



HAL
open science

Performance modelling of access control mechanisms for local and vehicular wireless networks

Paolo Ballarini, Benoît Barbot, Nicolas Vasselin

► To cite this version:

Paolo Ballarini, Benoît Barbot, Nicolas Vasselin. Performance modelling of access control mechanisms for local and vehicular wireless networks. 12th EAI International Conference on Pervasive Computing Technologies for Healthcare, Mar 2019, Palma, Spain. pp.111-118. hal-02131725

HAL Id: hal-02131725

<https://hal.science/hal-02131725v1>

Submitted on 15 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PERFORMANCE MODELLING OF ACCESS CONTROL MECHANISMS FOR LOCAL AND VEHICULAR WIRELESS NETWORKS

Paolo Ballarini*
Lab. MICS
University Paris Saclay
Gif-sur-Yvette, France.
paolo.ballarini@centralesupelec.fr

Benoît Barbot
Lab. LACL
University Paris Est Créteil
Ceréteil, France
benoit.barbot@u-pec.fr

Nicolar Vasselin
CentraleSupélec
Gif-sur-Yvette, France.
nicolas.vasselin@student.ecp.fr

ABSTRACT

Carrier sense multiple access collision avoidance (CSMA/CA) is the basic scheme upon which access to the shared medium is regulated in many wireless networks. With CSMA/CA a station willing to start a transmission has first to find the channel free for a given duration otherwise it will go into *backoff*, i.e. refraining for transmitting for a randomly chosen delay. Performance analysis of a wireless network employing CSMA/CA regulation is not an easy task: except for simple network configuration analytical solution of key performance indicators (KPI) cannot be obtained hence one has to resort to formal modelling tools. In this paper we present a performance modelling study targeting different kind of CSMA/CA based wireless networks, namely: the IEEE 802.11 Wireless Local Area Networks (WLANs) and the 802.11p Vehicular Ad Hoc Networks (VANETs), which extends 802.11 with priorities over packets. The modelling framework we introduce allows for considering: i) an arbitrarily large number of stations, ii) different traffic conditions (saturated/non-saturated), iii) different hypothesis concerning the shared channel (ideal/non-ideal). We apply statistical model checking to assess KPIs of different network configurations.

1 Introduction

Communication protocols regulate the behaviour of communicating nodes within a concurrent environment. The Open System Interconnection (OSI) model [11] defines a layered architecture for network protocols. The *Medium Access Control* (MAC) layer, part of the data-link layer, determines which node is allowed to access the underlying physical-layer (i.e. the medium) at any given moment in time. A MAC scheme is mainly concerned with reducing the possibility of *collisions* (i.e. simultaneous transmissions over a shared channel) from taking place. The basic mechanism used for reducing the likelihood of collisions, usually referred to as Carrier Sense Multiple Access (CSMA), is that, before starting a transmission, any node should sense the medium clear for a given period.

IEEE 802.11 WLAN. Wireless local area networks (WLANs) are wireless networks for which either the communication is managed by a centralised Access Point (AP) or, in the case of *ad-hoc*, nodes communicate in a peer-to-peer fashion through a distributed coordination function. The IEEE 802.11 [12] is a family of standards which specifies a number of MAC schemes and the *Physical* (PHY) layer for WLANs. The primary MAC scheme of the standard is called *Distributed Coordination Function* (DCF). It describes a de-centralised mechanism which allows network stations to

*Corresponding author.

coordinate for the use of a (shared) medium in an attempt to avoid collision. The DCF is a variant of the CSMA/CA MAC scheme developed for collision avoidance over a shared medium using a randomised backoff procedure. Two variants of the DCF have been defined in the standard: the *Basic Access* (BA), which uses a single *acknowledgement* to confirm the successful reception of a data packet, and the *Request-To-Send/Clear-To-Send* (RTS/CTS), which employs a *double-handshaking* scheme so to reduce costly collisions on large data packets.

IEEE 802.11p VANET. In the realm of Intelligent Transportation Systems (ITS) the main concern is one of improving the effectiveness as well as the safety of future transportation systems. ITS entail *hybrid* communication scenarios where both Inter-Vehicle-Communication (IVC), based on *ad-hoc* connections between moving vehicles, and Roadside-Vehicle-Communication (RVC), concerned with the exchanging of information between moving vehicles and fixed roadside infrastructures, co-exist. In order to cope with specific needs of such hybrid scenarios, an adaptation of the IEEE 802.11 MAC layer, named 802.11p Wireless Access in Vehicular Environments (WAVE) standard, has been introduced for Vehicular Ad Hoc Networks (VANETs). The 802.11p MAC is based on an adaptation of the CSMA/CA scheme, called Enhanced Distributed Channel Access (ECDA) protocol, to the case of a network whereby traffic with different level of priority, called Access Categories (AC), circulates between wireless nodes.

Our contribution. In this paper we present a formal modelling study for the analysis of performances of wireless networks using RTS/CTS based MAC protocols specifically i) WLANs (i.e. 802.11) and ii) VANETs (i.e. 802.11p). To this aim we develop formal models in terms of Generalized Semi-Markov Process (GSMP) expressed through an high-level stochastic Petri net formalism, namely Stochastic Symmetric Nets (SSN) [8], which we analyse through specific performance indicators defined through the HASL properties specification language [5]. The models we developed are configurable and allow for taking into account different scenario including the network dimension, different incoming traffic conditions, the possible presence of faults affecting the channel during the transmission of some packet. For both the 802.11 and 802.11p scenarios we developed and asses KPI expressed in terms of temporal logic specifications.

