

STATISTICAL MODEL CHECKING of CSMA/CA in WSNs

Zohra Hmidi¹, Laid Kahloul², Saber Benhazrallah³, and Cherifa Othmane⁴

LINFI Laboratory, Computer Science Departement, Biskra University, Algeria

¹ zohra_hmidi@yahoo.fr, ² kahloul2006@yahoo.fr, ³ sbharz@yahoo.fr,

⁴ major2010_ca@hotmail.com

Abstract. CSMA/CA (carrier sense multiple access with collision avoidance) reminds the basic method used in MAC level protocol in WSNs (wireless sensors network). Several extensions were proposed to enhance this method and to deal with specific problems in WSNs as energy consumption. The complexity of this method and its criticality motivate the formal specification and verification of its basic algorithms. In the current work, we opt for a stochastic formal model which provides more realism because it provides probabilistic aspect that often dominates several random events: collisions on the transmission medium, frame loss, loss of acknowledgements, possible disconnections, and loss of nodes. Hence, it is not enough to analyse qualitative properties where the answer will be yes or no, it is more appropriate to investigate whether certain properties are safe under particular constraints and with what probability; so that one seeks to evaluate the system and its probabilistic performance. Indeed, we model the protocol using probabilistic timed automata, a formalism in which both non-deterministic and probabilistic choice can be represented. Then a qualitative and quantitative verification of this protocol is made using the statistical model checking in the UPPAAL tool.

Keywords: WSNs, CSMA/CA, Statistical Model Checking, Probabilistic Automata, UPPAAL

1 Introduction

A Wireless Sensor Network (WSN) is a distributed system of cooperating devices that perform distributed monitoring applications in a physical environment over a self-organised wireless network topology (Intana et al. 2014)(Hai 2015). Unlike traditional networks, a WSN has its own design and resource constraints. Resource constraints include a limited amount of energy, short communication range, low bandwidth, and limited processing and storage in each node. Design constraints are application dependent, so that the WSN design is inherent to the monitored environment (Yick et al. 2008). Maximizing the network lifetime is a common objective of sensor network researchers. The MAC protocols have gained a lot of importance in the last few years because of their influence on

the lifetime of sensor nodes of WSN. Major source of energy waste in WSN is basically the collision. When a transmitted packet is corrupted due to interference, it has to be discarded and the follow on retransmissions increase energy consumption. Although collisions cannot always be prevented, randomized exponential backoff rules are used in the retransmission scheme of CSMA/CA to minimize the likelihood of repeated collisions (Patil et al. 2013).

In this paper, we illustrate a modelling approach tailored to the automatic verification of CSMA/CA protocol for wireless sensor networks. Specifically, we use probabilistic timed automata to model the protocol; we then describe how model analysis can be performed through SMC (statistical model checking) implemented in the UPPAAL tool version 1.4.19 (Uppaal 2016).

The remainder of this paper is organised as follows: In the next section, we report related work, in Section 3, we introduce the syntax and the semantics of probabilistic timed automata. Section 4 explains the modelling formalism used in UPPAAL SMC and the model checking queries. In section 5, we give an informal description of the CSMA-CA contention resolution protocol. Section 6 presents the modelling approach, the probabilistic timed automata of the protocol and the probabilistic model checking results. Finally, section 7 concludes this paper.

2 Related Works

To improve the reliability of WSNs protocols, proposed approaches include: simulation, testing and formal verification. Even if simulation and testing can be used to check large scale networks and discover errors; they can never guarantee the detection of all the bugs in a system. In (Elleuch et al. 2011), several works which are based on simulation are discussed. Formal modelling and verification are used to model the behaviour of a system and to prove its correctness. There have been several attempts to verify communication protocols for wireless networks.

For modelling and verification the MAC level protocols, the work presented in (Ho et al. 1996) is one of the first works that studied the IEEE 802.11 CSMA/CA protocol (IEEE 802.11,1995). The authors of (Ho et al. 1996) used the basic scheme of the protocol where the IFS (Inter-frame space) are introduced: "IFS, Random backoff, Data, IFS, ACK". In this work, the authors used Markov chains to compute the "throughput" and the "average delay". They made also a performance analysis of the protocol where hidden terminals are considered. Hidden terminals problem occurs when some nodes could not sense a neighbour node even if it is in its scope. They define a specific probability that a station does not hear another station. The paper showed the improvement in the throughput and the reduction of the delay when the CA (collision avoidance) is exploited in the protocol.

