Modular Analytical Performance Models for Ad Hoc Wireless Networks*

Fernando L. Dotti  Paulo Fernandes†  Afonso Sales  Osmar M. dos Santos

Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681 – 90619-900 – Porto Alegre, Brazil
{fldotti, paulof, asales, osantos}@inf.pucrs.br

Abstract

Wireless ad hoc networks raised a series of challenging research tracks. In order to analyze and validate research results achieved, thorough performance evaluation efforts are needed. In this context, the contribution of this paper is twofold: at the one side we investigate the effect of forwarding in wireless ad hoc networks, drawing the throughput behavior under variation of connectivity, interference and number of sources; on the other side we propose Stochastic Automata Networks (SAN) as a formalism well suited to the area because of its modularity, allowing model reuse and the structured stepwise representation of important aspects of the reality, and due to its ability to deal with models of considerable sizes, allowing the analytical evaluation of more complex situations.

1 Introduction

Computer networks is one of the areas of Computer Sciences in which performance evaluation is a mandatory step for the analysis of existing or planned distributed algorithms and protocols. It is often critical to assess performance figures for, e.g., scalability, throughput, delays, and packet loss rates, still during the development phase of the solutions. In fact, the indication that one or various parameters are out of expected ranges may imply in partial or total modification of the solution.

Ad hoc wireless networks provide infrastructure-free communication well suited for dynamic situations. Due to their intrinsic complexity, ad hoc wireless networks provide a wide range of important aspects to be addressed through performance evaluation, such as: the IEEE 802.11 MAC protocol [18]; the interaction of IEEE 802.11 MAC protocol and forwarding [7, 9]; routing mechanisms [1]; the effect of the size of the network in the average throughput; the impact of mobility patterns in the overall performance [4]; the influence of the locality of data transfers in the average throughput; the efficiency of quality of service mechanisms [12, 17]; among others.

The performance evaluation of wireless networks is often restricted to simulation studies - many of them using NS (Network Simulator) [16] - or Markov Chains (MC) [15] models that partially describe the problems [5, 11]. MC models provide reliable performance indices that are statistically significant, which many times is the main default of simulation studies. In this paper, we propose the use of another Markovian formalism, the Stochastic Automata Networks (SAN) formalism [14] to benefit of such statically-proven results. However, unlike MC, the SAN provide modular description of complex (and sometimes large) systems allowing the modeling of problems with nearly the same description power as the simulation techniques.

If compared to simulation techniques, the use of an analytical approach leads to the major known advantages found in performance evaluation, i.e., more confidence in the results. The modular approach allowed by SAN brings further advantages that lie not only in the size of the models that can be analyzed, if compared to MC, but also in the possibility of stepwise representing aspects of the reality of interest and analyzing the numerical results associated [8, 2]. In fact, the use of the SAN formalism attenuates the state space problem by furnishing efficient numerical methods to compute performance indices.

In this paper, we show the use of the SAN formalism to model wireless ad hoc networks and to obtain performance indices using analytical methods. Therefore, the proposed solution can furnish performance indices similar to those obtained by any simulation software tool. Actually, we compute stationary and/or transient solutions of the models in order to obtain the indices of interest.

We focus our study in the interaction of the IEEE 802.11 MAC protocol and packet forwarding in various configurations. More specifically, we investigate the throughput of ad hoc networks under variation of connectivity, interfer-

*The order of authors is merely alphabetical.
†Corresponding author. Paulo Fernandes is supported by CNPq/Brazil.
ence and number of relay nodes. In this contribution we do not consider the movability of nodes. This analysis is left for further studies. As an initial contribution, we show the performance analysis of a basic configuration already studied through simulation in [11]. The second contribution of this paper is the analysis of more complex configurations which are derivations of the basic one. This second part of our contribution is also didactic in the sense that it shows the reuse and extension of the basic configuration to perform the stepwise representation and analysis of the reality in investigation.

