Probabilistic Model Checking of IEEE 802.11 IBSS Power Save Mode

Pravati Swain
Department of CSE,
IIT Guwahati, India 781039
E-mail: pravati@iitg.ernet.in

Purandar Bhaduri
Department of CSE,
IIT Guwahati, India 781039
E-mail: pbhaduri@iitg.ernet.in

Sukumar Nandi
Department of CSE,
IIT Guwahati, India 781039
E-mail: sukumar@iitg.ernet.in

Abstract: The IEEE 802.11 standard for wireless local area network defines a power management algorithm for Independent Basic Service Set (IBSS). The wireless stations that form an IBSS communication mode create ad-hoc connections between themselves and communicate with each other through a contention based channel access mechanism. The wireless stations in IBSS are typically battery enabled low power devices. In the power management for IBSS, time is divided into Beacon Intervals (BIs) and each BI is divided into an Announcement Traffic Indication Message (ATIM) window and a data window. The stations that have successfully transmitted an ATIM frame within the ATIM window will compete to transmit a data frame in the date window. The rest of the stations go to sleep mode during the data window, thus saving power.

The power management algorithm for IBSS mode, based on a randomized exponential backoff procedure, is defined in the IEEE 802.11 standard for wireless LANs. The probabilistic behavior of the MAC mechanism enable us to design a probabilistic model suitable for model checking. This paper presents a finite state Markov Decision Process (MDP) and uses the tool PRISM to compute the performance of the IEEE 802.11 IBSS in PSM. A set of performance properties, such as expected delay and energy consumption are specified as queries in the probabilistic temporal logic PCTL and computed using PRISM.

Keywords: IEEE 802.11 standards; Markov Decision Process; PCTL; ATIM frame; power consumption.

1 Introduction

IEEE 802.11 (*IEEE*, 2007) is a standard for wireless LANs. The standard presents a contention based MAC protocol known as distributed co-ordination function (DCF) for ad hoc wireless networks. The DCF is a carrier sense multiple access/collision avoidance (CSMA/CA) based protocol and uses a binary exponential backoff (BEB) algorithm to access the channel. There are two operating modes for wireless stations. The first one is *Infrastructured Basic Service Set* (iBSS), where every wireless station communicates through an Access Points (AP) connected with the distribution system. The second communication mode is termed as *Independent Basic Service Set* (IBSS) where the stations create infrastructureless ad-hoc connections between themselves and communicate with each-other through a contention based channel access mechanism.
Most of the work on analyzing performance properties of the 802.11 DCF have focused on either simulation or analytical model based techniques (*Bianchi*, 2000; *Alshanyour and Agarwal*, 2009). The paper (*Kwiatkowska et al.*, 2002a) uses probabilistic model checking for analyzing the MAC protocol of the IEEE 802.11 standard. Probabilistic model checking is a formal verification technique for systems that exhibit stochastic behaviour (*Baier et al.*, 1997). In probabilistic model checking the system is modeled as a state

transition system where the transitions have associated probabilities. The performance properties to be analyzed are specified in a temporal logic, such as PCTL (*Hansson and Jonsson*, 1994). In recent years, probabilistic model checking has been used in different application areas such as, in communication (*Kwiatkowska et al.*, 2003), dynamic power management (*Norman et al.*, 2005), etc. The performance analysis of the communication system integrates several essential aspects, including concurrency between two components, randomization and real time constraints. Probabilistic model checking is used to determine the correctness and quantitative measures such as performance and reliability. This paper concentrates on the applicability of model checking techniques to the analysis of the IEEE 802.11 IBSS in power save mode (PSM).

Related Work

In the paper (*Kwiatkowska et al.*, 2002a), the authors have presented a *probabilistic timed automaton (PTA)* (*Kwiatkowska et al.*, 2002b) (an extension of the classical timed automata (TA) model (*Alur and Dill*, 1994)) for the basic channel access mechanism of the IEEE 802.11, assuming two senders and two receivers. A Markov decision processes (MDP) is obtained from the PTA through a property preserving discrete semantics (*Rutten et al.*, 2004), resulting in both nondeterministic and probabilistic choice in the model. The paper (*Kwiatkowska et al.*, 2002a) calculates two important properties viz., the minimum probability of both stations *eventually* sending their packet correctly and the maximum probability of *either* stations reaching the backoff counter value. The paper concludes that there is an exponential increment in the number of iterations against the backoff counter value. The paper (*Roy and Gopinath*, 2005) presents an optimized technique to counter the state space explosion problem. The authors present a deterministic path compression method to reduce the states in the model. The results obtained from the optimized model is compared with the original model for two stations. The optimized technique helps to build the model for three stations. The Carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocol is analyzed using probabilistic model checking in the paper (*Duflot et al.*, 2005). The system is modeled by a probabilistic timed automata (PTA) and the correctness of the probabilistic behavior of the protocol is verified by analyzing minimum and maximum probabilities for reachability and time-bounded reachability properties. (*Duflot et al.*, 2006) present a formal analysis of the device discovery phase of the Bluetooth wireless communication protocol by modeling a discrete time Markov chain (DTMC). (*Fruth*, 2006) presents the application of PTA to the IEEE 802.15.4, where the channel access is based on randomized exponential backoff. An analytical model of IEEE 802.11 DCF in Power Save Mode (PSM) has been presented in (*Swain et al.*, 2012). In this paper, the authors have used a discrete time Markov model to calculate the probability that an ATIM frame is transmitted successfully in the ATIM window. The throughput of the IEEE 802.11 PSM has been calculated using the probability of successful transmission of ATIM frames and the results are validated using simulation results obtained from the network simulator Qualnet (*QN*, 2011). To the best of our knowledge there is no probabilistic model checking work for the IEEE 802.11 IBSS PSM. In this paper we use probabilistic model checking to analyze the performance of stations in a wireless LAN following the IEEE 802.11 standard (*IEEE*, 2007) in Power Save Mode. The probabilistic model used here is a MDP (*Rutten et al.*, 2004) which is appropriate for modeling both nondeterministic choice (to model the asynchrony between subcomponent) and probabilistic choice (to model the probabilistic nature of the backoff procedure). The properties of MDP are specified in the Probabilistic Computation Tree Logic (PCTL) (*Hansson and Jonsson*, 1994). PCTL is an extension of Computation Tree Logic (CTL) (*Clarke et al.*, 1986) which is a branching time temporal logic. PCTL allows one to express quantities such as "the expected time for the stations to complete their packet transmissions successfully". The probabilistic model checking tool PRISM (*PRISM*, 2012) is used to construct the formal model from a high-level description of the system and then check its performance properties.

