Hierarchical Modeling of IEEE 802.11 Multi-hop Wireless Networks

Thiago Abreu
Université Lyon 1 - LIP
France
thiago.wanderley@ens-lyon.fr

Bruno Baynat
Université Pierre et Marie Curie - LIP6
France
bruno.baynat@lip6.fr

Thomas Begin
Université Lyon 1 - LIP
France
thomas.begin@ens-lyon.fr

Isabelle Guérin-Lassous
Université Lyon 1 - LIP
France
isabelle.guerin-lassous@ens-lyon.fr

ABSTRACT

IEEE 802.11 is implemented in many wireless networks, including multi-hop networks where communications between nodes are conveyed along a chain. We present a modeling framework to evaluate the performance of flows conveyed through such a chain. Our framework is based on a hierarchical modeling composed of two levels. The lower level is dedicated to the modeling of each node, while the upper level matches the actual topology of the chain. Our approach can handle different topologies, takes into account Bit Error Rate and can be applied to multi-hop flows with rates ranging from light to heavy workloads. We assess the ability of our model to evaluate loss rate, throughput, and end-to-end delay experienced by flows on a simple scenario, where the number of nodes is limited to three. Numerical results show that our model accurately approximates the performance of flows with a relative error typically less than 10%.

1. INTRODUCTION

IEEE 802.11, based on CSMA/CA principles, is one of the leading communication protocols for wireless networks. Its deployment has been largely supported by its distributed property as well as by the simplicity and the low cost associated with its implementation. Various underlying topologies for IEEE 802.11 exist. Clearly, the most frequent example is the infrastructure network, commonly referred to as a WiFi hotspot, where communications occur exclusively between nodes and the access point. Other examples include multi-hop wireless networks, where communications between nodes may involve relay nodes, that act like forwarding entities.

Throughout this paper we refer to the sequence of nodes involved in the communication between two nodes as being a chain. The behavior and associate performance of such a chain can not be easily derived since the latter is subject to many factors, e.g. the number and position of nodes in the chain, and it may exhibit complex phenomena like the freezing period of backoff, the hidden node problem, etc.

Despite these intrinsic complexities, analytical models can represent a convenient means to forecast and explain the performance of flows travelling along a multi-hop wireless network. However, only few models have been proposed for characterizing their behavior. Indeed, most of the analytical studies on IEEE 802.11 are devoted to either cell networks, or multi-hop networks with single-hop flow [2, 11, 5, 9]. Considering the few works dealing with multi-hop flows [1, 6], they consider simplifying assumptions (e.g. buffers with infinite length to queue extra datagrams, ideal physical layer for transmitting frames, specific scenarios where the network is overloaded or even totally saturated) that significantly deviate from some of the fundamental properties arising in IEEE 802.11 chains, and thus they may not be accurate when applied to realistic scenarios.

We present new a modeling framework to evaluate the performance of a flow conveyed through a chain. Our framework is based on a hierarchical modeling composed of two levels. The lower level is dedicated to the modeling of each node, while the upper level takes into account the actual shape (topology) of the chain. Thus, given the shape of the actual chain, the channel degradation pattern due to Bit Error Rate and the rate of the flow conveyed through the chain, the model delivers approximate values for customary performance parameters, e.g. loss rate, throughput, end-to-end delay, using a simple fixed-point iteration between the two modeling levels.

The remainder of the paper is organized as follows. Section 2 reminds the principles of IEEE 802.11 protocol and describes the considered scenario in this work. Section 3 presents our modeling framework. In Section 4, we compare the performance delivered by our model with the actual ones coming from a discrete-event simulator. Section 5 concludes this paper.

2. SYSTEM DESCRIPTION

2.1 IEEE 802.11

In this section, we remind the main principles of IEEE 802.11 DCF protocol [7]. Basically, the transmission of a datagram occurs as follows. First, the node continuously senses the channel until the latter is found to be idle for a time DIFS. Second, the node postpones the transmission of the frame associated to this datagram by a time whose length is randomly drawn within an interval determined by the contention window (also denoted CW hereafter). This latter delay is expressed as a number of slot times and is referred to as the *backoff*. Then, the backoff value is continually decremented (by one) whenever the channel is sensed idle during an entire slot time. If the channel is sensed as busy, the backoff value remains constant. It is said to be in *freezing* state. The backoff quits this freezing state, and hence resumes its decrement, if the channel is sensed idle during a DIFS period. As the backoff value reaches zero, the node can start its transmission. Note that a backoff period can be interrupted several times due to multiple transmissions in its neighborhood.

A node considers that its frame has been successfully transmitted to its neighbor node when it receives the acknowledgment. If the node does not receive an acknowledgment before a timeout, which includes a SIFS, it assumes the corresponding frame has been lost, and thus attempts to retransmit it up to a given limit [7]. Note that at each retransmission of a frame, the contention window size is doubled. If the transmission of a frame as well as the following retransmissions fail, the node simply discards the corresponding datagram. Finally, each successful transmission of a frame (as well as a complete discard) resets the contention window to its initial value.