This paper is organized as follows: the SAN formalism is presented in Section 2; Section 3 discusses details of ad hoc networks relevant for this paper and introduces the performance evaluation scenarios; in Section 4, the basic SAN model of a chain of ad hoc nodes is presented, and then extensions to investigate the impact of interference among nodes are discussed; the numerical results of the basic model and extensions are discussed in Section 5; and Conclusion assesses future work and emphasizes this paper main contributions.

2 Stochastic Automata Networks

The SAN formalism was proposed by Plateau [13] and its basic idea is to represent a whole system by a collection of subsystems with an independent behavior (local transitions) and occasional interdependencies (functional rates and synchronizing events). The framework proposed by Plateau defines a modular way to describe continuous and discrete-time Markovian models [14]. However, only continuous-time SAN will be considered in this paper, although discrete-time SAN can also be employed without any loss of generality.

The SAN formalism describes a complete system as a collection of subsystems that interact with each other. Each subsystem is described as a stochastic automaton, i.e., an automaton in which the transitions are labeled with probabilistic and timing information. Hence, one can build a continuous-time stochastic process related to SAN, i.e., the SAN formalism has exactly the same application scope as Markov Chain (MC) formalism [15, 6]. The state of a SAN model, called global state, is defined by the cartesian product of the local states of all automata.

There are two types of events that change the global state of a model: local events and synchronizing events. Local events change the SAN global state passing from a global state to another that differs only by one local state. On the other hand, synchronizing events can change simultaneously more than one local state, i.e., two or more automata can change their local states simultaneously. In other words, the occurrence of a synchronizing event forces all concerned automata to fire a transition corresponding to this event. Actually, local events can be viewed as a particular case of synchronizing events that concerns only one automaton.

Each event is represented by an identifier and a rate of occurrence, which describes how often a given event will occur. Each transition may be fired as result of the occurrence of any number of events. In general, nondeterminism among possible different events is dealt according to Markovian behavior, i.e., any of the events may occur and their occurrence rates define how often each one of them will occur. However, from a given local state, if the occurrence of a given event can lead to more than one state, then an additional routing probability must be informed. The absence of routing probability is tolerated if only one transition can be fired by an event from a given local state.

The other possibility of interaction among automata is the use of functional rates. Any event occurrence rate may be expressed by a constant value (a positive real number) or a function of the state of other automata. In opposition to synchronizing events, functional rates are one-way interaction among automata, since it affects only the automaton where it appears.

![Figure 1. Example of a SAN model](image)

Fig. 1 presents a SAN model with two automata, four local events, one synchronizing event, and one functional rate. In the context of this paper, we will use roman letters to identify the name of events and functions, and greek letters to describe constant values of rates and probabilities.

In the model of Fig. 1, the rate of the event $c_1$ is not a constant rate, but a functional rate $f$ described by the SAN notation\(^1\) employed by the PEPS tools [3]. The functional rate $f$ is defined as:

$$f = \begin{cases} \lambda & \text{if automaton } A^{(2)} \text{ is in the state } 0^{(2)} \\ 0 & \text{if automaton } A^{(2)} \text{ is in the state } 1^{(2)} \\ \gamma & \text{if automaton } A^{(2)} \text{ is in the state } 2^{(2)} \end{cases}$$

The firing of the transition from $0^{(1)}$ to $1^{(1)}$ state occurs with rate $\lambda$ if automaton $A^{(2)}$ is in state $0^{(2)}$, or $\gamma$ if automaton $A^{(2)}$ is in state $2^{(2)}$. If automaton $A^{(2)}$ is in state

\(^1\)The interpretation of a function can be viewed as the evaluation of an expression of non-typed programming languages, e.g., C language. Each comparison is evaluated to value 1 (true) and value 0 (false).
1(2), the transition from 0(1) to 1(1) state does not occur (rate equal to 0). It is important to observe that the use of functions allows a compact and flexible way to describe in one single event (local or event) alternative behaviors.