This paper is divided into four sections. Section 2 presents a short description of the IEEE 802.11 DCF in PSM and the CSMA channel contention resolution protocol. An introduction to Markov decision process (MDP) is presented in Section 3. Section 4 presents a description of PRISM. Section 5 presents the network configuration and explain the MDP model in PRISM code. In Section 6, the verification results are reported. Section 7 concludes the paper.

2 The IEEE 802.11 DCF in Power Save Mode

The IEEE 802.11 standard (*IEEE*, 2007) defines power saving techniques for IBSS. It is assumed that all the stations are synchronized and awake at the beginning of each beacon interval. The stations in power save mode (PSM) wake up periodically to listen to the beacon messages and stay awake for a period of time called the Announcement Traffic Indication Message (ATIM) window. The ATIM frame is a control frame which is exchanged by the stations within the ATIM window to determine whether to go for PSM or stay awake for data transmission after the end of the ATIM window. When a station has a data packet to transmit, it transmits an ATIM frame to the corresponding receiver during the ATIM window following the 802.11 CSMA/CA DCF backoff mechanism specified in the IEEE 802.11 (*IEEE*, 2007). The backoff time is chosen as follows;

$$\text{Backoff Time} = \text{Random}() \times \text{Slot Time}$$

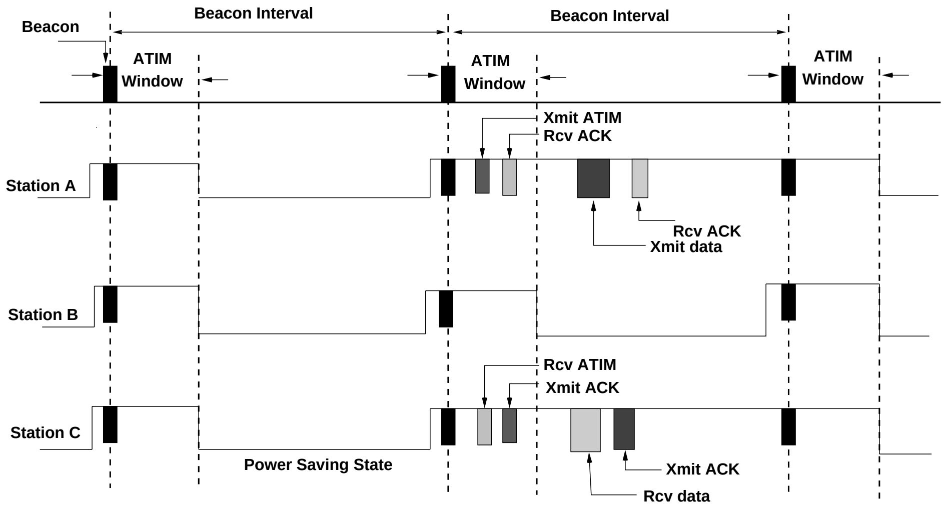

Figure 1 Power save mode in IBSS

where the random value is uniformly distributed in the interval $[0, \mathrm{CW}-1]$ where CW is the contention window size which satisfies $\mathrm{CW}_{\min} \leq \mathrm{CW} \leq \mathrm{CW}_{\max}$. Here $\mathrm{CW}_{\min}$ and $\mathrm{CW}_{\max}$ are the minimum and maximum contention window sizes respectively. In response to an ATIM, the receiver will respond with an ATIM-ACK frame. After this ATIM handshake, both stations will remain in power on mode in the following data window. If a station is unable to transmit an ATIM frame during the ATIM window, due to contention with another station or ending of the ATIM window, the data frame is buffered and an attempt is made to transmit the ATIM frame during the next ATIM window. A station may enter the power save state at the end of the ATIM window if it does not transmit or receive ATIM frame successfully. An example of the IEEE 802.11 PSM is illustrated in Figure 1. Station A announces a frame destined for station B by transmitting an ATIM frame during the ATIM window. Station B transmits ATIM-ACK to station A and remains awake for the rest of the BI. At the end of the ATIM window station A transmits a data frame to station B. The station B transmits an ACK after successfully receiving the data frame. Station C goes to the *power save* state at the end of the ATIM window, thus saving energy.

The IEEE 802.11 standard (*IEEE*, 2007) does not define the retry limit for ATIM frame transmission in IBSS. However, the paper (*Jung and Vaidya*, 2002) defined the retry limit of three for an ATIM frame transmission within an ATIM window and up to three BIs. This paper assumes that if the ATIM frame is not transmitted successfully the corresponding data frame is buffered for the next beacon interval. After the third ATIM window if the ATIM frame is not transmitted successfully then the data frame is dropped. The same assumption is made in this paper to model the ATIM frame transmission.