In (Kwiatkowska et al. 2002), the authors presented a formal study of a "two-way handshake" sub-protocol of the IEEE 802.11 standard. The authors studied a case where the net is composed of two sending stations and two destination stations. They made three hypothesis: (i) there is not an EIFS (Extended Inter-

frame Space) parameter, (ii) there is no timing synchronisation function where a specific station sends periodically short frames to synchronise the clocks of all the stations, (iii) and finally, the retry limits is supposed to be infinite. In their paper (Kwiatkowska et al. 2002), UPPAAL is used to make qualitative verification of some properties, then the PRISM tool is exploited to make a performance evaluation of the protocol. The PRISM tool can be used to give answer to several questions as: "what is the probability that a station sends correctly a packet within certain deadline". In another study on performance analysis of IEEE 802.11 CSMA/CA protocol in Wireless LANs, (Zhai et al. 2004), the authors studied two metrics: the "throughput" and the "delay" at various traffic loads. Basically, each station is seen as a queuing system with a packets-arrival process and a service time distribution (here, the service time is the MAC layer service time, which can be computed as the difference between the time a packet starts to contend for transmission and the time that the packet either is acknowledged for correct reception by the intended receiver or is dropped. Hence, dropping of packet is possible here). In order to study this queuing system, they model the "exponential backoff process" as a Markov Chain (more exactly a semi-Markov chain, where the service time is considered as the sojourn time in each state). Through this modelling, the authors computed a probability distribution for the time service. As a conclusion, they demonstrated that the exponential distribution is a good model for the MAC layer service time. In (Zhang et al. 2013), the authors used stochastic timed automaton in the study of WSNs protocols. As a case study, they considered the Timing-sync Protocol for Sensor Networks (TPSNs) (Ganerival et al. 2003).

The authors, of (Ganerival et al. 2003), used timed automata to describe the protocol in an ideal context where no messages can be lost and no node will be in fail. Based on this specification, the authors used UPPAAL to verify some qualitative properties of the TPSNs protocol. Then, the authors enrich this specification by introducing stochastic transitions, either to model losing of messages or to model damaged nodes. In (Ajith et al. 2014), the authors applied Model Driven Software Engineering (MDSE) to develop protocols dedicated to Wireless Sensors Actuators Networks. The use of MDSE aims to produce, in a short time, reliable protocols code. The authors used CPN (Coloured Petri Nets) and CPN-tool (Jensen et al. 2007) to realise a first model of the protocol then they exploit an extension of the PetriCode (Simonsen et al. 2013) tool to generate the code. This last approach can be used to produce the code for WSN platforms (TinyOS (Levis et al. 2005)) or to produce scripts for simulators (TOSSIM (Lewis et al. 2003), MiXiM (Köpke et al. 2008)).

In (Ballarini et al. 2006) a comparative analysis of the IEEE 802.11 protocol for wireless ad-hoc network and the S-MAC protocol is developed. In this work, DTMC (Discrete Time Markov Chain) models referring to a specific three-hop topology are developed with PRISM (Prism 2016). The authors have used the PCTL (Probabilistic Computation Tree Logic) logic to express relevant properties: probabilistic reachability, time-bounded probabilistic reachability and expected reachability properties are verified. In (Marta et al. 2002), the authors

studied the DCF (Distributed Coordination Function) protocol, in a fixed network topology consisting of two sending stations and two destination stations, using probabilistic timed automata. The probabilistic choice, in this work, is presented in the randomized backoff procedure. The authors in (Marta et al. 2002) used PRISM to automatically verify the probability of a station delivering a packet before a deadline.

Finally, (Hammal et al. 2014) proposed a new variant of the CSMA/CA with DCF mode. In this variant, each station has to disconnect whenever its signal-to-noise ratio is lower than a specific threshold. Such disconnections are intended to reduce the number of collisions and to improve the transmission rate. The authors in (Hammal et al. 2014) used the timed automata in UPPAAL and they analysed some safety and liveness properties, such as the absence of a deadlock and the successful termination of a transmission. However, the model checking shows a peculiar case where the protocol could not guarantee the successful transmission of packets. Furthermore, no upper bound exists for the required delay to successfully transmit a packet, and no performance evaluation was done.

In our work, we are interested to the qualitative/quantitative verification of IEEE 802.11 MAC protocol in particular, the variant proposed in (Hammal et al. 2014). We consider stochastic events that can occur during the execution of the protocol. These stochastic events concern failures to access the medium, collisions, messages lose, the choice made by a station to send or to receive, etc. In order to handle these stochastic aspects, Probabilistic Timed Automata are used as a modelling formalism and UPPAAL is exploited to check and to evaluate the protocol.

3 Probabilistic timed automata

Probabilistic timed automata (Kwiatkowska et al. 2001) are a modelling formalism for distributed systems that support dense time, non-determinism, and probabilistic choice. They represent an extension of timed automata, for which discrete probability distributions range over the edges of the control graph.