IEEE 802.11 includes an optional mechanism to avoid collisions, known as RTS/CTS. When activated, prior to each frame transmission, a node sends a RTS (Request-to-Send) frame and should receive a CTS (Clear-to-Send) frame in the aim of getting exclusive use of the channel. Note that since RTS/CTS has been shown to be inefficient in the case of a chain [12], we did not consider it in our study.

2.2 Studied chain and associated scenario

We consider the case of a wireless multi-hop chain with N nodes (labeled from 1 to N). We now itemize the main assumptions regarding our system. First, each node can only communicate with its 1-hop neighbors. Second, the carrier sense range of each node covers all its 2-hop neighbor nodes. Third, we consider a non-perfect physical layer that affects the channel quality, and can cause erroneous frame transmissions. We assume that the Bit Error Rate (BER) does only depend on the distance between nodes. Here, we use the values reported in [8].

We assume that the network chain conveys a single flow, starting at one border node and traveling up to the other border node. The flow may represent the traffic generated by a single (or multiple) user or application. A flow is composed of equally sized datagrams. The rate at which this flow generates datagrams determines the level of workload of the chain. Note that our system will be described as being in saturation if the flow always have datagrams to be sent across the chain. Figure 1 illustrates a possible example of our system with 6 nodes. As exhibited by this figure, nodes in the chain do not have to be aligned along a straight line, nor being equally spaced as long as they meet the assumptions mentioned above.

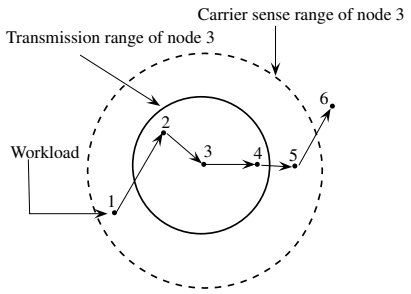


Figure 1: A multi-hop chain with a single flow and N=6 nodes.

We now detail the scenario that will be used as case study throughout the next sections. In this scenario, we consider a multi-hop chain with N=3 nodes that conveys a single flow travelling from node 1 to node 3 through node 2. We position node 2 at different distances in between nodes 1 and 3, and we consider several levels of rates for the flow workload. Clearly, the performance of the flow will vary against the distance between the nodes since each distance leads to a different BER, and thus to a different rate of erroneous frame transmission. Although this scenario does not face the hidden node problem (since all nodes are in the detection range of each other), it has to cope with several other issues, e.g. heavy dependence between nodes, freezing periods of the backoff, recurrent starvation of the relay node.

3. HIERARCHICAL MODELING

We have developed a hierarchical modeling framework for performance evaluation of a general multi-hop wireless chain. Our model is made of two levels: 1) a global high-level queueing network model and 2) several local low-level Markov chain models. In the global model, each queue is associated with a node of the wireless chain, and the routing between queues matches the topology of the chain. In order to parameterize the global model, we use several local Markov chain models, each one being associated to a given queue of the global model, i.e., to a given node of the chain. As a first step in the development of this general framework, we describe our hierarchical model on a simple scenario in Sections 3.1 and 3.2. As detailed in these two subsections, the inputs of the global model are the outputs of the local models, and vice versa. The performance of the resulting model will thus be the solution of a fixed-point iterative algorithm described in Section 3.3.

3.1 Global model

As explained before, in this paper we restrict the description of the model to a simple scenario made of three nodes, 1, 2 and 3, although we believe it can be extended to more complex scenarios. Node 1 generates a flow of datagrams to transmit to node 3, through relay node 2, which has an

intermediate position between 1 and 3. We naturally associate with this scenario the queueing network depicted in Figure 2. The customers of this queueing model are the datagrams of the wireless network and both queues have a single server modeling transmission of datagrams. Every single queue of this network corresponds to a node of the system that has to transmit datagrams, i.e., nodes 1 and 2. Indeed, in our simple scenario, node 3 is the receiver and only sends acknowledgment frames.

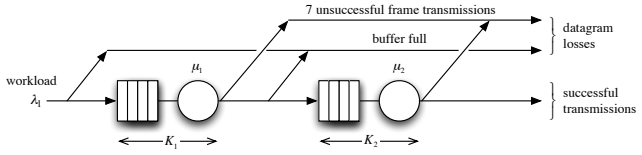


Figure 2: Global queueing model.

Each queue i of the model has a single server with rate μ_i and a finite capacity K_i that matches the buffer size of the wireless cards. These finite capacities will result in datagram losses due to buffer overflows. As illustrated in Figure 2, our model also takes into account the other potential cause of datagram losses, that again closely matches IEEE 802.11 specifications, which corresponds to excessive retransmissions. In our model, we consider that, after seven unsuccessful frame transmissions, a datagram is discarded [7]. Finally, it is important to emphasize that our model is a queueing network with a routing that matches the chain structure of our wireless system. Hence, the average number of datagrams that a relay node receives by unit of time and has to transmit over the radio channel, matches the average number of datagrams by unit of time that have been correctly transmitted without error by the preceding node in the chain. As will be seen in the Section 4, this implies that, even in a saturated case where one border node of the chain has always datagrams to transmit, nodes "in the middle" of the chain may not be saturated and can even suffer from starvation. This starvation has a significant impact on the performance of the system.

