.

2.2. TCP-Based Characterization. In this case we carried out 15 independent experiments, transmitting a 10 MByte file (with the FTP protocol) in each of them, using the TCP Reno with Selective Acknowledgment (SACK) and Timestamp options activated. In addition to the TCP performance (throughput) and error-related statistics (both at frame and datagram levels), we reported the following statistics for each of the measurements.

(i) Duplicate ACKs. Number of duplicated TCP acknowledgments which arrive at the transmitter entity. According to the TCP specification, every time an out-of-order segment is received, an ACK is automatically generated.

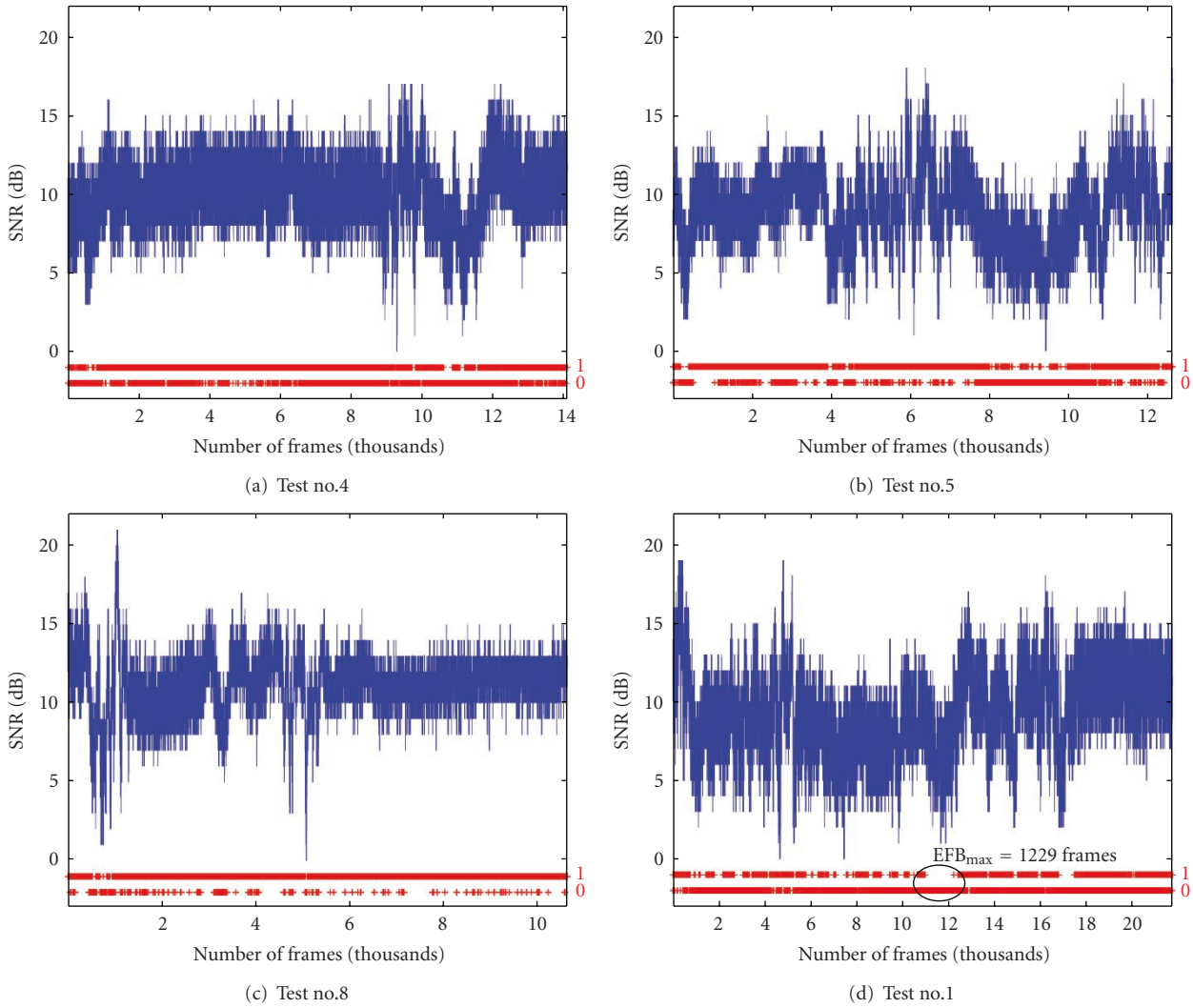
(ii) Triple ACKs. This case has a special relevance, since the reception of a triple ACK implies the immediate retransmission of a segment, according to the Fast Retransmit algorithm.

(iii) Maximum idle time. One of the aspects which cause most harmful impact on the TCP behavior is the presence of idle times in the transmitter. Since it was originally designed to overcome congestion situations, a TCP transmitter limits the rate at which it puts segments on the air as soon as it detects an unfavorable condition of the channel. Due to the algorithms it uses, it can actually remain quite a long time without any activity at all [3].

(iv) Retransmissions. Number of segments that the TCP transmitter needs to retransmit, either upon the reception of duplicated acknowledgments or when the Retransmission TimeOut (RTO) expires.

(v) Maximum retransmissions per segment. This corresponds to the maximum number of times that the same segment needs to be retransmitted, usually leading to relevant idle times when this metric is high.

As can be seen from the results of Table 2, the behavior of the TCP protocol is rather variable, despite the fact that all of the transfers have been carried out at the very same position. For instance, the performance ranges from almost 5 Mbps, which is the throughput that would have been achieved over an error-free wireless channel, to very low values (less than 1 Mbps). It is worth noting that one of



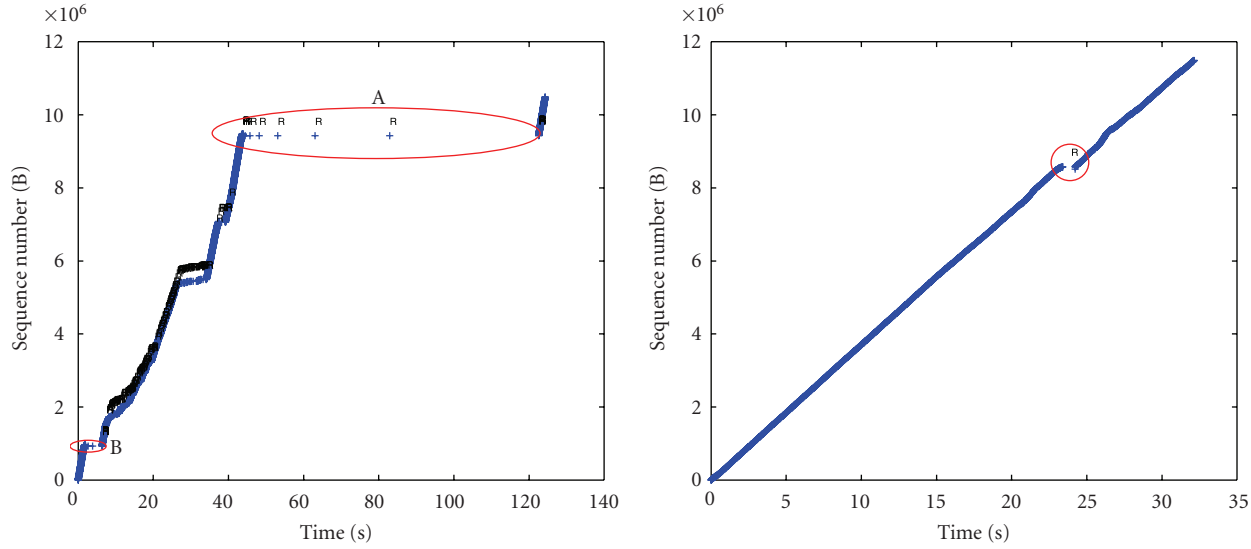
(a) Test no.4 (b) Test no.5

(c) Test no.8 (d) Test no.1

Figure 4: Instantaneous behavior (SNR and errors) of different measurements.

the aspects with greatest influence on the TCP behavior is, as discussed before, the appearance of idle times, which are usually associated with those situations in which the same TCP segment is retransmitted several times after consecutive RTO expirations. In this sense, the comparison between rows no.2 and no.5 is interesting, since the PER is, in both of them, very similar, but the impact of the idle time decreases the throughput of the former one almost four times. In such a case, the retransmission, up to nine times, of a single segment causes a maximum inactivity period of almost 40 seconds, which severely impairs the connection performance. Another example which is worth highlighting is measurement no.9; although the PER is null for the data segments, we can see that the performance is rather low; the reason is that the FER affecting the acknowledgements cannot be neglected. However, in the rest of the measurements, this is not the case, since the TCP ACKs, being much shorter than data segments, suffer from a much lower FER.

