Elsevier Editorial System(tm) for Performance Evaluation
Manuscript Draft

Manuscript Number:

Title: Performance analysis of multi-hop flows in IEEE 802.11 networks: A flexible and accurate modeling framework

Article Type: Full Length Article

Keywords: Multi-hop wireless network; Multi-hop path; IEEE 802.11; Modeling Framework; Performance Evaluation.

Corresponding Author: Dr. Thomas Begin, Ph.D.

Corresponding Author's Institution: LIP UMR ENSL, UCBL, Inria, CNRS

First Author: Thomas Begin, Ph.D.

Order of Authors: Thomas Begin, Ph.D.; Bruno Baynat, PhD; Isabelle Guerin-Lassous, PhD; Thiago Abreu, PhD

Abstract: Multi-hop wireless networks are often regarded as a promising means to extend the limited coverage area offered by WLAN. However, they are usually associated with poor and uncertain performance in terms of available bandwidth and packet losses, which clearly stands as a limitation to their use. In this paper, we consider the performance evaluation of a multi-hop path (also called chain), based on the IEEE 802.11 DCF. The proposed modeling framework is constructive and versatile, so that it can handle various types of multi-hop wireless paths, including scenarios with two flows in opposite directions, and topologies where nodes are exposed to the well-known hidden node problem. The models derived from our framework are conceptually simple, easy to implement and produce generally accurate results for the attained goodput of flows, as well as the datagram loss probability. Typical relative errors for these two quantities are below a few percent. Also, fundamental phenomena occurring in multi-hop wireless networks such as performance collapse and starvation, are well captured by the models.

Dear Editor(s),

Some aspects of this work have been partially and briefly presented at the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM) in 2013 and 2014 (see [1, 2]). The paper being submitted was completely overhauled and it contains many additional and important original elements. The main differences are the following:

- We fully revised the proposed modeling framework to make it as flexible and general as possible,
- We included an additionnal scenario, which tackles the case of two opposite flows on the path,
- We thoroughly extended the numerical results by experimenting thousands of new cases that yield novel figures and histograms,
- Finally, we introduced a new section devoted to the model exploitation, which brings deep insights on the behavior of wireless multi-hop networks.

Sincerely,

Thomas Begin
Associate Professor
Université Lyon 1
France

Bruno Baynat
Associate Professor
Université Paris 6
France

Thiago Abreu
PostDoc
Université Paris 6
France

Isabelle Guérin-Lassous
Full Professor
Université Lyon 1
France

References

[1] T. Abreu, B. Baynat, T. Begin, and I. Guérin-Lassous, "Hierarchical Modeling of IEEE 802.11 Multi-hop Wireless Networks," in *ACM MSWiM*, pp. 143–150, 2013.

[2] T. Abreu, B. Baynat, T. Begin, I. Guérin-Lassous, and N. Nguyen, "Modeling of IEEE 802.11 multi-hop wireless chains with hidden nodes," in *ACM MSWiM*, pp. 159–162, 2014.

Performance analysis of multi-hop flows in IEEE 802.11 networks: A flexible and accurate modeling framework

Thomas Begin*, Bruno Baynat†, Isabelle Guérin-Lassous* and Thiago Abreu†

*Université Lyon 1 / LIP (UMR ENS Lyon - INRIA - CNRS - UCBL), France

†Université Paris 6 / LIP6 (UMR UPMC - CNRS), France

Abstract

Multi-hop wireless networks are often regarded as a promising means to extend the limited coverage area offered by WLANs. However, they are usually associated with poor and uncertain performance in terms of available bandwidth and packet losses, which clearly stands as a limitation to their use. In this paper, we consider the performance evaluation of a multi-hop path (also called chain), based on the IEEE 802.11 DCF. The proposed modeling framework is constructive and versatile, so that it can handle various types of multi-hop wireless paths, including scenarios with two flows in opposite directions, and topologies where nodes are exposed to the well-known hidden node problem. The models derived from our framework are conceptually simple, easy to implement and produce generally accurate results for the attained goodput of flows, as well as the datagram loss probability. Typical relative errors for these two quantities are below a few percent. Also, fundamental phenomena occurring in multi-hop wireless networks such as performance collapse and starvation, are well captured by the models.

Index Terms

Multi-hop wireless network; Multi-hop path; IEEE 802.11; Modeling Framework; Performance Evaluation.

I. Introduction

Wireless networks are increasingly becoming a standard way for end users to access online-services hosted on the Internet. The volume of traffic passing through these network is rapidly

growing, fuelled by the introduction of new applications and connected devices which, in turn, drives the ever-increasing demand for bandwidth. Currently, most wireless local area networks (WLANs) are based on the IEEE 802.11 DCF standard, which implements a probabilistic medium access control (MAC) layer. WLANs are overwhelmingly deployed in the infrastructure mode, where all nodes, typically close to each other, communicate through an access point, which acts as a bridge between the wireless and wired networks.

Unlike WLANs operating on the infrastructure mode, multi-hop wireless networks include wireless relay nodes, and are often regarded as a promising means to extend the limited coverage area associated with WLANs. Flows may then have for source a node that is several wireless hops away from their ultimate destination. For instance, Figure 1 depicts an example of a multi-hop wireless network with 12 nodes and 3 multi-hop flows. Multi-hop wireless networks are usually associated with poor performance (e.g., available bandwidth is scarce and packet losses due to collisions are numerous). More importantly, not only the actual performance of a multi-hop network tend to be low but they are, in general, hardly predictable. This uncertainty mostly stems from many factors, e.g., radio channel quality, antenna power, buffer size, and workload to name a few, that affect the overall behavior of multi-hop networks. This lack of forecast in the expected performance of multi-hop networks is a clear limitation to their use.

A smart and virtually costless way to improve the performance of a multi-hop wireless network can simply consist to tune some of its parameters in a better way. Examples of possible beneficial actions are many and include: resizing the size of the buffer at nodes, changing the transmitting power of nodes, shifting the nodes position (assuming they are mobile). However, the discoveries of these actions require a deep analysis that accounts for all previously cited factors.

There is a large body of literature devoted to the analytical performance evaluation of wireless networks. Accurate and sound models have been proposed for the case of a single cell (i.e. no multi-hop) [1]. Relaxing the assumption that all nodes can sense each other's transmissions, relevant analytical models exist when flows are transmitted over a single-hop path. These models provide a better understanding of some of the fundamental principles ruling these wireless networks [2]. However, in the more general case, where flows may traverse paths with multiple hops, and several flows may compete to get access to nodes transmission resources, little has been made. This shortage may be due to the apparent complexity of these networks, where many factors need to be taken into account (e.g., hidden and exposed node problem, periods

where backoff periods are frozen are erratic), which precludes the design of adequate theoretical models. We postpone to Section V the discussion on the associated literature.

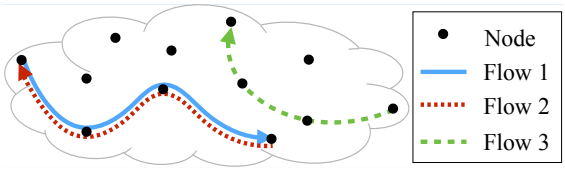


Fig. 1: Example of a multi-hop wireless network with 12 nodes and 3 multi-hop flows.

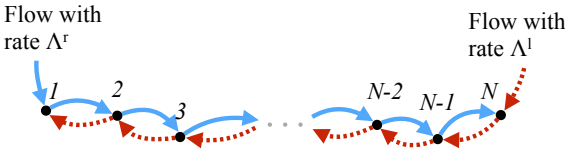


Fig. 2: Description of a multi-hop path with N nodes and 2 opposite flows.

In this paper, we focus our efforts on the performance evaluation of a single multi-hop path (also called chain), which is traversed by one or two flow(s). Though this clearly does not address the full scope of multi-hop wireless networks, we believe it can represent a significant milestone in this direction. More specifically, the contributions of this paper are threefold. First, we provide a constructive and versatile mathematical framework to derive easy-to-handle analytical models of multi-hop paths based on the IEEE 802.11 DCF. Its originality comes from its two-level abstraction: the high-level mostly seeks to mimic the buffer behavior of nodes along the path, while the low-level estimates the delay for transmitting a packet according to the IEEE 802.11 DCF, given the surrounding environment of a node. Second, we instantiate this framework on several case studies, including configurations where two flows in opposite directions travel along the path, and topologies where nodes are exposed to the well-known hidden node problem. Third, we assess the accuracy of the derived models using a discrete event simulator, and we show that these latter capture some of the fundamental phenomena occurring in multi-hop wireless networks such as performance collapse and starvation.

II. System description

A. IEEE 802.11 DCF

The IEEE 802.11 protocol specifies medium access control (MAC) and physical layer (PHY) for nodes within a WLAN. In its Distributed Coordination Function (DCF) scheme, it fetches the datagram, which is next in the waiting line for transmission (if any), and resorts to a

random access scheme, inspired by the carrier sense multiple access with collision avoidance (CSMA/CA), to access the medium. Note that, at this stage, the data link layer appends additional header and trailer to the datagram, which in turn becomes a frame. A node with a frame waiting for transmission senses the channel activity. Once the channel is detected as idle for a period of time equal to a distributed interframe space (DIFS) denoted by t_{DIFS}, the node draws a random backoff interval, which determines for how long its transmission will be postponed. This latter mechanism aims at reducing the probability of collision with frames sent by other nodes.

The backoff intervall is discretized, and its value is expressed in term of slot times. Note that the length of a slot time, denoted by t_{SlotTime}, depends on the physical layer being used (e.g., $t_{\text{SlotTime}} = 20\mu s$ for the IEEE 802.11b). The backoff timer is (i) decremented by one for each idle slot time, (ii) frozen when the node senses a transmission on the channel, and (iii) resumed if the channel is detected again as idle during a DIFS time. When the backoff timer hits 0, the node starts transmitting its frame.