One important feature of our model lies in the definition of the service times of the different queues. For queue i ($i = 1$ or 2 in our initial simple scenario), the service time of the queue represents the time separating the instant when a datagram is ready to be sent by node i over the radio channel, from the instant where it is either acknowledged or discard because of excessive retransmissions. The service time of a customer of the model thus includes transmission times of all frames that are necessary to transmit the datagram, as well as backoff times preceding any transmission, freezing times of backoff periods due to canal occupation by other transmissions, and all required protocol delays (DIFS, SIFS, timeout).

The originality of the model is to include the freezing time of a backoff in the definition of the service of a datagram. By doing so, the model works around the complexity of representing the strong synchronization between the different nodes of the system, namely the fact that, when a node transmits, it not only prevents nodes (that can sense its transmission) to transmit, but also freezes all ongoing backoffs. Instead, we opt for decoupling the two queues of the global model, by including in the definition of the service time of each queue, the average freezing time caused by transmission of the other one. We come up with a desynchronized queueing network model of a strong synchronized system. We really believe that this feature will be crucial to scale up our model to larger scenarios.

In order to solve our queueing model, we first decompose it into two isolated single queues, as illustrated in Figure 3. It is important to emphasize that this was not necessary for the scenario we consider (with only two queues), as we could, under certain assumptions, solve this open queueing network. However, we wanted our methodology to be easily generalizable to more complicated and realistic scenarios, including more than three nodes and independent flows in opposite directions. By decomposing the global model into single queues, we believe that this approach can be extended pretty easily to other scenarios.

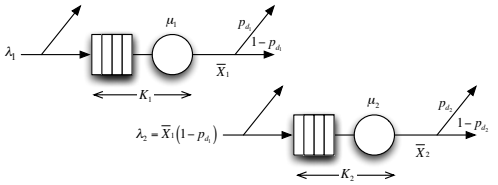


Figure 3: Decoupled queueing model.

The first assumption of this decomposition, consists in assuming that the arrival process at each queue of the network is a Poisson process, resulting in two isolated *M/G/1/K* queues. Assuming that datagrams are generated by the source (node 1) according to a Poisson process is a classical assumption, made in many papers [2, 4]. Let λ_1 be the rate with which datagrams are injected in the network. Though, in our original queueing model, the output process of queue 1 is generally not Poisson, this assumption will not have a significant impact on the accuracy of the model and drastically simplify its solution.

The second approximation we make, always with the objective of making the model easily tractable and expandable, is to consider that the service times of the two isolated queues are exponentially distributed. The resulting model thus consists in two isolated *M/M/1/K* queues (see Figure 3) and is therefore very easy to solve, provided that we know the values of the service rates μ_1 and μ_2, as well as those of the datagram loss probabilities p_{d_1} and p_{d_2} due to excessive retransmissions (outside buffer overflows that are performance parameters of the queueing model). We will derive in the next subsection the estimation of the service rates. Now, as we are only interested here in datagram loss probabilities resulting of seven consecutive unsuccessful frame transmissions, their expressions can directly be obtained from frame error probabilities p_{f_i} as:

$$p_{d_i} = p_{f_i}^{7} \tag{1}$$

A frame transmission error can be the result of the BER (Bit Error Rate) or a collision with another node transmission. In our simple scenario, we assume that the three nodes are within each other carrier sense range, and we can thus consider that no collisions take place in this system. The

frame error probability p_{f_i} can then be classically estimated from the BER that is function of the distance between the sender (say node i) and the receiver (node $i+1$), extracted from manufacturers card specifications [8].

All performance parameters of our global queueing model can then be obtained from the well known results of the $M/M/1/K$ queue. The output throughput of station i, corresponding to the average number of datagrams that are correctly transmitted by unit of time from node i to node $i+1$, is:

$$\overline{X}_i = \mu_i(1 - \pi_i(0)) \tag{2}$$

where $\pi_i(n)$ is the probability of having n customers in the i-th $M/M/1/K$ isolated queue.