For a better comparison, Figure 5 shows the time sequence graphs of both tests no.2 and no.9, where crosses with "R" on top represent TCP retransmissions. By looking at the first one, there are two relevant inactivity periods which can be highlighted. The first one ("A") corresponds to the aforementioned value reflected on the table (a maximum idle time of almost 40 seconds) and leads to an overall inactivity of 77 seconds. The second one ("B"), much less relevant, is caused by the retransmission of a particular segment up to five times, the first one being triggered by the first duplicate ACK, which was received with SACK information. Below, a more detailed explanation of both situations will be given. On the other hand, the behavior of measurement no.9 is clearly different; as can be seen in Table 2, there was not a single datagram lost in the TCP data direction, while the IP loss affecting the corresponding acknowledgements is slightly higher than 1%, leading to the worst situation of all the measurements (the FER was almost 20%). In Figure 5(b), it can be seen that there was only a single retransmission (highlighted with a circle in graph), due to the consecutive loss of 23 acknowledgments. Hence, that particular TCP data segment arrives duplicated at the receiver, which silently



(a) Test no.2 (b) Test no.9

Figure 5: Time Sequence Graphs for two TCP connections observed at the transmitter entity.

discards it. Besides, it can be concluded that unfavorable conditions in the ACK direction do not have such a strong impact on TCP performance: the throughput decreases due to the overhead of MAC retransmissions as well as the semiduplex nature of the 802.11 technology. Nevertheless, Table 2 also shows that in most cases, the FER in this sense of the communication can be considered as insignificant.

Regarding measurement no.2, Figure 6 details the two inactivity situations previously mentioned, showing the information captured both at the transmitter and at the receiver. Double arrows with "R" on top represent TCP retransmissions, while double arrows with "S" on top reflect the information provided by the SACK option in the corresponding ACKs. In the case of the TCP receiver, we represent the frames which arrived erroneously, by means of lines without arrows, so we can assess the length of the erroneous frame burst which caused the inactivity. As can be seen at the TCP transmitter entity, in both cases, there are a relevant number of new TCP segment transmissions after the last ACK with SACK information. Taking into account that the channel remains in an unfavorable condition during a certain amount of time, most of these new transmissions arrive erroneously at the receiver, thus resulting in a relatively long burst. In situation "A", this results in an 80-erroneous frame burst, to which we must add the ones that correspond to the consecutive retransmissions which were triggered afterwards, by the expiration of the RTO, as can be inferred by the longer separation between the groups of four erroneous frames at the receiver (resulting in a burst of 108 frames, as is shown in the figure).

We can say that the starting contexts in both inactivity situations are rather similar to each other. The main difference lies in the fact that in case "B", where the burst length is almost the same as in situation "A", the channel leaves the unfavorable condition after the fifth retransmission, while in situation "A" the channel stays in a "bad" state for a longer time, until the ninth retransmission takes place.

3. BEAR Channel Model

Using the results which were previously discussed, this section presents the novel channel model which was designed so as to reflect the behavior assessed over a real indoor wireless channel. In particular, two main requirements were pursued, as we discuss below.

First, the model should reflect some dependency on the link quality, in terms of the perceived SNR; in this sense, it should enable the analysis of cross-layer optimization techniques, in which the SNR could be employed as a metric to modulate upper layer procedures and mechanisms. It is worth mentioning that there are some works which have shown that, within indoor wireless environments, the SNR might be used as an appropriate metric to reflect the condition of the channel [2, 4] and another which advocates its use as a way to enhance routing decisions [5].

Secondly, the model should reflect the bursty behavior which characterizes real propagation environments. This emerges as the main distinctive feature of the proposal, since, as was discussed before, the performance of upper layer protocols (especially TCP) is heavily jeopardized by the presence of long error bursts.

3.1. Modeling the SNR. As a first approximation, the use of a Normal random variable seems to be a sensible choice for modeling the SNR, since it mimics quite well the values which were observed over the real channel (see Figure 2); in fact, `ns-2` uses the so-called Shadowing model, based on a Gaussian random variable. However, we shall see that this approach (being memoryless) fails to capture any correlation

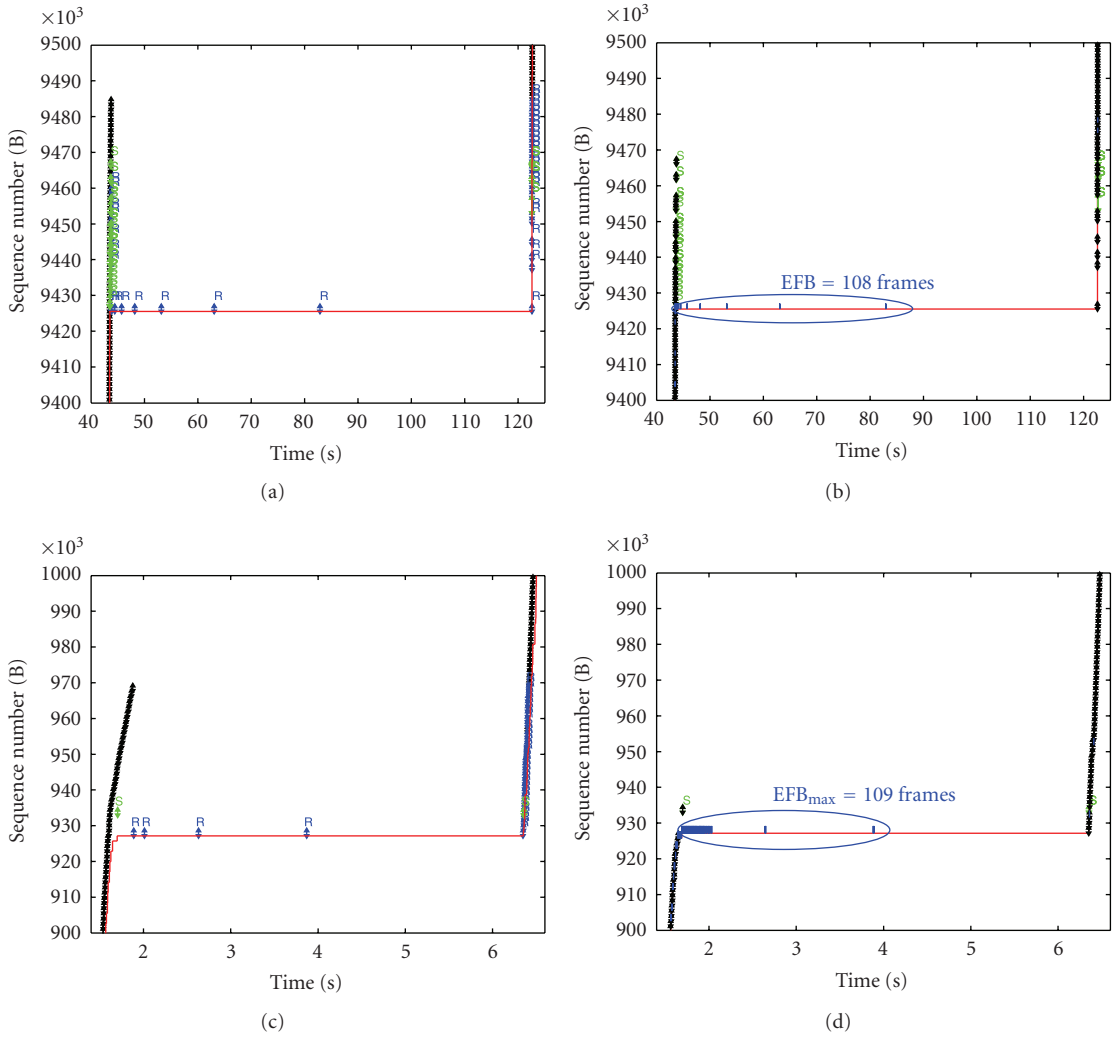


Figure 6: Details of situations *A* (a and b) and *B* (c and d) for TCP test no.2, at both the transmitter (a and c) and receiver (b and d).

between the SNR of consecutive frames. In order to better reflect this situation, we propose applying an autoregressive (AR) filter. Although other works in the literature have analyzed the possibilities of AR processes to model fading channels [6], there are no, to our knowledge, previous experiences in integrating this approach within network simulation platforms such as `ns-2`.

In order to correctly apply the AR filter, it is mandatory to decompose the received SNR. As can be seen in Figure 7, which shows the instantaneous SNR for a measurement comprising more than 12000 frames, it is possible to distinguish three different contributions. The first one reflects the dependence on the distance between transmitter and receiver. The second one, named slow varying (SV) component, captures slow variations of the channel, which are usually due to the presence of obstacles within the propagation channel; this contribution is the one which will be modeled by means of an AR filter. Lastly, there is also a fast varying (FV) component, which reflects the effect of multipath propagation, known to be relevant within indoor wireless environments. As can be seen, it is sensible to model this contribution using a Normal random variable.

