Combination of Random Linear Coding and Cross-Layer Opportunistic Routing: Performance over Bursty Wireless Channels

Pablo Garrido†, David Gómez†, Ramón Agüero† and Joan Serrat$

† University of Cantabria
Santander, Spain
{pgarrido,dgomez,ramon}@tlmat.unican.es

$Universitat Politècnica de Catalunya
Barcelona, Spain
{serrat}@tsc.upc.edu

***Abstract*—In this work we focus on the application of an intra-flow linear random coding scheme over wireless mesh networks. We propose a cross-layer technique to balance the load between relaying nodes, considering the quality of the wireless links. We assess the performance of our proposal by means of an extensive simulation campaign carried out over the ns-3 platform, exploiting a channel model based on a Hidden Markov Process, which accurately mimics the *bursty* behavior that is observed over real indoor channels. The results show the benefits of the proposed cross-layer technique. In addition, we also discuss the trade-off (between accuracy and overhead) that needs to be considered when obtaining the information that such scheme rely on. Our simulations yield a 10% performance gain when exploiting the link quality information, as it is already obtained by various routing protocols.**

***Index Terms*—Random Linear Coding, Opportunistic Routing, Cross-Layer, Wireless Mesh Networks, Simulation, Bursty Channel Models.**

I. Introduction

Wireless networks have become the most widespread communication alternative, due to (amongst other things) the recent advances in miniaturization and the increasing availability of advanced terminals (smartphones and tablets). However, the behavior exhibited by legacy protocols (mainly TCP) over such networks might be quite poor. As a consequence, several proposals have been made in the latest years to overcome the low performance that TCP shows over this type of networks and, in particular, over Wireless Mesh Networks (WMNs). One of the most promising techniques is Network Coding (NC), whose basic operating principle is that intermediate nodes modify (code) the received packets before sending them again. Network Coding is thus said to question the suitability of the legacy *store-and-forward* paradigm.

NC was originally coined by Ahlswede *et al.* in [1]. Since then, multiple proposals have been made; Chachulski *et al.* [2] proposed MAC-independent Opportunistic Routing & Encoding (MORE), a joint opportunistic routing/NC protocol, in which intermediate routers that overhear a transmission do not forward the same information, but a *recoded* version of the already received messages. Gomez *et al.* [3] proposed and studied the performance of a different approach, based on the combination of Random Linear Coding and the UDP protocol, to offer a reliable communication service.

Studying the performance of this type of techniques over real testbeds is usually rather complex; the size of the topologies and deployments that can be used is limited and, furthermore, there is little flexibility to tune some of the protocol operational parameters and to reproduce the same conditions in different experiments. On the other hand, the use of simulation tools, albeit overcoming most of the previous limitations, is said to offer less realistic results. One of the most widespread criticisms is the small accuracy of the most popular propagation models. Cardoso *et al.* [4] proposed a novel approach based on a Hidden Markov Process (HMP), showing that it is able to accurately reproduce the *bursty* behavior of real wireless environments. In [5], Gómez *et al.* improved such approach, decoupling it from a particular service pattern, by configuring the corresponding Markov chains on a time-basis. In this work we exploit this channel model to assess the performance of the combination of Network Coding and the proposed cross-layer technique.

The main contribution of this paper is therefore the systematic analysis of a novel probabilistic forwarding scheme to enhance a random linear coding solution, which can improve the performance only using information already exploited by most of current routing protocols. We study the advantages and potential limitations of using cross-layer techniques over opportunistic wireless networks. The analysis is carried out over the `ns-3` [6] simulator, and we also discuss the issues that may arise when applying such techniques over real networks.

The rest of this paper is structured as follows: Section II briefly describes the channel model and the *intra-flow* network coding solution used in this study and it also depicts the proposed probabilistic forwarding scheme. Section III discusses the performance of the cross-layer approach, identifying its advantages as well as its potential limitations; finally, Section IV concludes the paper, advocating some items that will be tackled in our future research.

II. Design and implementation

A. Hidden Markov Process Model

One of the contributions of this work is that we assess the performance of a Random Linear Network Coding (RLNC) scheme over *bursty* wireless links, modeled by means of *Hidden Markov Processes*, which have been shown [5] to provide an accurate emulation of real indoor channels. In the following we summarize the most relevant aspects of this model.