After successful transmission of ATIM frames, the ATIM sender and receiver remain awake for the rest of the BI and the rest of the stations go to sleep state during the data transmission to save battery energy. The data frame transmission is based on CSMA/CA with the binary exponential backoff (BEB) algorithm. In the paper (*Zheng et al.*, 2004) the authors have explained by theoretical analysis and simulation results that a single data window is sufficient to successfully transmit a data frame after transmitting an ATIM frame successfully in the ATIM window. The same assumption is made in this paper for data frame transmission.

Table 1 Symbol used for MDP in Figure 6

Symbol	Meaning
BN	Beacon number
$\mathrm{BN}_{\max}$	Maximum number of Beacon Interval
Ai	Backoff stage within ATIM window
$\mathrm{Ai}_{\max}$	Maximum retry limit within ATIM window
i	Backoff stage within data window
$\mathrm{i}_{\max}$	Maximum retry limit within data window
W	Minimum contention window
AT	Time duration of the ATIM window
DT	Time duration of the data window
ATIM	ATIM frame size (in terms of time)
DATA	DATA frame size (in terms of time)

3 Markov Decision Process (MDP)

A Markov decision process (MDP) is a probabilistic model with the addition of *nondeterminism.* Probability is used to quantify aspects of system behavior where probability distributions are known. In contrast, nondeterminism is used to model unknown environments, where such distributions are not known. MDPs are used to model concurrency, where it represents the different possible interleavings of multiple components operating in parallel.

Definition 3. A Markov Decision Process (MDP) (*Rutten et al.*, 2004) is a tuple $\mathrm{M} = (\mathrm{S}, \bar{\mathrm{s}}, \mathrm{Act}, \mathbf{Steps}, \mathrm{AP}, \mathrm{L})$ where:

- S is a set of *states*;

- $\bar{s}$ is the *initial state*;
- AP is a set of atomic propositions;
- $L : S \to 2^{AP}$ is a *labelling function*. $\forall s \in S$, L(s) is a set of atomic propositions.
- Act is a set of *action labels*;
- **Steps** $: S \times Act \to Dist(S)$ is a *transition probability function*, Dist(S) presents the set of all discrete probability distribution over S;

In an MDP, several actions may be available in a given state s, each corresponding to a probability distribution. We denote this set by $A(s) = \{a \in Act \mid \mathbf{Steps}(s, a) \text{ is defined}\}$. In each state s of an MDP, the successor state is decided in two steps: first, nondeterministically select an available action a from the alphabet Act and secondly randomly choose the successor according to the probability distribution **Steps**(s, a).

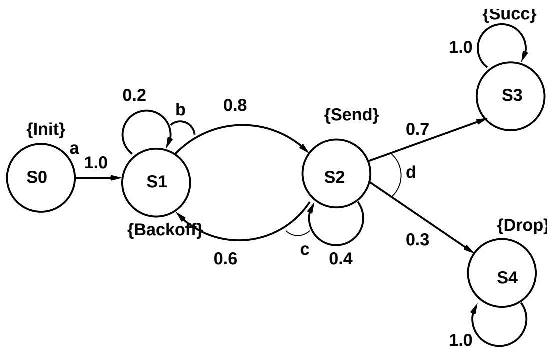

Figure 2 Markov Decision Process (MDP)

Figure 2 depicts an example of a MDP with states $S = \{s_0, s_1 \ldots, s_4\}$, s_0 is the initial state and the alphabet of actions is $\{a, b, c, d\}$. It can be observed from Figure 2 that state s_2 has nondeterministic choice between two actions, c and d.

The logic PCTL is used to specify the properties with an implicit quantification such as $P_{min} =?$ and $P_{max} =?$. For example:

> $P_{min=?}$ [F Succ]: the minimum probability that eventually the packet will be received successfully
>
> $P_{max=?}$ [F Drop] : the maximum probability that eventually the packet will be dropped.

Model checking of MDPs applies linear optimization problems for solutions, rather than linear equation systems. In practice, this is often done using dynamic programming.

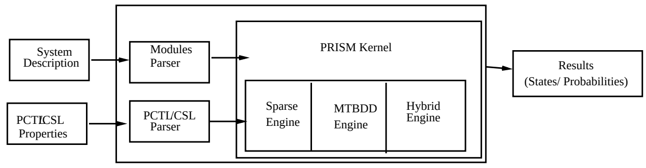

Figure 3 PRISM System Architecture (*Kwiatkowska et al.*, 2011)

4 PRISM: Probabilistic Model Checking Tool

In the probabilistic model checking tool PRISM (*PRISM*, 2012), models are described as the parallel composition of a set of modules and each module contains a set of commands describing transitions. Variables can be local or global to a single module or whole model. The transition from one state to another corresponds a choice of all enabled commands. A command is enabled in a state when its forerunner holds and all other commands with the same action synchronously hold. Depending on the type of model, the choice of transition is probabilistic, nondeterministic, or both. In contrast to other probabilistic model checkers, PRISM has provision for state and transition rewards, quantitative analysis and symbolic data structures. The system architecture of the PRISM tool is presented in Figure 3.

```
mdp

//-------------------------------------//
module node
x : [0..4] init 0;
// 0 Initial state
// 1 Backoff
// 2 Send
// 3 Drop (fail)
// 4 Succ
//Initial (Action "a")
[a] x=0 -> 1.0 : (x'=1);
// Backoff (Action "b")
[b] x=1 -> 0.2 : (x'=1) + 0.8 : (x'=2);
// Send (Action "c")
[c] x=2 -> 0.4 : (x'=2) + 0.6 : (x'=1);
// Send (Action "d")
[d] x=2 -> 0.3 : (x'=4) + 0.7 : (x'=3);
[]  x=3 -> (x'=3); // succ
[]  x=4 -> (x'=4); // Drop
endmodule
```

Figure 4 PRISM specification of the MDP in Figure 2

In order to reduce the memory requirements even further, PRISM supports symbolic data structures based on binary decision diagrams (BDDs) and in particular a generalization of them, Multi-Terminal BDDs (MTBDDs) (*Miner et al.*, 2004). PRISM provides a choice of three computation engines: (*Rutten et al.*, 2004), First the MTBDDs engine, which uses a symbolic representation, has the lowest memory requirements and is most suitable for large models containing a large degree of regularity. Second, the sparse engine, which uses a sparse-matrix representation, is the fastest engine but has the highest memory requirements and is most suitable for irregular models. Third, the hybrid engine, which uses extensions of MTBDDs, is faster than that engine and requires less memory than the sparse engine.