In order to model the SV component, an AR filter will be used. Hence, the current sample of SV (SV[i]) can be "predicted" based on a number of previous values, as is shown in

$$\mathrm{SV}[i] = \sum_{j=1}^{T} a[j]\mathrm{SV}[i-j] + \varepsilon[i], \tag{1}$$

where T is the order of the corresponding filter and ε is white noise with average power $P\varepsilon$. The problem is thus to obtain the filter coefficients that best reflect the real behavior. To solve this problem, the well-known Yule-Walker method was used. Although the different snapshots which were empirically obtained are rather different, the solutions for the filter coefficients were similar and, furthermore, there was not a relevant difference when using a larger number of coefficients. Since there was a benefit of around 5% (in terms of the mean square error) in the fitting process between

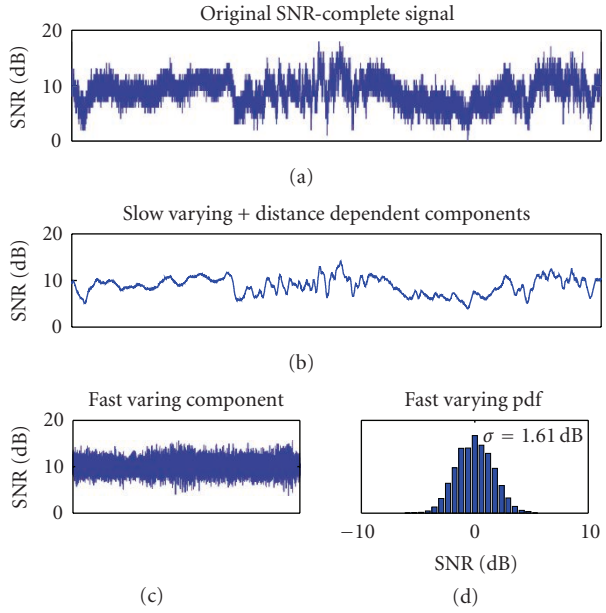


Figure 7: Decomposition of the SNR for a particular measurement.

$T = 2$ and $T = 3$, we opted for this latter configuration, being the particular filter coefficients the following ones: $a[1] = -1.106$, $a[2] = -0.02$, and $a[3] = 0.127$. Nevertheless, the implementation is flexible enough to be adapted to different configurations. On the other hand, the model also considers that when there is some time between consecutive frames, the validity of the previously stored AR samples should be limited; in this sense, we incorporated a timer, which was used to delete samples that should not influence the behavior of the channel, due to their longevity. We refer to this value as the coherence time of the model.

Taking all of the above into account, it is possible to compile the received SNR per frame as the joint contribution of three different components: the first one just captures the distance dependency (typically we can use an inverse relationship with d^{η}); the second component is the SV, the output of the AR filter, having the variance (power) of the white noise as an input parameter; finally, a Normal random process is used to model the third contribution (FV component), which is added to the two previous ones. Furthermore, considering that the `ns-2` framework does not have noise modeling at all, but just uses received power, we add a constant noise power, so as to better reflect the SNR values that were observed over the real channel.

3.2. FER Estimation. Once it is able to correctly simulate the received SNR per frame, the model also needs to deal with how to relate this value to the presence of errors in that frame. The original approach employed by the simulator does not reflect the real behavior which was discussed before, since it applies a hard threshold to decide whether the incoming packet is erroneous or not.

In contrast with this, Figure 2 implies that there is a clear relationship between the received SNR of a frame and the probability that it is corrupted. By examining such an empirically assessed correlation, we opted for a Logistic function, as can be seen below

$$\text{FER} = \begin{cases} 1, & \text{SNR} < lt, \\ \dfrac{a}{1 + e^{b(\text{SNR}-c)}} & \text{SNR} \in [lt, ht], \\ 0, & \text{SNR} > ht. \end{cases} \tag{2}$$

By correctly adjusting the parameters a, b and c, as well as the corresponding thresholds, the simulated FER shows an error which is less than $2 \cdot 10^{-4}$ when compared with the real behavior. In particular, the following values were used: $a = 1.24$, $b = 0.37$, $c = 6.88$, $lt = 3$ dB, and $ht = 16$ dB. Another advantage which is worth highlighting is that with this approach, various strategies might be used to handle different types of frames. For instance, we might find the parameters which best capture the behavior for frames of different lengths, which do not suffer the same degradation. In the simulation setup which will be used afterwards, this can be employed, for example to correctly model TCP acknowledgements (it has already been shown that they suffer a much lower FER than data segments). One aspect which is even more relevant derives from the differences which have been observed between the transmissions at various binary rates (see Figure 2). This characteristic, which has already been referred to as the "Gray Zones" effect [7], has a strong influence on the performance of wireless communication technologies, since in many cases control and broadcast frames are transmitted at a lower binary rate, thus leading to a larger coverage area. With the above expression, and selecting the appropriate set of parameters, it is possible to differentiate the frames which are broadcast, mimicking the behavior which was observed over the real propagation environment.

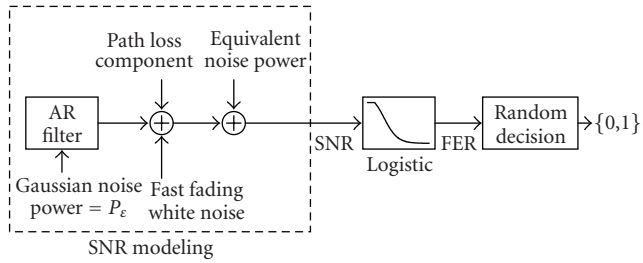


Figure 8: BEAR Architecture.

3.3. BEAR Architecture. The architecture of the novel channel model, which we name BEAR (Bursty Error model based on Auto-Regressive filtering), can be seen in Figure 8. When a frame is received, the SNR is calculated by combining the contributions from all the previously described components. This SNR is used to derive the probability of that frame being corrupted, depending on the specific type of frame and the corresponding BEAR configuration. Finally, a uniform random variable is used (together with the previous estimated FER) to establish the presence of errors within the frame.

It is also worth highlighting that the implementation of the channel has been done so that it can be used to model either symmetric or asymmetric wireless channels. In the former case, there is an instance of the AR filter per pair of nodes (whatever the direction is) while in the second case, two different processes are maintained for any pair of nodes, thus having a different behavior for each of the two senses of communication.

4. Simulation Result Discussions

Once BEAR has been introduced, this section compares its performance with that of other approaches which are traditionally employed by different network simulator frameworks, specifically by `ns-2`.

In particular, two other channel models will be included in the comparison analysis: the first one uses the Shadowing propagation which is intrinsically incorporated by the simulator. In this case, the comparison is relevant, since it will reveal the enhancements brought about by the memory which the AR filter adds on top of the Shadowing model.

On the other hand, another approach which is widely used in the literature is the so-called Markov chains. The idea is completely different, since the wireless channel is modeled by means of a number of states, as well as the probability, of each of them, receiving an erroneous frame. Hence, there is no direct correlation between the quality of the wireless link (e.g., the distance between transmitter and receiver) and the behavior of the model, which might be seen as a drawback of these approaches. This, for instance, would make a Markov model unsuitable for assessing any algorithm or protocol whose behavior was thought to depend on link quality. Another aspect which needs to be considered is that the corresponding chain is usually configured at a frame level, assuming that there is not a relevant time separation between consecutive receptions; this is obviously not the case for most of the currently used applications; in this sense, TCP procedures might introduce some time between consecutive transmissions (e.g., after an expiration of the retransmission timeout). These two disadvantages, which counter indicate the use of this approach for certain types of scenarios, affect any type of Markov model, independently of the complexity (number of states) of the subjacent chain. Needless to say, the most widespread Markov-based approach is the so-called Gilbert-Elliot (GE) model, which is still being used in a large number of current research works.

4.1. Results Based on UDP Traffic. Following the same ideas as the previously discussed measurement campaign, the main goal with UDP traffic is to assess the raw behavior of the wireless channel, without considering the influence that particular procedures employed by higher layer protocols (e.g., TCP) might have.

