Probabilistic Model Checking of Contention Resolution in the IEEE 802.15.4 Low-Rate Wireless Personal Area Network Protocol

Matthias Fruth
School of Computer Science, University of Birmingham,
Edgbaston, Birmingham, B15 2TT, United Kingdom
m.fruth@cs.bham.ac.uk

***Abstract*—The international standard IEEE 802.15.4 defines low-rate wireless personal area networks, a central communication infrastructure of pervasive computing. In order to avoid conflicts caused by multiple devices transmitting at the same time, it uses a contention resolution algorithm based on randomised exponential backoff that is similar to the ones used in IEEE 802.3 for Ethernet and IEEE 802.11 for Wireless LAN.**

We model the protocol using probabilistic timed automata, a formalism in which both nondeterministic and probabilistic choice can be represented. The probabilistic timed automaton is transformed into a finite-state Markov decision process via a property-preserving integral-time semantics. Using the probabilistic model checker PRISM, we verify correctness properties, compare different operation modes of the protocol, and analyse performance and accuracy of different model abstractions.

I. Introduction

Low-rate wireless personal area networks (WPANs) are a central communication infrastructure for pervasive computing. The recently published standard ZigBee [1] defines the upper network layers, while the lower layers are described in the IEEE standard 802.15.4 [2].

Crucial for the efficiency of a wireless network protocol is its contention resolution mechanism. When more than one station attempts to transmit a frame at the same time, a *collision* occurs, and subsequently all frames get corrupted. The standard mechanism for contention resolution in computer networks is called *carrier-sense multiple access (CSMA)*. CSMA algorithms attempt to break symmetries of failing transmissions being restarted at almost the same time by using randomised binary exponential backoff procedures. While wired devices can listen during their own transmissions and employ CSMA with collision detection (CSMA/CD), stations in wireless networks usually cannot listen to their own transmissions, and consequently colliding transmissions can only be detected after they have been completed. Thus wireless devices use CSMA with collision avoidance (CSMA/CA or CSMA-CA).

The contention resolution algorithm in IEEE 802.15.4 (CSMA-CA) is a variant of those used in IEEE 802.3 Ethernet (CSMA/CD) and IEEE 802.11 Wireless LAN (CSMA/CA). It contains a more complex logical structure than the other two, but involves much smaller numerical values, and is therefore more feasible for formal verification. Since wireless sensor networks are increasingly often used in safety critical applications, formal analysis is essential. So far, no comprehensive study of the CSMA-CA contention resolution protocol of the IEEE 802.15.4 networking standard has been published.

Probabilistic model checking is a technique for the automatic verification of probabilistic properties. Given a probabilistic model of a protocol, expressed as a probabilistic timed automaton with digital clocks, we can verify qualitative properties such as "the maximum probability of at most k collisions is 0.9" and compute quantitative properties such as "the expected time until two contending stations complete their transmissions successfully".

In this case study, we apply a range of performance measures to different scenarios in order to evaluate how the operation of low-rate wireless personal area networks is affected by different settings of protocol attributes and how model abstractions affect accuracy and complexity of probabilistic model checking. This work follows previous case studies of IEEE 802.3 [3], [4] and IEEE 802.11 [5]. Our modelling approach is based on an integral-time semantics for probabilistic timed automata [6]. We use the probabilistic model checker PRISM [7], which has proven to be successful in a wide range of case studies [8], [9].

This paper is divided into five sections. In the next section, we give an informal description of the IEEE 802.15.4 networking standard and the CSMA-CA contention resolution protocol. In Section 3, we define syntax and semantics of probabilistic timed automata and their representation in PRISM. Section 4 contains network configuration, modelling assumptions, and probabilistic timed automata of our models. In Section 5, we present our verification results. Section 6 concludes the paper.

Additional information about this case study can be found on the PRISM website [9].

II. Contention Resolution in IEEE 802.15.4

This section briefly introduces the IEEE 802.15.4 networking standard, defines its contention resolution protocol, and lists relevant numerical attributes.