3.1 Clocks and Zones

Let \mathcal{X} be a finite set of variables called clocks, ranging over the time domain $\mathbb{T} \in \{ \mathbb{R}, \mathbb{N} \}$ of either non-negative real or natural numbers. A function $v : \mathcal{X} \rightarrow \mathbb{T}$ is referred to as a clock valuation. For any v and $t \in \mathbb{T}$, the clock valuation $v \oplus t$ denotes the time increment for v with t . Let $Zones(\mathcal{X})$ be the set of zones over \mathcal{X} , which are conjunctions of atomic constraints of the form $x \sim c$ for $x \in \mathcal{X}$, $\sim \in \{ \leq, =, \geq \}$, and $c \in \mathbb{N}$. The clock valuation v satisfies the zone ζ , written $v \triangleleft \zeta$, if and only if ζ resolves to true after substituting each clock $x \in \mathcal{X}$ with the corresponding clock value from v .

3.2 Syntax of probabilistic timed automata

A probabilistic timed automaton is a tuple $(L, \bar{l}, \mathcal{X}, \Sigma, inv, prob)$ where:

- L is a finite set of locations;
- $\bar{l} \in L$ is the initial location;
- \mathcal{X} is a finite set of clocks;
- Σ is a finite set of events, such that $\Sigma_u \subseteq \Sigma$ are urgent;
- $inv: L \rightarrow Zones(\mathcal{X})$ is the invariant function associated for each location;
- $prob \subseteq L \times Zones(\mathcal{X}) \times \Sigma \times \text{Dist}(2^{\mathcal{X}} \times L)$ is the probabilistic transition relation. $\text{Dist}(2^{\mathcal{X}} \times L)$ denotes the couple: (probability intensity over the transition, the target location).

3.3 Semantics of probabilistic timed automata

A state of a probabilistic timed automaton is a pair (l, v) where $l \in L$ and $v \in \mathbb{T}^{|\mathcal{X}|}$ are such that $v \triangleleft inv(l)$ (i.e. v satisfies $inv(l)$). The model starts in the initial location \bar{l} with all clocks set to 0, and hence the initial state is $(\bar{l}, 0)$. In each location, there is a nondeterministic choice between two types of transitions: Delay transitions which correspond to the elapsing of time in a location. They are permitted as long as the invariant condition is satisfied and no urgent transitions are enabled. Discrete transition correspond to the execution of probabilistic transitions $(l, g, \sigma, p) \in prob$. If the current location l satisfies the clock constraint g and the current event is σ , then $p(X, l')$ is the probability of resetting all clocks in $X (X \subseteq \mathcal{X})$ to 0 and moving to the location l' .

4 SMC in the UPPAAL tool

Statistical Model Checking (SMC) (Koushik et al. 2004) is an innovative approach proposed as an alternative to avoid the exhaustive exploration of the state-space of the model. SMC is a simulation-based solution, which is less time and memory intensive than classical model checking. The idea behind SMC is to generate enough sample execution paths for the system and then to use the statistical hypothesis testing to decide whether the system satisfies the given property or not. Using SMC, one can verify qualitative properties (as in classical model-checking) and make quantitative evaluation too. UPPAAL SMC (David et al. 2015) is an extension of UPPAAL, proposed to represent systems via networks of priced timed automata (David et al. 2011) whose behaviours may depend on both stochastic and non-linear dynamical features. Each component of the system is described with an automaton whose clocks can evolve with various rates. Such rates can be specified with, e.g., ordinary differential equations.

4.1 Modelling

The modelling formalism of UPPAAL SMC is based on probabilistic timed automata. The stochastic interpretation replaces the non-deterministic choices between multiple enabled transitions by probabilistic choices (that may or may not be user-defined). The non-deterministic choices of time-delays are refined by probability distributions, which at the component level are given either uniform distributions in cases with time-bounded delays or exponential distributions in cases of unbounded delays.

4.2 Query Language

UPPAAL SMC supports five different analysis methods. Below we use N to denote natural number, M to denote number of simulations, P to denote a probability, and exp to denote an expression:

- Statistical evaluation (Hérault et al. 2004): SMC estimates the probability of the state property being satisfied, using the following query:

$$Pr[\leq N](\text{" } \langle \rangle \text{ } | \text{" } exp)$$

- Hypothesis testing (Younes 2005): SMC checks if the property is satisfied within a certain probability, using the following query:

$$Pr[\leq N](\text{" } \langle \rangle \text{ } | \text{" } exp) \leq | \geq P$$

- Statistical comparison: SMC compares the satisfaction possibilities over two properties, using the following query:

$$Pr[\leq N_1](\text{" } \langle \rangle \text{ } | \text{" } exp_1) \leq | \geq Pr[\leq N_2](\text{" } \langle \rangle \text{ } | \text{" } exp_2)$$

- Expected value: SMC computes the maximal or the minimal value of a certain variable while checking the system, using the following query:

$$E[\leq N; M](\text{" } min|max \text{" } : exp)$$

- Simulations: SMC simulates a system multiple times and computes trajectories of specified expressions over time, using the following query:

$$simulate M[\leq N]\{exp_1, exp_2\}$$

5 The Contention Resolution Protocol CSMA-CA

After having presented the necessary formalisms, in the above paragraphs, the current section aims to present informally the protocol IEEE 802.11 which will be our case study.