The average number of customers in queue i, corresponding to the average number of datagrams waiting or in transmission at node i, is given by:

$$\overline{Q}_i = \sum_{n=1}^{K_i} n\pi_i(n) \tag{3}$$

From Little's law, we can derive the mean sojourn time of an admitted customer in queue i, corresponding to the average time a datagram that is not lost because of buffer overflow, stays in node i before being transmitted to node $i+1$ (or discarded because of excessive retransmissions):

$$\overline{R}_i = \frac{\overline{Q}_i}{\overline{X}_i} \tag{4}$$

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

$$\overline{U}_i = 1 - \pi_i(0) \tag{5}$$

And finally, from PASTA theorem, we can obtain the rejection probability, corresponding to the probability that a datagram is rejected because the buffer of the destination node is full at its arrival instant:

$$p_{r_i} = \pi_i(K_i) \tag{6}$$

3.2 Local models

In order to estimate the missing parameters of the global queueing network model, i.e., the rates μ_i $(i = 1, 2)$ of the two $M/M/1/K$ queues, we propose to associate with each queue, a Continuous-Time Markov Chain (CTMC) describing precisely the transmission process of a node according to the IEEE 802.11 DCF protocol.

Throughout this subsection, we will then focus on the CTMC associated with each single queue i of the global decomposed model ($i = 1$ or 2). The objective of this CTMC is to provide an estimation of the mean service time S_i, inverse of the service rate μ_i, of the associated $M/M/1/K$ queue i. To do this, the CTMC has to precisely describe the succession of the different states a node has to go through in order to transmit a datagram over the wireless channel. The CTMC is depicted in Figure 4 and is globally made of seven "lines", each one corresponding to a given stage k of the backoff and modeling the backoff time preceding the k-th transmission of a given datagram (provided the $k-1$ preceding transmission of the datagram were in error). Recall that a maximum

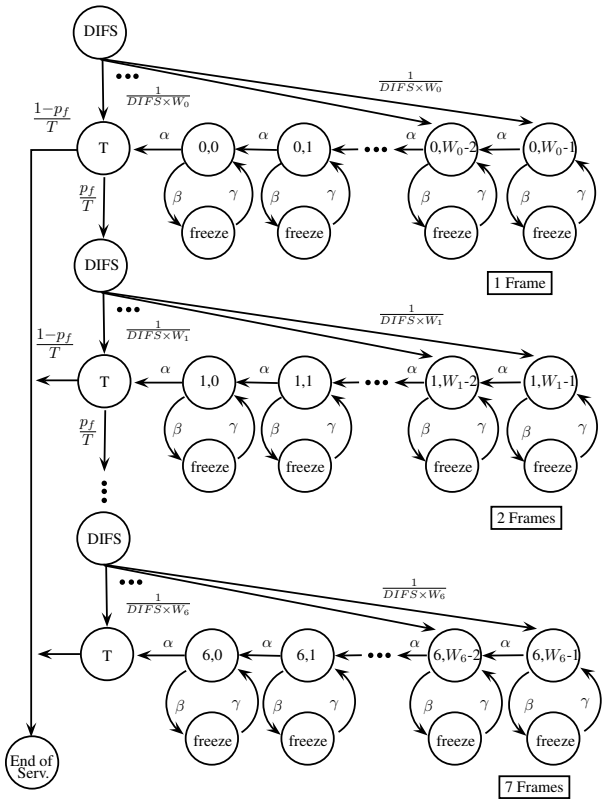


Figure 4: Local CTMC model for each node.

of seven unsuccessful transmissions is considered for a same frame associated to a datagram, after which the datagram is discarded. In order to simplify the drawing of this CTMC and the derivation of its performance parameters, we remove in the following, unless necessary, the index i from all notations (and use for example β instead of β_i). Let us however remind that there is a CTMC (with different parameters) associated with each transmitting node i.

As explained in Section 2, any transmission attempt starts with a DIFS time and then enters a backoff procedure, that can be interrupted anytime when the node senses a busy canal, and ends with a transmission time, that can be either successful leading to the "end of service" state with a rate $\frac{1-p_f}{T}$, or in error bringing the process to the next stage of the backoff with a rate $\frac{p_f}{T}$. In these rates, p_f is the frame error probability and T is the average of a time corresponding to the transmission of a frame plus the defer time SIFS plus a timeout value corresponding to the duration of an acknowledgment. A state $\{k, j\}$ corresponds to the k-th stage of the backoff (i.e., the k-th transmission attempt of the datagram) with an actual contention window equal to j ($j \in [0, W_k - 1]$). Exiting the k-th "DIFS" state, the process randomly chooses any state $\{k, j\}$ or the transmission state with a uniform probability (equal to $\frac{1}{W_k}$). From any state $\{k, j\}$, we can either reach state $\{k, j-1\}$ (or transmission state if $j = 0$) if the canal has remained idle during the current time-slot, or reach a "freeze" state if the canal has been sensed busy. The corresponding rates are denoted α and β. α is simply the inverse of a time-slot duration

and is fixed by IEEE 802.11 specifications. The inverse of β corresponds to the time separating two backoff freezing, provided that the node is in backoff. The rate out of any "freeze" state is γ, whose inverse corresponds to the freezing duration, which is nothing, in our simple scenario, but the transmission time of the other node (including the returning time of the acknowledgment) plus a DIFS time that is always deferred before any backoff retake.