Figure 2. Equivalent Markov Chain to the SAN model in Fig. 1

Fig. 2 shows the equivalent MC model to the SAN model in Fig. 1. Assuming the state 0(1)1(2) as an initial state, only 5 of the 6 states in this MC model are reachable. In order to express the reachable global states of a SAN model, it is necessary to define a (reachability) function. Although the reachable states could also be computed analysing all possible firing sequences, starting from a given initial state. For the model in Fig. 1, the reachability function must exclude the global state 1(1)1(2), thus:

Reachability = ! [ (st A(1) == 1(1)) && (st A(2) == 1(2)) ]

The use of functional expressions is not limited to event rates. In fact, routing probabilities also may be expressed as functions. The use of functions is a powerful primitive of SAN, since it allows to describe very complex behaviors in a very compact format. The computational costs to handle functional rates has decreased significantly with the developments of numerical solutions for the SAN models, e.g., the algorithms for generalized tensor products [3].

3 Case Study

In this section, we discuss characteristics of the 802.11 protocol standard [10] used in the case study. In Section 3.2, the case study is discussed.

3.1 IEEE 802.11 MAC

As defined in the protocol standard [10], the 802.11 MAC is composed of two different access methods, called: DCF (Distributed Coordination Function); and PCF (Point Coordination Function). DCF is the basic mechanism of the standard, whereas PCF is an optional mechanism which aims to support real-time traffic. In this work we focus in the DCF access method.

The DCF mechanism uses a CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) algorithm to control the access to the shared medium. Therefore, when a mobile node wants to transmit a packet, the node senses the medium for a given period (defined by the standard). If the medium is idle for such period, it can send the packet. Otherwise, it must adopt an exponential backoff scheme to transmit the packet (or re-transmit a collided packet). Moreover, DCF presents two techniques for packet exchange. The default technique is a two-way handshake, where the data packet is transmitted and an ACK (acknowledgement) is sent by the receiver if a successful reception occurred. The other technique is a four-way handshake, and is described in further details since it is the technique used in the case study.

Fig. 3 shows the four-way handshake technique, also called RTS/CTS (Request To Send/Clear To Send). Basically in this technique, when a mobile node wants to transmit a packet it sense the medium for a DIFS (Distributed InterFrame Space) period. Case the medium is idle for this period, a RTS packet is sent to the receiver. Otherwise, the node continues to sense the medium until it is idle for a DIFS period and generates a random backoff interval before transmitting the RTS packet. The backoff interval corresponds to a uniformly chosen value in the range of (0, w – 1) that is multiplied by the slot time size \( \sigma \). The value \( w \) is called contention window. When trying to transmit the first time, \( w \) is set equal to \( CW_{min} \) (minimum contention window). After every unsuccessful transmission \( w \) is doubled until it reaches the value \( CW_{max} \). Both values \( \sigma \), \( w \), \( CW_{min} \), and \( CW_{max} \) are PHY-specific (Physical Layerspecific) and are given by the standard [10].

Figure 3. Four-way handshake using RTS/CTS

After successfully transmitting the RTS packet, the receiver waits for a SIFS (Short InterFrame Space) period and responds with a CTS packet. When receiving the CTS packet, the transmitter waits for a SIFS period and sends the data packet. Finally, after receiving the data packet and waiting for a SIFS period, the receiver sends an ACK
packet to the transmitter ending the exchange. In this technique, both RTS and CTS packets carry an information of the amount of time the medium will be busy. This information is used by other mobile nodes to update their NAV (Network Allocation Vector) with the amount of time the medium will be busy. This way, if a mobile node is hidden from either the transmitter or receiver, when detecting a RTS or CTS packet being transmitted, it can avoid collision by setting the NAV [5].

3.2 Chain of Mobile Nodes

In [11], the authors examine interactions of the 802.11 MAC and ad hoc forwarding using various experiments. In order to evaluate the 802.11 standard for ad hoc networks using an analytical method, here we resemble one of the experiments presented in [11] using SAN. This experiment aims to obtain the throughput of a chain of mobile nodes. To carry out this experiment, we modeled a chain of six nodes as depicted in Fig. 4. The first node (node 1) generates packets as fast as the standard allows. The packets are forwarded through the chain to the last node (node 6).