1.1 Related work

Performance analysis of the IEEE 802.11 family of Wi-Fi protocols has been the subject of several research studies [7, 6, 14, 13] each of which considers specific modelling assumptions concerning e.g. the traffic model that is considered (i.e. saturated traffic, packets arrival following a Poisson law, etc), the presence/absence of errors on the channel and in case of errors how errors are modelled. In its pivotal work Bianchi [7], introduced a simple analytical model that, under given constraints (i.e. finite number of stations and ideal channel condition) allow for computing the *throughput* of the IEEE 802.11 Distribution Coordination Function (DCF) for both the Basic Access (BA) and the RTS/CTS versions of the DCF. Taking from the two-dimensional discrete-time Markov chain (DTMC) model of Bianchi, many extensions have been considered. In [2] a 4D DTMC, inspired by Bianchi's 2D model, has been introduced for considering the case of imperfect channels, that is, taking into account that transmission over a wireless medium is affected by errors. More specifically in [2] the imperfect nature of channels is modelled through a constant *bit error* probability P_b (i.e. the probability that an error occurs during the transmission of a single bit of data). In [13] the BA version of the 802.11 MAC is analysed through probabilistic model checking but taking into account only simple modeling assumption (2-nodes network dimension, ideal channel, no specific traffic model).

2 RTS/CTS carrier-sensing protocol

CSMA/CA MAC schemes are based on the simple idea that a station willing to transmit data packets has first to gain access to the channel through a *sensing phase* depending on which the station may either start transmitting, if the channel has been sensed free along the entire sensing phase, or, if the channel has been sensed busy, refrain from doing so for a randomly chosen duration (backoff). Collisions may take place whenever at least two stations ends the sensing phase at the same time, however by employing a randomised backoff delay, the probability of collisions decreases with the number of successive collisions. There are two versions of CSMA/CA scheme, the basic access (BA), which we do not consider in this paper, and the Request-to-Send/Clear-to-Send (RTS/CTS).

802.11 RTS/CTS: with the RTS/CTS DCF the transmission of data between a sender and a receiver is regulated by means of a *double handshaking* scheme which employs three small-sized, control-packets to regulate the transmission of (larger) data packets (Figure 1). The sender senses the channel for a randomly chosen duration before sending a RTS control packet to the receiver, on reception of which, the receiver, replies with a CTS control packet to the sender (first handshaking). The actual transmission of DATA packets starts as soon as the sender has received the CTS. Finally when the DATA transmission is over, the receiver acknowledges the sender with an ACK control packet (second handshaking). The RTS-CTS-DATA-ACK timing sequence is illustrated in Figure 1, which also points out the latency for a successful transmission of a DATA packet over 1-hop. To decrease the probability of collisions, the carrier-sensing time t_{CS} ,

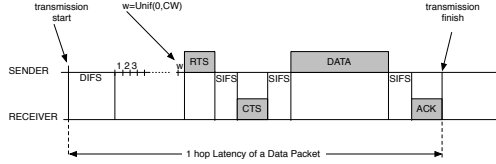


Figure 1: 1-hop carrier-sensing and packets-transmission timing in RTS/CTS 802.11

also named *Contention Window* (CW), which is discretely slotted in a finite number of slots of fixed duration $aSlot_t$ ($aSlot_t$ being a parameter of the standard whose value depends on the underlying PHY layer), is randomly chosen. A contention takes place when at least two stations are performing carrier-sensing at the same time. The one that has (randomly) chosen the shortest number of available slots in the CW, wins the contention. The loser, instead, goes into *backoff* and increases the dimension of the CW by a power of 2 (i.e. the new contention window size CW' is given by $CW' = (CW_m + 1) \cdot 2^{bc} - 1$ where bc , the *BackoffCounter*, increases with the number of consecutive unsuccessful transmissions). The minimum and maximum size of contention window, respectively CW_m and CW_M , are set by the standard and depends on the PHY layer. For the Frequency Hopping Single Spectrum (FHSS) PHY layer they are equal to $CW_{min} = 16$ (initial value of $bc = 0$) and $CW_{max} = 1024$ (maximum value of $bc_{max} = 6$). Apart from t_{CS} , two fixed-length time intervals are also relevant in the RTS/CTS DCF, namely: *Distributed InterFrame Space* (DIFS), the *Short InterFrame Space* (SIFS), where $SIFS < DIFS$ and which are defined by the PHY layer in the adopted networking stack (see Table 1). It should be noted that a collision can take place not only when two contending stations (randomly) pick the same carrier-sensing time (i.e. t_{CS}), but also, as pointed out by Heindl et al. [10], because of the existence of the so-called *vulnerable period*, which accounts for three factors: the time of radio waves propagation through the medium ($aAirPropagationTime$), the time a station takes for accessing the medium ($aCCATime$) and the time a station takes for switching from receiving to transmitting mode ($aRxTxTurnaroundTime$). As a consequence the duration of $aSlot_t$ is set by the IEEE standard to a value larger than the *vulnerable period* (i.e. the value $aSlot_t$ depends on the PHY-layer dependent parameters $aCCATime$ and $aRxTxTurnaroundTime$ plus the negligible $aAirPropagationTime$).

802.11p RTS/CTS with priority classes: The 802.11p MAC is based on an extension of the CSMA/CA RTS/CTS scheme, called Enhanced Distributed Channel Access (EDCA), which supports 4 priority levels, called Access Categories (AC), for data traffic, namely: Background (AC_BK), i.e. the lowest priority, Best effort (AC_BE), Video (AC_VI) and Voice (AC_VO), i.e. the highest priority. EDCA is designed so that higher-priority traffic is more likely (than lower-priority) to be granted access to the shared medium and successfully being transmitted. In practice this is obtained by associating each AC with a *contention window* (CW) whose size is inversely proportional to the corresponding priority level. Therefore, for example, the CW for AC_BK (lowest priority) has to be larger than that of AC_VO (highest priority). Table 2 depicts the CW characterisation for the ACs of EDCA, where aCW_{min} and aCW_{max} are parameters set by the 802.11p protocol. One can notice that AC_VO is the dominant AC as its CW is not only the minimal one but also it does not overlap to the CW of any other AC.