A. The Networking Standard IEEE 802.15.4

The international standard IEEE 802.15.4 [2] defines *low-rate wireless personal area networks (LR-WPANs)* as struc-

Table I
NUMERICAL ATTRIBUTES IN IEEE 802.15.4

Attribute	Value
CCA duration	8 symbol periods
PHY acknowledgement frame length	11 octets
PHY beacon frame length	23–100 octets
PHY data frame length	15–133 octets
aBaseSlotDuration	60 symbol periods
aMaxBE	5
aMaxFrameRetries	3
aMaxSIFSFrameSize	18 octets
aMinCAPLength	440 symbol periods
aMinLIFSPeriod	40 symbol periods
aMinSIFSPeriod	12 symbol periods
aTurnaroundTime	12 symbol periods
aUnitBackoffPeriod	20 symbol periods
macAckWaitDuration	120 or 54 symbol periods[1]
macBeaconOrder	0–15 (default 15)
macMaxCSMABackoffs	0–5 (default 4)
macMinBE	0–3 (default 3)
macSuperframeOrder	0–15 (default 15)

tures of "low data rate wireless connectivity with fixed, portable, and moving devices with no battery or very limited battery consumption requirements typically operating in the personal operating space of 10 meters". Devices conforming to this specification can operate on 27 channels in three frequency bands with bandwidths of 20, 40, and 250 kbit/s.

1) Numerical Attributes: In order to specify timing constraints of the contention resolution protocol, size parameters of the superframe structure, and length restrictions for different frame types, the standard uses a number of numerical attributes. Table I contains all parameters and constants that are used in our models. All values refer to the physical layer, taking into account an additional six *octets*[2] needed to transmit a frame that has been received from the media access control layer.

Depending on the modulation technique used, the transmission of one octet requires different numbers of *symbols*[3]: for the channels 0 to 10 (at 20 and 40 kbit/s), one octet corresponds to 8 symbols, while for the channels 11 to 26 (at 250 kbit/s), it corresponds to 2 symbols.

2) Superframe Structure: In order to synchronise devices and to assign *guaranteed time slots (GTSs)* for low-latency applications and applications requiring a specific data bandwidth, the coordinator can choose to use a *superframe* structure, as shown in Fig. 1. Each superframe consists of 16 equally sized slots and is bounded by network *beacons*, which are transmitted by the coordinator at the beginning of the first slot of each superframe. The superframe is divided into an active and an inactive portion. The former consists of a *contention access period (CAP)* and a *contention free period (CFP)* of guaranteed time slots. The CAP ends at a superframe slot

[1]The former value is used for the channels 0 to 10 and the latter for the channels 11 to 26.

[2]An octet is a grouping of eight bits.

[3]A symbol is the smallest unit of data that can be transmitted on a particular channel. The transmission time for one symbol is one *symbol period*.

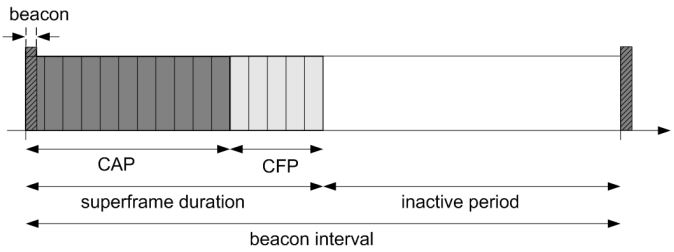

Figure 1. Superframe structure

boundary and has a minimum length of aMinCAPLength, although an exception of the latter is allowed for a temporary increase of the beacon frame to perform *GTS maintenance*[4].

If a superframe structure is used, the network is called *beacon-enabled*, otherwise *nonbeacon-enabled*. In beacon-enabled networks, all communication takes place *indirectly* via a designated *coordinator* device, while stations in nonbeacon-enabled networks can also communicate *directly* in a peer-to-peer mode.

B. The Contention Resolution Protocol CSMA-CA

The CSMA-CA protocol is used only for transmissions of data frames and MAC command frames within the CAP, unless the frame can be quickly transmitted following the acknowledgement of a data request command. It is not used for the transmission of beacon frames, acknowledgement frames, or data frames within the CFP.

Depending on the type of network, the protocol operates in either *slotted* or *unslotted* mode. In beacon-enabled networks, slotted CSMA-CA is used for transmissions between the coordinator and a device. In nonbeacon-enabled networks, or if no beacons can be located in a beacon-enabled network, unslotted CSMA-CA is used. Peer-to-peer transmissions always use unslotted CSMA-CA.

If slotted CSMA-CA is used, a so-called *battery life extension*[5] can be enabled, for which the contention resolution protocol is slightly different. In this paper, however, we only consider the case where the battery life extension is disabled.