For this experiment, the authors in [11] used the NS to simulate the model. Moreover, they also compared simulation results against same parameters obtained through monitoring a real system, which was configured according to the simulation model. One of the advantages of using the same experiment is that we can compare our analytical results with the ones obtained. Furthermore, we need to use the same parameters for the 802.11 standard used in the simulation with NS.

Thus, we used the DSSS (Direct Sequence Spread Spectrum) PHY specified by the 802.11 standard. Accordingly, the parameters of the 802.11 MAC are: DIFS = 50μs; SIFS = 10μs; σ = 20μs; CW_{min} = 32; CW_{max} = 1024. In addition, we also used the parameters for transmission and interference ranges used in [11], that are 250 and 550 meters respectively. Fig. 4 exemplifies the MAC interference and transmission ranges in the chain. Solid circles indicate mobile node’s transmission ranges, and dotted circles indicate mobile node’s interference ranges. This way, in this model, when node 4 starts to transmit, no other node can be transmitting a packet. This happens because node 4 can successfully transmit up to nodes 3 and 5. But its transmissions can interfere up to nodes 2 and 6. Thus, if node 1 tries to transmit when node 4 is transmitting, the transmission will fail because of the interference range.

4 Modeling

In this section, we show the SAN model for a chain of mobile nodes. After this, in Section 4.2, we propose extensions of this model to investigate phenomena of interest. The discussion of results achieved and their accuracy is given in Section 5.

4.1 Basic Model - Chain of Mobile Nodes

In this section, we define a SAN model to represent the chain of mobile nodes described in Section 3.2. For modeling the case study, each node of a chain of N nodes is represented by an automaton. For instance, we present in Fig. 5 a SAN model with N = 6 mobile nodes composing the chain. In this model, we define three different types of automata, called: source, relay, and sink.

![Figure 4. MAC interference among a chain of mobile nodes](image)

For this experiment, the authors in [11] used the NS to simulate the model. Moreover, they also compared simulation results against same parameters obtained through monitoring a real system, which was configured according to the simulation model. One of the advantages of using the same experiment is that we can compare our analytical results with the ones obtained. Furthermore, we need to use the same parameters for the 802.11 standard used in the simulation with NS.

Thus, we used the DSSS (Direct Sequence Spread Spectrum) PHY specified by the 802.11 standard. Accordingly, the parameters of the 802.11 MAC are: DIFS = 50μs; SIFS = 10μs; σ = 20μs; CW_{min} = 32; CW_{max} = 1024. In addition, we also used the parameters for transmission and interference ranges used in [11], that are 250 and 550 meters respectively. Fig. 4 exemplifies the MAC interference and transmission ranges in the chain. Solid circles indicate mobile node’s transmission ranges, and dotted circles indicate mobile node’s interference ranges. This way, in this model, when node 4 starts to transmit, no other node can be transmitting a packet. This happens because node 4 can successfully transmit up to nodes 3 and 5. But its transmissions can interfere up to nodes 2 and 6. Thus, if node 1 tries to transmit when node 4 is transmitting, the transmission will fail because of the interference range.

![Figure 5. Chain of nodes - SAN model (N = 6)](image)

The source automaton (MN^(1)) represents the first node of the chain and is responsible for generating packets. It
has two states: $I^{(1)}$ (idle) and $T^{(1)}$ (transmitting). The transition from state $I^{(1)}$ to $T^{(1)}$ represents the generation of packets from node 1 to 2 and it occurs through the firing of (synchronizing) event $g_{1,2}$. The transmission rate of a packet from node 1 to 2 is described by event $t_1$. Event $t_1$ leads the source automaton from state $T^{(1)}$ to $I^{(1)}$.

The relay automata ($MN^i(N)$, $i = 2 \ldots N - 1$) are responsible for forwarding the generated packets from node 1 through the chain. Each relay automaton has three states: $I^{(i)}$ (idle), $R^{(i)}$ (receiving) and $T^{(i)}$ (transmitting). The transition from state $I^{(i)}$ to $R^{(i)}$ represents the reception of a packet that must be forwarded to the next node in the chain. Such transition occurs through the firing of (synchronizing) event $r_{i-1,i}$. Hence, the transmission (forwarding) of the received packet is defined by the transition from state $R^{(i)}$ to $T^{(i)}$, which occurs by firing of (synchronizing) event $r_{i,i+1}$. The transition from state $T^{(i)}$ to $I^{(i)}$ signifies the transmission rate of the forwarded packet, described by event $t_i$.