The PRISM specification of the MDP in Figure 2 is presented in Figure 4. In Figure 4, **x** is an array which represents the states of the MDP, such as *Init, Backoff, Send, Succ* and *Drop.* In Figure 4, the equations

- [*c*] x=2 → 0.4 : (x'=2) + 0.6 : (x'=1);
- [*d*] x=2 → 0.3 : (x'=4) + 0.7 : (x'=3);

represent the *nondeterministic* choice between the action *c, d* from the state *send* (x=2).

5 Modeling the MDP of a Station in Power Save Mode in PRISM

This section presents MDP models for the shared channel condition shown in Figure 5 and a transmitting station in IBSS power save mode shown in Figure 6 along with the corresponding PRISM model. The symbols used in Figure 6 are explained in Table 1. The transmission of ATIM frame and data frame according to the CSMA/CA DCF follows the steps that are presented in Figure 6:

- **Sense**: When a station transmits an ATIM frame using CSMA/CA, the first step is to initialize the following variables:
  - BN := 0 (Beacon Number),
  - Ai := 0 (Initial backoff stage in ATIM window) and
  - i : = 0 (Initial backoff stage in data window).

  These steps are performed when the station transits from the state **Sense** to **Set Backoff** state in Figure 6. The integer variable x keeps track of the time elapsed since its last reset. It is incremented at every time step. Figure 7 presents the PRISM specification representing the initial value of the model definition which lists a set of finite ranging variables to determine the possible states that the module can be in.

- **Backoff**: As discussed in Section 2, the ATIM or data frame transmission is based on CSMA/CA with the binary exponential backoff (BEB) algorithm. Therefore, before transmitting an ATIM frame, the station picks up a random value between 0 and ($2^{\mathrm{Ai}}W-1$), shown in the transition from **Set Backoff** to **Backoff**. Here W is the initial contention window size and Ai is the Backoff stage within ATIM window (i is the backoff stage in data window). The backoff value is decremented at each slot time. The station tries to send the ATIM (or data) frame when the backoff value reaches zero (transition from the state **Backoff** to **Transmit ATIM** state or **Transmit DATA**). Before the end of the backoff process if the remaining ATIM window is less than or equal to SIFS (Short Inter Frame Space) time, *i.e.*, AT ≤ SIFS (the end of the ATIM window has been reached.) [1], the station goes to sleep mode in the following data window. Hence there is a transition from the state **Backoff** to the state **Wait End of DATA window**. A similar backoff process is used in data frame transmission in the data window after a successful transmission of the ATIM frame. The variable i is the backoff stage in the data window. For the above behavior, the corresponding PRISM fragment is shown in Figure 8.

- **Wait ATIM-ACK**: It can be seen from the MDP model in Figure 6 that
  - There is a transition from the state **Wait ATIM-ACK** to **ATIM-ACK RCV** when the channel satisfies the condition $c_i = 1$, *i.e.*, the i^{th} station transmits the frame successfully. The condition for the transition to take place, AT > SIFS, indicates that the remaining ATIM window is greater than SIFS. Hence it implies that ATIM-ACK is received successfully.
  - There is a transition from the state **Wait ATIM-ACK** to **No ATIM-ACK** when the condition $c_i = 2$ holds, *i.e.*, there is a collision at the time of transmission the i^{th} station within the ATIM window (*i.e.*, AT > SIFS).
  - If AT ≤ SIFS (*i.e.*, end of the ATIM window) then the station goes to the sleep mode in the following data window. Thus there is a transition from the state **Wait ATIM-ACK** to the state **Wait Endof DATA window**.

  The transition from the state **Wait ACK** in the data window is similar.

- **No ATIM-ACK**: From the state **No ATIM-ACK**, two transitions are possible, depending on whether the backoff stage (Ai) is less than the maximum retry limit within the ATIM window ($\mathrm{Ai}_{\max}$) or equal ($\mathrm{Ai} = \mathrm{Ai}_{\max}$). If $\mathrm{Ai} < \mathrm{Ai}_{\max}$ then there is a transition from the state **No ATIM-ACK** to the state **Set Backoff**, *i.e.*, the station will again try to transmit the ATIM frame within the same ATIM window. If $\mathrm{Ai} = \mathrm{Ai}_{\max}$ then there is a transition from the state **No ATIM-ACK** to the state **Wait End of ATIM window**. This transition represents the situation where within an ATIM window the station is unable to transmit the ATIM frame (y=1) and the station moves to idle mode in the remaining time of the ATIM window. The Figure 9 shows the corresponding PRISM fragment.