The contention resolution protocol consists of the following steps.

a) Initialisation: If a device wishes to transmit a frame using CSMA-CA, it first initialises the local variables $\mathtt{BE} := \mathtt{macMinBE}$ for the *backoff exponent* and $\mathtt{NB} := 0$ for the number of successive backoffs before the current transmission.

b) Backoff: Before a station attempts to send a frame, it has to wait for a random integer number of between 0 and $2^{\mathtt{BE}} - 1$ complete *backoff periods* of length aUnitBackoffPeriod. This process is called *backoff*. If slotted CSMA-CA is used, transmissions are synchronised with the beacon, and therefore the backoff starts at the beginning of the next backoff period; if unslotted CSMA-CA is used, the

[4]A beacon frame performs GTS maintenance by accommodating a list of up to seven descriptors of currently maintained GTSs.

[5]The battery life extension is a mechanism aimed on reducing coordinator receiver operation time during the CAP.

backoff starts immediately. The first backoff period of each superframe starts with the transmission of the beacon. If the backoff has not been completed at the end of the CAP, it resumes at the start of the next superframe.

c) Clear Channel Assessment: After completing its backoff, the station performs a *clear channel assessment (CCA)*. If, after eight symbol periods, the channel is assessed to be busy, both `BE` and `NB` are incremented by one, up to a maximum of `aMaxBE` for `BE` and `macMaxCSMABackoffs` + 1 for `NB`. If `NB` exceeds `macMaxCSMABackoffs`, the protocol terminates with a *channel access failure*; if not, it returns to the backoff step. If the channel is assessed free, the frame can be transmitted. In slotted CSMA-CA, two CCAs, each starting at the beginning of a backoff period, have to be performed.

d) Starting the Transmission: In slotted CSMA-CA, a transmission can only start at a backoff period boundary and only if all steps (two CCAs, frame transmission, and acknowledgement) can be completed at least one IFS period before the end of the CAP.

e) Acknowledgement: If the originator has not requested an acknowledgement, the transmission is assumed to have been successful. If an acknowledgement has been requested, the sender needs `aTurnaroundTime` to switch from sending to receiving mode and vice versa. The recipient starts the transmission of the acknowledgement `aTurnaroundTime` after the reception of the last symbol of the data or MAC command frame if unslotted CSMA-CA is used; it starts at a backoff period boundary between `aTurnaroundTime` and `aTurnaroundTime` + `aUnitBackoffPeriod` after the reception of the last symbol of the data or MAC command frame if slotted CSMA-CA is used. If the originator receives an acknowledgement from the recipient within a time of `macAckWaitDuration`, the data transfer has been successful. If no acknowledge is received within that time, the frame will be retransmitted up to a maximum of `aMaxFrameRetries` times, after which the protocol terminates and a *communications failure* is issued.

III. Probabilistic Timed Automata

Probabilistic timed automata [10] are a powerful modelling formalism for distributed systems that supports dense time, nondeterminism, and probabilitic choice. They are a generalisation of *timed automata* [11] that is obtained by adding a probabilistic transition relation. We include the notion of *urgent events* [5], a common feature of classical timed automata [12], [13]. Our presentation follows [6].

A. Syntax

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 $v : \mathcal{X} \to \mathbb{T}$ is referred to as a *clock valuation*. For any $v : \mathcal{X} \to \mathbb{T}$ and $t \in \mathbb{T}$, the clock valuation $v \oplus t$ denotes the *time increment* for v with t.

Let $C(\mathcal{X})$ be the set of *clock constraints* 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 $c \in \mathbb{N}$. A clock valuation $v \in \mathbb{T}^{\mathcal{X}}$ satisfies a clock constraint ζ if and only if ζ resolves to true after substituting each clock $x \in \mathcal{X}$ with the corresponding value $v(x)$. Note that we consider the syntax of *closed, diagonal-free clock constraints*, which does not allow atomic constraints of the form $x > c$ or $x < c$ (not closed) or $x - y \sim c$ (not diagonal-free).

A *discrete probability distribution* over a countable set Q is a function $\mu : Q \to [0, 1]$ such that $\Sigma_{q \in Q} \mu(q) = 1$. For a possibly uncountable set Q', let $\mathsf{Dist}(Q')$ be the set of discrete probability distributions over countable subsets of Q'.

A *probabilistic timed automaton* is a tuple $\mathsf{PTA} = (L, \bar{l}, \mathcal{X}, \Sigma, inv, prob)$ 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*, of which $\Sigma_u \subseteq \Sigma$ are *urgent*;
- $inv : L \to \mathcal{C}(\mathcal{X})$ is the *invariant condition* function;
- $prob \subseteq L \times C(\mathcal{X}) \times \Sigma \times \mathsf{Dist}(2^{\mathcal{X}} \times L)$ is the *probabilistic transition relation*.