The primary MAC scheme of the standard IEEE 802.11 is called Distributed Coordination Function (DCF)(Ballarini et al. 2006). It describes a decentralised mechanism which allows network stations to coordinate for the use of a medium in an attempt to avoid collision. Three time periods are considered: the DCF InterFrame Space (DIFS), the Short InterFrame Space (SIFS) and the Extended InterFrame Space (EIFS), where $SIFS < DIFS < EIFS$. A station can start a transmission of data packets only after sensing the medium free for a DIFS. On reception of a data packet, the destination station, after sensing the channel free for SIFS, sends an acknowledgement packet (ACK) back to the sender. A collision is recognised by the sending station if either: on termination of the transmission the channel is sensed occupied by another station, or if an ACK packet is not received within a given time. In order to avoid collisions, the MAC protocol obliges the stations to enter in backoff stage before sending if either (Marta et al. 2002):

- the channel is not sensed idle for a DIFS;
- the channel is sensed busy after the station finishes a data transmission;
- a positive acknowledgement of successful transmission is not received from the destination station before a timeout;
- the station receives an acknowledgement and wishes to send another packet.

As soon as a backoff condition becomes true, the deferring station selects a *BackoffTime* composed of a random number *backoffvalue* of slot times, where each slot has size *SlotTime*. This value indicates the number of time periods which must be passed before the station can start transmitting. If the channel is detected idle for a some *SlotTime*, the *backoffvalue* is decremented by 1. This decrementing procedure is temporarily suspended if a transmission is detected (the reduction of *backoffvalue* is frozen) and is resumed only after the channel is sensed free for DIFS time units. When the *backoffvalue* reaches 0, the station can start its transmission. The value of *backoffvalue* is a pseudo-random integer drawn from a uniform distribution over the interval $[0, CW]$, where *CW* is the Contention Window which has initial value *CWmin* and takes values of ascending powers of 2 minus 1. Thus $CW = (CWmin + 1) \times 2^{bc} - 1$ where *bc* (the *BackoffCounter*) increases with the number of consecutive unsuccessful transmissions. Note that the likelihood of a longer backoff delay for repeatedly detected collisions (where *bc* is large) is increased. The value of *CW* has an upper bound of *CWmax*. Once this value has been reached, *CW* will remain at this value until it is reset.

6 Modelling using UPPAAL

In this section, we present the formal model of the studied protocol. The model is composed, basically, of three synchronised PTA (Probabilistic Timed Automata): Wireless station (WS) model, Backoff model, and Medium Model. In the following, we present only the WS model and the Medium model.

6.1 Model of wireless station and wireless medium

When analysing the model presented in the previous work (Hammal et al. 2014), we have found several cases where the transmission operation fails forever. These fails do not reflect real anomalies in the studied protocol, but they are due to the modelling process:

1. A station may have *fail* as a response of the medium forever. In the model of the station, there is no mechanism that allows it to get out of this particular case. The station can only retry continuously sending the PDU, and the medium continues to meet fail. This is an unrealistic case, but the model proposed may make it a possible case in the protocol. Indeed, this way of simulating the fail causes the random simulations made by UPPAAL enter in infinite loops choosing each time a transition fail.

2. A station can get infinitely in a loop (*Disconnected, idle_Connected*) without advancing the protocol. Another unrealistic behaviour but a possible path in the model.
3. From the idle state, there are three possible transitions, one towards the *Disconnected* state, the second towards the *WaitingUntilTransmit* state, and the third towards the *after_begin_receivePDU* state, but this last one is not possible unless there is a synchronization (with the medium). Following this modelling, the station can always take the transition towards *WaitingUntilTransmit* and never take the third one; this leads to a deadlock problem that will be explained in the next fourth point.
4. When the station st wants to send a PDU to the station st' , along st' wants to send a PDU. Both stations are leaving their states Idle to the state *WaitingUntilTransmit*. The two stations can not return to their Idle states before the end of their transmissions. However, terminating the transmission of the st PDU requires receiving an ACK from st' . The latter can never send this ACK, because it can never return to its Idle state, for several reasons (including those described in 1 and 2 as examples).