- **Wait End of ATIM window**: There are two possible transitions to the state **Wait End of ATIM window**. After the maximum retry limit, either the station is unable to transmit the ATIM frame within the ATIM window (y=1) or the

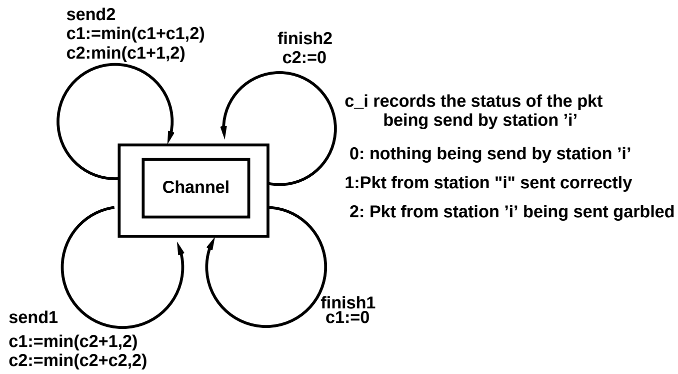

Figure 5 MDP model for channel condition with two stations (transmitter)

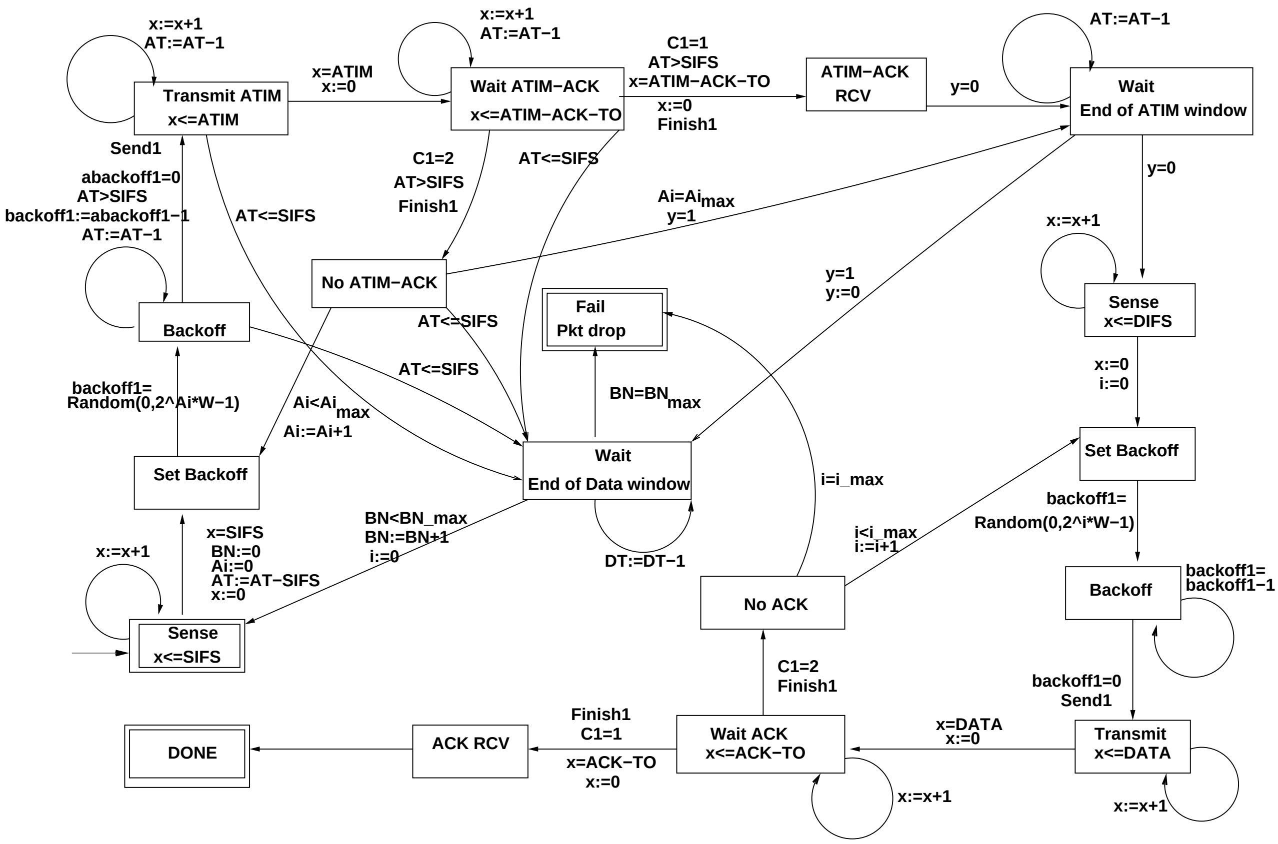

Figure 6 MDP model for a station in IBSS PSM

Table 2 Parameters used in the simulation

Parameter	value
ATIM frame	28 bytes + PHY header
PHY header	24 bytes
MAC header	28 bytes
Slot time	$20\mu s$
SIFS	$10\mu s$
DIFS	$50\mu s$
Data rate	2Mbps
$PW_{\text{tx/rx}}$	2.25W (Watt)
PW_{idle}	1.35W (Watt)
PW_{sleep}	0.07W (Watt)

station successfully transmits the ATIM frame (y=0). In both of these cases the station will wait for the remaining time of the ATIM window. For the former case (y=1) the station goes to the sleep mode in the following data window and resets the value y=0. So there is a transition from the state **Wait End of ATIM window** to the state **Wait End of DATA window**. The successful transmission of the ATIM frame by the station prepares the station to compete for transmitting the data frame in the data window using the CSMA/CA mechanism.