3 Background: stochastic symmetric Petri nets

To carry out the performance modelling study of 802.11 protocols we used a *coloured* stochastic Petri net formalism, namely the Stochastic Symmetric Net (SSN) [8], to model networks using 802.11/802.11p MAC, and we applied the Hybrid Automata Specification Language (HASL) [5] formalism (through the COSMOS statistical model checker [4]) to assess relevant performance indicators against the SSN models. For the sake of space we only give a very succinct overview of the SSN and HASL formalisms referring the reader to the literature for a detailed treatment.

Stochastic Symmetric Nets. Like any Petri net formalism an SSN model is a bi-partite graph consisting of *place* nodes (circles) and *transition* nodes (bars). Places may contain (countably many) *tokens* and are connected to transitions through arcs which are labelled with *arc functions*. An SSN model describes a continuous-time, discrete-state stochastic process whose states correspond with the possible *markings* of the SSN (a marking gives the content of each place of the SSN). The peculiarity of SSN models is that tokens may be associated with information, hence they may have different *colours*, instead of being all indistinguishable as with non-coloured PN formalisms. Therefore places and transitions of an SSN are associated with a *color domain* (CD) built from elementary types called *color classes* ($\{C_1, \dots, C_n\}$) with $cd(p)$, resp. $cd(t)$, denoting the CD of place p , resp. transition t (see Table 4 and Table 3 for the CDs of 802.11p SSN model). SSN color classes are finite, non empty and disjoint sets, they may be ordered (in this case a successor function is defined on the class, inducing a circular order among the elements in the class), and may be partitioned into (static) subclasses. An SSN transition may be immediate (drawn as a thin filled in bar) or timed which means its

firing delay is sampled (the moment it gets enabled) from a probability distribution that may be *exponential* (thick empty bar) or *general* (thick filled in bar). A coloured transition may be associated with a *guard*, i.e. a boolean-valued expression built on top of *colored variables* by means of the following basic predicates: $x=y$, $x \in subclass$, $d(x)=d(y)$ where x and y are variables of the transition with same type, and $d(x)$ denotes the static subclass x belongs to. A valid *transition binding* is an assignment of values to its variables, satisfying the predicate expressed by the *guard*. A pair (transition, binding) is called *transition instance*. Each arc connecting a place p and a transition t is labeled with an expression denoting a function $arcf : cd(t) \rightarrow Bag(cd(p))$ where $Bag(A)$ is the set of all possible multisets that may be built on set A . The valuation of $arcf$ given a legal binding of t gives the multiset of colored tokens to be withdrawn from (in case of input arc) or to be added to (in case of output arc) the place connected to that arc upon firing of such transition instance. The arc expressions in SSNs are built upon a limited set of primitive functions whose domains must be color classes. Typically an arc expression is a linear combination of function tuples (denoted $\langle f1, \dots, fn \rangle$), and each element of a tuple is either a projection function, denoted by a variable in the transition color domain (e.g. sa and sb in the tuple $\langle sa, sb, p, pt \rangle$ appearing as the labelling in several arcs of the SSN of Figure 2), a successor function, denoted $x++$ where x is a variable whose type is an ordered class; a constant function, denoted $C_i.All$ returning all elements of (sub)class C_i ; a complement function denoted $C_i.All - x$ where x is a variable of type C_i . The dynamics of an SSN model is defined in terms of transition instances *enabling* and *firing*: a transition instance is *enabled* if the marking of all of its input places is *compatible* with it (i.e. if the marking enables the transition instance). Upon *firing* a transition instance modifies the state of the SSN by removing (resp. adding) tokens from its input (resp. output) places. For example, w.r.t. the SSN in Figure 2, assuming colour classes $St = \{st1, st2\}$ and $Pr = \{pr1\}$, the initial marking of place *Idle* (colour domain $St \times Pr$) being $\langle St, Pr \rangle = \{\langle st1, pr1 \rangle, \langle st2, pr1 \rangle\}$ enables two instances of timed transition *PacketArrival* (given its guard $sa \neq sb$), namely $\langle st1, st2, pr1, rts \rangle$ $\langle st2, st1, pr1, rts \rangle$. The firing of the first instance consumes $\langle st1, pr1 \rangle$ from *Idle* and adds $\langle st1, st2, pr1, rts \rangle$ to *Sense*.

4 Modelling of 802.11/802.11p networks

To analyse the performance of two versions of the protocol we developed two SSN models, one modelling a network whose stations use the 802.11 MAC, the other where stations that use the 802.11p MAC. The models we developed are based on the following assumptions: **1) Clique network topology**: the network consists of N stations arranged in a clique (i.e. every station can overhear every other station). **2) Traffic direction**: each station has a unique target station to which it addresses its incoming traffic. **3) Incoming traffic**: each station is either under a) a *saturated regime* (i.e. a packet ready to be transmitted is invariably present) or b) its incoming traffic is given by an Poisson process with parameter λ . **4) Perfect/Imperfect channel**: the wireless medium is supposed to behave either a) as a *perfect channel* (i.e. transmitted data are never affected by errors) or b) as an *imperfect channel* with some error probability (details below). **5) Vulnerable period**: the radio device of each station is supposed to exhibit a certain delay for switching between transmission/reception mode. The duration of such period is called the *vulnerable period* and is one source of traffic collisions within the network.

Channel error model. To model the possibility that transmissions undergo errors due to the channel we adopted the burst-noise binary channel model [9]. A burst noise channels may be subsumed by a two-states Markov chain where one state (G as in good) represents the absence of noise while the other (B as in burst errors) represents the presence of an error spike which affects the channel impeding a transmission to correctly take place. We equipped our Petri nets models with an implementation of such burst-noise binary channel model.

4.1 SSN models of 802.11/802.11p networks

We developed two SSN models one for 802.11 networks the other for 802.11p's. For the sake of space we only present the 802.11p version of the model, which is somehow a generalisation of the 802.11's. The 802.11p SSN model relies on the definition of a number of color classes (Table 3) and color domains (Table 4) that are used to characterise the type of the various elements (places and transitions). The model consists essentially of three parts: the *802.11p core module*, representing the prioritised carrier-sensing and RTS/CTS handshaking mechanism, the *backoff module*, representing the behaviour of a station in case of a collision and the *medium module* representing the state of the wireless channel. We present these 3 modules separately bearing in mind that they are part of the same SSN model.