978-1-4673-6782-0/15/$31.00

A *Hidden Markov Process* is defined as a system with N states, S_i, $i = 0 \dots N-1$. The transitions between two states are defined by a set of stochastic probabilities, which are referred to as transmission probabilities, where $a_{i,j}$ represents the probability of shifting from the current state S_i to S_j. In these particular Markov models, states are said to be *'hidden'*, since they do not yield a single output value, but each of them might lead to various potential outputs, according to a probability given by $b_i(k)$, where i refers to the current state, and k corresponds to the system output value.

Technically speaking, a *Hidden Markov Process* can be modeled by the operational parameters enumerated below.

- Number of states, N
- Number of output values, M
- Transmission matrix (A), a matrix of dimension $N \times N$, where each element, $a_{i,j}$, corresponds to the transmission probability between states S_i and S_j.
- Emission matrix (B), matrix of dimension $N \times M$, where each row corresponds to the probabilities associated to the M potential outputs for a particular state.
- The initial probability distribution, $\vec{\Pi}$, a column-vector with N elements, which establishes the probability of being at a particular state at $t = 0$.

One of the most relevant features of this approach is that it translates the frame-based (discrete) configuration to a time-based (continuous) one. For that, it establishes the probability density function of the sojourn time at each of the states, which is modeled as a negative exponential random variable, $f_{T_i}(t_i) = \lambda_i \cdot e^{-\lambda_i \cdot t_i}$, where its mean, corresponding to the average time at a particular state i, can be calculated as $\bar{T}_i = \frac{1}{\lambda_i} = \frac{\varphi_i}{1-a_{i,i}}$, where φ_i is the average time between consecutive packets at state i [5].

We use a $4-state$ hidden Markov chain and we assume that there exist two possible outputs: a correct or erroneous reception. The corresponding parameters are obtained by training the system with traces obtained during a measurement campaign carried out over a real wireless channel.

B. Intra-Flow network coding

In this work we start from the solution proposed in [3], a combination of the legacy UDP protocol and a RLC scheme, to offer a reliable communication service. Besides, we allow the relaying nodes to recombine the received packets, as explained in [2] and [7]. Below we briefly describe the protocol operation.

The NC module at the transmitter receives *native* packets, p_i, from the upper layer. Once it has received the K^{th} one, it encodes the whole block (composed by those K packets) using the following scheme: $p'_j = \sum_{i=0}^{K-1} c_i \cdot p_i$, being c_i random coefficients obtained from a finite field $GF(2^q)$. Afterwards, the source node sends downwards the generated encoded packets, p'_i. It will keep sending coded packets until the destination is able to decode the original information.

When the destination receives a packet, it extracts the coefficient vector, which is transmitted within the *Intra-flow* NC header, and checks whether this vector is linearly independent from those stored previously. In such case, the destination would keep the coded packet in its *reception buffer*. Once the destination has received K linearly independent coefficient vectors, it sends an acknowledgment to the source node, since it can then decode the complete block of K packets, retrieving the original information. Afterwards, the receiver's *NC* entity sends up the K native packets to the application. If the acknowledgment packet was lost before reaching the source node, this would keep sending coded packets and, each time the destination receives a packet belonging to an already decoded block, an acknowledgment would be immediately sent.

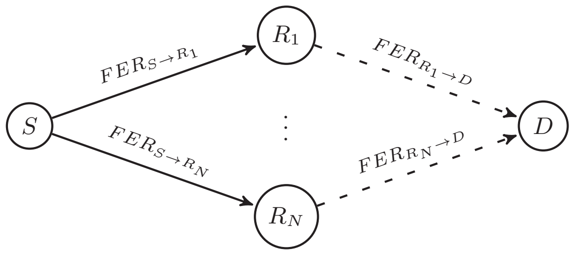


Figure 1: Scenario with multiple intermediate nodes

The intermediate nodes also process the information. When one relaying entity receives a coded packet it checks whether the coefficient vector is linearly independent from those it has previously stored, following the procedure that was described for the receiver node. Each time a relaying entity gets a transmission opportunity, it builds a *re-coded* packet, $p''_{,} = \sum_{i=0}^{T} c'_i \cdot p'_i$, where T is the number of coded packets already stored in these nodes. As can be easily inferred, these *re-coded* packets are indeed a linear combination of the originals ones.