The transmission of a frame and its successful reception by the destination node triggers the transmission of a positive acknowledgement (ACK), that lasts for a time t_{ACK}, after a short interframe space (SIFS) labelled by t_{SIFS} (dimensioned such that $t_{\text{SIFS}} < t_{\text{DIFS}}$, in order to refrain other nodes from starting their frame transmission in the meantime). If the sending node does not receive an ACK after a given time $t_{\text{ACKTimeout}}$, it deems the frame as lost.

The retransmissions of lost frames are managed according to binary exponential backoff rules. Note that the IEEE 802.11 DCF protocol sets a limit to the number of attempted transmissions. We denote by m this maximum number of transmission attempts. All frame transmissions are postponed by a random backoff time, uniformly drawn in the interval $[0, w]$. Here, w refers to the size of the contention window. The actual value of w depends on the number of former failed transmissions for this frame. For the first try, $w = W_1 = W_{\text{min}}$, where W_{min} designates the minimum size of the contention window. If subsequent transmissions are needed, then the contention window is gradually expanded. More precisely, for the i-th transmission of a frame ($i = 2, ..., m$), the value of w is set at follows: $w = W_i = \min(2^{i-1} \times (W_{\text{min}} + 1) - 1, W_{\text{max}})$, where W_{max} is an upper-bound on the contention window size. We report the numerical values of these constants in the case of the IEEE 802.11b protocol in Table I.

The IEEE 802.11 DCF protocol includes two additional features, namely RTS/CTS and EIFS. If a node operates in RTS/CTS mode, it sends a Request-to-Send (RTS) frame, and waits, in

t_{SlotTime}	Duration of a slot time	20μs
t_{DIFS}	Duration of distributed interframe space (DIFS)	50μs
t_{SIFS}	Duration of short interframe space (SIFS)	10μs
t_{ACK}	Transmission time of an ACK	202μs with a link of 11Mb/s
$t_{\text{ACKTimeout}}$	Time after which a frame is considered as lost if no ACK returns to the sender meanwhile	212μs
m	Maximum number of transmission attempts for a frame	7
W_{min}	Minimum size of the contention window	31
W_{max}	Maximum size of the contention window	1023
W_i	Size of contention window for the i-th transmission of a frame	
N	Number of nodes within the multi-hop path	
$d_{n,n+1}$	Distance between nodes n and $n+1$	
K_n	Buffer size (in datagrams) at node n	
s	Mean size of the frames (in bytes)	
Λ^r (resp. Λ^l)	Rate of datagrams arrivals at node 1 (resp. node N)	
$c_{n,n+1}$	Transmission speed of the link connecting nodes n and $n+1$	
$T_{n,n+1}$	Transmission time of a frame from node n to node $n+1$	
BER_n^r (resp. BER_n^l)	Probability that a bit is misinterpreted when node n transmits to its right-hand neigbor (resp. on its left-hand neighbor)	
FER_n^r (resp. FER_n^l)	Probability that a frame sent by node n to its right-hand neigbor (resp. left-hand neighbor), is received with error because of the BER	
COL_n^r (resp. COL_n^l)	Probability that a frame sent by node n to its right-hand neigbor (resp. left-hand neigbor), collides with another frame	
f_n^r (resp. f_n^l)	Probability that a frame sent by node n to its right-hand neigbor (resp. left-hand neigbor), is received with error (due to the BER or to a collision)	

TABLE I: Notation used for the multi-hop path and the associate values for IEEE 802.11b.

return, a Clear-to-Send (CTS) frame (sent by the destination node) before transmitting its "data" frame. While the use of RTS/CTS mechanism is a clear asset when hidden nodes communicate with an access point, its efficiency is questioned in the case of a multi-hop wireless network [3]. Therefore, we do not include the RTS/CTS mechanism in this paper. As for the EIFS, the standard defines an extended interframe space (EIFS) in lieu of DIFS when frame are erroneously received. Yet, studies show that several IEEE 802.11 cards do not implement this feature (e.g., [4], [5]). Therefore, even if our proposed framework could accommodate to the EIFS option, we present it in the next section without this feature.

In multi-hop wireless networks, two nodes can be within each other's transmission range, within each other's carrier sensing range, or out of range of each other. The size of these ranges mostly depends on physical factors such as the strength of the radio signal, the channel quality and the used antennas. Clearly, the position of nodes with regard to these zones has strong implications on their backoff timer decrement. Indeed, the backoff timer of a given node will be frozen each time that a node, whose transmission or carrier sensing range contains this given node, transmits.

B. Multi-hop path

As discussed in the introduction, this paper is devoted to the performance evaluation of a multi-hop path. We define a multi-hop path as a sequence of N nodes such that, for any $n = 1, ..., N-1$, nodes n and $n+1$ are within each other's transmission range, while, for any $n = 1, ..., N-2$, nodes n and $n+2$ are not. We refer to nodes $n+1$ and $n-1$ as the "right-hand" and "left-hand" neighbor nodes of node n, and define $d_{n,n+1}$ as the distance between node n and node $n+1$. In a multi-hop path, a node n can thus only directly communicate with its right-hand neighbor and with its left-hand neighbor. Conversely, at this point, we do not make any restricting assumptions about the carrier sensing ranges of nodes, i.e., we do not restrict the set of nodes which transmissions can be sensed by a given node n. The only certainty is that this set must contain nodes $n-1$ (if $n > 0$) and $n+1$ (if $n < N$).

Each node n is equipped with a single IEEE 802.11 communication interface, and its buffer size is given by K_n, which corresponds to the maximum number of datagrams being queued for transmission. A multi-hop path carries datagrams from node 1 to node N, and vice-versa. Note that, with this definition, no datagram enters or quits the path somewhere else than at the border nodes, unless it gets lost. For sake of readiness, we gather all datagrams entering at node 1 (resp. node N) into one flow, referred subsequently to as the "left-to-right flow" (resp. the "right-to-left flow"), and we let Λ^r (resp. Λ^l) be its rate of datagrams arrivals. The superscript r (resp. l) stands for "towards right" (resp. "towards left"), and it allows to discriminate precisely each flow. It follows that the workload of the multi-hop path consists of two flows in opposite directions, originating at each border node. Figure 2 illustrates the multi-hop path we consider in this paper.

The transmission speed of links depends on the radio channel quality. This mechanism is

referred to as the Auto Rate Fallback (ARF) algorithm. For the sake of easiness, we consider that the transmission speed of a link is a decreasing function of the distance of a link. We use $c_{n,n+1}$ to denote the transmission speed of the link connecting the nodes n and $n+1$. Since we naturally assume that the transmission speed only depends on the distance, we have: $c_{n,n+1} = c_{n+1,n}$. Let us denote by s the mean size of the frames (in bytes). We define $T_{n,n+1}$ as the time for node n to transmit a frame to its right-hand (resp. left-hand) neighbor, once the backoff timer has expired. Thus, it follows: $T_{n,n+1} = \frac{8s}{c_{n,n+1}} + t_{\text{SIFS}} + t_{\text{ACK}}$ (resp. $T_{n,n-1} = \frac{8s}{c_{n-1,n}} + t_{\text{SIFS}} + t_{\text{ACK}}$).

Wireless communication channels are prone to bit alterations due to multipath propagation, fading and scattering. Typically, the greater the distance between two nodes, the lower the probability that a frame arrives with no bit errors. These transmission impairments are taken into account by the Bit Error Rate (BER), which expresses the probability that a bit is misinterpreted at a receiver node due to the propagation process. Since the distance between the successive nodes of a multi-hop path is not necessarily constant, we let BER_n^r (resp. BER_n^l) be the value of the BER when node n transmits to its right-hand neighbor node $n+1$ (resp. left-hand neighbor node $n-1$). The data we used for the BER computation are reported in Section IV. Based on the value of the BER_n^r, and on the mean size of the frames (in bytes), we can then derive the frame error rate FER_n^r, which stands for the probability that a frame transmitted by node n to its right-hand neighbor node $n+1$, is received with error (due to the BER). This frame error rate reads as:

$$\text{FER}_n^r = 1 - (1 - \text{BER}_n^r)^{8s}. \tag{1}$$

Clearly, the same approach applies also for the computation of FER_n^l.

In addition to the BER, transmitted frame can also be received with errors due to collisions. A collision occurs when "nearby" nodes are active on the channel simultaneously. The well-known hidden node problem causes collisions, and, in general, it occurs for multi-hop paths, unless all nodes from 1 to N are within the same carrier sensing range. Unlike the frame error rate FER, the odds of collisions highly depend on the workload level. Typically, if every node of the path has only very few datagrams to transmit, then the probability of collisions for the corresponding frames is expected to be low. We denote by COL_n^r (resp. COL_n^l) the probability that a frame sent by node n to its right-hand neighbor (resp. its left-hand neighbor), collides with another frame, and thus gets lost. Note that there is, in general, no simple closed-form expression to

assess the value of COL_n^r because of its complexity.

In order to ease the subsequent computations, we introduce a single parameter denoted as f_n^r (resp. f_n^l), and referred to as the "frame loss probability", which stands for the probability that a frame transmitted by node n to its right-hand neighbor (resp. its left-hand neighbor), is received with error, regardless of the cause. As detailed below, a frame can be lost because of the BER or because of collisions. The frame loss probability f_n^r can thus be calculated as the probability that a frame is lost because of the BER, i.e., FER_n^r, plus the probability that the frame is lost because of a collision, i.e., COL_n^r, minus the probability of both events occurring simultaneously. If we further make the assumption that BER and collisions are independent events, which sounds like a reasonable assumption, we can estimate the frame loss probability as:

$$f_n^r = \text{FER}_n^r + \text{COL}_n^r - \text{FER}_n^r \times \text{COL}_n^r. \tag{2}$$

Table I summarizes the detailed notations used in this section for the description of multi-hop path.