As a result, most of the parameters of this CTMC come directly from IEEE 802.11 specifications, except the frame error probability p_f and the rate β. As developped in Section 3.1, the frame error probability can be classically estimated from the BER and the distance between the consider node and the next one in the chain. Estimating the last remaining parameter β is much more difficult and we now turn our attention to it.

As explained before, it is easier to consider the inverse of parameter β which corresponds to the mean time between two successive backoff freezing (provided a node is in backoff). This quantity is related to the average number of freezing of the backoff of the considered node i, denoted as $\overline{np}$ and the mean backoff duration $\overline{B}$. This is illustrated in Figure 5, where the hashed areas correspond to freezing periods of the backoff. If we assume that a) each backoff is paused

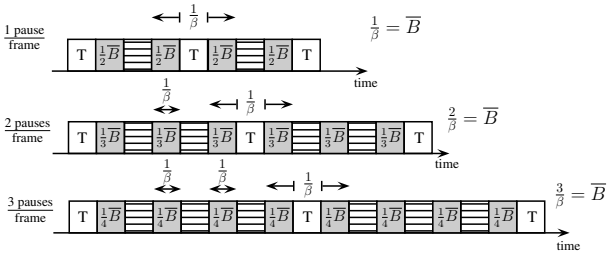


Figure 5: Illustration of the relation between β, $\overline{np}$ and $\overline{B}$.

exactly one time, we can see from the figure that, in average, $1/\beta = \overline{B}$; b) each backoff is paused exactly two times, $2 \times 1/\beta = \overline{B}$; and c) each backoff is paused exactly three times, $3 \times 1/\beta = \overline{B}$. We can easily extend these results to an average number of pauses in the backoff, $\overline{np}$, and obtain the following (pretty intuitive) relation:

$$\frac{\overline{np}}{\beta} = \overline{B} \tag{7}$$

It turns out that parameter β can be directly obtained from the estimation of the average number of backoff freezing and the mean backoff duration. This last parameter can easily be estimated from the following average:

$$\overline{B} = \frac{\frac{CW_1}{2} f_1 + \frac{(CW_1+CW_2)}{2} f_2 + \cdots + \frac{(CW_1+CW_2+\cdots+CW_7)}{2} f_7}{f_1 + 2f_2 + \cdots + 7f_7} \tag{8}$$

where CW_j is the size of the contention window at the j-th frame transmission attempt by assuming an initial CW of 31 and a maximal CW of 1023, see Section 4):

$$CW_j = \max(2^{4+j} - 1, 1023) \tag{9}$$

and f_j is the probability that the transmission of a datagram requires exactly j frames, and can be derived from the frame error probability as:

$$f_j = p_f^{j-1}(1 - p_f) \text{ for } j \leq 6 \text{ and } f_7 = {p_f}^6 \tag{10}$$

Now it remains to estimate the average number of pauses in the backoff of a node. Let us remind that the backoff of a given node i is paused whenever any node j in the carrier sense range of i makes a transmission. Now going back to our simplified scenario (with 3 nodes, but where only node 1 and 2 transmit datagrams), let us consider backoff of node 1. Figure 6 illustrates the successive transmissions of node 1 in a saturated case, i.e., when node 1 has always datagrams to transmit. In this case, between two frame transmissions there is always a backoff that is paused by transmissions of node 2. The average number of pauses of each backoff is thus equal to the average number of frame transmissions of node 2 in between two conscutive frame transmissions of node 1. If we denote by $\overline{F}_i$ the average number of frames transmitted by node i by unit of time ($i = 1, 2$), we thus have: $\overline{np}_1 = \frac{\overline{F}_2}{\overline{F}_1}$. (Note that we need here to put back the indexes corresponding to nodes in the notations.) We now

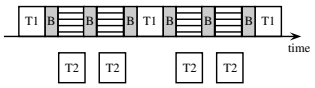


Figure 6: Relation between transmissions in neighbor nodes and backoff freezing in a saturated case.

consider the general case where node 1 is not saturated, illustrated in Figure 7. Between two transmissions there may occur an idle time (of average duration x on the figure) before the beginning of the backoff. Some transmissions of node 2 freeze the backoff of node 1, whereas others do not (because they happen during idle periods of node 1). The average number of pauses in the backoff of node 1 has thus to be adjusted with a corrective factor δ_1 as: $\overline{np}_1 = \frac{\overline{F}_2}{\overline{F}_1} \times \delta_1$. This corrective factor corresponds to the proportion of time node 1 is idle between two successive frame transmissions: $\delta_1 = \frac{y}{x+y}$. The final question relies on estimating the frame

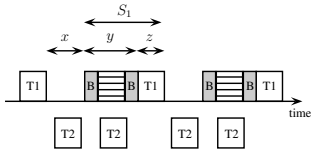


Figure 7: Relation between transmissions in neighbor nodes and backoff freezing in a non-saturated case.

throughput $\overline{F}_i$, as well as the corrective factors δ_i. Intuitively, the average number of frame transmissions by unit of time $\overline{F}_i$ is related to node i datagram throughput $\overline{X}_i$ and the corrective factor δ_i is related to node i utilization $\overline{U}_i$. It turns out that both quantities are performance parameters that can be derived from the global model described in the previous subsection.