802.11p core SSN module Figure 2 depicts the portion of the SSN model representing the traffic generation and prioritised RTS/CTS handshaking protocol. It consists of three main parts: the generation of incoming data traffic (timed transition *PacketArrival*), the carrier-sensing phase (timed transitions *WaitAIFS_i*, *WaitSIFS*), the packets transmission phase (timed transitions *sendRTS*, *sendCTS*, *sendDATA*, *sendACK*), the handling of packets received/overheard by a station (the sub-net in between places *WaitForResponse*, *Receiving* and *ReadingPacket*). Let us describe the various parts of the model. Place *Idle* (color domain SP) is initially filled up with tokens corresponding

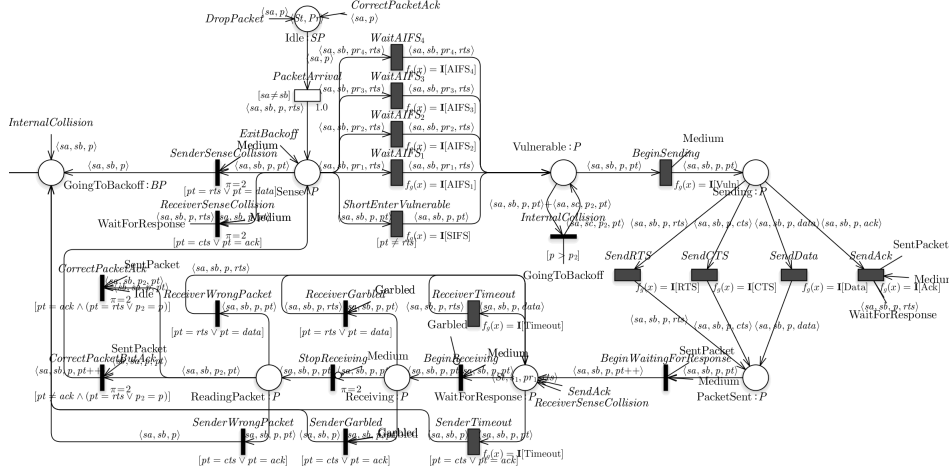


Figure 2: SSN model of the 802.11p prioritised RTS/CTS scheme

with all pairs from $St \times Pr$, representing that each station may generate incoming traffic with any level of priority. The firing of the outgoing timed (exponentially distributed) transition *PacketArrival* consumes a $\langle sa, p \rangle$ token and produces the $\langle sa, sb, p, rts \rangle$ token (added to place *Sense*) indicating that station sa is ready to send an rts request with priority p to station sb ($sa \neq sb$). A token $\langle sa, sb, p, pt \rangle$ in place *Sense* (representing an ongoing transmission of a packet of type pt and priority p from station sa to sb) is then moved either to i) place *Vulnerable* through either one of the four mutually-exclusive timed (deterministic) transition *WaitAIFS_i* (only that corresponding to the actual priority value p is enabled), if packet type $pt = rts$ and the medium remains free for $AIFS_i$ time units, or through *WaitSIFS* if packet type $pt \neq rts$ and medium stay free for $SIFS$, or if the medium gets occupied in the meantime ii) to *GoingToBackoff* (through immediate transition *SenderSenseCollision*) if sa is the sender of an rts or $data$ packet or iii) to *WaitForResponse* (through immediate transition *ReceiverSenseCollision*) if sa is a station responding with either a cts or ack to sb . A token $\langle sa, sb, p, pt \rangle$ from place *Vulnerable* moves to *Sending* (after $Vuln$ time units through timed transition *BeingSending*), and in so doing it adds one (uncoloured) token to *Medium* indicating that the number of transmitting stations has increased. Observe that the immediate transition *InternalCollision* (which is also consuming tokens from *Vulnerable*) deals with the case of competing packets sent, with different level of priorities, by a common station sa : in this case only the packet with the highest priority stays in *Vulnerable*, while packets with lower priority are moved to *GoingToBackoff* (i.e. highest priority packet wins the internal competition by “overtaking” lower priority packets). From place *Sending* a token $\langle sa, sb, p, pt \rangle$ moves to *PacketSent* if the packet type is either rts , cts or $data$, after a delay corresponding with the kind of packets, and then, with no delay, it is further moved to *WaitForResponse* through immediate transition *BeginWaitingForResponse* which while removing a token from *Medium* (hence decreasing the occupation of the wireless channel) and storing the $\langle sa, sb, p, pt \rangle$ in place *SentPacket* for later retrieval, also updates the packet type of token $\langle sa, sb, p, pt \rangle$ to $\langle sa, sb, p, pt++ \rangle$, indicating that station sa is now ready to wait for a reply message (of type $pt++$, i.e. cts in reply to rts or ack in replay to $data$) from sb . Place *WaitForResponse* is initialised with marking $\langle St, s1, pr1, rts \rangle$ indicating that each station of St is, initially, ready to get engaged into replying to an rts request ($s1$ and $pr1$ being irrelevant at this stage). As the medium gets occupied but not garbled all tokens in *WaitForResponse* are moved (with no delay) to *Receiving* and from there, as soon as the channel gets free (and assuming it hasn't been garbled before) they move on to *ReadingPacket*. Thus, at the end of a transmission phase, place *ReadingPacket* contains all tokens representing all possible combinations of responses that all stations may be engaged in. On completion of the transmission corresponding to token $\langle sb, sa, p, pt \rangle$ being added to *SentPacket* the corresponding token $\langle sa, sb, p_2, pt \rangle$, if present, is consumed from *ReadingPacket* (through either one of the prioritised transitions *CorrectPacketButAck* or *CorrectPacketAck*) representing the creation of the response, by the destination station sa , to the transmitted packet pt . Notice that *CorrectPacketButAck* triggers the response to an rts , cts or $data$ packet by adding the token $\langle sa, sb, p, pt++ \rangle$ to *Sense* hence moving to the next step of the RTS/CTS protocol for the processed packet. Conversely *CorrectPacketAck* represents the end of the RTS/CTS handshaking hence it puts back a $\langle sa, p \rangle$ token to *Idle* which restart the cycle for the transmission of a priority level p packet by station sa . On the other hand all tokens $\langle sa, sb, p, pt \rangle$ that remain in *ReadingPacket* after a response to the transmitted packet has been treated (through either *CorrectPacketButAck* or *CorrectPacketAck*) are either put back to *WaitForResponse* (if they represent a receiver, i.e. transition *ReceiverWrongPacket*) or they are moved to