C. Behavior and performance

In a multi-hop path, a given node n can transmit to a set of nodes reduced to its right-hand neighbor and to its left-hand neighbor, but node n can sense the transmission of a larger set of nodes. All nodes that belong to this last set have transmissions that are mutually exclusive with those of node n (unless they start exactly at the same time) and that, in addition, can freeze the backoff decrement of node n. This implies a very strong synchronization between nodes, that must be taken into account in the model. In addition to this phenomenon, nodes that do not belong to this set can transmit frames that can collide with frames transmitted by node n. This is the well-known hidden problem, that also introduces a strong dependency between the individual behaviors of each node (through the calculation of the collision probability).

Another factor that has a strong implication on the development of realistic models, is that the workload is often non-uniformly distributed among the nodes of the path. Indeed, because links between nodes are exposed to different conditions (e.g., BER and collisions), the time necessary to successfully send a datagram can considerably vary from one node to another (and also from one direction to the other). As a result, the level of workload at each node, corresponding to the number of datagrams waiting for transmission, can be very disparate among the nodes of

the path. Some nodes may have their buffer fully filled of queued datagrams, while in the same time, others (prior or subsequent in the path) may be in starvation (i.e., idle buffers).

Losses should also be carefully considered when dealing with modeling. First it is important to clearly make the difference between frame losses and datagram losses. As detailed in the previous subsection, a frame can be lost either because of the BER or because of collisions. A lost frame will be retransmitted (with an increased contention window size), unless the maximum number of allowed retransmissions is reached. The collision probability is much more difficult to estimate than the frame error probability (due to BER), and represents a crucial step in the modeling process. Let us now consider datagram losses. A datagram can be lost because of two different factors: 1) if the corresponding frame is transmitted to a receiver when its buffer is full, or 2) if all successive retransmissions of the frames corresponding to the same datagram are lost. As discussed above, the first case is very likely to happen, because of the high non-uniformity of the load in the path. On the opposite, the second case is very unlikely to happen, as the probability of having m consecutive frame losses is typically very low (remember that m is the maximum number of transmission attempts allowed by the protocol for a same frame)[1].

We conclude this section by precisely defining all the key performance parameters we want to obtain for our multi-hop path. For sake of readiness, we only give the definition of the performance parameters pertaining to the flow travelling from node 1 to node N (i.e., the left-to-right flow). We denote by G^r, the goodput of this flow, corresponding to the mean number of datagrams that reach destination node N by unit of time, and denote by L^r the probability that a datagram gets lost in its path from node 1 to node N (which, as discussed earlier, is mainly due to buffer overflow). Obviously, these parameters vary with the rates of the workload Λ^r and Λ^l, with the position of nodes $d_{n,n+1}$, as well as with virtually all the other parameters. In the next section, we show how we can evaluate all of these performance parameters (for both flows), namely, G^r, G^l, L^r and L^l, based solely on the parameters of the used version of IEEE 802.11, namely t_{SIFS}, t_{DIFS}, t_{ACK}, t_{SlotTime}, W_{min}, W_{max} and m, as well as those coming from the multi-hop path, namely K_n, $d_{n,n+1}$, $c_{n,n+1}$, s, Λ^r, and Λ^l.

[1]Assuming that at each transmission attempt, a frame gets lost with constant and independent probability q, the probability of loosing a datagram by that means is: q^m. For $m = 7$ and $q = 0.6$, it accounts for only a couple of percents.

III. Modeling Framework

We have developed a general modeling framework for the performance evaluation of a multi-hop path in a wireless network. Our model is made up of two levels: 1) a unique high-level queueing model and 2) several low-level Markov chain models. At the high-level, we associate one queue with each transmitting node of the wireless network. The high-level queueing model mostly aims at capturing the workload level and the buffer overflows at intermediate nodes. As for the low-level, we use multiple continuous-time Markov chains, each one being associated to a given node and to a given flow. Each Markov chain precisely describes the transmission delay of a datagram according to the IEEE 802.11 DCF protocol. As detailed in the two next subsections, if the input parameters of the high-level model are supposed to be known, its output performance parameters allow to parameterize the low-level Markov chain models. Conversely, once the low-level Markov chain models are parameterized, they provide an estimation of the missing input parameters for the high-level model. As a result, our overall model can be solved using the fixed-point iterative procedure presented in the last subsection.

A. High-level modeling

The studied wireless network is made up of N nodes. Each node n has to transmit datagrams to its right-hand neighbor node $n+1$ (except node N) for the left-to-right flow, and to its left-hand neighbor node $n-1$ (except node 1) for the right-to-left flow. At a high-level, we associate with each node n of the wireless network, a queue with a single server and a finite capacity. Hence, customers of the latter queueing network model represent the datagrams of the wireless network. An arriving customer in a queue n "in the middle" of the path, i.e., for any $n = 2, ..., N-1$, corresponds to a datagram successfully transmitted by node $n-1$ to node n (along the left-to-right flow), or by node $n+1$ to node n (along the right-to-left flow). As for the border nodes, even though they deal with datagrams from both flows, their queues are only fed by datagrams belonging to one of the two external workload since the others have already reached their final destination and thus do not need to be queued for a further transmission. The waiting room of queue n matches the buffer size of transmitting node n, i.e., K_n, as defined in Section II-B. Because of this finite buffer, an arriving customer can be rejected, whether it belongs to the left-to-right flow or to the right-to-left flow, resulting in datagram loss for the corresponding flow. As datagrams from the two flows compete for the same MAC access, the

server of queue n models the transmission of a datagram in the wireless network, either to node $n+1$ (unless $n = N$) or to node $n-1$ (unless $n = 1$). Note that the service time of each server corresponds to the total time node n needs to transmit a datagram that is ready to be sent over the radio channel, and includes all successive frame (re)transmissions (associated to the considered datagram), as well as all IEEE 802.11 DCF protocol delays (DIFS, backoff, SIFS, timeout) and all freezing times due to other nodes transmissions during the backoff of node n.

For the sake of computational simplicity, we assume that the service times are exponentially distributed, and that the arrival process at each queue follows a Poisson process. It follows that the high-level model comprises of a set of N *M/M/1/K* queues as illustrated on Figure 3. We denote by μ_n the service rate of queue n, whose inverse value equals the mean service time S_n to transmit a datagram, and we denote by λ_n the workload of queue n, i.e., the arrival rate of customers at the entrance of the queue.

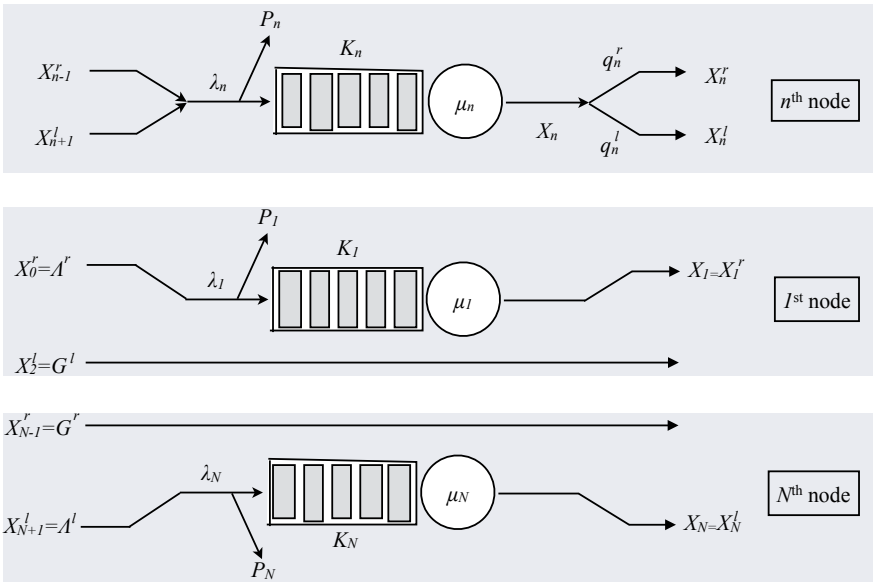


Fig. 3: High-level modeling of a given node n.

Assuming that λ_n and μ_n are known for all $n = 1, ..., N$, then all performance parameters of our high-level queueing model can be obtained from the well-known results of the *M/M/1/K* queue. The output throughput of queue n, i.e., X_n, can be derived from the probability $\pi_n(i)$ of having i customers in the n-th *M/M/1/K* queue, as:

$$X_n = \mu_n(1 - \pi_n(0)) \tag{3}$$

As noted in Section II-C, the probability of loosing a datagram because the maximum number of unsuccessful retransmissions of the corresponding frame has been reached, is very low. We will thus neglect this possibility and consider that a datagram can only be lost because, upon its arrival on the next node, it finds the buffer full. As a result, X_n can be considered as the average number of datagrams that are correctly transmitted by unit of time from node n either to node $n+1$ (if $n \leq N-1$) or to node $n-1$ (if $n \geq 2$).

The utilization of queue n, i.e., U_n, corresponding to the proportion of time node n is not empty, is:

$$U_n = 1 - \pi_n(0). \tag{4}$$

And finally, from PASTA theorem [6], we can obtain the buffer overflow probability, i.e., P_n, corresponding to the probability that a datagram is lost because the buffer of node n is found full upon its arrival instant:

$$P_n = \pi_n(K_n). \tag{5}$$

In order to estimate the missing parameters λ_n and μ_n, let us first consider a node n "in the middle" of the path (meaning not in the first or in the last position), i.e., for $n = 2, ..., N-1$. Remember that transmissions of node n can correspond either to datagrams that are destined to $n+1$ in the left-to-right flow, or to datagrams that are destined to node $n-1$ in the right-to-left flow. Let us define q_n^r the proportion of datagrams sent by node n that move to the right (and that are thus destined to node $n+1$) and q_n^l the proportion of datagrams sent by node n that move to the left (and that are thus destined to node $n-1$). We can thus split the output throughput of queue n into two opposite flows, one to the right and one to the left, as illustrated in Figure 3:

$$X_n = X_n^r + X_n^l \tag{6}$$

where

$$X_n^r = X_n.q_n^r \text{ and } X_n^l = X_n.q_n^l. \tag{7}$$

Obviously, we have: $q_n^r + q_n^l = 1$. Reminding that the flow of customers (viz. datagrams) arriving at the entrance of queue n is the superimposition of the left-to-right flow leaving queue $n-1$, and of the right-to-left flow leaving queue $n+1$, we derive the value of the workload λ_n as:

$$\lambda_n = X_{n-1}^r + X_{n+1}^l \tag{8}$$

as well as the proportion of customers moving to the right and those moving to the left:

$$q_n^r = \frac{X_{n-1}^r}{X_{n-1}^r + X_{n+1}^l} \text{ and } q_n^l = \frac{X_{n+1}^l}{X_{n-1}^r + X_{n+1}^l}. \tag{9}$$

These last relations are true because the PASTA property ensures that the proportion of lost datagrams is the same for the right-to-left flow and for the left-to-right flow (the Poisson assumption ensures that, despite having different rates, both flows undergo equal probability to find the buffer of queue n full).