B. Semantics

In each location of a probabilistic timed automaton, there is a nondeterministic choice between two types of transitions: *Delay transitions* correspond to the elapsing of time in a location. They are permitted as long as the invariant condition is satisfied and no urgent transitions (transitions under urgent events) are enabled. *Event transitions* 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' to 0 and moving to the location l'.

This notion of a probabilistic timed automaton is strong enough to represent several higher-level features such as *urgent locations* and *integer variables*. In urgent locations, only event transitions are allowed, that is, such locations have to be left immediately without time passing. They can be modelled using an additional clock [13], [14]. Integer variables with bounded ranges, which can be tested within enabling conditions and reset by event transitions, can be represented by encoding their values within locations [15].

Formally, the semantics of probabilistic timed automata is defined in terms of *timed probabilistic systems* [6]. The traditional *dense-time semantics*, where $\mathbb{T} = \mathbb{R}$ and $\oplus = +$, is generally uncountable. Kwiatkowska et al. [6] have extended the concept of a finite *integral-time semantics* [16] from classical to probabilistic timed automata: for $\mathbb{T} = \mathbb{N}$ and $\oplus = \oplus_{\mathbb{N}}$, let $v \oplus_{\mathbb{N}} t \stackrel{\text{def}}{=} \min\{v(x) + t, \mathbf{k}_x + 1\}$, with $\mathbf{k}_x$ denoting the largest value that the clock $x \in \mathcal{X}$ is compared to in all clock constraints of PTA.

C. Representation in PRISM

When automatic verification techniques are applied to complex systems, abstraction methods can help to achieve theoretical and practical feasibility.

Probabilistic model checking requires a finite system model. Models of probabilistic timed automata obtained using the

presented integral-time semantics always have a finite number of states. By removing delay transitions where the delay is zero, finite branching can also be ensured, and hence the model is finite. Kwiatkowska et al. [6] have shown that the integral-time semantics preserves probabilistic reachability and expected reachability properties of closed, diagonal-free probabilistic timed automata.

Unfortunately, the integral-time semantics contributes to the state explosion problem, as it leads to models of a size exponential in the number of clocks and the largest constant that the clocks are compared to. In order to cope with that, *timescale abstraction* can be used to reduce the size of a model by dividing all constants clocks are compared to by the value of a new time unit and then rounding lower bounds down and upper bounds up. Alur et al. [17] have shown that the original model is a refinement of the reduced model, and thus the maximum and minimum probabilistic and expected probability measures of the reduced model are upper and lower bounds of those for the original model.

Finite probabilistic timed automata are represented in PRISM as *Markov decision processes* [18], a formalism that supports nondeterminism and probabilistic choice. Due to the compositionality property of the integral-time semantics [6], a parallel composition of probabilistic timed automata can be modelled as the parallel composition of their respective Markov decision processes.

IV. Modelling

In this section, we present basic network configuration, modelling assumptions, and probabilistic timed automata for our models of the CSMA-CA contention resoultion protocol.

A. Network Configuration

For all scenarios, we consider a personal area network consisting of a fixed configuration of sending and receiving devices. Each sending station s_i intends to send, using CSMA-CA, a single data frame to its corresponding receiving station r_i. Both stations start sending at the same time. As in the probabilistic timed automata model for contention resolution in the IEEE 802.11 protocol [5], we conclude that the behaviour of the destination stations is deterministic and incorporate it into that of the sending stations, removing the destination stations from the model. Also, we assume a channel of 20 kbit/s bandwidth, which uniquely determines all timing parameters of the model.

Other network activity, such as data transmissions from a coordinator to a station, including indirect transmissions of pending messages by the coordinator, has not been modelled. Communication activity within the CFP has been modelled indirectly as follows: dynamic allocations of GTSs in the CFP lead to varying lengths of the CAP in different superframes. In order to reflect this, the size of the CAP is determined nondeterministically for each superframe. Temporary decreases of the CAP length below `aMinCAPLength` due to GTS maintenance are not modelled.

B. Modelling Assumptions

We have implemented all features of the network protocol in as much detail as possible, considering their contributions to both accuracy of results and complexity of model construction and verification.

1) Ideal Channel: For all models in this case study, we assume a perfect medium and ideal channel conditions, that is, no messages get lost. For beacon-enabled networks, we assume that only one personal area network (PAN) is present in the personal operating space, no PAN conflicts occur, and all stations in the PAN are and remain synchronised; in particular, there are no synchronisation problems related to PAN ID, association and disassociation, or security.