The resolution of these deficiencies can be made in several ways. Indeed, solving the problems mentioned in 1, 2 and 3 can be achieved by making the model more realistic by introducing stochastic factors to reduce unrealistic non-determinism in the model. Disconnections, failures and collisions must happen with probabilities (rare events under normal conditions). We can also assign two different probabilities for a station, one for transmitting and one for receiving. This allows for a more realistic model. However, the problem cited in the fourth point requires the updating of the model structure in order to avoid this deadlock. Hence, the station st can return to its Idle state from the *WaitingUntilTransmit* state after passing a time limit without having access to the medium. Indeed, if st returns to its Idle state it will unlock other stations waiting its ACK.

The following figures (Figures 1 and 2) show the probabilistic models after consideration of probabilities on some transitions (fail in the model of the medium and disconnected, transmitted and received in the model of the station). In these models, we have associated a probability of 1/10 for a fail event and a probability of 2/10 for a disconnect event. For the transmission and reception we will discuss the different values.

6.2 Model verification

The statistical model-checker associated with UPPAAL 4.1.19 required to make all channels "broadcast channels" and to define the probability laws governing elapsing time for all locations with output transitions. In our model, we have associated a rate = 1 for all locations without invariants on clocks. In this work, we are interested to verify qualitative and quantitative properties using the UPPAAL SMC.

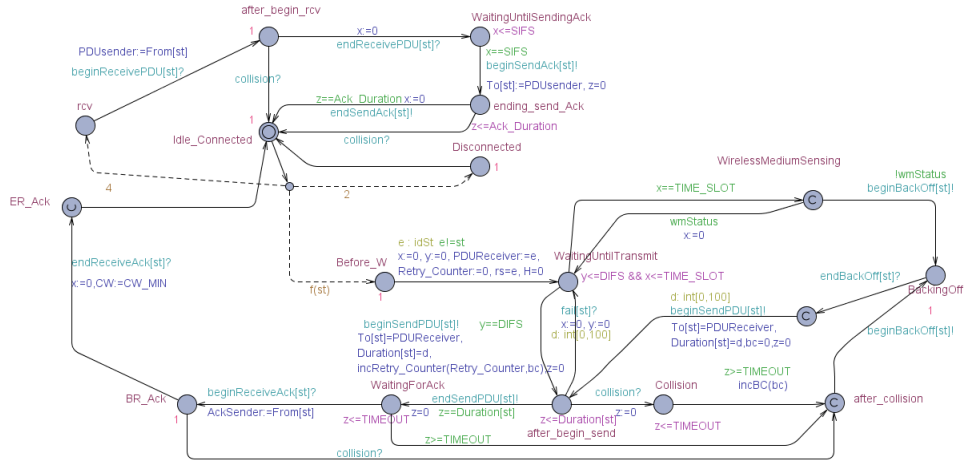


Fig. 1: Probabilistic model of wireless station

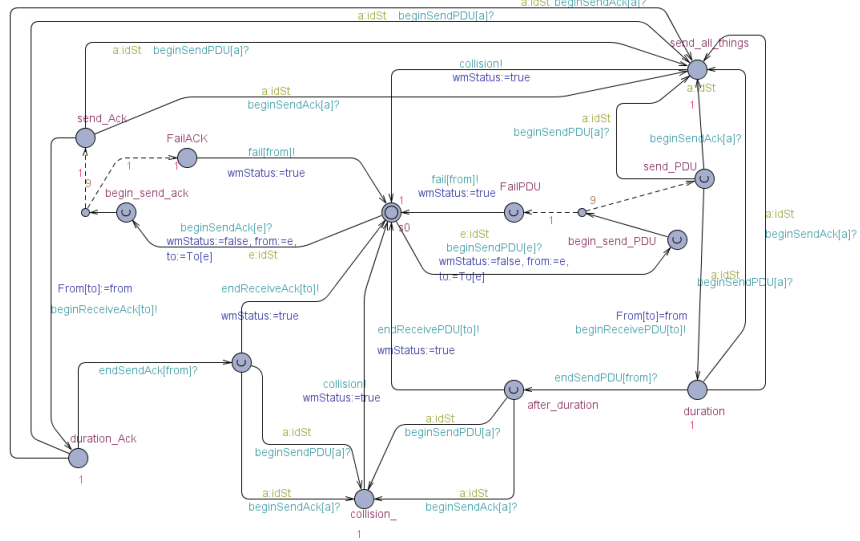


Fig. 2: Probabilistic model of wireless medium

Verification of qualitative properties. Qualitative properties can be classified into reachability, safety, liveness, and fairness.