When it comes to the special case of node 1 (resp. node N), as it is the first (resp. last) node of the left-to-right (resp. right-to-left) flow and the last (resp. first) node of the right-to-left (resp. left-to-right) flow of the wireless network, node 1 (resp. node N) only has to transmit datagrams to node 2 (resp. node $N-1$). As a result, its queue is only fed by the external workload process with rate Λ^r (resp. Λ^l). This is illustrated by Figure 3. It follows that for node 1, we have: $X_1 = X_1^r$, $q_1^r = 1$, $q_1^l = 0$ and $\lambda_1 = \Lambda^r$. Similarly, for node N, we have: $X_N = X_N^l$, $q_N^l = 1$, $q_N^r = 0$ and $\lambda_N = \Lambda^l$.

At this stage we turn to the single parameter that is left to determine within the high-level queueing model, i.e., the service rates μ_n. To begin with, we consider a value of n such that $n = 2, ..., N-1$. We use S_n to denote the mean time node n needs to transmit a datagram. S_n is just the inverse of the service rate μ_n, and can be expressed as a weighted sum of two components, each of which accounting for one of the two possible next hops for the transmitted datagrams:

$$S_n = \frac{1}{\mu_n} = q_n^r.S_n^r + q_n^l.S_n^l \tag{10}$$

where S_n^r (resp. S_n^l) is the mean service time to handle a datagram headed to its right-hand (resp. left-hand) neighbor. As stated before, S_n^r and S_n^l are parameters that depend on many factors and it will be the objective of the low-level Markov chain models to provide an estimation of these parameters. Note that for the border nodes, i.e., $n = 1$ or $n = N$, the computation of S_n still obeys to relation (10), even though one component is void.

If we assume that these missing mean service times are correctly estimated, then all the input parameters of the high-level model are now known. We can then solve it and derive from its performance parameters (relations (3)-(7)), all the key performance parameters of our multi-hop path network defined in Section II-C. First, the goodput of the left-to-right (resp. right-to-left)

flow corresponds to the output throughput of queue $N-1$ (resp. queue 2) toward the right (resp. toward the left):

$$G^r = X^r_{N-1} \text{ (resp. } G^l = X^l_2). \tag{11}$$

Then, the datagram loss probability along the left-to-right (resp. right-to-left) flow, is the probability that a datagram is lost by buffer overflow in any of the first (resp. last) $N-1$ queues:

$$L^r = 1 - \prod_{n=1}^{N-1}(1-P_n) \text{ (resp. } L^l = 1 - \prod_{n=2}^{N}(1-P_n)). \tag{12}$$

Overall, we have shown throughout this subsection that if the service rates μ_n of all queues of the high-level model were to be known, then we can easily derive all the key performance parameters of our wireless network.

B. Low-level modeling

In order to estimate the missing parameters of the high-level queueing model, i.e., the mean service times S^r_n and S^l_n involved in the derivation of the service rates μ_n of all *M/M/1/K* queues (see relation (10)), we associate with each queue and with each direction (left-to-right or right-to-left) a Continuous-Time Markov Chain (CTMC) describing precisely the transmission process of a node according to the IEEE 802.11 DCF protocol. Therefore, any node n "in the middle" of the path, i.e., n such that $n = 2, ..., N-1$, results in two CTMCs, one describing the transmission time of a datagram from node n to node $n+1$ (along the left-to-right flow), and the other one describing the transmission time of a datagram from node n to node $n-1$ (along the right-to-left flow). Obviously, since buffers of transmission for nodes 1 and N are only fed by a flow, we associate a single CTMC with each of these border nodes.

For the sake of readiness, we restrain our analysis in the remainder of this subsection to the CTMC that accounts for a queue n of the high-level model and for the left-to-right direction. However, the same analysis can be carried out for the case of the right-to-left direction. All parameters of the chain will then be possibly indexed with a subscript n corresponding to the considered node and with a superscript r corresponding to the considered direction. The goal of this CTMC is to provide an estimation of the mean transmission time S^r_n. To do this, the CTMC has to precisely describe the succession of the different states node n has to go through in order to transmit a datagram to node $n+1$ over the wireless channel. The CTMC is depicted

in Figure 4 and overall, consists of m "lines", each of which corresponding to the backoff time interval preceding the k-th transmission of a given datagram (this implies that the $k-1$ preceding transmissions of the datagram were in error). Let us remind that m denotes the maximum number of transmission attempts (see Section II-A).

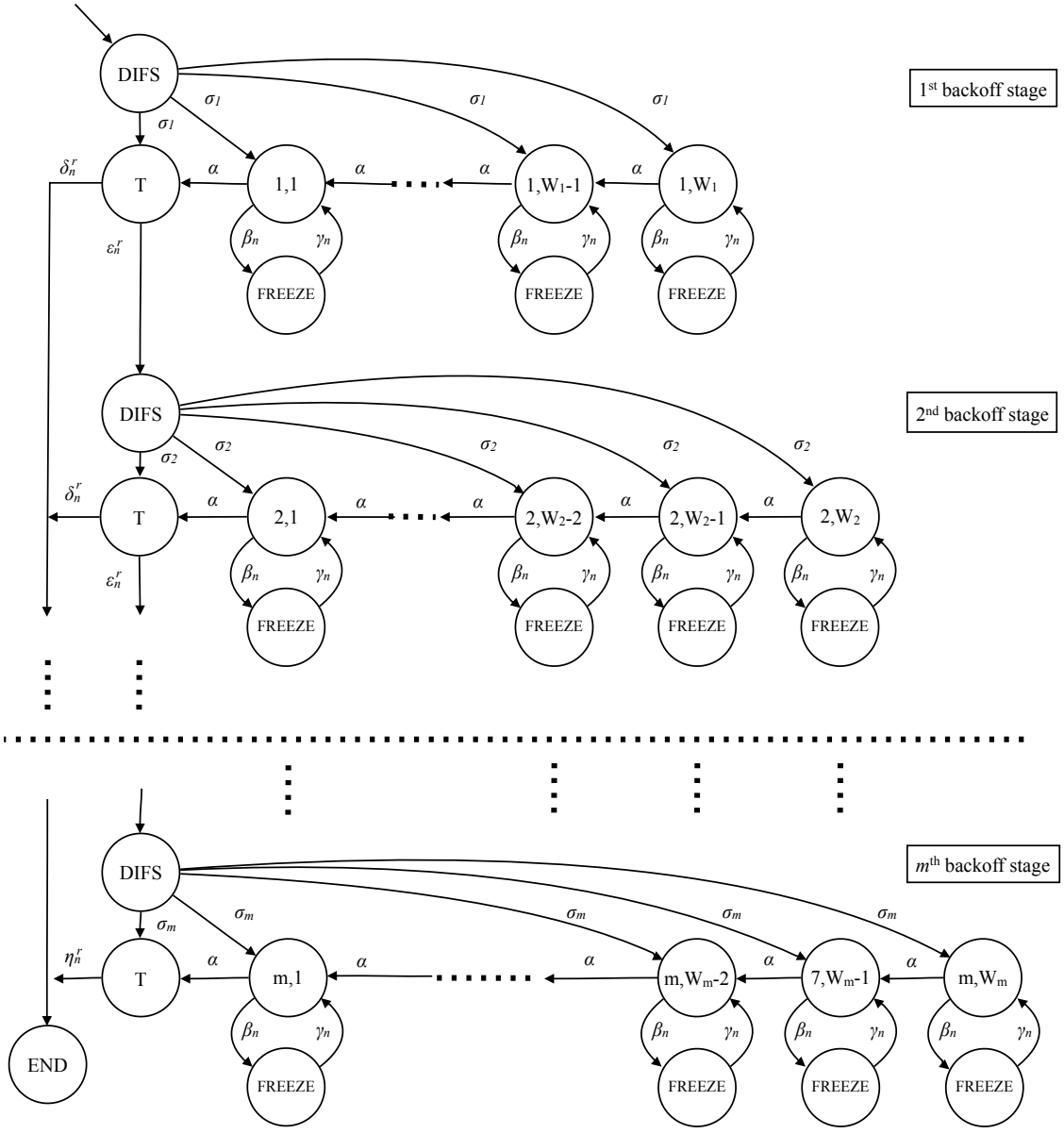


Fig. 4: Low-level modeling of the datagram transmission time at node n towards its right-hand neighbor, namely S_n^r.

As explained in Section II-A, any transmission attempt starts with a DIFS time and then enters

a backoff procedure, that can be interrupted anytime the node senses the channel as being busy, and ends with a transmission time, that can be either successful and thus leading to the END state with a rate δ_n^r, or in error bringing the process to the next stage of the backoff with a rate ϵ_n^r. Any state $\{k, j\}$ refers to the k-th stage of the backoff ($k \in [1, m]$) (i.e., the k-th transmission attempt of the datagram) with a current contention window size equal to j ($j \in [1, W_k]$). Exiting the k-th *DIFS* state, the process is uniformly routed towards any state $\{k, j\}$ or the transmission state with a rate σ_k. From any state $\{k, j\}$, it can either reach state $\{k, j-1\}$ (or transmission state if $j = 1$) if the channel was sensed idle during a complete slot time, or reach a FREEZE state if the channel has been sensed busy. We denote the corresponding rates by α and β_n, respectively. The rate out of any FREEZE state is γ_n bringing back the process to the state $\{k, j\}$ where it came from before entering the FREEZE state.