First, the frame throughput $\overline{F}_i$ can easily be obtained from

the datagram throughput $\overline{X}_i$ of *M/M/1/K* queue i of the global model, by multiplying it by the average number of frame transmissions necessary to transmit a datagram:

$$\overline{F}_i = \overline{X}_i \; n_{f_i} \tag{11}$$

where n_{f_i} is expressed from the f_n probability as:

$$n_{f_i} = \sum_{n=1}^{7} n \; f_n \tag{12}$$

Then in order to obtain the correcting factor δ_i, we first express the utilization $\overline{U}_i$ of queue i with notations of Figure 7:

$$\overline{U}_i = \frac{S_i}{S_i + x} \tag{13}$$

and use it in the expression of the corrective factor:

$$\delta_i = \frac{y}{x+y} = \frac{S_i - z}{S_i \frac{1-\overline{U}_i}{\overline{U}_i} + S_i - z} \tag{14}$$

where z is just the actual transmission time of a frame.

Finally, by using together previous equations, we get the following expression for the missing parameters β_i:

$$\beta_i = \frac{\overline{X}_j n_{f_j}}{\overline{X}_i n_{f_i}} \frac{\delta_i}{\overline{B}} \tag{15}$$

with the convention that $j = 2$ if $i = 1$, and $j = 1$ if $i = 2$.

Now that all the parameters of the CTMC have been estimated, we can use it to get a fair estimation of the service rate μ_i of the associated *M/M/1/K* queue i. The inverse of this rate corresponds to the average time that is necessary to reach the state "end of service", starting from the first "DIFS" state of the Markov chain, and is given by:

$$\frac{1}{\mu_i} = t_0 + p_{f_i} \times (t_1 + p_{f_i} \times (t_2 + ... + p_{f_i} \times t_6))))))) \tag{16}$$

where t_k corresponds to the average time spent by the process in "line" k of the CTMC:

$$t_k = \text{DIFS} + \frac{W_k}{2} \times r + T \tag{17}$$

and r is the average time spent in any pair of loop states $(\{k, j\}, \{freeze\})$:

$$r = \frac{1}{\alpha} \times \left(1 + \frac{\beta}{\alpha}\right) \tag{18}$$

3.3 Fixed-point solution of the model

As developed in the two previous subsections, the global model takes as an input the service rates μ_i of all queues and provides the global performance parameters, among which datagram throughputs $\overline{X}_i$ (relation (2)) and node utilizations $\overline{U}_i$ (relation (5)). Conversely, the local models assume known values of datagram throughputs $\overline{X}_i$ and node utilizations $\overline{U}_i$ to parameterize the CTMCs (thanks to relations (8)-(15)) and give estimations of the service rates μ_i (relations (16)-(18)). It is then quite obvious to resort on a fixed-point iteration to obtain the values of the subsequent parameters. We detail the iterative procedure below:

0. Initialize rates β_i (with non-absurd values, e.g. by taking $np_i = 1$).

DIFS	50μs
SIFS	10μs
Time slot	20μs
Contention window size (min,max)	31, 1023
Frame transmission limit	7

Table 1: IEEE 802.11 parameters

1. Calculate the service rates μ_i by solving the CTMC associated with each node i of the chain (equation (16)-(18)).
2. Solve the global queueing model and derive its performance parameters (equations (2)-(6)).
3. Update rates β_i (equations (8)-(15)).
4. Repeat algorithm until convergence.

Once the algorithm has converged, we can obtain the performance parameters of interests from the well parameterized global model: the average throughput of datagrams that have reached the destination without error, $\overline{X}_2$, the datagram rejection probabilities at both nodes, p_{r1} and p_{r2}, and the end-to-end delay of correctly transmitted datagrams, $\overline{R}_1 + \overline{R}_2$.

4. NUMERICAL RESULTS

We simulate the scenario described in previous sections with *Network Simulator 2* (ns-2.35) [10], which is tuned to incorporate a non-perfect physical layer implementation in wireless transmissions [3]. Nodes 1 and 3 are fixed, 500m away from each other. The relay node can be positionned in several spots within the interval [110m,390m] away from node 1. The communication and carrier sense ranges cover 399m and 700m, respectively. The links capacities are configured at 11Mb/s, using a 2-ray ground propagation model.

We compare the results of our model with those delivered by simulation, for different levels of workload. In our simulations, similarly to our model, datagrams are generated by the source according to a Poisson process. Datagrams have a size of 1500 bytes while acknowledgments are limited to 14 bytes. For both queues we set the buffer size to 50 datagrams. We remind that no RTC/CTS handshake mechanism is used. All the other used values of the IEEE 802.11 parameters are given in Table 1. Note that our model can use and be adapted to any value of the IEEE 802.11 parameters.