The sink automaton ($MN^N(N)$) represents the last node of the chain. Unlike the source automaton, it also has two states: $I^{(N)}$ (idle) and $R^{(N)}$ (receiving). The reception of a packet from chain is indicated by transition from state $I^{(N)}$ to $R^{(N)}$ and it occurs by (synchronizing) event $r_{N-1,N}$. The transition from state $R^{(N)}$ to $I^{(N)}$ occurs by (synchronizing) event $t_{N-1}$, since the sink automaton must be able to immediately receive new packets from the relay automaton $MN^{(N-1)}$.

### 4.1.1 Assignment Parameters

The proposed model represents the following main aspects of the reality:

- packet generation and forwarding rates;
- transmission and interference ranges between mobile nodes;
- different packet sizes; and
- packet consumption.

Since we want to model the maximum usage of the network, we consider that packets are generated having a minimum time interval between successive packets, i.e., DIFS time. Bringing this value to the model, we transform it to the frequency of generation of packets once the source automaton is on the $I$ state or relay automata are on the $R$ state. Thus, the rates assigned to events $g_{1,2}$ and $r_{i,i+1}$ is $DIFS = \frac{1}{90+504+256+240} \text{ms} = 20.000$ times per second.

The number mentioned is the value assumed by $\lambda$ in the functions of Fig. 5. However, since we model the transmission and interference ranges, such transitions are not always allowed to occur. The communication between stations on the same transmission range is modeled by synchronizing events. For instance, $g_{1,2}$ in automata $MN^{(1)}$ and $MN^{(2)}$ or $r_{2,3}$ in automata $MN^{(2)}$ and $MN^{(3)}$. Moreover, the interference range is modeled by functions. For example, $r_{4,5}$ is modeled by function $f_{4,5}$. This function states that automata $MN^{(2)}$, $MN^{(3)}$ and $MN^{(6)}$ have not to be in $T$ state such that the source automaton may generate packets according to the rate $\lambda$ (as discussed at the end of Section 3.2).

The rate of events $t_i$ ($i = 1 \ldots N - 1$) is obtained considering how long a mobile node stays in state $T^{(i)}$, i.e., how much time is needed to accomplish the transmission of a packet. The successful transmission of a packet, as presented in Fig. 3, takes:

- Interframe size: $(3*\text{SIFS}) + \text{DIFS} = (3*10)+50 = 80\mu\text{s}$;
- Data overhead: $\text{RTS} + \text{CTS} + \text{ACK} + \text{MAC} = 126$ bytes, considering bandwidth 2 Mbps, 504$\mu\text{s}$;
- Data packet: 64 bytes, 500 bytes, or 1500 bytes, considering bandwidth 2 Mbps, respectively 256$\mu\text{s}$, 200$\mu\text{s}$, and 600$\mu\text{s}$;
- Backoff time$^2$: 240$\mu\text{s}$.

Therefore, we will assume the transmission rate in transmissions per second as:

- $\frac{10^6\mu\text{s}}{(80+504+256+240)\mu\text{s}} = 925.9259$ for 64 bytes packet size;
- $\frac{10^6\mu\text{s}}{(80+504+2000+240)\mu\text{s}} = 354.1076$ for 500 bytes packet size; and
- $\frac{10^6\mu\text{s}}{(80+504+6000+240)\mu\text{s}} = 146.5416$ for 1500 bytes packet size.

### 4.2 Extended Model

The proposed model is quite simple and the information extracted from its analysis, although accurate, is limited (see Section 5.1). As stressed in the introduction one of the main advantages of SAN is the possibility of dealing with modular analytical models. This characteristic is highly desirable from the modelers perspective, since it allows a stepwise reasoning about the reality, as well as the reuse and extension of existing models. To exemplify such

---

$^2$In our model the backoff mechanism is not modeled considering all details involved. Instead, we consider an average backoff time obtained assuming a 50% probability of success. Obviously, the probability of success will depend on the probability of congestion which is not known before evaluating the model.
extensions, we prospect enhancements to the basic model presented in Section 4.1.