Before describing the expression of all the rates appearing in the CTMC and seing how they can be estimated, let us first assume that the values of all of these rates are known. It is then straightforward to compute the mean transmission time S_n^r, i.e., the time needed to go from the first DIFS state up to the END state of the CTMC:

$$S_n^r = t_{1,n}^r + f_n^r \times (t_{2,n}^r + f_n^r \times (t_{3,n}^r + ... + f_n^r \times t_{m,n}^r)) \tag{13}$$

where $f_n^r = \frac{\epsilon_n^r}{\epsilon_n^r + \delta_n^r}$ is the loss frame probability (see rel. (2) in Section II-B), and $t_{k,n}^r$ corresponds to the mean time spent by the process in the k-th "line" of the CTMC. The main approximation here is that, at each transmission attempt, and regardless of the number of retransmissions experienced, each frame gets lost with constant and independent probability. By construction of the CTMC, we get:

$$t_{k,n}^r = t_{\text{DIFS}} + \frac{W_k}{2}.r_n + T_{n,n+1} \tag{14}$$

where $T_{n,n+1}$ is the mean transmission time of a frame between node n and node $n+1$ (see Section II-B), and r_n is the average time spent in any pair of loop states ($\{k, j\}$, FREEZE) at node n. We can express r_n as:

$$r_n = \frac{1}{\alpha}.\left(1 + \frac{\beta_n}{\gamma_n}\right). \tag{15}$$

Before going any further, it is very important to point out that one important parameter involved in the derivation of the CTMC is the frame loss probability f_n^r. As detailed in Section II-B, this probability is related to the frame error rate FER_n^r and to the collision probability COL_n^r (see relation (2)).

As discussed above, the CTMC involves several rates that need to be correctly set. Some of them, namely σ_i and α, are constant, and they depend neither on node n, nor on direction r. Other rates, namely δ_n^r, ϵ_n^r and η_n^r, depend both on node n and direction r. Finally, the remaining rates, namely β_n and γ_n, only depend on node n (they do not depend on direction r).

The first set of parameters comprises constant rates, which are very simple to estimate from the IEEE 802.11 specifications. σ_i is just the product of the inverse of the time spent in state DIFS by a uniform probability deriving from the current size of the contention window:

$$\sigma_i = \frac{1}{t_{\text{DIFS}}.(W_i + 1)}, \ i = 1, ..., m. \tag{16}$$

α is simply the inverse of a slot time duration:

$$\alpha = \frac{1}{t_{\text{SlotTime}}}. \tag{17}$$

The second set of parameters consists of rates leaving out of states T, which correspond to frame transmissions. For the first $m-1$ "lines" of the CTMC, ϵ_n^r is the rate from T to the next DIFS state, corresponding to an unsuccessful frame transmission. It can thus be expressed as:

$$\epsilon_n^r = \frac{f_n^r}{T_{n,n+1}} \tag{18}$$

where $T_{n,n+1}$ is mean transmission time of a frame between node n and node $n+1$, and f_n^r is the frame loss probability (see Section II-B). Similarly, δ_n^r is the rate from T to the END state, corresponding to a successful frame transmission, and can thus be expressed as:

$$\delta_n^r = \frac{1 - f_n^r}{T_{n,n+1}}. \tag{19}$$

Finally, η_n^r is the rate leaving out of the last T state, corresponding to the last frame transmission attempt, undoubtfully leading to the END state (whether the transmission is a success of a failure). It follows that:

$$\eta_n^r = \epsilon_n^r + \delta_n^r. \tag{20}$$

The third set of parameters is made up of two types of rates, i.e., β_n and γ_n, whose estimation is, in general, more difficult. At this stage, we do not include the corresponding generic formulas for these two parameters and just give some intuition on how to derive them. We present in the next section, how to instantiate these formulas when considering specific configurations of our wireless network.

The inverse of β_n has an easier interpretation than β_n itself. It corresponds to the mean time between two successive backoff freezings of node n, conditioned by the fact that node n is in backoff. This quantity is naturally related to the mean backoff duration of node n before any frame transmission attempt, and to the mean number of times backoff of node n is freezed by transmissions of other nodes. The first parameter (mean backoff duration) can easily be related to the frame loss probability, whereas the second one (the mean number of freezings) is much more complicated to estimate, as it depends on transmissions of all other nodes that are within the carrier sensing range of node n.

Again, the inverse of γ_n is easier to be interpreted, as it corresponds to the mean freezing duration of the backoff of node n. For many cases this quantity merely corresponds to the mean frame transmission time of neighbor nodes (including the returning time of the acknowledgment), and thus is easy to be estimated. However, in more complicated scenarios, frame transmissions of two nodes that are not in the carrier sensing range of each other can overlap, resulting in longer freezing durations.

As a conclusion of this subsection, let us underline that among all the parameters of the two CMTCs associated to node n (in both directions), only three are not easy to estimate since they do not derive straightforwardly from the IEEE 802.11 specifications and systems characteristics. These are the frame collision probability COL_n^r (and similarly COL_n^l), the rate β_n and the rate γ_n. In Section IV, we show how to estimate them in specific configurations.

C. Fixed-point solution

As stated in the introduction of this section, the solution to the high-level queueing model requires input values for μ_n, which directly derives from the solution of the low-level CTMC. Conversely, the low-level CTMC models have input parameters, whose values can be derived from the performance parameters of the high-level queueing model, namely X_n, D_n, U_n and P_n. As a result, the overall model can be solved using the fixed-point iterative procedure described by Algortihm 1. The stopping criterion is triggered whenever the deviation for the estimated values of parameters μ_n between two successive iterations is less than a given threshold. Once the algorithm has converged, it returns the performance parameters of interest for the network understudy, namely G^r, G^l, L^r and L^l.

With the exception of steps 5, 6 and 7, whose instantiation depend on the specific network

Algorithm 1 Fixed-point solution for the overall model

initialize μ_n and λ_n for all $n = 1, ..., N$ with non-absurd values
repeat
  solve the high-level queueing model and obtain X_n, D_n, U_n and P_n for all $n = 1, ..., N$ (rel. (3)-(5))
  calculate q_n^r and q_n^l for all $n = 1, ..., N$ (rel. (9))
  estimate COL_n^r and COL_n^l for all $n = 1, ..., N$
  estimate β_n for all $n = 1, ..., N$
  estimate γ_n for all $n = 1, ..., N$
  for $n = 1$ solve the low-level "right" CTMC associated with node 1 and calculate S_1^r (rel. (13))
  for $n = 2, ..., N-1$ solve the 2 low-level CTMCs associated with node n and calculate S_n^r and S_n^l (rel. (13))
  for $n = N$ solve the low-level "left" CTMC associated with node N and calculate S_N^l (rel. (13))
  update μ_n for all $n = 1, ..., N$ (rel. (10))
  update λ_n for all $n = 1, ..., N$ (rel. (8))
until convergence of the algorithm
return G^r, G^l, L^r and L^l (rel. (11)-(12))

under consideration, every other step of this algorithm has been described in details in this section and can be applied for any given path.

IV. Case studies

In order to highlight the versatility of our modeling framework as well as its accuracy, we consider 3 case studies of multi-hop wireless paths, which differ by their number of nodes and of flows. For each of these case studies, we detail the modeling finish that is specific to the considered case study, and we assess the models accuracy for the key performance parameters of a path. This section ends with two examples for the model exploitation, which provide, at a very low cost of computation, insights in the behavior of wireless multi-hop paths.

Throughout this section, we use the standard parameter values of IEEE 802.11b as reported in Table I. To ease the readiness of this section, the links have a constant capacity (physical rate) of 11 Mb/s (i.e., no Rate Adaptation) corresponding to the modulation CCK 11. We consider that all datagrams have a constant size of 1500 bytes. Hence, the transmission time for a frame (with its associate ACK) is identical for all nodes, and we denote it simply by T. The received signal power at each node is computed using a transmission power of 31.6 mW, and an antenna gain of 1 dBi. These latter values match the specifications of the ORiNOCO11b card [7]. Using

the two-ray ground reflection model, it follows that the ranges of communication and carrier sensing cover 399 and 700 meters, respectively.

To account for the non-perfect physical layer, and the fact that transmission errors are more likely when the distance between two transmitting nodes is large, each link experiences a Bit Error Rate (BER), whose actual value depends of the distance of the link. The derivation of the BER is based on a relation between the received signal power and the used modulation. For more details on this computation, the reader can refer to [8]. Figure 5 shows the evolution of BER and that of the corresponding FER (Frame Error Rate) for datagrams of 1500 bytes as a function of the distance between two communicating nodes. Remind that the FER derives directly from the BER and the length of frames (see rel. (1)). In our examples, the BER rises from $4e^{-9}$ to $8e^{-5}$ as the distance between two communicating nodes widens from 150 to 399 meters. In the latter case, the FER reaches values close to 60%.

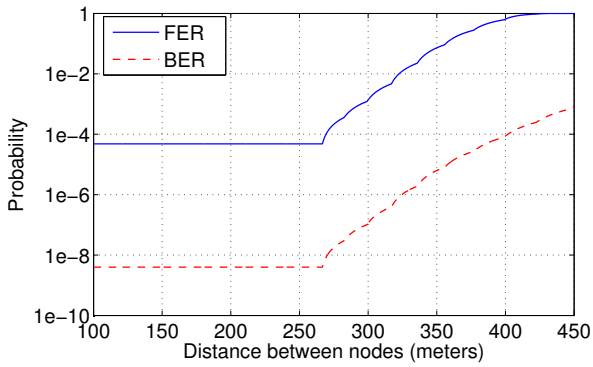


Fig. 5: Evolution of BER and FER as a function of the distance for datagrams of 1500 bytes.