First of all, it is worth looking at the way the BEAR model is able to reliably mimic the instantaneous SNR. Figure 9 shows the SNR which was obtained for a simulation comprising more than 10^4 frames for three different propagation models. The first one is the traditional Two Ray Ground,

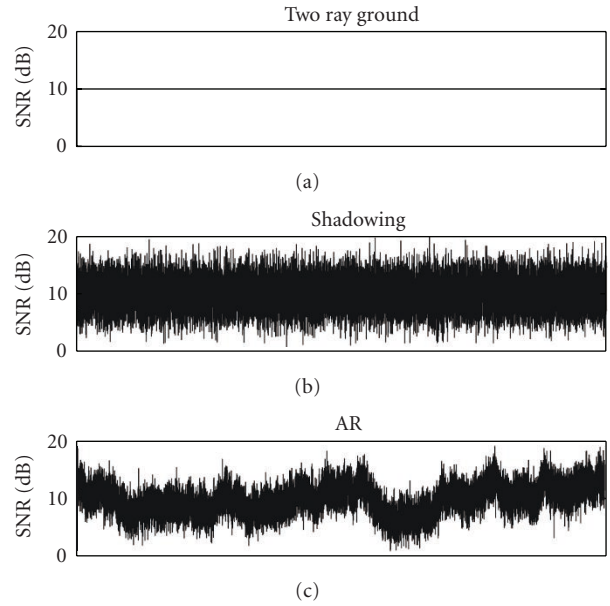


Figure 9: Simulated SNR with different channel models.

which, although it does not reflect any randomness in the perceived SNR, is widely used by the scientific community; this approach, together with the hard threshold originally used by the simulator to decide whether the incoming frame was erroneous or not, leads to a very deterministic wireless channel, which might not be appropriate for certain types of analysis. In contrast, it can be seen that the Shadowing model incorporates some randomness into the received SNR, although it fails to capture any correlation between consecutive frames. On the other hand, thanks to the memory provided by means of the AR filter, the BEAR model is able to reflect some degree of correlation between consecutive frames, capturing with a greater degree of accuracy the behavior which was empirically assessed. In this case, it is worthless analyzing the behavior of the GE model, since its performance does not depend at all on the link quality.

We performed, for each case, 500 independent runs, each of them comprising the transmission of 20 000 IP datagrams over an IEEE 802.11b single hop. There is always traffic to be sent at the transmitter, thus saturating the wireless channel. In addition, the MTU size was fixed at 1500 Bytes, and the BEAR channel was configured so as to emulate the same situation which was previously characterized over the real platform. On the other hand, we will use a set of different configurations for the two-state Markov chain, based on both the FER and the average length of consecutive erroneous frames (EFB) which were empirically observed. In this case, and in contrast to the traditional tendency to configure this type of channel models using frames to measure the duration of the corresponding two states, we do it at a time level, since, in this way, we are able to reflect, more accurately, the dynamics shown in real channels (e.g., if there is some additional time between two frames, whenever non-saturated traffic is considered).

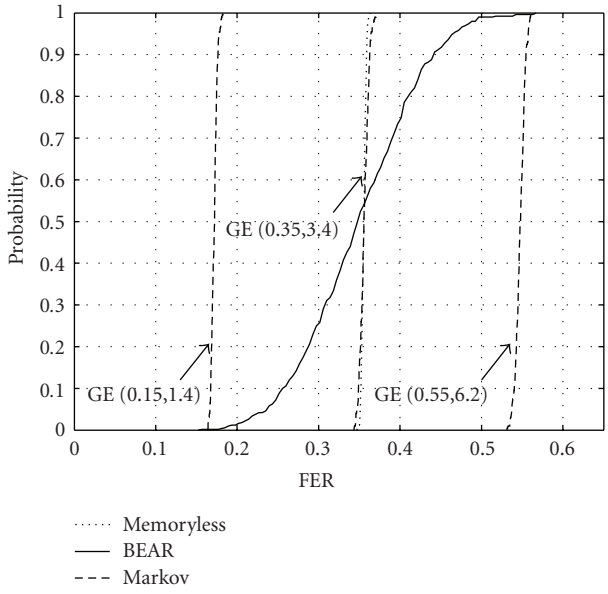


Figure 10: FER cdf for different channel models.

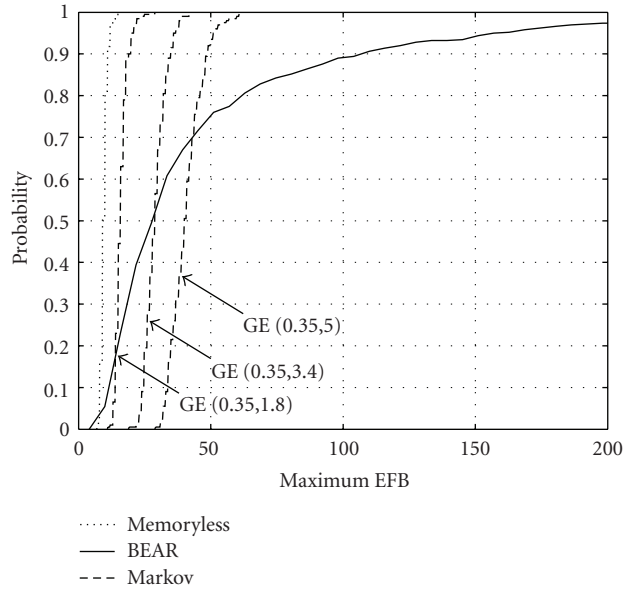


Figure 12: Maximum EFB cdf for different channel models.

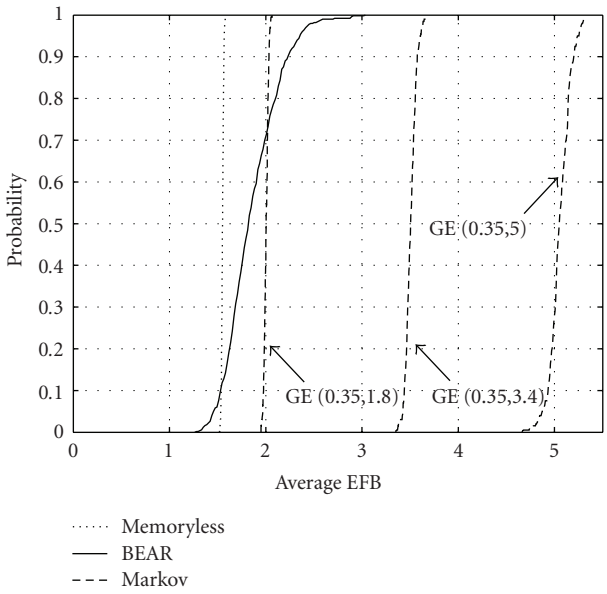


Figure 11: Average EFB cdf for different channel models.

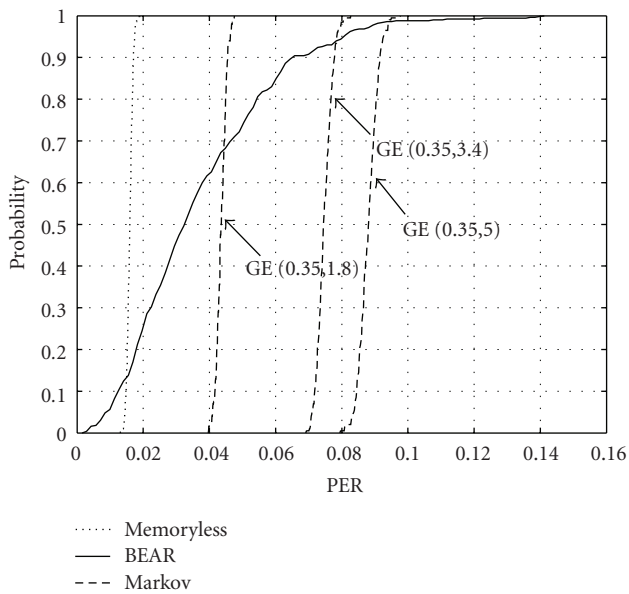


Figure 13: PER cdf for different channel models.

Figure 10 shows the cumulative distribution function (cdf) of the FER which was obtained with the different channel models. As can be seen, the one approach able to mimic the large variability observed in the real channel is BEAR, since for a single configuration it assesses quite a wide range of different FER values. On the other hand, it is clear that the Gilbert-Elliott channel does not capture this relevant variability. Although it may be argued that it is possible to establish different FER values, thus reflecting particular instances of the performance assessed over the real channel, none of them mimics the broad range which was empirically seen. Finally, the Shadowing-based approach (Memoryless) also shows very predictable FER values.