2) Vulnerable Period: Before starting a transmission, stations have to perform a clear channel analysis and to switch from receiving to sending. Concurrent transmissions that start during this period can lead to collisions. As air propagation times of $16-50\mu s$ for one symbol are negligible, we adapted Heindl and German's formula for the vulnerable period [19] to

$$\begin{aligned} VULN &\stackrel{def}{=} CCA + \texttt{aTurnaroundTime} \\ &= \texttt{aUnitBackoffPeriod} \end{aligned}$$

for unslotted CSMA-CA and

$$VULN \stackrel{def}{=} 2 \times \texttt{aUnitBackoffPeriod}$$

for slotted CSMA-CA, where CCA is the duration of a clear channel assessment.

C. Probabilistic Timed Automata Models

In this study, we developed high-level generic models for slotted and unslotted operation mode of the CSMA-CA protocol. Each model is defined as a probabilistic timed automaton, that is, a parallel composition of smaller modules. Various high-level features of timed automata have been used: urgent locations, urgent events, and integer variables. The model for unslotted CSMA-CA (see Fig. 2) consists of three modules: the channel, taken from [5]), and two stations. Beacon synchronisation, which is part of slotted CSMA-CA, is realised using an extra coordinator module and modified station modules.

1) Generic PRISM Models: Although PTAs are already a concise representation formalism, the PRISM code of the models includes additional model details and optimisations: for example, the invariant conditions on transitions have been simplified after expanding high-level features.

Contrary to previous case studies, our models are generic with respect to many aspects of network configuration, transmission types, and timing parameters. This allows a much wider range of scenarios to be investigated. For both unslotted and slotted mode, the channel characteristics (frequency band and modulation technique), as well as the minimum and maximum values for the backoff procedure `macMinBE`, `aMaxBE`, `macMaxCSMABackoffs` can be modified.