C. Probabilistic Scheme

Many works ([2], [5], just to name a few) have studied the performance improvement when multiple intermediate nodes participate in the transmission process, using a *recoding* mechanism (RLNC). Figure 1 shows an illustrative example, in which the source node S transmits packets to the destination, D, using N intermediate nodes $(R_1, R_2, \cdots, R_N)$. Each of these relaying entities receive coded packets from S, and forward a random re-combination of them to the destination. Let us assume that there are two forwarding nodes, and that the links $S \to R_1$ and $S \to R_2$ are characterized by the same quality, with an error rate of 0.5; under these circumstances, the probability that one of them sends different information to the destination is 0.25, thus increasing the overall network performance. A similar conclusion could be drawn when the links $R_i \to D$ are prone to induce errors, since the information sent by the various intermediate nodes might be slightly different, increasing the probabilities that the receiver gets new information with each incoming packet.

In order to further enhance the system performance, we also propose a cross layer technique to prioritize those forwarding entities whose links exhibit the better qualities. The transmission opportunities are granted according to a probability $\mathcal{P}_i^{\text{forward}}$ that is inversely proportional to the link error rate, as shown in Eq.1. The γ parameter can be seen as an aggressiveness factor: the higher γ is, the more transmission opportunities will be granted to those nodes with better quality links. Note that the sum of these probabilities, considering all the intermediate nodes, equals 1.

$$\mathcal{P}_i^{\text{forward}} = \frac{1}{FER_i^\gamma} \cdot \frac{1}{\sum_{j=1}^{N} \left(\frac{1}{FER_j^\gamma} \right)^{-\gamma}} \tag{1}$$

It is important to highlight that, in order to use the aforementioned cross-layer technique, we could exploit the fact that there are several routing protocols proposed for wireless mesh networks that require the dissemination of loss probabilities across the network. Some of the most remarkable examples are [2] and [8]; both of them aim at reducing the number of transmissions that are needed to deliver a packet from the source to the destination. They exploit the Expected Transmission Count (ETX) metric, originally proposed by De Couto *et al.* in [9]. For example, in [2] the authors assume that the source is aware of the ETX metric of every other node in the network.

The ETX (which can be seen as an estimation of the link loss probability) could be measured over a real network by means of Probe packets of a fixed size that are periodically sent. Each node broadcast one Probe packet every τ seconds, while it also remembers the number of Probe packets it has received during the last ω seconds; with these two parameters, it can estimate the delivery ratio r from a particular sender at any time t, as can be seen in Eq. 2, proposed in [9], where $\xi(t_a, t_b)$ is the number of Probe packets received in the interval between t_b and t_a. It is worth highlighting that the ω and τ parameters allow establishing a certain trade-off between the accuracy of the link measurements and the additional overhead that would be required in the network.

$$r(t) = \frac{\xi(t-\omega, t)}{\omega/\tau} \tag{2}$$

III. Results

In this section we discuss the most relevant results after carrying out an extensive simulation campaign over the `ns-3` simulator, which was modified to integrate the entities that are required to perform the coding and decoding procedures, as detailed in [7]. First, we assess the performance of the RLNC scheme over a scenario similar to that shown in Figure 1; in this case, we assume that all relaying entities have already all the information to be sent to the receiver and therefore they act as source nodes that are sending the same data to a single destination node.

Table I summarizes the parameters that define the simulation setup. The wireless channels are configured according to the *IEEE 802.11b* specification and the corresponding link qualities are randomly selected from a set of configurations that are available for the *HMP* model: *ideal* ($FER = 0.0$), *good* ($FER \approx 0.163$), *average* ($FER \approx 0.298$), and *bad* ($FER \approx 0.517$). Regarding the RLNC scheme configuration, we use blocks of 32 packets and we work with a finite field $GF(2)$, since these parameters, which were studied in previous works, [7], lead to a good performance. In [3] we also concluded that it was worth disabling the 802.11 retransmission scheme. The data to be transmitted on a single session (file) has a size of 1000 blocks. On the other hand, the probe packets are generated by a virtual application that sends (at a fixed rate) 1000 Byte packets, which are granted a higher priority when contending for the wireless channel.