- Reachability Properties: they are the simplest form of properties. They ask whether a given state formula, ϕ , possibly can be satisfied by any reachable state. We express that "a state satisfying ϕ " is reachable by the use of the path formula, $E \langle \rangle \phi$. For example, the formula $E \langle \rangle ws0.WaitingForAck$ determines whether the protocol makes it possible for the sender $ws0$ to send a message, and therefore it will pass to the "waiting acknowledgement" state. The figure 3.a below shows that this formula is satisfied.
- Safety Properties: they are of the form: "something bad will never happen". For example, the formula $A \square notdeadlock$ determines that the model is deadlock-free. The figure 3.b shows that this formula is not satisfied (it was discussed in Section 6.1).
- Liveness Properties: These properties are of the form: "Something will eventually happen". In the model of the communication protocol, any message that has been sent should eventually be received. In its simplest form, the liveness is expressed using the path formula $A \langle \rangle \phi$, meaning that ϕ is always eventually satisfied. For example, the station $ws0$ always reaches the state where it will expect an ACK. The figure 3.c shows that the formula $A \langle \rangle ws0.BRAck$ is not satisfied. This means that the station can pass into collision paths and never reach this state.

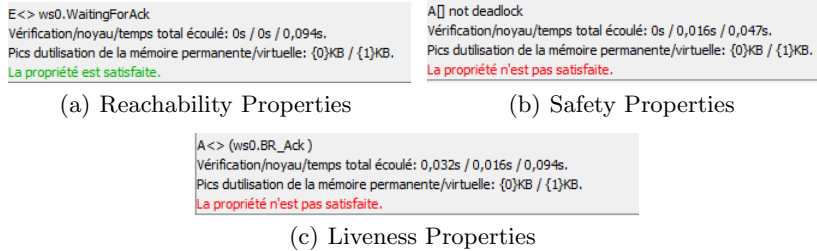
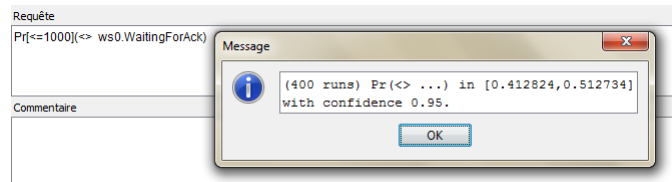


Fig. 3: Verification of qualitative properties

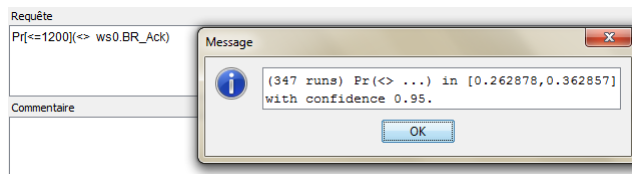
Verification of quantitative properties. In this case, the question is to estimate the probability that a certain property will be satisfied by the system prior to the violation of some constraint. In other words, we are not interested here in whether a station reaches a state or not, but to know with what rate this station will reach this state. To check the model, we added to the model of the station, a probability to be the sender and another to be the receiver. Here after, we will show the results in two different cases:

- Both probabilities are equal: we assigned a probability of 4/10 for both transitions, and we will make a set of queries as following:

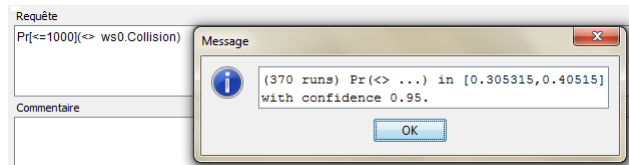
- The query: $Pr[\leq 1000](\langle \rangle ws0.WaitingForAck)$, which allows to know the probability that the station reaches a state where it has finished its transmission and waits for an ACK before 1000 units of time during the run of the protocol. Figure 4.a shows the response of UPPAAL to this request.
- The query: $Pr[\leq 1200](\langle \rangle ws0.BR_Ack)$, which allows to know the probability that the station reaches a state where it started receiving an ACK before 1200 units of time during the run of the protocol. Figure 4.b shows the response of UPPAAL to this request.
- The query: $Pr[\leq 1000](\langle \rangle ws0.Collision)$, which allows to know the probability that the station reaches a collision state before 1000 units of time during the run of the protocol. Figure 4.c shows the probability interval.



(a) sending PDU



(b) receiving ACK



(c) collision

Fig. 4: The probability confidence interval of case1

- The two probabilities are different: in the case of two stations, we assigned to each one two different probabilities. For example, for station $ws0$, we assigned a probability of $4/14$ to be the receiver and a probability of $8/14$ to be the sender. Rather, for the station $ws1$, we assigned a probability of $4/8$ to be the receiver and a probability of $2/8$ to be the sender. We will present some queries to compare the results with the first case. Figures 5.a, 5.b and

5.c show the response of UPPAAL to the following requests, respectively: $Pr[\leq 1000](\langle \rangle ws0.WaitingForAck)$, $Pr[\leq 1200](\langle \rangle ws0.BR_{Ack})$ and $Pr[\leq 1000](\langle \rangle ws0.Collision)$.