In order to better understand why the Markov chain-based model is not able to reflect the high variability which characterizes real indoor wireless channels, Figure 11 shows the distribution of the average EFB length for different instances of wireless channels. In the case of the Gilbert-Elliott model, we fixed the FER at the value which best characterizes the average behavior of the real channel (around 0.35) and we modified the average EFB length to reflect diverse situations; in particular, we fixed this parameter at 1.8, 3.4, and 5.0 frames (it has to be considered that these values were translated to their corresponding duration, so as to configure the chain at a time level, as was mentioned before). We shall see, however, that for a single instance of the

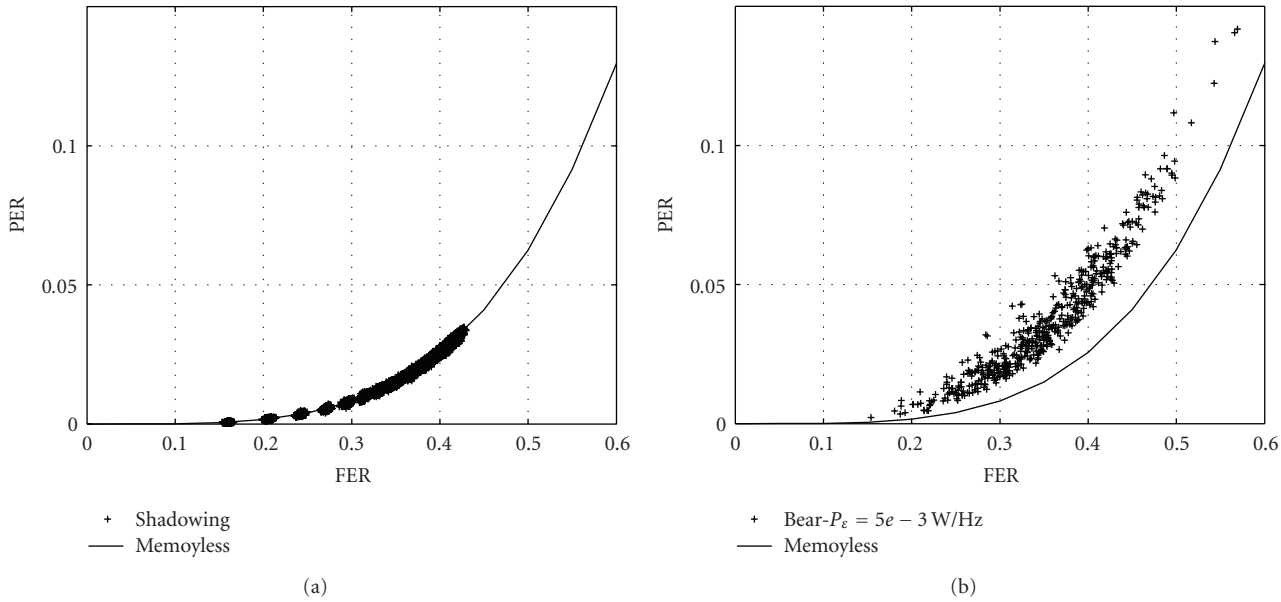


Figure 14: Relationship between FER and PER for the Shadowing (a) and BEAR (b) channel models.

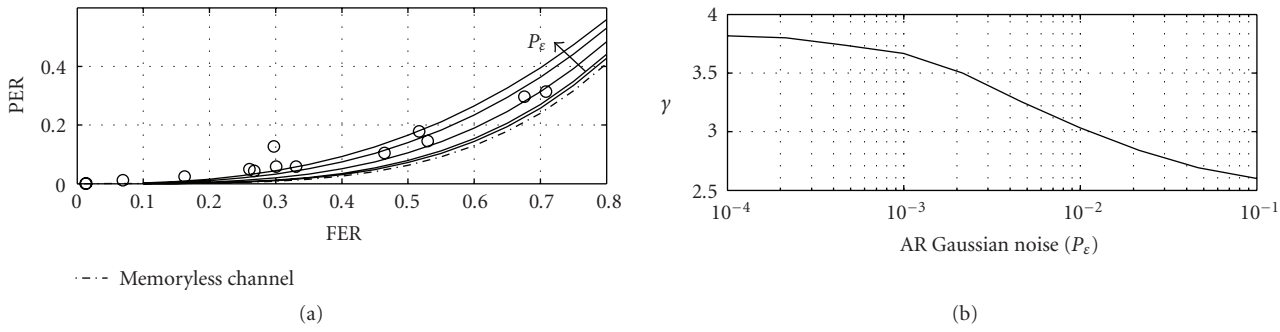


Figure 15: BEAR channel memory.

Markov chain, the EFB length remains quite close to the value used to configure the model, while BEAR is able to mimic the unpredictability which was observed in the real channel. Moreover, the values exhibited by the Shadowing model do not accurately reflect the empirically assessed behavior, providing a rather low EFB length.

Figure 12 shows the cdf of the maximum EFB length which was measured for each of the analyzed channel models. BEAR is able to simulate bursts longer than 100 frames, with a probability of 0.1, while the average EFB length stays below 3 frames (see Figure 11). However, in the most pessimistic GE configuration, characterized by an average EFB length of 5 frames, the maximum burst stays below 60 frames. This is a rather important aspect, since the presence of long erroneous frame bursts may lead to severe performance degradation. Again, we observe a very predictable behavior of all the approaches, except BEAR. Moreover, the most remarkable aspect is that the maximum erroneous bursts for the different GE instances and the Shadowing propagation are much lower than the values which are observed in the case of the real channel, as can be seen by comparing those results with the values presented before (see, e.g., Table 1 and Figure 3).

In most of the 802.11 configurations which are usually employed, the MAC layer specifies that before discarding a MSDU, it can be retransmitted up to a certain number of times. As was discussed before, in the particular wireless cards which were used during the measurement campaign, this parameter was set to 3 frames, that is a datagram would be lost only after four consecutive frames do not correctly arrive at the receiver. The consequence is that, as was also seen before, the PER does not match the FER. This is also reflected in the simulation, as can be seen on the PER cdf (Figure 13), which also yields the same conclusions which were discussed before for the FER regarding the predictability of all the approaches except BEAR.

One interesting result which can be inferred by analyzing these results is that in the case of the Shadowing channel model (memoryless) the PER values are rather low. In general, the PER could be calculated as

$$\mathrm{PER} = \mathrm{FER}^{\gamma}, \tag{3}$$

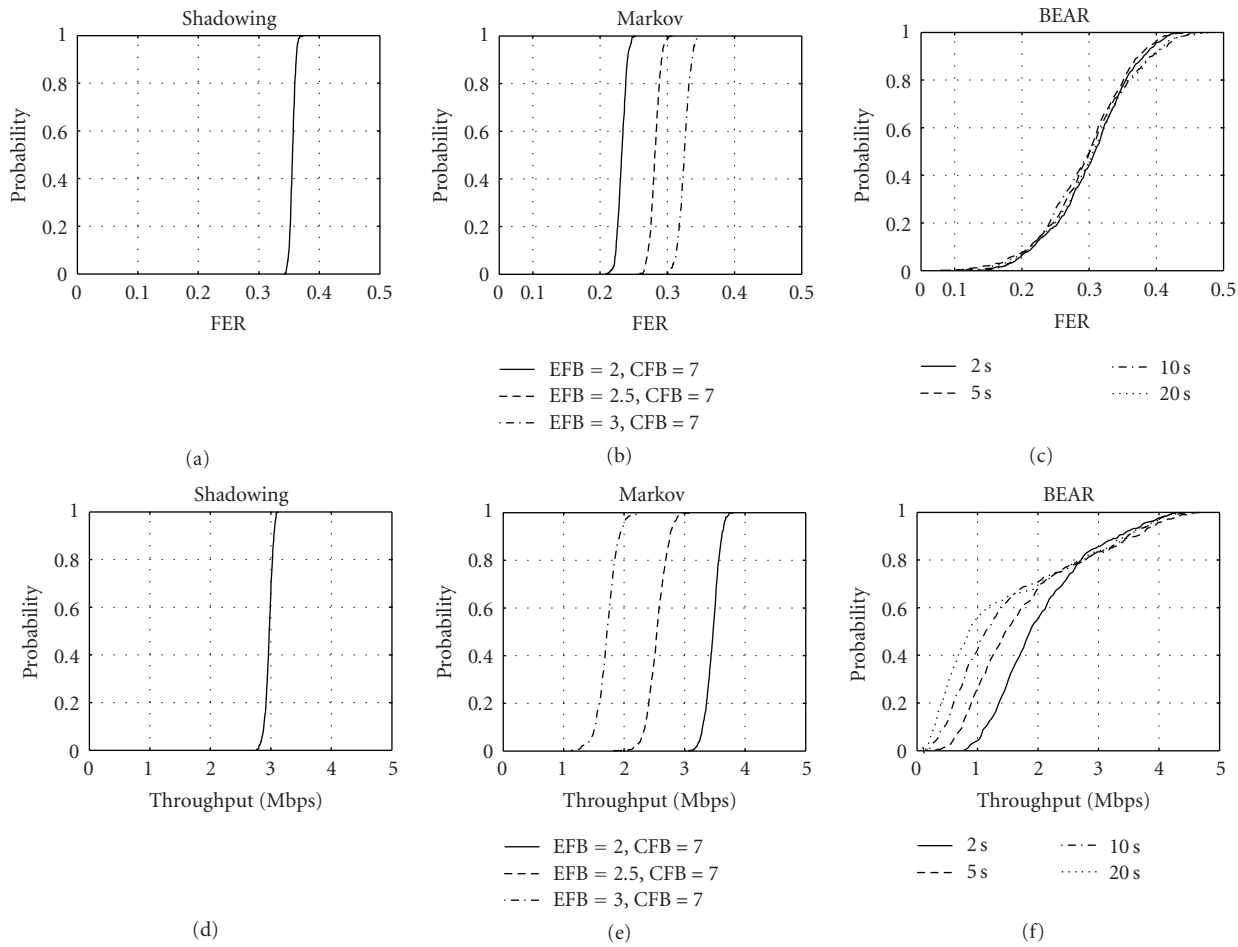


Figure 16: FER (a-c) and Throughput (d-f) for TCP connections using the three channel models.

where γ gives an idea of the channel's memory. For instance, on a memoryless channel γ equals 4, since there must be four consecutive corrupted frames before a datagram gets lost and, in this case, having an error on a frame is a completely independent process. As can be seen in Figure 14, the Shadowing follows the performance of a memoryless channel. On the other hand, and just for a single BEAR instance, a relatively large range of FER/PER pairs are obtained, implying some degree of memory in the channel's behavior. In this case, it is possible to estimate the gamma value which best fits the overall set of points. We observed that this memory could be tweaked by changing the power of the white noise (P_ε) which served as an input to the BEAR channel model.