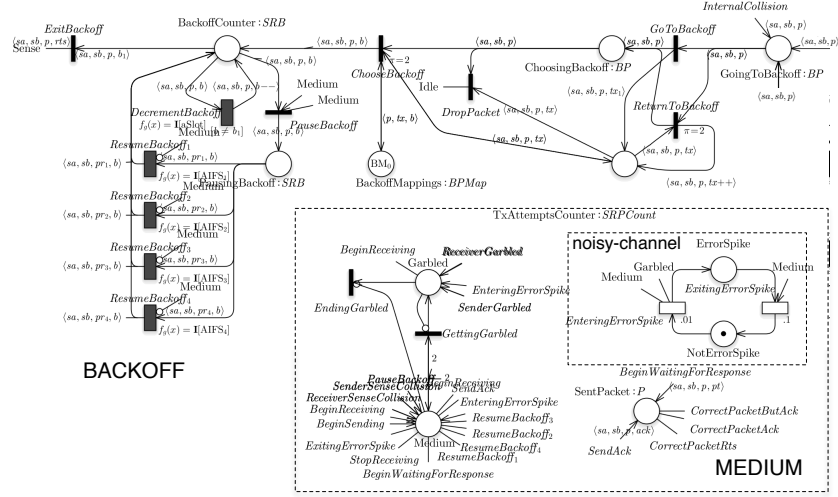


Figure 3: SSN model of the 802.11p backoff mechanism (top left) and the medium (bottom right).

GoingToBackoff (if they represent a sender that was expecting either a *cts* or an *ack* response to a previously sent packet *rts* or *data*).

802.11p backoff SSN module With the 802.11p DCF the selection of the *backoff* duration, i.e. the selection of the number of *time slots* a packet has to wait before starting a new transmission attempt, depends on the priority class of the packet as well as on the backoff round, i.e. the number of times a packet has unsuccessfully went through the carrier-sensing phase without winning the contention. Therefore a model of 802.11p backoff procedure must be equipped with necessary means to take into account the priority level of each packet hence the size of the contention window for each priority level. Figure 3 (top left) depicts the SSN subnet representing the randomised selection of the backoff delay for a packet of a given priority level p . A backoff begins when a token $\langle sa, sb, p \rangle$ is added to place *GoingToBackoff* (color domain BP) indicating that something went wrong (e.g. a collision or a lack of handshaking) during the transmission of a priority p packet from station sa towards sb . The first step then is the updating of the backoff round counter which boils down to adding of a token $\langle sa, sb, p, tx \rangle$ to place *TxAttemptsCounter* (whose color domain is $SRPCount$ and where variable $tx \in TxCount$ stores the backoff counter) through either transition *GoToBackoff* (enabled if the failed transmission attempt is the first one, hence initialising tx to color tx_1) or through transition *ReturnToBackoff* (enabled by any further failed transmission attempt and increasing tx by 1). Once a token $\langle sa, sb, p \rangle$ enters place *ChoosingBackoff* having, in the process, also added (or updated) a $\langle sa, sb, p, tx \rangle$ token to place *TxAttemptsCounter*, either one between the mutually exclusive transitions *ChooseBackoff* or *DropPacket* transition is enabled. Transition *ChooseBackoff* realises the probabilistic selection of the backoff counter value b , corresponding to the priority level p of the *backoffing* packet, by adding token $\langle sa, sb, p, b \rangle$ to place *BackoffCounter*. Such a selection is achieved through randomly choosing (through the test arc between transition *ChooseBackoff* and place *BackoffMappings*) a token $\langle p, tx, b \rangle$ among those specified by the invariant marking of *BackoffMappings* (Table 5). The marking of *BackoffMappings* associates with each priority level pr_i and re-transmission attempt tx_j the corresponding CW expressed as the union of color subclasses $\cup_{k=1}^{k'} bs_k$ (where k' is a function of pr_i and tx_j). Therefore, for example, for the first re-transmission attempt (i.e. $tx = tx_1$) of a priority p_1 packet the backoff value b is chosen randomly as $b \in bs_1 = \{b_1, \dots, b_4\}$ (which corresponds with the narrowest CW i.e. the AC_V0 category) as the initial marking BM_0 contains tokens $\langle pr_1, tx_1, bs_1 \rangle$. Similarly for the second re-transmission attempt (of a priority p_1 packet) b is chosen randomly as $b \in bs_1 \cup bs_2 = \{b_1, \dots, b_8\}$ as marking BM_0 contains tokens $\langle pr_1, tx_2, bs_1 + bs_2 \rangle$ and so on. Once a $\langle sa, sb, p, b \rangle$ token is added to place *BackoffCounter* (i.e. the backoff delay b is selected) the actual backoff (sensing) period begins: b is decremented every $aSlot$ time units (deterministic transition *DecrementBackoff*) however, as soon as the medium gets occupied, the decrement of b is suspended by moving token $\langle sa, sb, p, b \rangle$ from *BackoffCounter* to *PausingBackoff* (immediate transition *PauseBackoff*). Token $\langle sa, sb, p, b \rangle$ is moved back to *BackoffCounter* only after the channel got freed and stayed free for $AIFS_i$ time units (where $i \in \{1, 2, 3, 4\}$ denotes the priority level p of the backoff-ing packet). The backoff for a token $\langle sa, sb, p, b \rangle$ ends as soon as the backoff counter has reached 1 (i.e. $b = b_1$): at this point token $\langle sa, sb, p, b_1 \rangle$ is removed from *BackoffCounter* (immediate transition *ExitBackoff*) and $\langle sa, sb, p, rts \rangle$ is pushed back to *Sense* which represent the re-start of the transmission procedure for station sa to sent a priority p packet to sb .