Figure 8 compares the mean values of the service time of a datagram (as described in Section 3.1) obtained with our model with the values collected from simulation, with a workload of 6Mb/s and for different relay positions. As seen in this figure, the service time, which is a key parameter for our model, closely matches the values of the simulation, with a difference between both results smaller than 10%. Overall, the model presents good performance for most of the positions of the relay node. The existence of a high BER affects the results of our model for the extreme positions. As expected, when node 1 (source node) is distant of the relay node, we see the service time in the first queue being multiplied by a factor of 10. Nevertheless, our model satisfactorily represents this behavior.

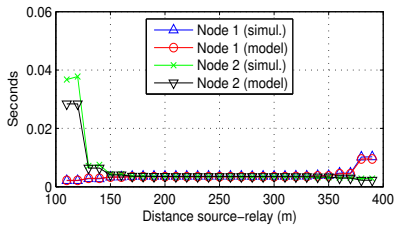


Figure 8: Mean service times of datagrams for a workload of 6Mb/s.

Figure 9 shows the mean time spent in backoff freezing before transmitting a frame, another key parameter of our model. For a workload of 6Mb/s the accuracy of our model. The relative error between the results is less than 10% for the worst cases and typically less than 5%. For the relay node, the mean values of backoff freezing increase substantially when this node is distant from the destination (node 3). This is due to losses by BER, which increases with the distance. The necessary retransmissions yield greater contention windows, which increases the probability that a neighbor node transmits and freezes the backoff decrement. For the source node, this behavior is not observed, despite the elongation of the mean backoff duration due to losses. If few datagrams are correctly transmitted from the source, clearly the relay node will very likely go to starvation. Therefore, it will not be able to freeze the backoff decrement at the source node. This behavior shows that there is no symmetry between the source and relay nodes, which may not be obviously initially.

Figure 10 shows the performance of our model, regarding the mean throughput of a node, datagram rejection probability by buffer overflows and mean end-to-end delay. These performance parameters correspond to characteristics of our model, as seen in Equations (2), (6) and (4). Figure 10(a) shows the throughputs of both nodes obtained from our model and from simulation, for a workload of 3Mb/s. Under these circumstances, depending on the actual distance between nodes 1 and 2, node 1 can be in saturation. We clearly see that the relative error between the results of our model and those from simulation is very low (around 3%). In a wide range of relay positions (between 150m and 350m), we see that the system is capable of conveying the entire workload. On the other hand, when source and relay nodes are close, the throughput of node 2 decreases due to high BER and the network can not cope with the workload. In the

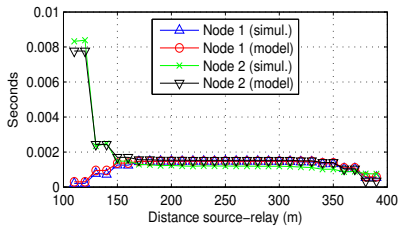


Figure 9: Mean backoff freezing duration for a workload of 6Mb/s.

other extreme position (source and relay nodes are distant), due to BER, there are more retransmissions for node 1 with consequent service time elongation and buffer overflows in node 1. Figure 10(d) shows that, when the system workload is set to 6Mb/s, the best performance of the system are obtained in several points around the middle position for node 2, where the effect of BER is less significant. With this workload, the system is not capable of conveying all datagrams and is in saturation (buffer overflows).

Figure 10(b) and 10(e) compare the datagram rejection probabilities of our model with simulation, for the same levels of workload. The accuracy of our model is clearly seen, since the relative error, when comparing to the simulation results, remains low for both cases (less than 5%). The rejection probability for the relay node is significantly more important when it is close to source node. This is due to the high rate of arriving datagrams from node 1 and to the frame losses between nodes 2 and 3 due to high BER. These losses increase the frame retransmissions and the mean service time of a datagram for node 2, which leads to a higher buffer occupation. Regarding the source node, it also loses several datagrams by buffer overflow whenever the BER is important. Moreover, for the workload of 6Mb/s, the chain is not capable of conveying the entire workload, and datagrams are rejected by the source node.