Table I: Simulation setup (common attributes)

Feature	Value
Lower layers	IEEE 802.11b (11 Mbps)
Error model	*Hidden Markov Process*
RTS/CTS	Disabled
IEEE 802.11 RTX	0
Routing	Static routing tables
Transport level	UDP + RLNC
Number of Packets	32000
Packet length	MTU 1500 B
Block Size (RLNC)	32 packets
Galois Field	$GF(2)$
Simulations	1000 independent runs/point

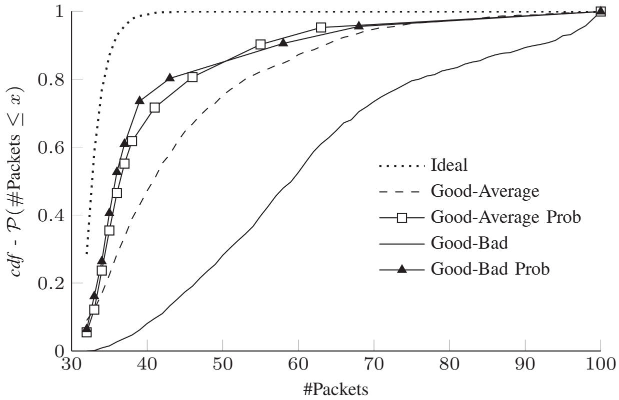
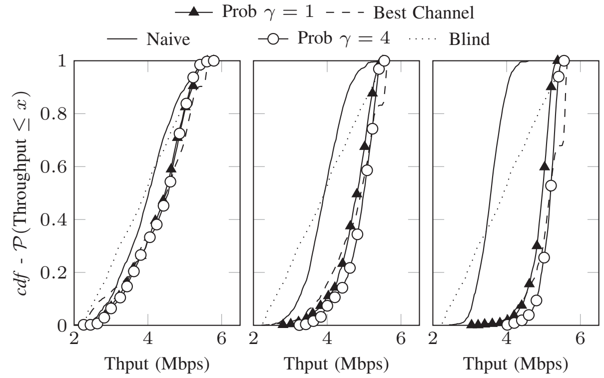


Figure 2: *cdf* of the # of transmitted packets to decode a block

We initially study the benefits of using the optimal probabilistic scheme. We assume that each node is aware in real-time of the various links' qualities, by exploiting the possibilities brought by the simulation framework. Although we know that this is an unrealistic situation, we are interested in understanding the potential loss that might be attributed to the limitations of the link quality estimation schemes that are usually employed by different routing protocols. Figure 2 shows the decoding probability as a function of the overall number of packets received by the destination. We assume that there are two sources transmitting the same information to the destination, and we establish the same channel conditions for the two links (this means that they exhibit the same average behavior, but since they are indeed independent random processes, their instantaneous conditions might not be alike), using the four configurations that were introduced earlier. First, we can see that the number of excess packets that are needed to successfully decode a block grows as the links' quality gets worse. In addition, the results also yield that the proposed probabilistic scheme is able to heavily reduce the amount of excess packets that are needed; we observe that it is able to decouple the performance from the particular characteristics of the wireless links and, as can be seen, the *good-bad* configuration equals the performance achieved with the *good-average*, being always higher than the one seen with this latter setup, if the probabilistic scheme was not used.

In a second experiment, we keep the optimal probabilistic scheme, but we increase the number of source nodes, $N = 2, 4, 8$ to assess the impact of having a larger number of nodes contending to access the shared wireless medium. Figure 3 shows the *cdf* of the throughput that was observed during the connection. We study five different configurations: a *naive*



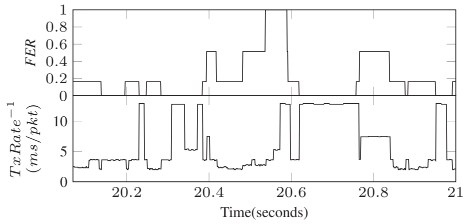
(a) 2 Nodes (b) 4 Nodes (c) 8 Nodes

Figure 3: *cdf* of the throughput for the different transmission schemes

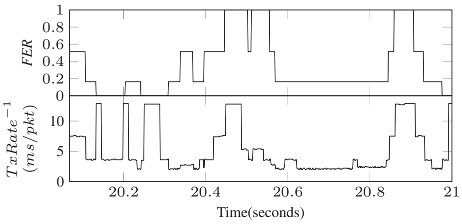
scheme, where all source nodes try to access the wireless channel equally; the *probabilistic scheme* with two different γ values, $\{1, 4\}$; the *best solution*, in which we ensure that only the node having the best link to the destination is sending packets; and finally, a *blind scheme*, where a single source is randomly selected, without considering the corresponding qualities. For the *naive scheme* there is a clear decrease on the system performance when the number of source nodes increases, due to the larger number of entities contending for the channel. On the other hand, the probabilistic scheme is able to maintain the performance, even for a larger number of transmitters, due to the fact that it effectively distributes the use of the wireless channel. In fact, the throughput that is seen for the *probabilistic scheme* almost reaches the one that would have been obtained with the *best solution*, especially if $\gamma = 4$.