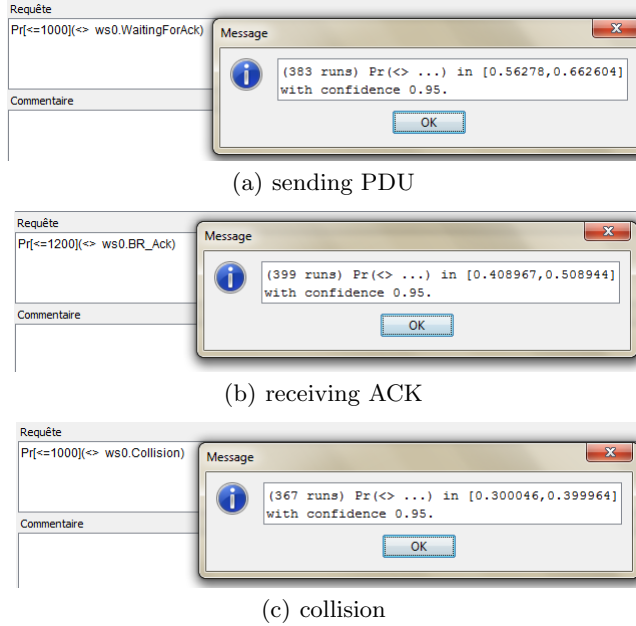


Fig. 5: The probability confidence interval of case2

If we make a comparison, we will notice that the rates have changed considerably. The rate of sending a pdu will be 66% instead of 51%, the rate of receiving of an ACK will be 50% instead of 36% and finally the collision rate will be 39% instead of 40%. Finally, 75% of sent packets are received correctly, unfortunately, the collision rate remains high.

It is also possible to plot the characteristic function of probability using the tool UPPAAL, as shown in figure 6. Based on these charts, we see that the probability of sending for ws_0 is the largest, and the collision probability is the smallest (of course, using the current parameters). In addition, the ws_1 station is less likely to perform the transmission first.

7 Conclusion

In this paper, we have described the use of statistical model checking to analyse and evaluate the WSN MAC protocol. We have modelled stochastic uncertainties which are quite common in WSN systems behaviour, like disconnections and

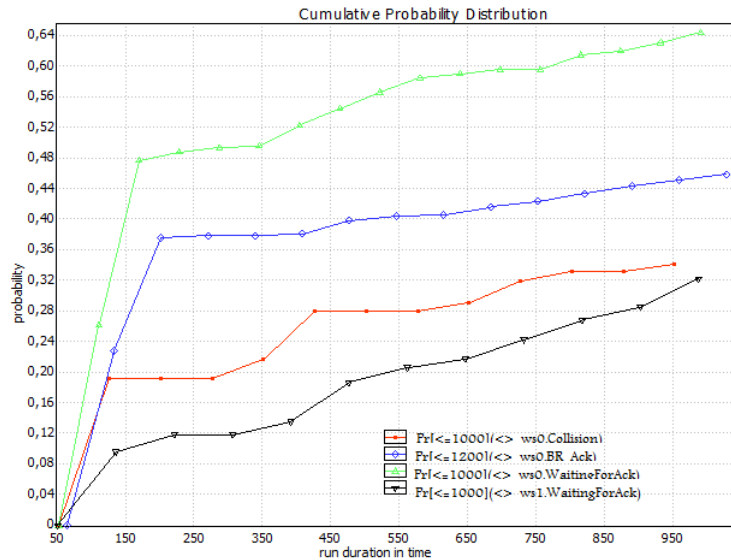


Fig. 6: The characteristic function of probability

failures using probabilistic timed automata. Besides of checking the correctness of the protocol, we have used statistical model checking, in the UPPAAL tool, to evaluate the performance of the protocol numerically as well. In a first step of performance evaluation, we have detected that there is some anomalies in the proposed model, inspired from the work (Hammal et al 2014). In a next work, we plan to make more evaluation of the protocol model, to detect all anomalies in the proposed model, to correct and update the model and then to make a complete verification process.