To evaluate the accuracy of our model, we compare its performance parameters obtained from an implementation in MATLAB with those delivered by a discrete-event simulator (Network Simulator version 2.35 - *ns-2.35* [9]). All simulation results have been performed by generating 100,000 packets at the source node(s). Throughout this section, we define the absolute error of our model versus the actual values (delivered by *ns-2.35*) as $|$*model - actual*$|$. The percentage relative error is determined as the ratio $100 \times |$ *model - actual* $|$ */ actual*.

A. *"Basic" case study*

We start considering a multi-hop path with N=3 nodes that conveys a single flow travelling from node 1 to node 3 through node 2. We remind that this left-to-right flow of datagrams constitutes the workload for the path, and that the datagrams generation at node 1 follows a Poisson process with a rate Λ^r. The positions of the nodes, which determine their carrier sensing and transmission ranges, are such that any node can communicate only with its 1-hop neighbor(s), yet it can sense transmissions from the two other nodes.

Since only two of the three nodes are effectively transmitting frames (node 3 is solely returning ACK), the high-level model comprises two queues. And as all frames belong to the same flow, originating at node 1 and ending at node 3, many quantities, related to the possible existence of a second flow such as X_n^l, q_n^l, S_n^l, are void here. Hence, relations (6), (8), (10), (11) and (12) are instantiated as follows:

$$\begin{aligned} X_n &= X_n^r, & G^r &= X_2, \\ \lambda_n &= X_{n-1}^r, & L^r &= 1 - \textstyle\prod_{n=1}^{2}(1-P_n), \\ S_n &= \tfrac{1}{\mu_n} = S_n^r. \end{aligned}$$

As said at the end of Section III-B, three parameters of our modeling framework were left without formula since they strongly depend on the actual network configuration. We now provide ad hoc formulas for the context of this case study.

COL_n^r, which represents the probability that a frame sent by node n to its right-hand neighbor collides with another frame, is virtually void here since any node can sense each other's transmission. Relation (2) becomes simply: $f_n^r = \text{FER}_n^r$. By doing this, we overlook the possibility that two nodes start their transmission exactly at the same time. The corresponding probability has been found to be very low [10].

The derivation of γ_n is also straightforward. Having in mind that $1/\gamma_n$ corresponds to the mean freezing duration, i.e., the frame transmission time of another node plus a DIFS time, we have: $1/\gamma_n = T + t_{\text{DIFS}}$.

Finally, let us consider the case of β_n. Here, we resort on the fact that $1/\beta_n$ is equal to the mean time between two successive backoff freezing, given that node n is in backoff. It follows that: $1/\beta_n = \overline{B}_n/\overline{np}_n$ where $\overline{B}_n$ denotes the mean backoff duration for a frame of node n, and $\overline{np}_n$ indicates the average number of freezing of the backoff of node n.

The computation of $\overline{B}_n$ is based on [11]:

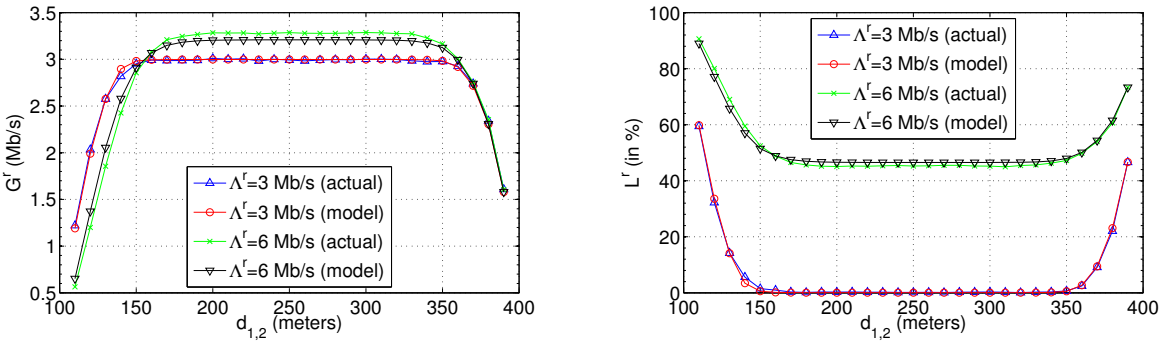
$$\overline{B}_n = \frac{W_1/2.\phi_{1,n} + (W_1+W_2)/2.\phi_{2,n} + \cdots + (W_1+W_2+\cdots+W_m)/2.\phi_{m,n}}{\Phi_n}.t_{\text{SlotTime}}$$

where W_i is the size of contention window for the i-th transmission of a frame, $\phi_{k,n}$ denotes the probability that the transmission of a datagram at node n requires exactly k frames, and Φ_n is the mean number of frame transmissions required per datagram at node n. Clearly, $\phi_{k,n}$ and Φ_n can be derived from the frame loss probability f_n^r as follows: $\phi_{k,n} = (f_n^r)^{k-1}.(1-f_n^r)$ for $k \leq m-1$, $\phi_{m,n} = (f_n^r)^{m-1}$ and $\Phi_n = \sum_{k=1}^{m} k.\phi_{k,n}$. Let us remind that m denotes the maximum number of transmission attempts.

As for $\overline{np}_n$, let us remind that the backoff of a given node n is paused whenever any node j in the carrier sense range of n makes a transmission. We designate by F_n the frame throughput of node n, which corresponds to its number of frame transmissions by unit of time. Clearly, we have: $F_n = X_n.\Phi_n$ (remind that X_n denotes the datagram throughput of node n). Assuming node n has a datagram to transmit, we have: $\overline{np}_n = \frac{F_j}{F_n}$ with the convention that $j = 2$ if $n = 1$, and $j = 1$ if $n = 2$. Hence, $\overline{np}_n = \frac{X_j.\Phi_j}{X_n.\Phi_n}$. In the more general case, where node n is not necessarily saturated, we refine this relation with a corrective factor η_n such that: $\overline{np}_n = \frac{X_j.\Phi_j}{X_n.\Phi_n}.\eta_n$. This corrective factor amounts to the ratio of time node n is non-idle. This quantity is related to the node utilization U_n and it reads as: $\eta_n = \frac{S_n - T}{S_n \frac{1-U_n}{U_n} + S_n - T}$. For more details, the reader can refer to [11].

We now study the ability of our model to provide fair predictions for the key performance parameters of a 3-nodes path with a single flow. Figure 6 illustrates the accuracy of the proposed model for two different levels of workload Λ^r, as well as for a large range of distances between node 1 and relay node 2. In these examples, the distance between nodes 1 and 3 is kept constant to 500 meters, and the buffers at nodes are of length $K_n = 20$. As shown by Figures 6a and 6b, the values delivered by our model are very close to the actual values of goodput G^r and of loss probability L^r.

To give a more comprehensive view on the accuracy of our model, we consider hundreds of examples with different values of buffer sizes, workload levels and nodes positions. For each scenario we compute the error committed by our model. We report in Table II the overall distribution of relative errors for the goodput. The mean error is around 4%, and it virtually does not exceed 10%. Similarly, Table III indicates the distribution of the absolute errors for the



(a) Goodput.

(b) Loss probability.

Fig. 6: Accuracy of the proposed model for the case of a path with 3 nodes and 1 flow of rate Λ^r for many positions of the relay node.

datagram loss probability. In average, the error is less than 1%, and only 6% of the hundreds examples led to an error greater than 3%, while none of them exhibit an error exceeding 4%.

TABLE II: Overall accuracy on the goodput for a path with 3 nodes and 1 flow.

Average	<5%	5-10%	10-15%	> 15%
4.12%	93.82%	5.95%	0.23%	0.0

TABLE III: Overall accuracy on the loss probability or a path with 3 nodes and 1 flow.

Average	<1%	1-2%	2-3%	3-4%	>4%
0.85%	60.0%	10.0%	24.0%	6.0%	0.0%

B. Extension to account for 2 opposite flows

Our second case study extends the former scenario since we consider an additional flow, originated in node 3. Thus, the workload consists of 2 independent flows with rate Λ^r and Λ^l, respectively. The remaining assumptions are the same as before, which state, among other things, that the 3 nodes can sense each other's transmissions.

From the standpoint of our model, this case study presents two additional complexities. First, the freezing of the backoff of a node may be caused by the activity of the two other nodes. Second, node 2 handles datagrams from two distinct flows, so that the corresponding frames will experience different channel conditions for their next hop transmission.

As the 3 nodes are effectively transmitting datagrams, the high-level modeling has 3 queues. Regarding the low-level modeling, the values of the parameters COL_n^r and COL_n^l are void, while γ_n is set to $1/(T + t_{\text{DIFS}})$ (see Section IV-A for more explanation). The only parameter left to determine is β_n. Its computation is performed in a very analogous way to the former case study with a single flow. Indeed, we resort again on the formula $1/\beta_n = \overline{B}_n/\overline{np}_n$ where $\overline{B}_n$ denotes the mean backoff duration for a frame of node n, and $\overline{np}_n$ indicates the average number of freezing of the backoff of node n.

Since node 2 transmits datagrams belonging to both flows, we adapt the computation of $\overline{B}_2$ by simply splitting it in 2 components: $\overline{B}_2 = q_2^r.\overline{B}_2^r + q_2^l.\overline{B}_2^l$ where $\overline{B}_2^r$ and $\overline{B}_2^l$ are the mean backoff durations over the links of distance of $d_{1,2}$ and $d_{2,3}$, respectively. Note that, since the BER does only depend on the distance, $\overline{B}_2^r = \overline{B}_3$ and $\overline{B}_2^l = \overline{B}_1$. The quantities $\overline{B}_1$ and $\overline{B}_3$ are easily obtained using the formula given in the previous section.

For $\overline{np}_n$, we simply need to extend the formula related in the former section since each node has 2 neighbor nodes, which can interrupt its backoff decrement. We have: $\overline{np}_n = \sum_{j \neq n} F_j/F_n.\eta_n$. Let us remind that the values of F_n are easily obtained as they simply denote the frames throughput of node n and derive directly from the datagram throughput X_n. The computation of the non-idle probability for node n η_n remains identical to the description given in Section IV-A.