802.11p medium SSN module Figure 3 (bottom right) shows the part of the SSN model representing the shared channel. It consists of 2 uncoloured places: *Medium*, whose marking corresponds with the number of transmitting stations, and *Garbled*, which contains a token whenever a collision has taken place or an error spike has occurred on the channel. Transition *GettingGarbled* sets the state of the medium to *garbled* (adding a token in place *Garbled*) as soon as *Medium* contains at least 2 tokens (i.e. at least two stations are transmitting). Transition *EndingGarbled* ends the *garbled* state of the medium by removing a token from *Garbled* as soon as the *Medium* is emptied. Place *SentPacket* (domain P) stores the information relative to an ongoing transmission in terms of a token $\langle sa, sb, p, pt \rangle$ where sa is the sender station, sb the receiver, p the priority level p and pt the packet type). The medium SSN is also equipped with a subnet for modelling the presence of error spikes affecting an ongoing transmission. To study the effect of a noisy channel on the network performances it suffices that place *NotErrorSpike* initially contains a token in which case transition *EnterinErrorSpike* reproduce the occurrence of errors on the channel (following an Exponential distribution with configurable rate, 0.01 in the picture), by adding a token in *Medium* (hence triggering a collision if a transmission is going on). The end of an error-spike is modelled by transition *ExitingErrorSpike* (also Exponentially distributed with a configurable rate).

5 analysis of 802.11/802p models

We analysed the performances of both the 802.11 and 802.11p network models² by means of the COSMOS statistical model checker [4, 1]. We considered a number of key performance indicators (KPI) including 1) the **throughput** (THR) of a network station (i.e. the number of successfully transmitted packets per time unit); 2) the **busyTimeRatio** (BTR, i.e. the ratio between the time the channel is occupied by some transmitting station and the total operation time of the network). We encoded such KPIs in terms of HASL specifications for SSN models [3] and used them to run a number of experiments to assess the impact that 1) a faulty channel, 2) the incoming traffic, 3) the network dimension have on the KPIs. In the remainder we present an excerpt of the results obtained. Figure 4 shows an example of an HASL specification (a linear hybrid automaton, LHA) we used to assess the THR for priority p traffic on the 802.11p model. The LHA in Figure 4 has an initial (l_1) and a final (l_2) location and uses a clock (t) plus a counter of successfully

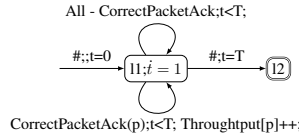


Figure 4: The LHA for measuring the throughput

terminated transmissions for a priority p packet ($Throughput[p]$). On processing a trajectory $Throughput[p]$ is incremented each time an instance of the *CorrectButAck* transition (Figure 2) with the corresponding priority p level occur (which coincides with the termination of the transmission of a priority p packet). The LHA ends measure at time T and the value of THR is obtained as $Throughput[p]/T$.

5.1 Results

Impact of network dimension and incoming traffic. Figure 5(a) and 5(b) depict the estimated THR and BTR for the 802.11 model computed (for different network dimensions) in function of the traffic arrival rate λ (i.e. under a non-saturated regime) and assuming the channel is not affected by errors (*ideal channel*). Both the THR and the BTR exhibit an asymptotic profile (the faster the inter-arrival rate the higher the throughput, resp. the BTR) and both are upper bounded by the *nominal maximum throughput under saturated regime* (which is ~ 640 packets per second), resp. the *saturated nominal maximum BTR*³Figure 5(e) and 5(f), refer to the same kind of experiment for the 802.11p model (for a network with $N = 2$ nodes). Plots in Figure 5(e) highlight the effect of prioritised management of data traffic:

²SSN models and HASL properties used to run such experiments on COSMOS are available at <https://sites.google.com/site/pballarini/models>.

³the BTR of a network using 802.11 RTS/CTS MAC has a *nominal maximum upper bound* ($NomMaxBTR$) corresponding with an ideal (hypothetical) situation where a continuous flow of DATA packets are transmitted without inter-arrival delay and in absence of collisions. In such situation the BTR is given by the ratio of busy-time over total-time for successfully transmitting, where the latter is given the sum of delays of sequence DIFS-RTS-SIFS-CTS-SIFS-DATA-SIFS-ACK sequence, that is $NomMaxBTR = TXtime / (SENSEtime + TXtime)$, where $TXtime = RTS + CTS + DATA + ACK$, is the total transmission time and $SENSEtime = DIFS + 3SIFS$ is the total carrier-sensing time. From parameters in Table 1, we have that $NomMaxBTR = 80/88 = 0.909$, meaning that the optimal channel utilisation for a WiFi based on RTS/CTS 802.11 MAC cannot trespass $\sim 90\%$.