In the extended model presented in this section, we consider the performance evaluation of the chain of mobile nodes under the interference of another chain. Basically, we add to the model a new stream of communication, i.e., a new pair of source and sink automata is added, along with a chain of relay automata. This modification is depicted in Fig. 6, where two chains of mobile nodes are shown. The first chain consists of a stream from left to right, whereas the second chain (added to the model and below the first chain) adds a stream from right to left. In order to compare with the results of the basic chain of nodes, we set the size of the chain to be \( N = 6 \). This value was chosen because adding more nodes to the basic chain has almost no impact to the throughput (see Fig. 11).

In Section 5.2, we use data packets of size 64, 500, and 1500 bytes to obtain the performance results. We also use the parameters for the basic model as discussed in Section 4.1. Moreover, in order to model the interference, we modify the basic model of Fig. 5 considering different states for the functions defined.

For the first scenario shown in Fig. 7, only mobile nodes that are approximated can interfere with each other. We call this a coarse interference between mobile nodes from both chains.

In the second scenario presented in Fig. 8, we consider that mobile nodes (relay automata) that are approximated as pairs are closer to the chain, and can interfere more than in the first scenario, we call this a fine interference.

In the definition of the configurations, we adopt the following notation. Above the mobile nodes we present the nodes that can interfere with the node’s communication.

The interference is shown inside “()”, where the first part presents the interference of nodes in the same chain and the second part (if it exists), separated by a “;”, presents the interference of nodes from the other chain. For instance, in the coarse interference scenario in which only one pair of nodes is approximated, the second mobile node of the first chain has interference from nodes one, four and five of the same chain, and node five of the other chain, i.e., \((1, 4, 5; 5)\).

Finally, in the third scenario, we consider data streams sharing same nodes (Fig. 9). In order to compare sharing with interference, we model analogous scenarios with one, two, three, and four nodes being shared, instead of pairs of interfering nodes. This increases the interfer-
ence range of the chains even more, since nodes are shared. Moreover, the relay automata of the shared scenario has to be modified with the addition of new states and definition of functions.

Figure 9. Automata configurations for scenario with shared mobile nodes

Basically, the extended relay automaton for the shared scenario has to forward packets in two streams of communication. In order to deal with this functionality, we define the extended relay automaton as a product of a pair of basic relay automata. The result is shown in Fig. 10.

Figure 10. Extended relay automaton for shared mobile nodes

When generating the extended relay automaton, we eliminate states and transitions that are not reachable in the model. This way, the state $T^{(2)}T^{(5)}$ (obtained from the product of two basic relay automata $MN^{(2)}$ and $MN^{(5)}$) is not considered. Besides, transitions from states in which a relay automaton transmits to another state whose the automaton continues in the same (transmit) state are also eliminated (e.g., transition from state $T^{(2)}I^{(5)}$ to state $T^{(2)}R^{(5)}$). Using this approach, in case we add another stream of communication to the model, we would have the product of three relay automata. Of course we would have to eliminate some states not reachable (and some transitions), reducing the state space of the generated model.

5 Numerical Issues

First we show the results achieved with the basic model. After that, we discuss the results of the modeled extensions.

5.1 Numerical Validation of the Basic Model

Fig. 11 shows the throughput observed with the analytical model proposed in this paper considering data packet sizes of 64, 500, 1500 bytes and size of the chain varying from 2 to 6 and 10 nodes.

Figure 11. Throughput achieved using the proposed SAN model

In [11] compatible performance figures were obtained for the same experiments using simulation and real hardware. In a more detailed analysis, the throughput curves observed in our model are closer to the ones observed in the real hardware experiments than the simulated ones. Nevertheless, our obtained results seems to be slightly higher than the ones from [11]. This difference may be caused by the backoff estimation assumed in our model, but we cannot establish further conclusions without exact values that are not provided in [11].
5.2 Numerical Results of the Extended Model

Now we present the throughput observed with the three scenarios of extended models. The reader should note that in the graphics presented in this section (see Fig. 12, Fig. 13, and Fig. 14), the start point of the plotted curves is the throughput result of the basic model, i.e., the message passage with two chains without any interference (zero node pairs).