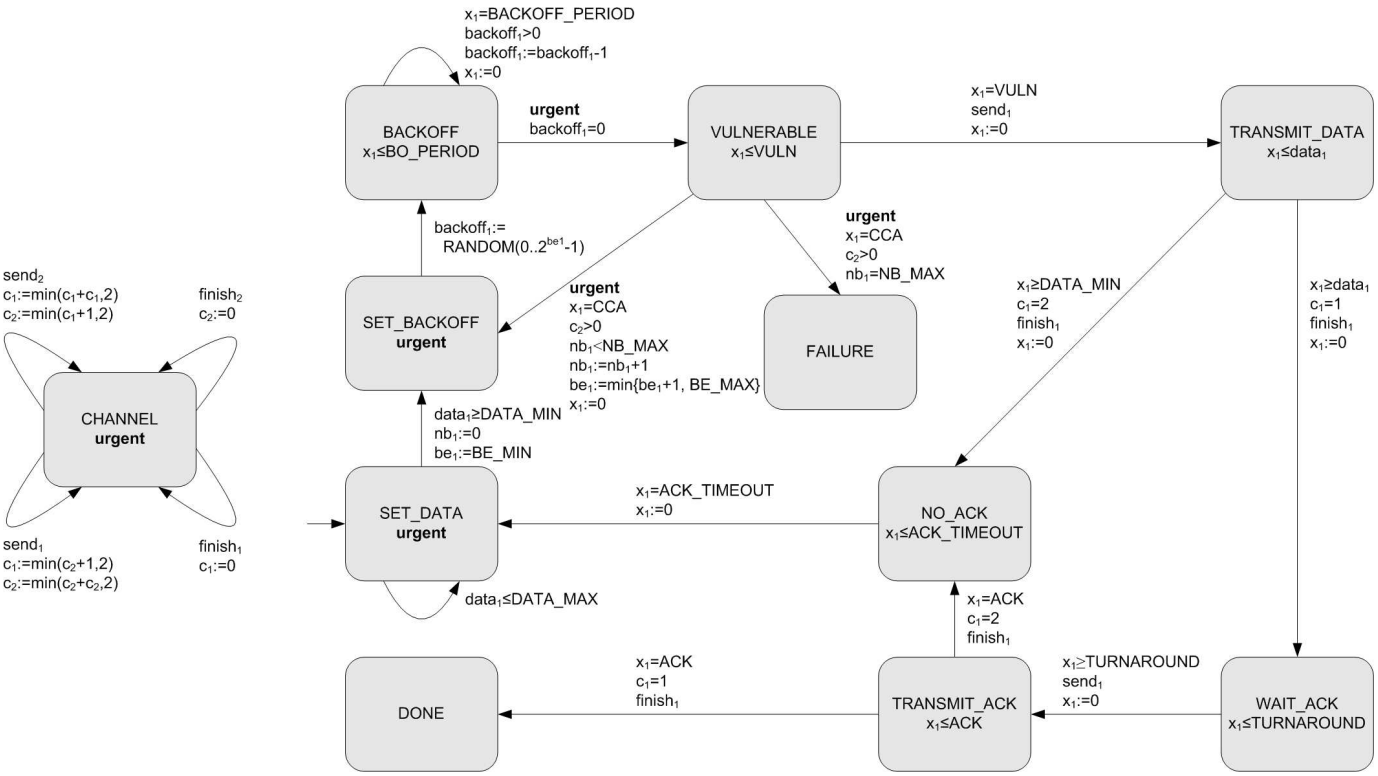

Figure 2. Probabilistic timed automata models for channel and station in unslotted CSMA-CA

In addition to that, beacon synchronisation in the slotted mode can be controlled by modifying the parameters `macBeaconOrder` and `macSuperframeOrder`. Finally, acknowledgements, failures due to too many collisions, and acknowledgement failures can all be separately enabled or disabled.

2) Beacon Synchronisation: For the first time, we have applied probabilistic model checking to a model of a contention resolution protocol that includes beacon synchronisation. The beacon synchronisation process synchronises the timing of all devices in the PAN and defines the lengths of CAP, CFP, and inactive period in the following superframe. Beacon synchronisation is essential for large PANs, and although our model is relatively small, it is generic enough to be adapted to larger scenarios focussing on this particular feature.

3) Timescale Abstraction: When we applied timescale abstraction to the probabilistic timed automata models, we encountered a number of crucial modelling issues. The abstraction granularity (the new unit of time) should be a common divisor of all constants appearing in clock constraints. Otherwise, some constants would have to be rounded up or down. If this affects the same constant in invariant conditions of a sequence of transitions, the imprecisions of these roundings can sum up to more than 1, which may add spurious behaviour to the abstracted model. These implicit delays over sequences of transitions have to be modelled by further approximating the respective constants downwards and upwards.

For example, when we scale down the PTA for slotted CSMA-CA and 20 kbit/s frequency band, using a granularity of `aUnitBackoffPeriod`, all constants are first divided by 20. Consequently, the turnaround time after WAIT_ACK is scaled down from 12–32 symbol periods to 0–2, the acknowledgement time `ACK` from 88 to 4–5, and the acknowledgement timeout time `ACK_TIMEOUT` (that is, the maximum time for turnaround plus acknowledgement) from 120 to 6. We can see that the acknowledgement timeout delay of 6 conflicts with the possible time of $2 + 5 = 7$ for turnaround plus acknowledgement. This problem can be resolved by setting `ACK_TIMEOUT` to 7, since, according to [17], upper bounds may be approximated upwards. This only causes an imprecision of 1 time unit per execution of this transition, which is acceptable since this is the only imprecision caused by timescale abstraction of this model, and it only affects transmissions in slotted CSMA-CA where an acknowledgement is being requested and transmitted successfully. In order to model the concurrent delay constraints `ACK` and `ACK_TIMEOUT` exactly, two clocks would be necessary.

When the granularity for timescale abstraction is not a common divisor of CCA and $VULN$, a similar problem can occur. Then, the period from the beginning to the end of a CCA (which has a duration of 8 symbol periods), where transmissions can still take place without necessarily causing a collision, cannot be distinguished from the point when the 8 symbol periods of the CCA are complete and a collision would take place.

The highest granularity for an exact timescale abstraction

of our models is 4 symbol periods.

V. Verification and Results

In this section, we present our verification methodology, experiments, and results. Using probabilistic model checking, we investigated three aspects of the protocol: the performance impact of beacon synchronisation, the performance impact of the backoff procedure, and the performance and accuracy impact of model abstractions. For our experiments, we used version 3.1 of the probabilistic model checker PRISM, in particular its symbolic verification engine, "MTBDD" [20].

A. Properties

In order to compare different models, we used a set of probabilistic reachability and expected reachability properties that were expressed in the probabilistic temporal logic PCTL [21]. As our models are nondeterministic, probabilistic properties typically refer to minimum and maximum probabilities over all possible adversaries, rather than to one single probability as in the deterministic case.

In order to evaluate the probabilities for a transmission to be finished correctly and for a transmission to contain at least k collisions, we define the probabilistic reachability properties PR1 and PR2 as follows:

- PR1 Minimum probability of both stations successfully completing their transmissions.
- PR2 Maximum probability of at least k collisions.