- **Wait End of DATA window**: If the Beacon number (BN) is less than the maximum Beacon Number (BN_{max}) then the station will retransmit

```
module station1
        //flag bit used in transmitting data frame in data window or goes to sleep mode
        //  z=1 (succ) & z=2 (unsucc) trans. of ATIM frame
        y1 : [0..2] init 0;
        //Time counter for station
        x1 : [0..TIME_MAX];
        // local state
        a1 : [1..18];
        // 1 Sense (ATIM window)
        // 2 Set Backoff counter
        // 3 Backoff
        // 4 Transmit ATIM frame
        // 5 Wait ATIM-ACK
        // 6 No ATIM-ACK (Collision in ATIM window)
        // 7 ATIM-ACK RCV (succ. transmition of ATIM frame)
        // 8 Wait for End of the ATIM window
        // 9 Wait for End of the Data window
        // 10 Sense (Data window)
        // 11 Set Backoff counter
        // 12 Backoff
        // 13 Transmit Data frame
        // 14 wait ACK (DATA ACK)
        // 15 No ACK (Collision in Data window)
        // 16 ACK RCV (succ. transmition of DATA frame)
        // 17 DONE
        // 18 Fail (Packet drop)
        bn1 : [1..4] init 1;
        abackoff1 : [0..3] ; // backoff value in ATIM window
        Ai1 : [0..2] init ai_min; //
        backoff1 : [0..31]; // backoff value in Data window
        i1 : [0..6] init i_min; // BACKOFF COUNTER for DATA
```

Figure 7 PRISM specification representing the initial value of the model

```
// SENSE (state 1)
// let time pass
[time] a1=1 & x1<SIFS -> (x1'=min(x1+1,TIME_MAX)) ; // pravati
// set the backoff counter if the channel free for SIFS time
[] a1=1 & x1=SIFS & AW > SIFS -> (a1'=2) & (x1'=0) & (AW'=AW-SIFS); //pravati
// chosen slot now set backoff (state 2)
// backoff exponent 0
[time] a1=2 & Ai1=0 -> (a1'=3) & (x1'=0) & (abackoff1'=0);
// backoff exponent 1
[] a1=2 & Ai1=1 -> 1/2 : (a1'=3) & (x1'=0) & (abackoff1'=0)
+ 1/2 :(a1'=3)  & (y1'=0) & (abackoff1'=1);
// backoff exponent 2
[] a1=2 & Ai1=2 -> 1/4 : (a1'=3) & (x1'=0) & (abackoff1'=0)
+ 1/4 :(a1'=3)  & (x1'=0) & (abackoff1'=1)
+ 1/4 :(a1'=3) & (x1'=0) & (abackoff1'=2)
+ 1/4 :(a1'=3) & (x1'=0) & (abackoff1'=3);

// BACKOFF (state 3)
// let time pass
[time] a1=3 & x1<SIFS -> (x1'=min(x1+1,TIME_MAX));
// decrement backoff
[] a1=3 & x1=SIFS  & abackoff1>0 & AW >  SIFS   -> (a1'=3) & (x1'=0) & (abackoff1'=abackoff1-1) & (AW'=AW-1) ;
//pravati
[send1] a1=3 & x1=SIFS  & abackoff1=0 -> (a1'=4) & (x1'=0);
[] a1=3 & x1>SIFS -> (a1'=9) & (x1'=0);
```

Figure 8 Transitions in PRISM specification from the states **SetBackoff**, **Backoff** and **Transmit ATIM**

the ATIM frame for the same data frame in the next ATIM window. If the Beacon number (BN) is equal to the maximum Beacon Number (BN_{max}) then the data frame will be dropped.

6 Verification And Results

This section presents the verification performed on the MDP model in Section 5 using PRISM (*PRISM*, 2012)

```
// Wait for ATIM ACK, let time pass (state 5)
[time] al=5 & xl < ACK_TO-> (xl'=min(xl+1,TIME_MAX));
[finishl] al=5  & xl>= ACK_TO & cl=1  & AW >=SIFS+1 -> (al'=7) & (xl'=0);
[finishl] al=5  & xl>= ACK_TO & cl=2 & AW >=SIFS+1 -> (al'=6) & (xl'=0);
//[] a1=5  & y1>= ACK_TO & c1=2 & AW >=SIFS+1 -> (a1'=6) & (y1'=0)& (AW'=AW-1);
[finishl] al=5 & xl>= ACK_TO & cl=2 & AW < SIFS-> (al'=9);

// NO ATIM ACK (state 6)
// Z used as a flag bit z=1 (succ) & z=2 (unsucc) trans. of ATIM frame
[time] al=6  & Ail < ai_max & AW >= SIFS  & yl=0 -> (al'=3)& (Ail'=Ail+1) &(xl'=0)&(yl'=0) ;
[] al=6  & Ail >= ai_max & AW >= SIFS  & yl=0-> (al'= 8) & (xl'=0)& (Ail' =0) & (yl'=2);

// ATIM-ACK RCV (state 7)
[] al =7 & yl=0  -> (al'=8)&(yl'=1);

//Wait for End of the ATIM window (state 8)
[time] al=8 & xl<ASLOTTIME -> (xl'=min(xl+1,TIME_MAX));
[] al =8 & xl=ASLOTTIME & AW > ASLOTTIME -> (al'=8)& (AW'=AW-1) & (xl'=0);
[] al =8 & AW = ASLOTTIME & yl=1 -> (al'=10)&(yl'=0);
[] al =8 & AW = ASLOTTIME & yl=2 -> (al'=9) &(yl'=0) ;
```

Figure 9 Transitions in PRISM specification from **Wait ATIM-ACK**, **ATIM-ACK RCV**, **No ATIM-ACK** and **Wait End of ATIM window**

Table 3 Size of model for data frames of different length

	2 stations		3 stations		4 stations	
Data Packet size (bytes)	**States**	**Tran.**	**States**	**Tran.**	**States**	**Tran.**
64	71906	73329	96023	105990	177873	238202
128	72049	73472	96439	106406	179992	240633
256	72324	73747	97239	107206	184067	245308
512	72874	74297	98839	108806	192217	254658
1024	73974	75397	102039	112006	213893	284950

and discusses the results obtained. The objective of this work is to analyze the performance of IEEE 802.11 DCF in IBSS power save mode, *i.e.*, the time required for a station for transmitting the data frame successfully after the successful transmission of an ATIM frame. More specifically, as this protocol is probabilistic in nature, the expected time that the stations complete their transmission successfully is calculated. The expected power consumption is also computed. The probabilistic model checker PRISM version 4.0.3 was used for carrying out the model checking. structure. The *Gauss-Seidel* method was used for convergence of iterations (*PRISM*, 2012). The probabilistic temporal logic PCTL was used to express the expected reachability properties. The model checker PRISM supports assignment of costs to states and transitions of the model to compute the expected cost before reaching the final states. In this presented model, time and power consumption are measured as cost to reach the final states. In case of expected time calculation, we assign a cost of $20\mu s$ to all *time* action and zero to others. For expected energy consumption, the values assigned to different states (transmit/receive, idle, sleep) are taken from the data-sheet of CISCO Aironet 350 Series Client Adapters (*CISCO*, 2011).