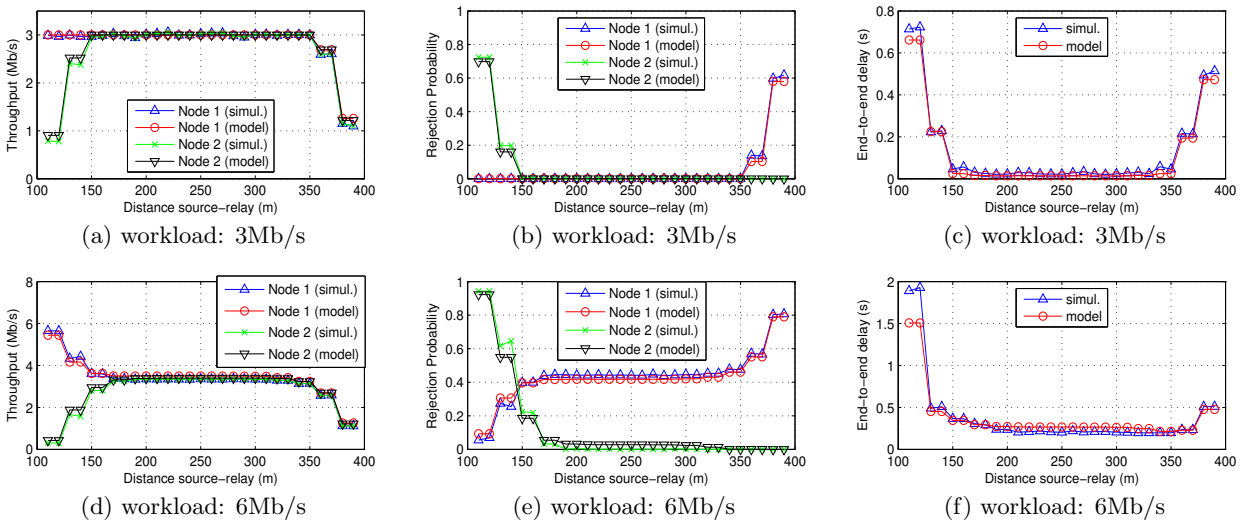
Finally, Figure 10(c) and 10(f) show the mean end-to-end delay of a datagram. The relative errors between our model results and those from simulation are low, ranging from 3% (best cases) up to 10% (worst cases). As expected, these values increase significantly when the BER is more important (long distances between nodes), leading to values up to 12 times larger when compared to the best cases around the middle position. To the best of our knowledge, our model is the only one to also estimate this performance parameter.

Given the accuracy of our model, we believe it can be used as a prediction tool to forecast the flow performance for different positions of the relay node. Figure 11 shows the behavior of the system throughput for several workloads, ranging from 0.1Mb/s up to 8Mb/s, and for five different distances between source and relay nodes. We can see the impact of saturation in the overall throughput, when the relay node is 150m away from the source. This scenario presents a performance degradation when the workload exceeds a certain limit (around 3.2Mb/s). This behavior can be exacerbated for other scenarios, and a prediction tool is thus of high importance, in order to to prevent the system from such a performance degradation.

We have also developed and validated a model for the scenario where two independent flows are conveyed throughout the chain, each starting at one border node and traveling up to the other border node. However, due to space constraints, related results are not presented in this work.

5. CONCLUSION

In this paper, we present a hierarchical modeling framework to evaluate the performance of flows conveyed through a multi-hop wireless chain. This latter approach is thought to deal with different topologies of the chain, can handle various channel degradation patterns due to Bit Error Rate



(a) workload: 3Mb/s (b) workload: 3Mb/s (c) workload: 3Mb/s

(d) workload: 6Mb/s (e) workload: 6Mb/s (f) workload: 6Mb/s

Figure 10: Mean throughput (a,d), rejection probability by buffer overflow (b,e) and end-to-end delay (c,f).

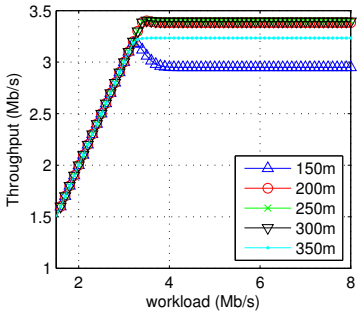


Figure 11: Evolution of the mean throughput as a function of the workload.

and can be applied to flows with rates ranging from light to heavy workloads. The resulting model is simple to implement and to parameterize. Its solution is easily obtained using a simple fixed-point iteration.

We validate our modeling framework on a simple scenario where the number of nodes in the chain is limited to three, yet the positioning of the relay node and the rate of the incoming flow may vary significantly. We compare the values provided by our modeling approach for customary performance parameters, e.g. loss rate, throughput, end-to-end delay, with simulation. Numerical results show that our model is able to accurately approximate the performance of flows. The relative difference is in general less than 10%.

Acknowledgments

This work was partially funded by the French National Research Agency (ANR) under the project ANR VERSO RESCUE (ANR-10-VERS-003). The authors wish to thank Nghi Nguyen for his valuable help with ns-2.

6. REFERENCES

[1] 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.

[2] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. In *IEEE JSAC*, 18(3), 2000.

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

[4] 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.

[5] 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.

[6] 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.

[7] IEEE 802.11 Working Group and others. IEEE 802.11-2012 Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. In *IEEE 802.11 Wireless LAN Standards*, 2012.

[8] H. Intersil. Direct sequence spread spectrum baseband processor, February 2002.

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

[10] NS2. http://www.isi.edu/nsnam/ns/.

[11] I. Tinnirello, G. Bianchi, and Y. Xiao. Refinements on IEEE 802.11 distributed coordination function modeling approaches. In *IEEE Transactions on Vehicular Technology*, 59(3), 2010.

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