In order to evaluate the expected number of collisions and the expected time for a correct transmission, PRISM supports *reachability rewards* [20], [22], which we use to define the expected reachability properties ER1 and ER2 as follows:

- ER1 Maximum expected number of collisions until both stations have successfully completed their transmissions.
- ER2 Maximum expected time until both stations have successfully completed their transmissions.

B. Model Abstractions

Owing to the state explosion problem, most of our properties could only be verified in simplified versions of the models. In order to overcome state space explosion, we used the following abstractions.

a) Timescale Abstraction: Though the optimal granularity for timescale abstraction of our models is 4 symbol periods, we have used a granularity of 20 symbol periods except where stated otherwise. This substantially reduces the size of the model without sacrificing much precision and decreases verification times and memory requirements.

b) Fixed Beacon Frame Length: In the slotted-mode model, the length of beacon frames is chosen nondeterministically for each beacon interval. This not only increases the state space immensely, but can also prevent transmissions when both beacon and data frame are large but superframe and CAP are small. This situation can be exploited by pathological adversaries that permanently block transmissions by choosing the respective values for beacon and data frame length.

Although a sufficiently large superframe could be defined by assigning higher values to `macBeaconOrder` and `macSuperframeOrder`, this would only worsen the state space explosion. Instead, we fixed the length of the beacon frame to the minimum value permitted by the specification, and thereby resolved both permanent prevention of transmissions and further state space explosion. Considering our small scenarios of two network stations, this is a reasonable assumption which only slightly reduces the generality of our results.

c) Fixed Data Frame Length: In unslotted CSMA-CA, the length of a data frame is chosen nondeterministically within each beacon interval, while in slotted CSMA-CA, it is chosen nondeterministically before the first transmission and then maintained during possible retransmissions. As in our scenarios each station only sends one message, this nondeterminism can be replaced by data frames of fixed lengths which constitute separate models that can then be analysed separately. This abstraction reduces the state space while preserving our properties.

Table II shows model sizes (in terms of minterms and nodes) and verification results for unslotted and slotted mode models with different abstractions for the data frame length. In order to obtain comparable results for all models, we assumed data transmissions without acknowledgement. For the expected reachability properties, we set the maximum number of successive backoffs `macMaxCSMABackoffs` to infinity.

We observed that many verification tasks only became feasible using fixed data frame lengths and timescale abstraction. Timescale abstraction renders results less precise and should be used with care.

C. Beacon Synchronisation

In order to study the impact of beacon synchronisation, we evaluated our set of properties on models of unslotted and slotted mode using a timescale granularity of 20 symbol periods and data frames of different lengths. For the slotted mode models, the superframe parameters `macBeaconOrder` and `macSuperframeOrder` were set to 1. As expected reachability properties are evaluated to infinity if there exist adversaries where the respective state is not reached (here, $DONE$ is not reached when a transmission fails), we set the maximum number of successive backoffs `macMaxCSMABackoffs` and the maximum number of retransmissions of data frames `aMaxFrameRetries` to infinity. Fig. 3 shows the verification results.

These experiments confirmed our intuition that the slotted mode helps avoiding collisions while slightly increasing the transmission time. The plot for the probability of successful transmission in the slotted mode shows an interesting anomaly: while for small data frames, an average backoff is long enough to avoid most collisions, and for large ones, a station resuming from backoff does not have enough remaining time in the CAP to start a new transmission, a scenario with data frames the length of which is half of the length of the usable (non-beacon) period of the CAP accounts for the worst case.

Table II
PERFORMANCE AND ACCURACY OF DIFFERENT MODEL ABSTRACTIONS

Model	data frame length	stations	time unit	PR1 nodes	PR1 min-terms	PR1 result	ER1 nodes	ER1 min-terms	ER1 result	ER2 nodes	ER2 min-terms	ER2 result
unslotted	fixed[1]	2	4	22k	120k	1.0	22k	120k	0.125	93k	210k	112.8 ms
unslotted	nondet	2	4	180k	960m	1.0	180k	960m	0.125	280k	1.6bn	-[2]
unslotted	fixed[1]	2	20	6.8k	13k	1.0	6.9k	13k	0.125	10k	17k	123.1 ms
unslotted	nondet	2	20	56k	19m	1.0	56k	19m	0.125	83k	26m	123.1 ms
slotted	fixed[1]	2	20	29k	47k	1.0	29k	47k	0.125	45k	67k	166.0 ms
slotted	nondet	2	20	1m	130m	1.0	1m	130m	0.125	1.6m	180m	166.0 ms