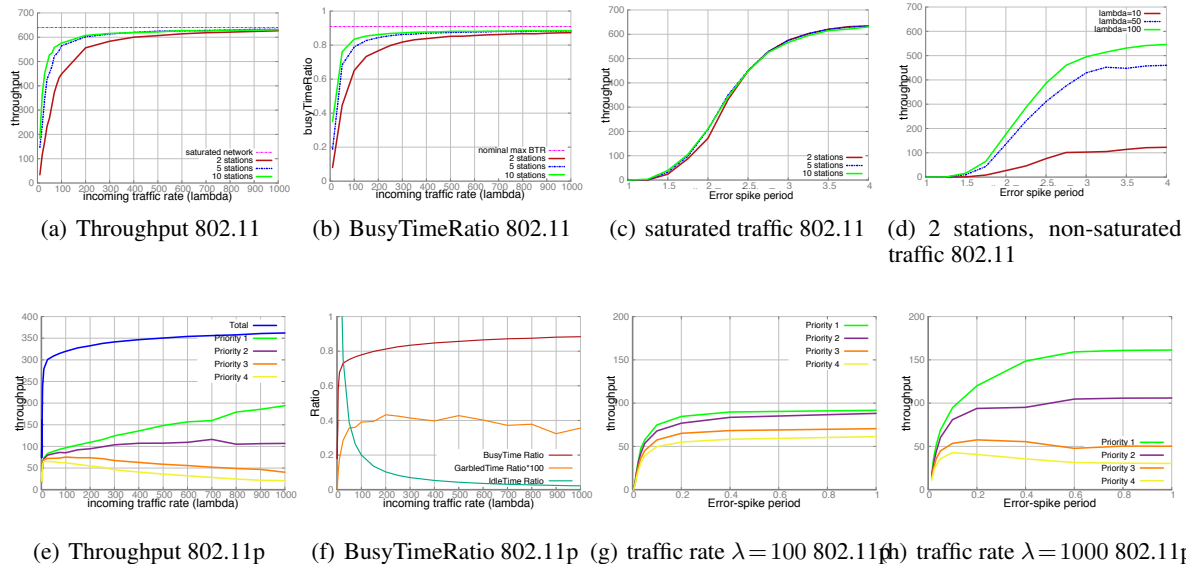


Figure 5: Impact of network dimension and noisy-channel on throughput and BTR for 802.11 and 802.11p networks

the higher the traffic the larger the difference between high priority and low priority throughput. Figure 5(f) shows the BTR as well as the ratio between the time the channel is idle (over the total observation time) and the ratio between the time the channel is garbled (collision).

Impact of faulty channel. Figure 5(c), 5(d), 5(g) and 5(h) report on assessing the effect that a faulty channel has on the network performances. The THR and BTR are measured in function of the period of an error spike (determined by the rate of transition *EnteringErrorSpike*). Results show that the throughput increases with the error-spike period however (for the 802.11 model under saturated regime Figure 5(c)) the THR is unaffected by the network dimension (identical plot for different number of stations) indicating that the performances gradient induced by the network dimension in case of ideal channel (Figure 5(a)) fades away in presence of a faulty channel. Conversely under a non-saturated regime (802.11 model Figure 5(d)) the error-spike period affects the throughput differently depending on the traffic arrival rates. Finally Figure 5(g) and 5(h) show the effect of the error-spike on the prioritised throughput in a 802.11p network showing that the effect of a faulty channel, in terms of the gradient between higher and lower priority throughput is more evident under a high traffic regime (Figure 5(h)) than under a low traffic regime (Figure 5(g)).

6 Conclusion

We presented a performance modelling study of two versions of MAC protocol for wireless networks: the 802.11 MAC and its prioritised extension 802.11p devoted to VANETS. We developed our models using a high-level stochastic Petri nets formalism which allowed us to encode the complexity of the 802.11 prioritised scheme in a model of reasonable size. The models we presented are highly configurable and allow for the analysis of the performance of networks in different respect (traffic conditions, ideal or imperfect channel, network dimension). We analysed the performance on the network models by means of statistical model checking based on the HASL specification language. Future work include the extension of the models to N -hops topologies, which would allow us to take into account the effect of routing on the performances of a given network.

References

- [1] COSMOS home page. <http://cosmos.lacl.fr>.
- [2] Ahed Alshanyour and Anjali Agarwal. Performance of IEEE 802.11 RTS/CTS with finite buffer and load in imperfect channels: Modeling and analysis. In *Proceedings of the Global Communications Conference, 2010. GLOBECOM 2010, 6-10 December 2010, Miami, Florida, USA*, pages 1–6. IEEE, 2010.

- [3] Elvio Gilberto Amparore, Benoit Barbot, Marco Beccuti, Susanna Donatelli, and Giuliana Franceschinis. Simulation-based verification of hybrid automata stochastic logic formulas for stochastic symmetric nets. In *Proceedings of the 1st ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, SIGSIM PADS '13, pages 253–264, New York, NY, USA, 2013. ACM.
- [4] P. Ballarini, H. Djafri, M. DufLOT, S. Haddad, and N. Pekergin. COSMOS: A statistical model checker for the hybrid automata stochastic logic. In *Proceedings of the 8th International Conference on Quantitative Evaluation of Systems (QEST'11)*, pages 143–144. IEEE Computer Society Press, sep. 2011.
- [5] Paolo Ballarini, Benoît Barbot, Marie DufLOT, Serge Haddad, and Nihal Pekergin. Hasl: A new approach for performance evaluation and model checking from concepts to experimentation. *Performance Evaluation*, 90:53 – 77, 2015.
- [6] Frederico J. R. Barboza, Aline Maria Santos Andrade, Flávio Morais de Assis Silva, and George Lima. Specification and verification of the IEEE 802.11 medium access control and an analysis of its applicability to real-time systems. *Electr. Notes Theor. Comput. Sci.*, 195:3–20, 2008.
- [7] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J.Sel. A. Commun.*, 18(3):535–547, September 2006.
- [8] G. Chiola, C. DuTheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed colored nets and symmetric modeling applications. *IEEE Trans. Comput.*, 42(11):1343–1360, November 1993.
- [9] E. N. Gilbert. Capacity of a burst-noise channel. *Bell System Technical Journal*, 39:1253–1265, 1960.
- [10] Armin Heindl and Reinhard German. Performance modeling of ieee 802.11 wireless lans with stochastic petri nets. *Performance Evaluation*, 44:139–164, 2000.
- [11] IEEE. *The OSI reference model.*, 1983.
- [12] IEEE. Ieee wireless lan medium access control (mac) and physical layer(phy) specification std 802.11-1997. Technical report, Institute of Electrical and Electronic Engineers, 1997.
- [13] M. Kwiatkowska, G. Norman, and J. Sproston. Probabilistic model checking of the IEEE 802.11 wireless local area network protocol. In H. Hermanns and R. Segala, editors, *Proc. 2nd Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification (PAPM/PROBMIV'02)*, volume 2399 of LNCS, pages 169–187. Springer, 2002.
- [14] A. Lyakhov and F. Simatos. Hybrid rts/cts mechanism in wi-fi ad hoc networks with correlated channel failures. *17th IMACS*, 2015.