We study the accuracy of the proposed model for a spectrum of configurations of the path. With Λ^r set to 3.5 Mb/s and Λ^l set to 1.5 Mb/s, leading to a total workload of 5 Mb/s, and a buffer size K_n of 20 packets at each node, we consider the relative error on the goodput G^r for values of $d_{1,2}$ and $d_{2,3}$ ranging from 110 to 390 meters in steps of 10 meters. Table IV reports the distribution of the relative errors, as well as the mean error values. In the over 800 examples studied, the relative error remain below 5% in over 80% of the cases explored. The mean error is around 4%.

As for the datagram loss probability L^r, we explore the model accuracy for values of K_n ranging from 5 to 50 in steps of 5, and for workload values of the flow at node 1, Λ^r going from 0.2 to 5 Mb/s in steps of 0.2. The rate of the second flow Λ^l is kept to 1.5 Mb/s and the distance between nodes is such that $d_{1,2}$ = 360 meters and $d_{2,3}$ = 250 meters. Table V displays the corresponding results for the absolute error on the loss probability. Here we observe that the mean error is at 1.58% and that it remains below 4% in over 98% of the 250 examples studied.

TABLE IV: Overall accuracy on the goodput for a path with 3 nodes and 2 flows.

Average	<5%	5-10%	10-20%	>20%
4.05%	81.69%	5.83%	7.99%	4.49%

TABLE V: Overall accuracy on the loss probability or a path with 3 nodes and 2 flows.

Average	<1%	1-2%	2-3%	3-4%	>4%
1.58%	23.6%	48.4%	21.2%	5.2%	1.6%

C. Extension to account for hidden node problem

In our third case study, we consider a path with $N = 4$ nodes, which forwards datagrams from node 1 up to node 4 through nodes 2 and 3. Datagrams are generated at node 1 at a rate Λ^r. We assume that the communication and carrier sensing ranges of a node cover its 1-hop and 2-hop neighbor nodes, respectively. Therefore, nodes 1 and 4 are exposed to the hidden node problem. Frames sent by node 1 can collide with the acknowledgments (ACK) returned by node 4 (to node 3). For some locations of the nodes, the collision probability can be as large as 30%. Thus, and unlike previous scenarios, we need to take into account the risk of frames collision in our modeling.

We estimate the probability that a frame sent by node 1 will collide with an ACK of node 4 as follows: $\text{COL}_1^r = \text{HID}_1^r + \text{ST}_1^r$ where HID_1^r denotes the probability of collision due to the hidden node, i.e. node 4, and ST_1^r refers to the collision probability because of a node within its carrier sensing range, i.e. nodes 2 and 3, which finishes its backoff countdown at the exact same time as node 1, and thus simultaneously starts its transmission. Note that this latter decomposition is made possible because it involves two disjoint events. Although the full details can be found in [12], we indicate here that HID_1^r is expressed as the ratio between the duration of a collision and the full period that elapses between two frames transmissions by node 1. The computation of ST_1^r is as follows: $1 - \prod_{j \neq 1}(1 - U_j/\overline{B}_j)$ where j iterates over the nodes within the carrier sensing range of node 1. More explanations are given in [12].

As for γ_n and β_n, there is no specific difficulties associated to this scenario. γ_n is computed as previously described. In the case of β_n, we rely again on the relation: $1/\beta_n = \overline{B}_n/\overline{np}_n$. More precisely, $\overline{B}_n$ is evaluated as in Section IV-A, while the computation of $\overline{np}_n$ follows the scheme given in Section IV-B.

We consider the relative error on the goodput G^r with a workload rate set to $\Lambda^r = 2$ Mb/s and buffers at nodes with size $K_n = 20$ datagrams for hundreds of positions of the two relay nodes.

$d_{1,2}$ and $d_{2,3}$ vary from 110 to 350 meters in steps of 10 meters, while $d_{1,4}$ is constantly kept to 750 meters. This leads to 750 possible configurations. Corresponding results are presented in Table VI. The mean error is less than 3% and the error is always less than10%.

As a further matter, we set the distances between the nodes to $d_{1,2} = 100$, $d_{2,3} = 300$ and $d_{1,4} = 750$ meters, and we consider a wide spectrum of values for Λ^r, ranging from 0.2 to 4 Mb/s in steps of 0.2 and for K_n, ranging from 5 to 50 in steps of 5. We report the found values for the absolute error on the loss probability in Table VII. The mean error is found to be around 1.5%, and only 3.5% of the hundreds of considered examples yield to an error exceeding 7%. There are no examples whose error exceeds 10%.

TABLE VI: Overall accuracy on the goodput for a path with 4 nodes and 1 flow.

Average	<2%	2-5%	5-10%	>10%
2.73%	35.36%	52.86%	11.78%	0.0

TABLE VII: Overall accuracy on the loss probability or a path with 4 nodes and 1 flow.

Average	<1%	1-4%	4-7%	7-10%	>10%
1.49%	36.0%	44.0%	16.5%	3.5%	0.0%

D. Model exploitation

Following the validation of the model accuracy, we provide two practical examples that illustrate how the proposed model can help in the deployment of a multi-hop wireless network with 4 nodes and a single flow. In these examples, we decide to let only the node 2 mobile. The nodes 1, 3 and 4 are kept fixed with $d_{1,3}$ = 400 and $d_{1,4}$ = 750 meters. The buffers at nodes are set to a size of K_n = 20 datagrams.

First, we rely on our model to study the behavior of the queue length in the buffer of nodes, Q_n. Let us remind that datagrams are enqueued in buffers, waiting for transmission over the next link. We set the rate of the workload to Λ^r = 2.5 Mb/s. Figure 7 describes the evolution of the mean queue length at nodes 1, 2 and 3 as a function of the distance between nodes 1 and 2, $d_{1,2}$. Because of the BER, as the distance $d_{1,2}$ grows, the quality of the link between nodes 1 and 2 (resp. nodes 2 and 3) decreases (resp. increases). Therefore, the bottleneck link in the path is gradually shifting. With small values of $d_{1,2}$, i.e. less than 50 meters, the BER for the link between nodes 2 and 3 is very large (more than 60% of frames transmissions fail), and in turn, this latter link tends to become the bottleneck. A queue of datagrams is built up at

node 2 (clearly shown by Figure 7 where the mean buffer occupation is close to 20). For larger values of $d_{1,2}$, say between 60 and 350 meters, the BER is relatively low for all links. However, with the exception of the link between nodes 2 and 3, the two other links are exposed to the hidden node problem and thus experience a slowdown pace of transmission. This is the reason why the queues at nodes 1 and 3 are close to the saturation while node 2 is not. Finally, for $d_{1,2}$ larger than 350 meters, the link between nodes 1 and 2 is greatly flawed, and thus it becomes the bottleneck link. Consequently, a large queue is built up at node 1. This first example clearly points out the high sensitivity of a path to the BER of its links, hence making its behavior hardly predictable without a fair performance analysis.

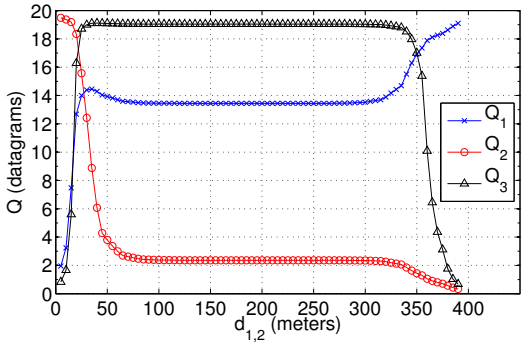


Fig. 7: Mean buffers occupation as a function of the distance between nodes 1 and 2 with $K_n = 20$ and $\Lambda^r = 2.5$ Mb/s.

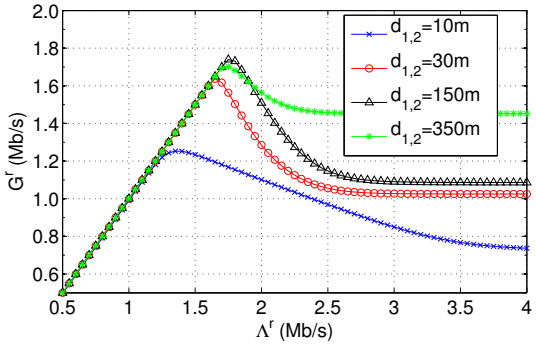


Fig. 8: Goodput of the path as a function of the workload level Λ^r for different positions of node 2 with $K_n = 20$ and $d_{1,3} = 400$ meters.

Multi-hop wireless paths are known to undergo their best rate of goodput when the workload is capped under a certain threshold, e.g. [13], [14]. Our second example focuses on the existence of this goodput "optimum" and on the associated performance collapse which occurs when the workload is excessive. We compute the goodput G^r attained by a 4-nodes path for various values of Λ^r and for different positions of node 2, $d_{1,2}$. The corresponding values of G^r are displayed in Figure 8. We observe that our proposed model captures this performance optima. The magnitude of these tipping points (turnaround) widely varies depending on the specific location of node 2. For instance the gap between the maximum value and the asymptotic value (attained when the workload is close to 4 Mb/s) goes up to nearly 50% for $d_{1,2} = 10$ meters, while it stands around 15% in the case of $d_{1,2} = 350$ meters. Our proposed model allows to find out the optimal value of Λ^r as well as the corresponding level of goodput at the expense of a very low computational

cost. This knowledge can be useful for control mechanisms (such as admission control and traffic shaping policies), which aim to protect the network from congested and "counter-productive" states.