Figure 15 shows the results achieved for different values of $P\varepsilon$, compared with those from a memoryless channel, and the ones which were empirically observed. Each of the corresponding lines represents the best fit of an overall set of 500 points, which were obtained after the corresponding independent simulation runs. The figure also shows the relationship between $P\varepsilon$ and the corresponding γ parameter. As can be seen, an increase in $P\varepsilon$ yields a higher correlation between consecutive erroneous frames, since γ decreases.

4.2. Results Based on TCP Traffic. For the TCP case, the configuration of the different channel models is the same as the one used before with UDP traffic, although the coherence time will be modified, so as to assess its influence on TCP performance, especially over the idle times. In order to generate TCP traffic, a 10 Mbyte file was transferred, using FTP, and 500 independent runs were used for each of the cases, so as to guarantee tight confidence intervals. In addition, according to what was discussed before, we assume that all TCP ACKs arrive free of errors.

First, Figure 16 compares the frame error rate as well as the TCP throughput which was observed with the different channel models. The main conclusion which can be obtained is that the only channel which is able to model the relevant variability observed over the real channel is BEAR, since the two traditional approaches offer, for a particular configuration, a very predictable behavior, both for the FER and the throughput, as was also seen in the UDP-based characterization. On the one hand, we can see how the proposed channel is able to accurately capture the FER empirically observed, since it ranges between 0% and 50%, which are exactly the same values assessed during the measurement campaign, as can be seen in Table 2. For the two legacy approaches (both Shadowing and Gilbert-Elliot)

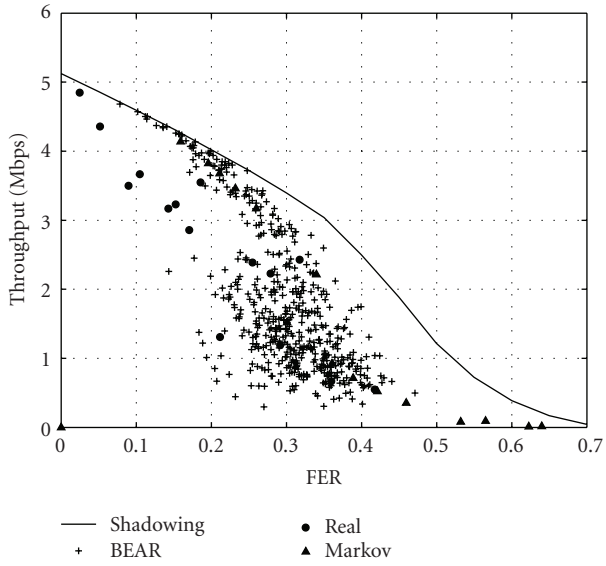


Figure 17: Relationship between FER and TCP Throughput.

the behavior is rather predictable, as was also seen before. The figure also implies, for the BEAR case, that the effect of the coherence time is not very relevant, since there is not much difference between the various configurations. One interesting aspect which we can observe is that the FER values are, in the case of the BEAR model, a bit lower than the ones observed for UDP traffic; this result, which actually reflects what is experienced over a real link (the TCP transmitter stops sending traffic when it detects unfavorable conditions in the link), is not correctly captured by the Shadowing model.

The above discussion can also be applied to the throughput results which are observed with the three channel models. As can be seen, both the Shadowing and the Markov (for a particular configuration) approaches show a rather predictable TCP performance, while the proposed BEAR model is able to capture the variability which was observed over a real channel: we can see how the performance ranges from rather low values (close to 0.5 Mbps) to almost 4.5 Mbps, reflecting the empirically measured performances; in addition, the effect of the coherence time can be seen (at least for the worst performance values); for these situations, the figure shows that the longer the coherence time, the lower the throughput. The main reason for this is that the relevance of the idle times becomes higher when we increase the memory of the channel.

In order to better reflect the influence of the FER on the TCP performance, and to compare the different approaches, Figure 17 shows the relationship between both parameters. In the case of the BEAR channel model, we chose a coherence time of 5 s, which is a sensible value (according to the previous results), and since the behavior is rather unpredictable, we plotted the 500 values which were obtained for this single configuration of the model. However, for both the Shadowing and the GE channels, we represented their average values, and for the former, we modified the configuration, so as to be able to capture a range of FER from 0 to 70%. We can see that the Shadowing model establishes an upper limit for the TCP throughput, since it corresponds to a memoryless channel. When the FER remains below 10%, the results provided by the BEAR model are pretty close to the Shadowing ones and it is within this FER value when we observe a greater difference with the real behavior of TCP. For greater FERs the relevant variability which is captured by the proposed model covers the broad range of results empirically obtained. However, we see that the Markov model is either rather close to the memoryless bound (up to FER around 25%) or far from it (greater FER values). It is worth recalling that neither of the two traditional approaches would be able to reflect the large variability which was observed over the real channel.

As was discussed before, two of the key parameters with greatest influence on the TCP performance are the number of retransmissions performed by the TCP transmitter so as to overcome the different IP losses, and, especially, the presence of idle times in the transmitter. Figure 18 shows the probability functions of both metrics, for the three channel models which we are analyzing. The general statements which were made before for the performance and the FER still apply here, since we only observe some degree of variability for the BEAR channel. The number of retransmitted segments for both the Shadowing and the Markov channels is very predictable, while real wireless links are usually characterized by presenting a broad range of values (as was seen before), which is indeed reflected by our proposed model (from almost 0 to >500 retransmissions). In this case, the results do not imply a relevant influence of the coherence time on the overall number of retransmissions. It is even more interesting to analyze the inactivity time which we observed using the different wireless channels; we clearly see how neither of the two traditional approaches we are studying is able to capture the real behavior previously described. When the Shadowing model is used, there are almost no idle times at all, due to the fact that this approach fails to reflect any memory; hence, it is not able to reflect the relevant erroneous bursts which lead to the aforementioned inactivity times at the TCP transmitter due to consecutive RTO expirations. On the other hand, for the Markov there were some relevant idle times but only for the most pessimistic configuration, although they do not reach values higher than 12 seconds, which do not reflect the real behavior of the channel either, despite that the number of retransmissions for this particular configuration is greater than the values which were empirically observed. The BEAR model, again, succeeds in reflecting the empirical behavior, since the probability functions show that it captures the times observed during the measurement campaign; on the other hand, it can be seen that in this case there is a clear impact of the coherence time; according to the different probability functions, it seems that a coherence time of 5 s seems to be a sensible choice for this parameter.

In TCP, the retransmission of a segment can be triggered either by Fast Retransmit or by the expiration of the retransmission timeout. In both cases, the retransmissions are used to recover from the IP losses which the MAC retransmission scheme was not able to deal with. Figure 19