In order to assess how the probabilistic scheme behaves, Figure 4 shows the instantaneous evolution of the Frame Error Rate (FER) for the two considered links (i.e. $N = 2$), and how the corresponding transmission rates (we represent the time between consecutive transmissions) are dynamically modified. We assume that the two links are equally configured and that the probabilistic scheme is configured with $\gamma = 4$. It can be seen that when the qualities of both links are alike (same FER value), a packet is sent approximately every $3.6\ ms$ (i.e. the two nodes together can be said to saturate the wireless channel), while if one of the links is error-free, the corresponding source acquires the control of the channel, sending a new packet every $\approx 1.8\ ms$.

As was already mentioned, we have so far assumed that nodes are able to monitor all link qualities in real-time, by exploiting the possibilities that are offered by the simulation platform. Since this situation does not reflect a realistic situation, Figure 5 can be used to assess the accuracy of the link quality estimation schemes that are used by different routing protocols, as was explained in Section II. In particular we use two different configurations, sending either 10 or 100 probe packets per second, that are used to estimate the Packet Error Rate (PER) and, afterwards, the ETX. In order to help the readability of the graphs, we apply a low-pass filter to smooth the real instantaneous FER; we are mostly interested in assessing whether the estimation scheme is able to follow

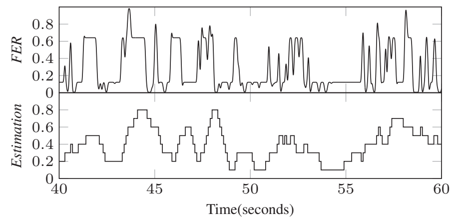


(a) Link $S_1 \rightarrow D$

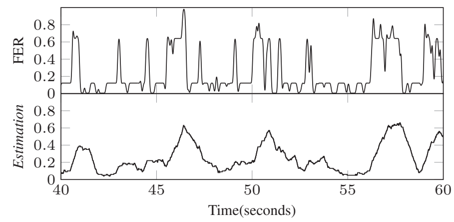


(b) Link $S_2 \rightarrow D$

Figure 4: Transmission rate evolution Vs. wireless links instantaneous FER



(a) 10 packets per second

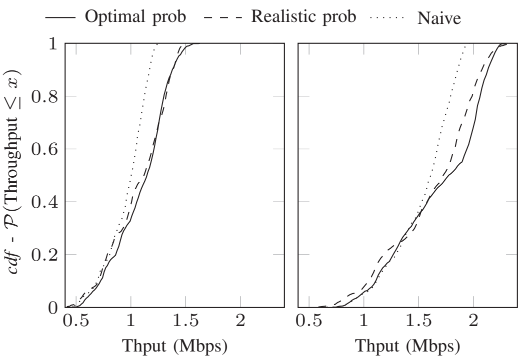


(b) 100 packets per second

Figure 5: Impact of the *Probe* transmission frequency over the FER estimation

the long-term variations of the link quality. As can be seen, both configurations are able to provide an accurate estimation of the FER value; the figure shows that a higher frequency of probe packets leads to a slightly more precise estimation, although there is a clear trade-off between this accuracy and the increase of the required overhead, as will be discussed later in this section.

Finally, we deploy the scenario depicted in Figure 1, with $N = 4$, randomly selecting the quality of those links between R_i and D, assuming an error-free condition for all links between the source and the forwarding nodes, and we carry out a Monte-Carlo analysis, repeating the experiment 1000 times. Figure 6 compares the results that were obtained with three different configurations: a *naive scheme*, in which the probabilistic mechanism is disabled; an *optimal probabilistic scheme*, where we again exploit the possibilities of the simula-