V. Related Work

There is a large body of literature devoted to the performance modeling of the IEEE 802.11 protocol. Many of the proposed models address different problems though. The differences include the specific topology of interest (cells or multi-hop networks), the assumptions on the generated traffic (while some studies restrict their analysis to the sole case of saturated traffic sources at nodes, others cover the more general case of nodes in non-saturated regime), the length of paths on which flows are routed (single-hop path versus multi-hop path), the properties of the physical layer (e.g., perfect or non-perfect with BER), the length (if any) of the buffer at nodes (to enqueue datagrams before transmission), as well as some precise DCF mechanisms (e.g., retransmission limit).

In the case of a cell (i.e., a single-hop network in which each node senses the transmission of each other), Bianchi [1] developed a seminal model based on a Discrete-Time Markov Chain to evaluate the attainable throughput of the cell. In this work, the author assumes that every node runs in a saturated regime, and that frame transmissions are not exposed to a BER, neither to a retransmission limit. Several follow-up studies were carried out to better match some IEEE 802.11 mechanisms [15] (and the references therein), to include a more realistic radio channel with the presence of a BER [16], or to cover non-saturated scenarios [17]. However, all these latter studies are devoted to a single cell scenario, thereby bypassing the hidden node problem (since each node senses all other transmissions) as well as the strong correlations between nodes in a chain.

For multi-hop wireless networks, where nodes are not necessarily in each other carrier sensing range, existing works typically fall into two categories depending on the length of the paths on which flows are routed, i.e., single-hop versus multi-hop path. In the case of a single-hop path, in which the source and the destination are only one hop away, Wang and Kar [18] developed a Markovian model to evaluate the average throughput of flows. To help their analysis, the authors assume no binary exponential backoff, a perfect physical layer (no BER) and a RTS/CTS mechanism, which never experiences failures. Moreover, they consider all nodes to be

identical in terms of workload. In [19], the authors proposed an hybrid approach to predict the achievable throughput between sources and destinations that are one hop away. They designed an analytical model, whose parameterization is based on a empirical pre-processing step that measures the actual radio condition of each node. In another set of works, the authors of [10], [20], [21] proposed analytical models to derive the throughput of flows transmitted on single-hop paths. They come up with a closed-form expression for the achievable throughput in [21] under the assumption that all nodes are in saturation. Their approach does not seem to allow the computation of other performance parameters of interests, e.g., the datagram loss probability. In a prior work [10], that generalizes the approach proposed in [20], the authors use an iterative approach to estimate parameters like flow throughput and frame loss probability. However, the non-saturated case study is based on the assumption that the sending throughput of a node equals the receiving throughput of the same node. This is clearly not the case when the links do not have the same quality.

When it comes to the case of multi-hop flows, i.e., flows where source and destination can not directly communicate, there is only a handful of works, which have tackled through an analytical standpoint the performance analysis of flows conveyed through a path. In [22], the authors highlighted some of the complex phenomena undergone by a wireless path (e.g., mutual exclusion of links, unfairness between the nodes). In particular, they studied the length evolution of the queues at the buffer nodes, and they emphasized their correlation (some queues may be empty, while others are not). In the same work, the authors also proposed an analytical model, which applies only when every node of the path, including the relay nodes, is in a saturated regime (i.e., it always has a datagram waiting to be transmitted), which appears as a strongly simplifying assumption. In a separate work [23], the authors proposed to estimate the maximum attainable throughput of the path by considering that its value approximately corresponds to the actual capacity of the bottleneck link (i.e., the link that takes the longest time to transmit a datagram). By successively iterating on the nodes of the path, they estimate the delay incurred by the transmissions of neighbor nodes on the transmission time of a datagram in oder to update its current value. Such an approach does not seem to allow the computation of the datragam loss probability. The authors of [24] proposed an original approach, which aims to partition a multi-hop network into two-edge topologies (i.e. two distinct edges not sharing the same transmitter) in order to characterize the achievable rate of each edge of the network. Unlike our modeling

framework, they assume a large buffer size in order to overlook the potential datagram losses.

To summarize, there is only a few works devoted to the analytical performance evaluation of IEEE 802.11 in the case of multi-hop networks with multi-hop flows. Besides, it seems that none of them handles at the same time realistic assumptions regarding the behavior of the MAC protocol, the inter-dependencies in the distribution of the workload among the nodes (some nodes may be in saturation while others may be in starvation), the hidden node problem, and the finite size of buffers. In a previous work [11], we presented an analytical model for the performance analysis of a multi-hop flow in a three-nodes path. Because our model features buffers of finite size at each node, it captures the potential starvation state (i.e., empty buffers) of relay nodes, even if the first node of the path may be in saturation. However, this latter model was restricted to the case of a three-nodes scenario, in which the hidden node problem does not exist, thereby preventing virtually all frame collisions. In a follow-up work [12], we extended this model to a four-nodes scenario, in which collisions due to hidden nodes may arise. Compared to our former studies [11], [12], this paper provides a significant step to demonstrate the accuracy and the versatility of our modeling framework as it includes much more numerical results (thousands of examples) for the output throughput and for the datagram loss probability, as well as a complete new scenario with 2 opposite flows.

VI. Conclusions

In this paper, we consider the performance evaluation of a single multi-hop path (also called chain), based on the IEEE 802.11 DCF. The proposed modeling framework is constructive and versatile, so that it can handle various types of multi-hop wireless paths. Its originality comes from its two-level abstraction: the high-level mostly seeks to mimic the buffer behavior of nodes along the path, while the low-level estimates the delay for transmitting a packet according to the IEEE 802.11 DCF, given the surrounding environment of a node. With only small refinements, we apply the proposed modeling framework on three case studies, which together cover various potential difficulties, e.g., two flows in opposite directions, and topologies where nodes are exposed to the well-known hidden node problem. We performed thousands of discrete event simulations to assess the accuracy of the derived models for the attained goodput of flows and for the datagram loss probability. It is our conclusion that their accuracy is generally good, and that they capture fundamental phenomena occurring in multi-hop wireless networks such as

performance collapse and starvation. Finally, in all the thousands of examples we explored in this section, the fixed-point iteration involved in the solution to the proposed models never failed to converge within typically just a few tens of iterations, which corresponds to an execution time smaller than a second on a standard machine with no code optimization. Having in mind that multi-hop wireless networks can be viewed as a set of inter-dependent multi-hop paths, our future works aim to extend our modeling framework to handle more general topologies.

REFERENCES

[1] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE JSAC*, vol. 18, no. 3, pp. 535–547, 2000.

[2] M. Garetto, J. Shi, and E. W. Knightly, "Modeling media access in embedded two-flow topologies of multi-hop wireless networks," in *ACM MobiCom*, 2005.

[3] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Ad Hoc Networks*, vol. 1, 2003.

[4] C. Chaudet, D. Dhoutaut, and I. G. Lassous, "Experiments of some Performance Issues with IEEE 802.11b in Ad Hoc Networks," in *IEEE/IFIP WONS*, pp. 158–163, 2005.

[5] G. Bianchi, A. D. Stefano, C. Giaconia, L. Scalia, G. Terrazzino, and I. Tinnirello, "Experimental assessment of the backoff behavior of commercial IEEE 802.11b network cards," in *IEEE INFOCOM*, 2007.

[6] R. W. Wolff, "Poisson arrivals see time averages," *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982.

[7] W. Xiuchao, "Simulate 802.11b Channel within NS2," tech. rep., National University of Singapore, 2004.

[8] M. Fiore, "http://perso.citi.insa-lyon.fr/mfiore/research.html."

[9] NS2, "http://www.isi.edu/nsnam/ns/."

[10] M. Garetto, T. Salonidis, and E. W. Knightly, "Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks," in *IEEE INFOCOM*, 2006.

[11] T. Abreu, B. Baynat, T. Begin, and I. Guérin-Lassous, "Hierarchical Modeling of IEEE 802.11 Multi-hop Wireless Networks," in *ACM MSWiM*, pp. 143–150, 2013.

[12] T. Abreu, B. Baynat, T. Begin, I. Guérin-Lassous, and N. Nguyen, "Modeling of IEEE 802.11 multi-hop wireless chains with hidden nodes," in *ACM MSWiM*, pp. 159–162, 2014.

[13] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of Ad Hoc Wireless Networks," in *ACM MobiCom*, pp. 61–69, 2001.

[14] T. Abreu, N. Nguyen, T. Begin, I. Guérin-Lassous, and B. Baynat, "Substitution Networks: Performance Collapse Due to Overhead in Communication Times," in *ADHOCNETS*, pp. 1–16, 2012.

[15] I. Tinnirello, G. Bianchi, and Y. Xiao, "Refinements on IEEE 802.11 distributed coordination function modeling approaches," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 3, 2010.

[16] V. Vishnevsky and A. Lyakhov, "802.11 lans: Saturation throughput in the presence of noise," in *IFIP Networking*, pp. 1008–1019, 2002.

[17] D. Malone, K. Duffy, and D. Leith, "Modeling the 802.11 distributed coordination function in nonsaturated heterogeneous conditions," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 1, pp. 159–172, 2007.

[18] X. Wang and K. Kar, "Throughput modelling and fairness issues in CSMA/CA based ad-hoc networks," in *IEEE INFOCOM*, pp. 23–34, 2005.

[19] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, "A general model of wireless interference," in *ACM MobiCom*, pp. 171–182, 2007.

[20] M. Garetto, J. Shi, and E. W. Knightly, "Modeling media access in embedded two-flow topologies of multi-hop wireless networks," in *ACM MobiCom*, 2005.

[21] B. Nardelli and E. W. Knightly, "Closed-form throughput expressions for CSMA networks with collisions and hidden terminals," in *IEEE INFOCOM*, 2012.

[22] A. Aziz, M. Durvy, O. Dousse, and P. Thiran, "Models of 802.11 multi-hop networks: Theoretical insights and experimental validation.," in *IEEE COMSNETS*, 2011.

[23] M. M. Hira, F. A. Tobagi, and K. Medepalli, "Throughput analysis of a path in an IEEE 802.11 multihop wireless network," in *IEEE WCNC*, 2007.

[24] A. Jindal and K. Psounis, "The achievable rate region of 802.11-scheduled multihop networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 4, pp. 1118–1131, 2009.