A Parameters of the SSN model of the 802.11 and 802.11p MAC

name	meaning	value ($\times 10\mu s$)
aslot	the time unit of the backoff procedure	2
nStations	# of stations	2
packetSizeInAslot	# of time slot for sending a DATA packet	variable
DIFS	the length of the DCF interframe space	5
SIFS	the length of a short interframe space	1
vuln	delay to switch radio from RX-to-TX mode	2
RTS	the time it takes to send a RTS packet	16
CTS	the time it takes to send a CTS packet	11
ACK	the time it takes to send an acknowledgement packet	11
timeout	delay a station waits for handshaking packet	5
CWmin	min. size of the contention window (in aslot)	15
backoffMax	# of TX attempts before dropping a packet	6
deadline	# of time slots for entering the livelock (end) state	5000

Table 1: Timing parameters for the SSN models of RTS/CTS

AC	CWmin	CWmax
Background (AC_BK)	aCWmin	aCWmax
Best effort (AC_BE)	aCWmin	aCWmax
Video (AC_VI)	$[(aCWmin+1)/2]-1$	aCWmin
Voice (AC_VO)	$[(aCWmin+1)/4]-1$	$[(aCWmin+1)/2]-1$

Table 2: Contention windows boundaries for ECDA access categories.

name	description	definition	ordered
PT	Packet Type	$PT = \{rts, cts, data, ack\}$	YES
St	Station ID	$St = s\{1, \dots, N\}$	NO
Pr	Priority level (of a packet)	$Pr = p\{1, \dots, 4\}$	NO
$TxCount$	max. num. of re-transmissions before dropping a packet	$TxCount = tx\{1, \dots, 20\}$	YES
$BackoffStage$	backoff counter domain	$BackoffStage = bs_1 \cup \dots \cup bs_9$	YES
bs_1	backoff counter domain stage 1	$bs_1 = b\{1, \dots, 4\}$	YES
bs_2	backoff counter domain stage 2	$bs_2 = b\{5, \dots, 8\}$	YES
bs_3	backoff counter domain stage 3	$bs_3 = b\{9, \dots, 16\}$	YES
bs_4	backoff counter domain stage 4	$bs_4 = b\{17, \dots, 32\}$	YES
bs_5	backoff counter domain stage 5	$bs_5 = b\{33, \dots, 64\}$	YES
bs_6	backoff counter domain stage 6	$bs_6 = b\{65, \dots, 128\}$	YES
bs_7	backoff counter domain stage 7	$bs_7 = b\{129, \dots, 256\}$	YES
bs_8	backoff counter domain stage 8	$bs_8 = b\{257, \dots, 512\}$	YES
bs_9	backoff counter domain stage 9	$bs_9 = b\{513, \dots, 1024\}$	YES

Table 3: Color classes for the SSN model of the 802.11p protocol.

name	description	definition
SR	Sender-Receiver	$SR = St \times St$
SP	Station-Priority	$SP = St \times Pr$
BP	Backoff-Priority	$BP = St \times St \times Pr$
P	Packet sending	$P = St \times St \times Pr \times PT$
SRB	Sender-Receiver-BackoffStage	$SRB = St \times St \times BackoffStage$
BPM_{ap}	Mapping-Priority-Backoff	$BPM_{ap} = Pr \times TxCount \times BackoffStage$
SRP_{count}	Sender-Receiver-Priority-Count	$SRP_{count} = St \times St \times Pr \times TxCount$

Table 4: Color domains for the SSN model of the 802.11p protocol.

marking	definition
BM_0	$\langle pr1, tx1, bs1 \rangle + \langle pr1, tx2, bs1 + bs2 \rangle + \langle pr1, tx3, bs1 + bs2 \rangle + \langle pr1, tx4, bs1 + bs2 \rangle$ $+ \langle pr2, tx1, bs1 \rangle + \langle pr2, tx2, bs1 + bs2 \rangle + \langle pr2, tx3, bs1 + bs2 + bs3 \rangle +$ $+ \langle pr2, tx4, bs1 + bs2 + bs3 \rangle + \langle pr2, tx5, bs1 + bs2 + bs3 \rangle + \langle pr3, tx1, bs1 + bs2 \rangle +$ $\langle pr3, tx2, bs1 + bs2 + bs3 \rangle + \langle pr3, tx3, bs1 + bs2 + bs3 + bs4 \rangle$ $+ \langle pr3, tx4, bs1 + bs2 + bs3 + bs4 + bs5 \rangle + \langle pr3, tx5, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 \rangle$ $+ \langle pr3, tx6, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 \rangle +$ $\langle pr3, tx7, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 + bs8 \rangle$ $+ \langle pr3, tx8, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 + bs8 + bs9 \rangle +$ $\langle pr3, tx9, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 + bs8 + bs9 \rangle +$ $\langle pr3, tx10, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 + bs8 + bs9 \rangle$ $+ \langle pr4, tx1, bs1 + bs2 + bs3 \rangle + \langle pr4, tx2, bs1 + bs2 + bs3 + bs4 \rangle$ $+ \langle pr4, tx3, bs1 + bs2 + bs3 + bs4 + bs5 \rangle + \langle pr4, tx4, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 \rangle$ $+ \langle pr4, tx5, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 \rangle$ $+ \langle pr4, tx6, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 + bs8 \rangle$ $+ \langle pr4, tx7, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 + bs8 + bs9 \rangle +$ $\langle pr4, tx8, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 + bs8 + bs9 \rangle +$ $\langle pr4, tx9, bs1 + bs2 + bs3 + bs4 + bs5 + bs6 + bs7 + bs8 + bs9 \rangle$

Table 5: Invariant marking of place *BackoffMapping*.