(a) 100 packets per second (b) 10 packets per second

Figure 6: *cdf* of the throughput for the probabilistic transmission scheme

tion platform and we use the FER to update the transmission on a real-time basis; finally we use a *realistic probabilistic scheme*, where the FER value is estimated by means of Probe packets that are sent by each of the source nodes, as it would happen over real networks. To ensure a fair comparison, the source nodes send probe packets in all the studied configurations, regardless of whether they are really needed or not. Figure 6a corresponds to the configuration where 100 Probe packets are sent per second, while the results shown in Figure 6b were obtained when only 10 Probe packets are transmitted per second. Since the estimated FER is more accurate when the frequency of the Probe packets is higher (as was seen earlier), the performance achieved with the realistic configuration almost matches that observed for the ideal case. On the other hand, when a less accurate estimation scheme is used (sending 10 probe packets per second), there is a larger loss compared to the optimal configurations; we can also see that in all cases, including the *naive scheme*, the throughput is higher, as a consequence of the lower number of Probe packets that are sent, causing less contention. In any case, the probabilistic scheme that would have been used in a realistic implementation always leads to a higher throughput (the gain is approximately 10 %) while the difference with the optimum configuration is not very large.

IV. Conclusions and Future Work

In this work we have presented and analyzed a probabilistic forwarding scheme that has been conceived to be applied over opportunistic networking scenarios in combination with an *intra-flow* network coding solution. These two techniques have a great potential when working together, since the potential advantages brought about by the opportunistic routing can be exploited. Furthermore, considering that all nodes participate in the transmission procedure, the use of the available resources can be optimized by being aware of the channel qualities. We have also assessed that the estimation techniques that are promoted by several routing protocols, in particular the ETX metric, which are mostly based on the transmission of Probe packets, are enough to leverage a performance improvement, even if we compare it to an ideal situation, in which all nodes are aware of the instantaneous link qualities in real-time. One of the key contributions of this paper is that the performance of the proposed solution has been studied over a channel model that accurately mimics the *bursty* behavior that characterizes real indoor environments, based on a *HMP* process.

Using an extensive simulation campaign over the `ns-3` framework we have first studied the number of required transmissions to successfully decode a block of original information, analyzing the impact of having different link qualities. We also assessed the relevant performance enhancement brought about by the probabilistic retransmission scheme.

In order to challenge the proposed solution over realistic conditions, we also thoroughly analyzed the link quality estimation procedures that are used by several routing schemes, all of them based on the periodic transmission of Probe packets. In particular we studied the impact of the transmission frequency, comparing the estimated FER with the real one. The results yield that there is a trade-off between the accuracy that can be obtained and the additional required overhead. In any case, the estimated FER mimicked quite well the real one, and the performance of the complete solution (using a realistic estimation solution) was 10 % higher than the one that would have been obtained without the probabilistic forwarding scheme.

In our future work we are planning to study the performance of the proposed solution over more complex network deployments, including wireless mesh networks. In addition, we will also study the possibility to exploit packets belonging to existing data flows (including the corresponding acknowledgments) to estimate link qualities, thus avoiding the additional overhead that is caused by the probe packets.

Acknowledgements

This work has been supported by the Spanish Government by its funding through the project **COSAIF**, *"Connectivity as a Service: Access for the Internet of the Future"* (TEC2012-38754-C02-01).

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[2] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 169–180, Aug. 2007. [Online]. Available: http://doi.acm.org/10.1145/1282427.1282400

[3] D. Gomez, E. Rodriguez, R. Aguero, and L. Munoz, "Reliable communications over lossy wireless channels by means of the combination of udp and random linear coding," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, June 2014, pp. 1–6.

[4] K. V. Cardoso and J. F. De Rezende, "Accurate hidden markov modeling of packet losses in indoor 802.11 networks," *IEEE Communications Letters*, vol. 13, no. 6, pp. 417–419, 2009.

[5] D. Gomez, R. Aguero, M. Garcia-Arranz, and L. Munoz, "On the use of hidden markov processes and auto-regressive filters to incorporate indoor bursty wireless channels into network simulation platforms," *Wireless Networks*, pp. 1–18, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11276-015-0909-0

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

[7] D. Gómez, E. Rodríguez, R. Agüero, and L. Muñoz, "Reliable communications over wireless mesh networks with inter and intra-flow network coding," in *Proceedings of the 2014 Workshop on ns-3*. ACM, 2014.

[8] S. Biswas and R. Morris, "Opportunistic routing in multi-hop wireless networks," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 69–74, Jan. 2004. [Online]. Available: http://doi.acm.org/10.1145/972374.972387

[9] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.