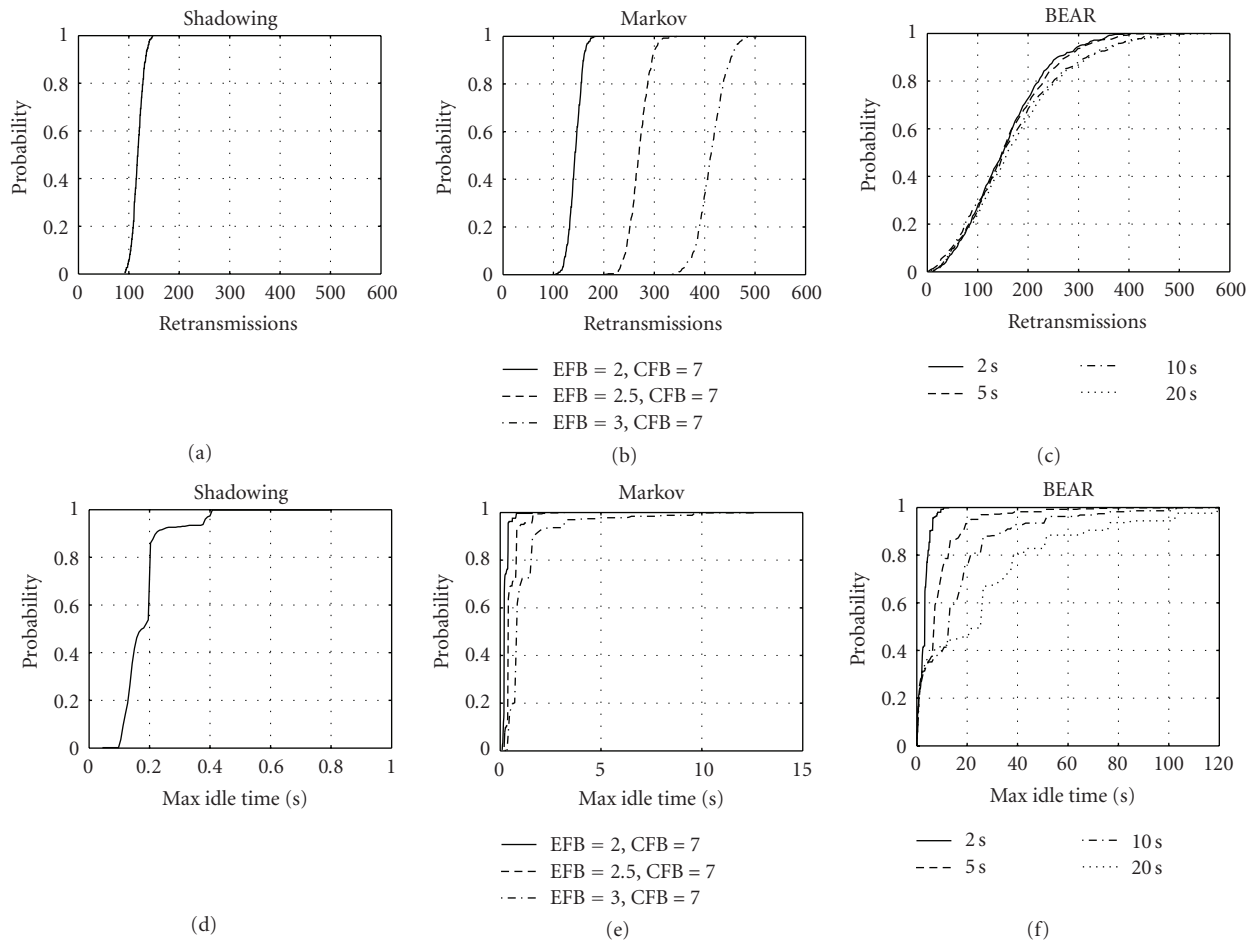


Figure 18: TCP retransmissions (a–c)and maximum idle time at the transmitter entity (d–f) for TCP connections over the three different channels.

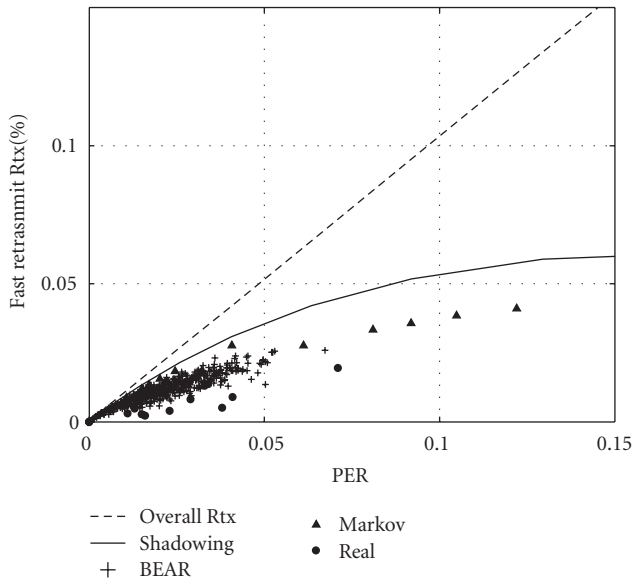


Figure 19: Comparison of Fast Retransmit and RTO expiration retransmissions.

shows the relationship between the percentage of retransmissions (compared to the overall number of segments) and the packet error rate. In all cases, this relationship follows a linear trend; this was expected, since TCP retransmissions are recovering formerly lost IP datagrams, considering (as we have done) that TCP ACKs do not suffer any FER. However, the presence of Fast Retransmit is much higher in the Shadowing model, and therefore, in this case, there would be fewer retransmissions triggered by the expiration of the timeout, which causes a less relevant inactivity time (as we have indeed seen before). We also observe that, in the case of the GE model, although without reaching the limit established by the memoryless channel, the influence of the retransmissions triggered by the reception of a triple acknowledgment is rather more relevant than in the real measurements, especially when the PER remains below 5%. The BEAR channel model, though, since it is able (for a single configuration) to provide a much broader range of behaviors, can mimic, with a greater degree of accuracy, the empirical performance.

A rather significant assessment of the variability which the BEAR model is able to reflect, as opposed to the more traditional approaches, can be inferred by analyzing the

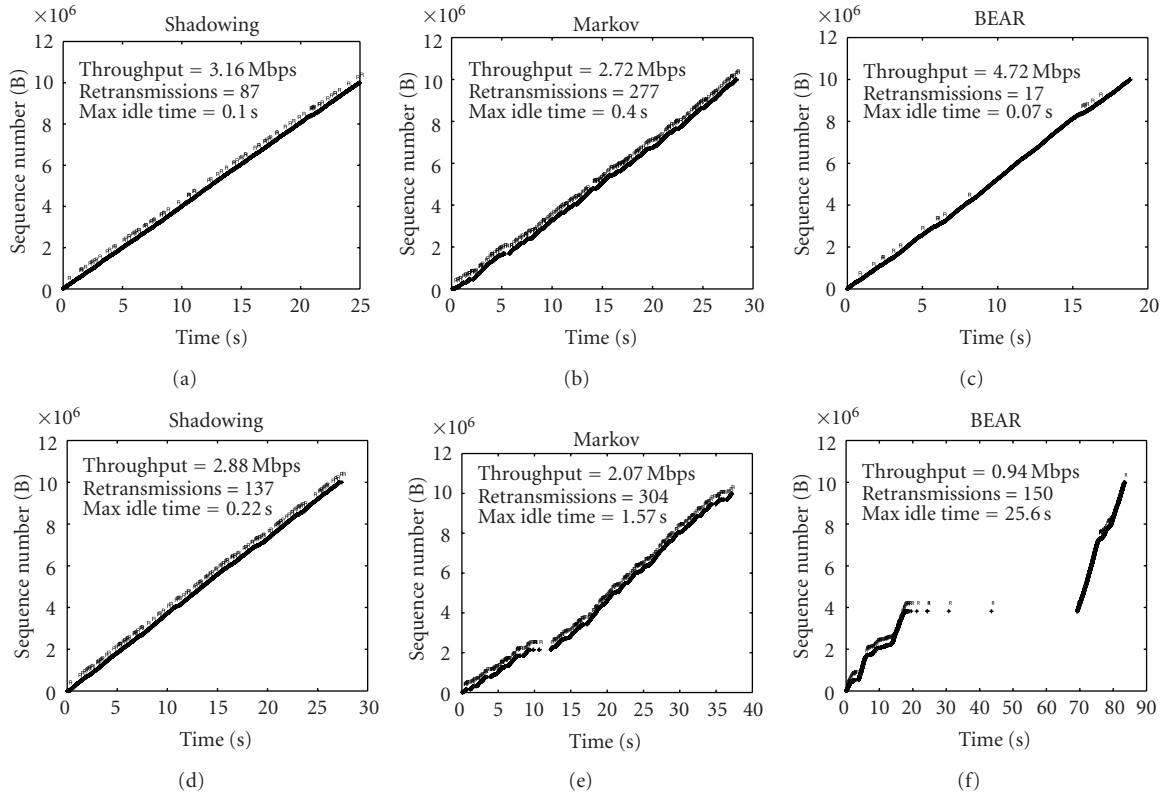


Figure 20: Different TCP Connection evolution for the three channel models.

temporal evolution of particular TCP connections. In this sense, Figure 20 shows the Time Sequence Graphs of two representative runs performed for each of the analyzed models. In particular, we seek to show one connection with a high performance, compared with one in which a poor throughput was achieved. As can be seen, the rather predictable behavior which characterizes both the Shadowing and the Gilbert-Elliot models is again easily recognizable, as well as the little similarity they have with the real performance; we can see, for instance, that the retransmission of TCP segments is almost constant during the whole transfer. On the other hand, and for a single BEAR configuration, the figure shows two completely different situations: in the first one the TCP performance is rather high, and the transmitter does not incur in any relevant inactivity period; however, in the second one, the performance is heavily impaired by the presence of quite a long idle time, similar to the ones which were empirically observed due to consecutive RTO retransmissions of the same segment. Furthermore, the number of retransmissions is one order of magnitude higher in the latter case.