The expected reachability properties verified are as follows:

ER1: The minimum expected time until all stations successfully complete their data frame transmissions:
$R\{\text{"time"}\}min =?[F$ s1= DONE $\wedge$ s2=DONE $\cdots \wedge$ sn=DONE].

ER2: The minimum expected time for any one station to successfully complete its data frame transmission:
$R\{\text{"time"}\}min =?[F$ s1= DONE|s2=DONE $\cdots$ |sn=DONE].

ER3: The minimum expected energy consumption when all the stations successfully complete their data frame transmissions :
$R\{\text{"power"}\}min =?[F$ s1= DONE $\wedge$ s2=DONE $\cdots \wedge$ sn=DONE].

ER4: The minimum expected energy consumption when any one station successfully completes its data frame transmission :
$R\{\text{"power"}\}min =?[F$ s1= DONE|s2=DONE $\cdots$ |sn=DONE].

Here s_i is a state variable of the i^{th} station in the MDP indicating the stage of the protocol.

The system parameters used in the model are listed in Table 2. The ATIM and date window sizes are $2ms$ and $18ms$, respectively. A time scale abstraction is used to further reduce the state space. In Table 2 the slot time

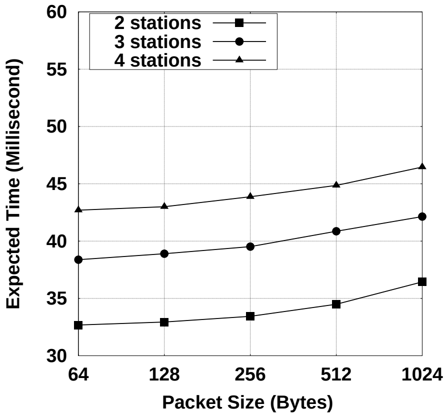

Figure 10 ER1

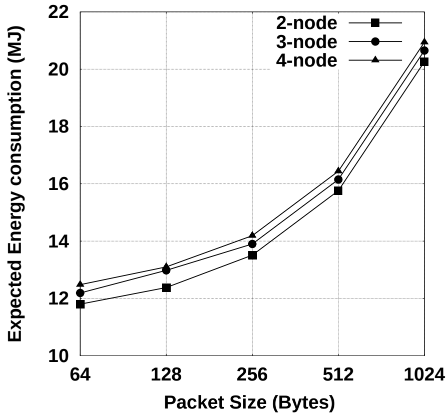

Figure 12 ER3

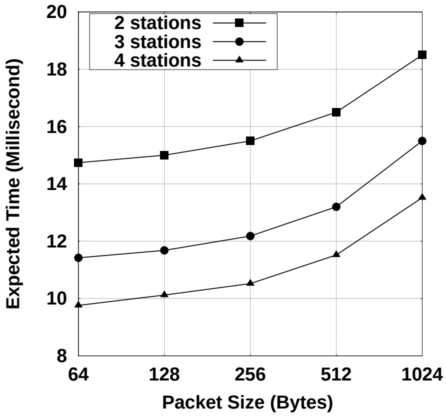

Figure 11 ER2

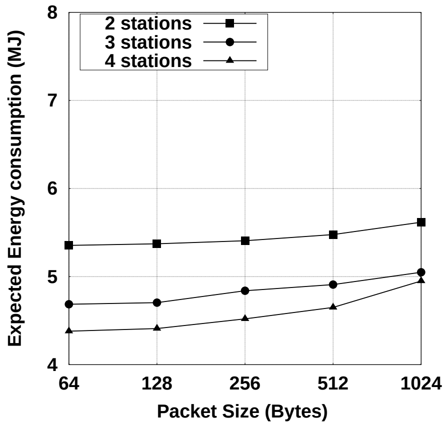

Figure 13 ER4

is assumed to be $20\mu s$, so all constants are first divided by 20 then rounded up and down.

Table 3 shows statistics of the models for different network size (2 stations, 3 stations, 4 stations) along with the number of states and transitions. Different packet size (bytes) are considered, in particular the values 64 bytes, 128 bytes, 256 bytes, 512 bytes and 1024 bytes (again rounding by a factor of 20 μs and 2 Mbps data rate). It can be inferred from Table 3 that the number of stations and the size of the data packet result in an increase in the model size. We note that the above reachability properties for 5 stations could not be verified within 2 GB of memory.

Figure 10 shows the results of the expected time until all stations successfully complete their data frame transmission against different data frame sizes (property ER1). Here the data rate is 2 Mbps. Similarly, Figure 11 presents the relationship between the expected time for any one of the stations to completes its data frame transmission and data frame sizes. It can be noted from Figure 11 that the minimum expected time for any one station to successfully complete its data frame transmission (property ER2) for a network with 4 stations is less than the corresponding expected time for a network with 3 stations and 2 stations. This is because when the network size is very small (here the maximum network size is 4 stations), the packet collision in the channel is negligible and the expected time for any one of the successful transmission improves with the increase in the number of stations.