Figure 12. Throughput achieved using the proposed SAN extended models with packets of 64 bytes

These graphics show the throughput of the first chain. In fact, in the conclusion a small discussion about the differences of the throughput for the first and the second chain is discussed.

In all analytical evaluations using the different extended models and configurations, the throughput achieved in the chain decreases with the increase of interference. As expected, the throughput of the shared nodes option is lower than the throughput of the fine interference extended model. Though, the difference is not as significant as the results obtained for the coarse and the fine interference extended models.

Figure 13. Throughput achieved using the proposed SAN extended models with packets of 500 bytes

6 Conclusion

The accuracy of the model presented in Section 4.1, and discussed in Section 5.1, is verified by the similarity of the results obtained by [11]. Using this more reliable model, we have extended it to observe other phenomena. As expected, in scenarios and configurations evaluated in Section 5.2, the greater the interference between nodes the less the throughput of the chains. Moreover, the throughput achieved in the shared nodes scenario is lower than in the fine interference extended model, but as we can see from the results, they are not as low as one might expect by sharing nodes.

The contribution of our paper also lies in the new possibilities of performance prediction due to the use of SAN. Many authors in the area use NS and compare simulation results with partial MC models. We do believe that the use of SAN models allows more confident results due to the use of analytical solution to the whole or, at least, a large part of the problem. Moreover, the modularity of the formalism
allows a reasonable modeling and analysis flexibility. However, our model still suffers from the state space explosion problem. As said in the introduction, the use of the SAN formalism to model complex systems only attenuates the state space problem in the numerical point of view.

As shown for the shared nodes scenario, the addition of new streams of communication for a single relay automaton increases the complexity of the SAN model. In fact, as the number of streams increases, new states must be added to represent all possible combinations of the relay automata. Those extensions could result in intractable models. Nevertheless, the evolution of performance evaluation technologies could cope with such problems. Future work must address alternative modeling for such aspects.

Another possible future work may address the description of node mobility. As discussed in Section 2, functional rates allow subsystems (or automata) to interact. Functional rates may take into consideration the state of other automata of the SAN. Keeping the existing basic automata proposed to model wireless nodes, we could think of an additional orthogonal set of automata to model the position and mobility of the nodes of the ad hoc network (location automata). Each mobile node may be in one of many positions, each position regarded as a possible state of the corresponding location automaton. In order to consider the degree of interference suffered by a mobile node in a given position (state of the location automaton) we should write functional rates taking the location of the other nodes into consideration. The modification of the functional rate should be the only modification in the existing basic automata modeling mobile nodes. Although this is feasible, the number of global states of the model would drastically increase.

One interesting phenomenon we have found in the solution of our extended models is the different throughput rates obtained for different chains in similar situations. Fig. 15 depicts the evaluation of the coarse interference extended model using 1500 bytes data packets.

Both chains start with the results obtained from the basic model of a 6 node chain. However, as we can see from the graphic, the first (from left to right) and second (right to left) chains present different curves. In opposition, when having four pairs of approximated nodes, both streams suffer the same degree of interference. Actually, streams with a first interference within the same distance from the starting node seems to have identical behaviors. Although we have no concrete explanation for this phenomenon, one hypothesis is that the more a source node suffers interference, the higher is the impact on the throughput. However, more accurate results using real case monitoring should be obtained in order to bring further explanations for this phenomenon.

Generally speaking, the comparison of the analytical to the simulation results may back up our confidence partially obtained by the comparison with [11] and also a comparison with real wireless implementations could be foreseen as natural future work. In another aspect of future work, we may suggest the study of even more complex extensions, such as dynamic disposition of mobile nodes, in order to reveal the modeling limits of the SAN formalism, and therefore to encourage its evolution to model even more realistic ad hoc wireless networks.

References