5. Discussion and Related Work

With the channel model proposed in this paper, we have addressed the specific needs which some earlier works have identified [8]. For instance, the authors in [9] use video transmission to demonstrate that legacy channel models, for example, GE, do not capture periods characterized by high packet loss rates, which are relevant for both the perceived user quality and the design of error control methods. In [10], the GE error model is used to assess the perceived voice quality, advocating an adaptive FEC scheme to enhance it; the authors conclude that they would like to extend their work to more realistic channel models.

BEAR is able to use link quality (based on the SNR) to modulate its behavior, and it also depends on the distance between transmitter and receiver; in this sense, it is suitable to be used to analyze scenarios where node mobility needs to be considered, as well as cross-layer optimization techniques. For instance, in [5], the authors advocate the use of SNR to modulate routing algorithms to be used over multi-hop networks. The authors of [2, 4] conclude that for indoor propagation environments, the SNR is an appropriate metric to assess link quality.

Although the use of AR filters to model fading channels was presented for example in [6], there are no other works which have integrated such an approach within a network simulator platform such as ns-2. It is logical that predicting link quality based on any parameter might lead to inaccurate results, but according to the conclusions presented in [11], it seems that being able to tweak the configuration of the channel model based on the binary rate and the frame length gives some degree of flexibility and trustworthiness to the model. In that work, the authors claim that the Packet Delivery Ratio depends on both the packet length and the binary rate, and BEAR can be configured to correctly account for the two parameters, as was discussed before.

The authors in [12] follow a similar approach, but we differentiate from them mainly in two key aspects: first, they do not consider any memory in the channel, and thus, their modeling might not be able to capture the "bursty" behavior of the real propagation channel; secondly, although they state that there are differences between the various transmission rates, they find the relationship between the delivery ratio and the SNR for the lowest PHY-layer rate. However, BEAR can handle 11 Mbps and 2 Mbps transmissions.

In the same sense, the work presented in [13] states that in order to promote reliable results in their simulation-based analysis, the MANET research community should be able to include asymmetric links as well as being able to model time-varying fluctuations in link quality. We have seen that BEAR also fulfills these two requirements.

Finally, it is also worth mentioning the work presented in [14, 15]. In this case, the authors focus on an outdoor rural scenario to derive an empirical frame error model. They advocate the need to perform the same research for indoor environments, and they also share our view that currently used channel models (e.g., GE) do not accurately reflect real propagation characteristics.

The authors of [14, 15] have also made their traces available at the CRAWDAD project. We have looked at the data sets available at this project, but none of them fulfill our requirements. The one which was closest to our measurement campaign [16] (it uses 11 Mbps transmissions in the 2.4 GHz band) does not monitor the SNR of erroneous frames. In this sense, we can say that our traces might be used to obtain a more accurate estimation of the relationship between SNR and FER.

Other works which seek the improvement of the 802.11 wireless channel modeling in Network Simulator are [17, 18]. The former pays more attention to improving the interference model; however, it does not look into propagation aspects, thus failing to consider any memory in the wireless channel, which is the most distinguishing aspect of our proposal. It is also worth highlighting the effort which is being made in the development of ns-3; currently, the propagation models which are available do not account for the memory characteristic which was previously reported; furthermore, since the operation of the propagation module is almost orthogonal, we expect that BEAR could be integrated without major difficulties into this new version of the simulator.

6. Conclusions

Traditional research on wireless networks is usually based on different simulation platforms. The results achieved and performances are then extrapolated to real environments. This approach, although widely employed, is also known to have some shortcomings; one of the main criticisms lies on the lack of proper channel models. In order to address this aspect, in this work we have followed a different approach, in which the real performance of TCP and UDP protocols over an indoor wireless environment has been used so as to design and assess the validity of a novel channel model.

BEAR has a number of advantages compared to other approaches. First, it is able to reflect the "bursty" behavior which characterizes real indoor propagation and, in addition, it also captures the high variability which was empirically observed. On the other hand, since it accurately uses link quality (based on SNR) as a means to modulate the behavior, it can be used to analyze cross-layer optimization techniques. Another relevant advantage of BEAR is that it can be configured to manage different types of frames, considering their length, binary rate, and so forth. Furthermore, it correctly handles the dynamics of the wireless channel, and can be used to simulate either symmetric or asymmetric links.

The paper has shown, after an extensive simulation campaign in which BEAR was compared with two other traditional approaches (Shadowing and Gilbert-Elliot models), that it fulfills the requirements we had at the beginning, accurately reflecting the empirical behavior. In this sense, and for both UDP and TCP protocols, the performance achieved with the BEAR model is much closer to the real one than those of the two legacy approaches.

As future work, we would like to propose two main directions. The first one would be to benefit from BEAR's ability to accurately reflect real indoor wireless propagation, based on SNR, to analyze the performance of different protocols and algorithms, considering cross-layer optimization techniques. On the other hand, we would like to complement BEAR, by adding support for different 802.11 technologies; besides, since BEAR does respect the way the simulator deals with interference (which is handled by the MAC module), incorporating interference in the scenario should be rather straightforward. Furthermore, the comparison analysis will be extended, incorporating more complex channel models, such as Hidden Markov Models, with a greater number of states. Last, since the code will be made available to the scientific community, it is envisaged that its functionalities and capacity will be extended and improved, as long as it is used by other researchers.

Acknowledgments

The authors would like to thank the Spanish Government for its funding in the Mobilia (CELTIC Program) and "Optimización de Técnicas de Descubrimiento de Servicios sobre Plataformas Inalámbricas Heterogéneas" (TEC2006-05819) projects. The authors would also like to express their gratitude to the anonymous reviewers, since their detailed comments have helped them to improve the quality of the paper. This work was presented in part at the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007) and IEEE Wireless Communications and Networking Conference (WCNC2008).

References

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," *SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 121–132, 2004.

[2] J. De Bruyne, W. Joseph, L. Verloock, and L. Martens, "Evaluation of link performance of an indoor 802.11g network," in *Proceedings of the Consumer Communications and Networking Conference (CCNC '08)*, pp. 425–429, January 2008.

[3] M. García, R. Agüero, and L. Muñoz, "On the unsuitability of TCP RTO estimation over bursty error channels," in *Proceedings of the IFIP TC6 9th International Conference on Personal Wireless Communications (PWC '04)*, pp. 343–348, Delft, The Netherlands, September 2004.

[4] M. R. Souryal, L. Klein-Berndt, L. E. Miller, and N. Moayeri, "Link assessment in an indoor 802.11 network," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '06)*, pp. 1402–1407, April 2006.

[5] C. E. Koksal and H. Balakrishnan, "Proceedings of the Quality-aware routing metrics for time-varying wireless mesh networks," *IEEE Journal on Selected Areas on Communications*, vol. 24, pp. 1984–1994, 2006.

[6] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulation," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1650–1662, 2005.

[7] H. Lundgren, E. Nordström, and C. Tschudin, "Coping with communication gray zones in IEEE 802.11b based ad hoc networks," in *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia (WoWMoM '02)*, pp. 49–55, ACM Press, September 2002.

[8] H. Bai and M. Atiquzzaman, "Error modeling schemes for fading channels in wireless communications: a survey," *IEEE Communications Surveys and Tutorials*, vol. 5, no. 2, 2003.

[9] G. Convertino, S. Oliva, F. Sigona, and L. Anchora, "An adaptive FEC scheme to reduce bursty losses in a 802.11 network," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, November 2006.

[10] 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*, vol. 7, no. 10, pp. 1184–1199, 2008.

[11] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and M. Faloutsos, "Assessing link quality in IEEE 802.11 wireless networks: which is the right metric?" in *Proceedings of the 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '08)*, September 2008.

[12] A. Kashyap, S. Ganguly, and S. R. Das, "Measurement-based approaches for accurate simulation of 802.11-based wireless networks," in *Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '08)*, pp. 54–59, Vancouver, Canada, October 2008.

[13] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '04)*, pp. 78–82, Venice, Italy, October 2004.

[14] P. Barsocchi, G. Oligeri, and F. Potortì, "Measurement-based frame error model for simulating outdoor wi-fi networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1154–1158, 2009.

[15] P. Barsocchi, G. Oligeri, and F. Potortì, "Frame error model in rural wi-fi networks," in *Proceedings of the International Symposium on Modeling and Optimization (Wiopt '07)*, pp. 41–46, Limassol, Cyprus, April 2007.

[16] A. Prabhu Subramanian, J. Cao, C. Sung, and S. R. Das, CRAWDAD data set sunysb/multi_channel (v. 2009-02-24), February 2009.

[17] N. Baldo, F. Maguolo, and M. Miozzo, "A new approach to simulating phy, mac and routing," in *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS '08)*, October 2008.

[18] "The ns-3 network simulator," http://www.nsnam.org/.