Energy consumption is an essential point to analyze the performance of IEEE 802.11 power save mode and the properties used for energy analysis allow insight into the behavior. Figure 12 shows the results for the minimum expected energy consumption when all the stations successfully complete their data frame transmissions (property ER3) against different data frame sizes. The results that are presented in Figure 12 make a comparison of expected power consumption between different network size. Similarly, Figure 13 presents the minimum expected energy consumption when any one station successfully completes its data frame transmission (property ER4).

7 Conclusion

This paper presents the application of probabilistic model checking to the IEEE 802.11 IBSS Power Save

Mode. The Markov decision process model is designed for a few stations and a channel which is shared by the stations. Two reachability properties relating to expected time to deliver a packet and expected energy consumption are verified. The impact of the size of data frame on the two properties and the impact on model construction are also demonstrated.

References

Qualnet 5.0.1 Network Simulator, `http://www.scalable-networks.com/products/qualnet/`, 2011.

Cisco Aironet 350 Series Client Adapters, `http://www.cisco.com/en/US/prod/collateral/wireless/ps6442/ps4555/ps448/product_data_sheet09186a0080088828.html`, 2011.

PRISM, `http://www.prismmodelchecker.org/.`, 2012.

Alshanyour, A., and Agarwal, A., Three-dimensional markov chain model for performance analysis of the IEEE 802.11 distributed coordination function, in *28th IEEE conference on Global telecommunications (GLOBECOM)*, pp. 2510–2516, IEEE, 2009.

Alur, R., and Dill, D. L., A theory of timed automata, *Theoretical Computer Science*, *126*(2), 183–235, 1994.

Baier, C., Clarke, E., Hartonas-Garmhausen, V., Kwiatkowska, M., and Ryan, M., Symbolic model checking for probabilistic processes, in *Proc. 24th International Colloquium on Automata, Languages and Programming (ICALP'97)*, vol. 1256 of *LNCS*, pp. 430–440, Springer, 1997.

Bianchi, G., Performance analysis of the IEEE 802.11 distributed coordination function, *IEEE Journal on Selected Areas in Communications*, *18*(3), 535–547, 2000.

Clarke, E., Emerson, E., and Sistla, A., Automatic verification of finite-state concurrent systems using temporal logic specifications, *ACM Transactions on Programming Languages and Systems*, *8*(2), 244–263, 1986.

Duflot, M., Fribourg, L., Herault, T., Lassaigne, R., Magniette, F., Messika, S., Peyronnet, S., and Picaronny, C., Probabilistic model checking of the CSMA/CD protocol using PRISM and APMC, *Theoretical Computer Science*, *128*(6), 195–214, 2005.

Duflot, M., Kwiatkowska, M., Norman, G., and Parker, D., A formal analysis of bluetooth device discovery, *STTT*, *8*(6), 621–632, 2006.

Fruth, M., Probabilistic model checking of contention resolution in the ieee 802.15.4 low-rate wireless personal area network protocol, in *ISoLA*, pp. 290–297, IEEE, 2006.

Hansson, H., and Jonsson, B., A logic for reasoning about time and reliability, *Formal Asp. Comput.*, *6*(5), 512–535, 1994.

IEEE, *IEEE Std 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, 2007.

Jung, E. S., and Vaidya, N. H., Energy efficient MAC protocol for wireless LANs, in *IEEE INFOCM*, 2002.

Kwiatkowska, M., Norman, G. and Sproston, J., Probabilistic model checking of the ieee 802.11 wireless local area network protocol, in *Process Algebra and Probabilistic Methods: Performance Modeling and Verification*, vol. 2399 of *LNCS*, pp. 169–187, Springer, 2002a.

Kwiatkowska, M., Norman, G., Segala, R. and Sproston, J., Automatic verification of real-time systems with discrete probability distributions, *Theoretical Computer Science*, *282*, 101–150, 2002b.

Kwiatkowska, M., Norman, G., and Sproston, J., Probabilistic model checking of deadline properties in the IEEE 1394 firewire root contention protocol, *Formal Asp. Comput*, *14*(3), 295–318, 2003.

Norman, G., Parker, D., Kwiatkowska, M., Shukla, S. and Gupta, R., Using probabilistic model checking for dynamic power management, *Formal Asp. Comput*, *17*(2), 160–176, 2005.

Roy, A., and Gopinath, K., Improved probabilistic models for 802.11 protocol verification, in *Proc. 17th International Conference on Computer Aided Verification (CAV'05)*, vol. 3576 of *LNCS*, pp. 239–252, Springer, 2005.

Rutten, J. J. M. M., Kwiatkowska, M., Norman, G. and Parker, D., *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems*, American Mathematical Society, 2004.

Kwiatkowska, M., Norman, G. and Parker, D., Stochastic model checking. In Formal Methods for Performance Evaluation, in *Proceedings of the th International School on Formal Methods for the Design of Computer, Communication and Software Systems*, pp. 220–270, Springer, 2007.

Kwiatkowska, M., Norman, G. and Parker, D., PRISM 4.0: Verification of Probabilistic Real-time Systems, in Proceedings *23rd International Conference on Computer Aided Verification (CAV'11)*, pp. 585–591, Springer, 2011.

Swain, P., Chakraborty, S., Nandi, S. and Bhaduri, P. , Performance analysis of IEEE 802.11 IBSS power save mode using a discrete-time markov model, in *Proceedings of the ACM Symposium on Applied Computing, SAC 2012, Italy*, pp. 631–633, ACM, 2012.

Miner, A.and Parker, D., Symbolic representations and analysis of large probabilistic systems, in *Validation of Stochastic Systems*, pp. 296–338, Springer, 2004.

Zheng, R., Hou, J. C. and Sha, J. C., Performance analysis of the IEEE 802.11 power save mode, in *Proceedings of IEEE International Symposium of Computer Networks and Distributed Systems*, 2004.