References

- Prism. Available (seen in 2016) at: <http://www.prismmodelchecker.org/>
- Uppaal. Available (seen in 2016) at: <http://www.uppaal.org/>
- Ballarini, P., Miller, A.: Model checking medium access control for sensor networks. In: Leveraging Applications of Formal Methods, Verification and Validation, 2006. ISoLA 2006. Second International Symposium on. pp. 255–262. IEEE (2006)
- David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B.: Uppaal smc tutorial. International Journal on Software Tools for Technology Transfer 17(4), 397–415 (August 2015)
- David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B., Van Vliet, J., Wang, Z.: Statistical model checking for networks of priced timed automata. In: Formal Modeling and Analysis of Timed Systems. p. 8096. Springer Berlin Heidelberg (2011)

- Elleuch, M., Hasan, O., Tahar, S., Abid, M.: Formal analysis of mac protocols for wsns: a review. Tech. rep. (May 2011)
- Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-sync protocol for sensor networks. In: Proceedings of the 1st international conference on Embedded networked sensor systems. pp. 138–149 (2003)
- Hai, N.V.: Formal verification of a distributed routing protocol for wireless sensor networks wire on a large scale (2015)
- Hammal, Y., Ben-Othman, J., Mokdad, L., Abdelli, A.: Formal modeling and verification of an enhanced variant of the ieee 802.11 csma/ca protocol. *Communications and networks* 16(4), 385–396 (August 2014)
- Hérault, T., Lassaigne, R., Magniette, F., Peyronnet, S.: Approximate probabilistic model checking. In: International Workshop on Verification, Model Checking, and Abstract Interpretation. pp. 73–84. Springer (2004)
- Ho, T.S., Chen, K.C.: Performance analysis of ieee 802.11 csma/ca medium access control protocol. In: Proceedings of PIMRC 96 - 7th International Symposium on Personal, Indoor, and Mobile Communications. vol. 2, pp. 407–411 (1996)
- IEEE_organisation: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. (July 1995)
- Intana, A., Poppleton, M.R., Merrett, G.V.: A formal co-simulation approach for wireless sensor network development. In: Proceedings of the 14th International Workshop on Automated Verification of Critical Systems (AVoCS 2014). vol. 70. Electronic Communications of the EASST (2014)
- Jensen, K., Kristensen, L., Wells, L.: Coloured petri nets and cpn tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer* 9, 213–254 (2007)
- Köpke, A., Swigulski, M., Wessel, K., Willkomm, D., Haneveld, P.T.K., Parker, T.E.V., Visser, O.W., Lichte, H.S., Valentin, S.: Simulating wireless and mobile networks in omnet++ the mixim vision. In: Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops (Simutools). vol. 71, p. 18 (2008)
- Koushik, S., Mahesh, V., Gul, A.: Statistical model checking of black-box probabilistic systems. In: Computer Aided Verification. pp. 202–215. Springer (2004)
- Kumar, S., Simonsen, K.I.: Towards a model-based development approach for wireless sensor-actuator network protocols. In: Proceedings of the 4th ACM SIGBED International Workshop on Design, Modeling, and Evaluation of Cyber-Physical Systems. pp. 35–39 (2014)
- Kwiatkowska, M., Norman, G., Sproston, J.: Process Algebra and Probabilistic Methods: Performance Modeling and Verification, vol. 2399. Berlin, Heidelberg: Springer Berlin Heidelberg (2002)
- Kwiatkowska, M., Norman, G., Sproston, J.: Symbolic computation of maximal probabilistic reachability. In: International Conference on Concurrency Theory. pp. 169–183. Springer (2001)
- Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., Brewer, E., Tinyos, D.C.: An operating system for sensor networks. In: Ambient Intelligence. p. 115148. Springer Berlin Heidelberg (2005)
- Lewis, P., Lee, N.: TOSSIM: A Simulator for TinyOS Networks. TinyOS documentation (2007)
- Marta, K., Gethin, N., Jeremy, S.: Probabilistic Model Checking of the IEEE 802.11 Wireless Local Area Network Protocol, vol. 2399, chap. Lecture Notes in Computer Science, pp. 169–187. Springer-Verlag Berlin Heidelberg (July 2002)

- Patil, U.A., Modi, S.V., Suma, B.: A survey: Mac layer protocol for wireless sensor networks. *International Journal of Emerging Technology and Advanced Engineering* 3(9), 203–211 (September 2013)
- Simonsen, K.I.F., Kristensen, L., Kindler, E.: Generating protocol software from cpn models annotated with pragmatics. In: *In Formal Methods: Foundations and Applications*. vol. 8195, pp. 227–242. Springer Berlin Heidelberg (2013)
- Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. *Computer Networks* 52(12), 2292–2330 (August 2008)
- Younes, H.L.: Verification and planning for stochastic processes with asynchronous events. Tech. rep., DTIC Document (2005)
- Zhai, H., Kwon, Y., Fang, Y.: Performance analysis of ieee 802.11 mac protocols in wireless lans. In: *Wirel. Commun. Mob. Comput.* vol. 4 (December 2004)
- Zhang, F., Bu, L., Wang, L., Zhao, J., Chen, X., Zhang, T., Li, X.: Modeling and evaluation of wireless sensor network protocols by stochastic timed automata. In: *Electron. Notes Theor. Computer Science*. vol. 296, pp. 261–277 (August 2013)