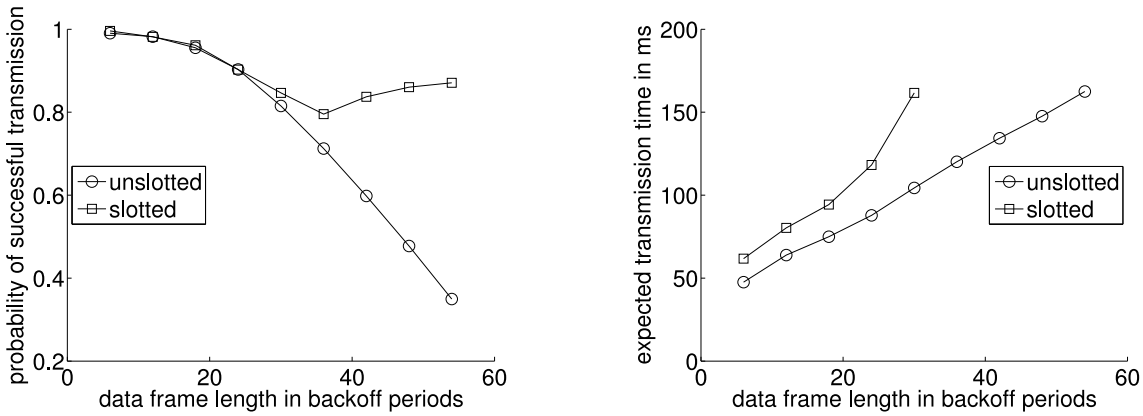

Figure 3. Performance for data frames of different lengths

For large sizes of the data frames and a small size of the superframe, the expected transmission time evaluates to ∞, due to the existence of pathological adversaries where transmissions are not completed successfully. However, for larger superframe sizes, it evaluates to a finite value.

D. Backoff Procedure

In another experiment (see Table III), we have studied the impact of the backoff parameter `macMinBE`, which determines the minimum value of the backoff exponent `BE`, for data frames of different lengths with the same models as in the previous table.

As expected, a high value of `macMinBE` (the default is 3) decreases both collision probability and expected transmission time. However, longer backoff times often result in higher energy consumption. This connection could be investigated further in combination with an analysis of the battery life extension.

VI. CONCLUSIONS

We have presented the first application of probabilistic model checking to the IEEE 802.15.4 networking standard. In a comprehensive case study, we have developed high-level generic models for the CSMA-CA contention resolution protocol, evaluated performance properties, and compared different abstraction techniques, thereby providing a better understanding of both the protocol and modelling issues. Contrary to test and simulation, our formal approach provides provably correct results that cover the full behaviour of the models.

In comparison to previous applications of probabilistic model checking to contention resolution procotols [3]–[5], our models are more realistic but, for that reason, also more complex. We have shown that previous modelling techniques for timescale abstraction are inadequate here and produce pathological adversaries and consequently misleading results.

There are many ways to continue this case study. New scenarios could contain a larger number of stations and more complex behaviours of the stations, such as transmitting more than one message per station and allowing stations to send and to receive. Other interesting features that have not been studied in this work include the optional battery life extension for slotted CSMA-CA, different values of the superframe parameters `macBeaconOrder` and `macSuperframeOrder`, and different channels. Properties describing energy consumption (see, for instance, [23]–[25]) such as minimal, maximal, and expected power consumption for concrete devices could easily be added if data about the device's power state were available.

However, probabilistic model checking is always limited by the only too apparent state space explosion problem. Efficient techniques to deal with that, such as abstraction, symmetry reduction, partial-order reduction, symbolic representations, and induction could greatly improve the scalability of this

[1]The values given for *nodes* and *minterms* are for data frames of maximal length.

[2]This property could not be verified within 2 GB of memory.

Table III
PERFORMANCE FOR DIFFERENT VALUES OF macMinBE

Model	data frame length	macMinBE	PR1	PR2					ER1	ER2
				$k=0$	$k=1$	$k=2$	$k=3$	$k=4$		
unslotted	6	0	0	1	1	1	1	1	∞	∞
unslotted	6	1	0.7361	1	0.5817	0.3293	0.1828	0.0999	1.3094	66.50 ms
unslotted	6	2	0.9287	1	0.3784	0.1300	0.0424	0.0134	0.5698	53.87 ms
unslotted	6	3	0.9904	1	0.2165	0.0438	0.0087	0.0017	0.2710	47.53 ms
slotted	6	0	0	1	1	1	1	1	∞	∞
slotted	6	1	0.8353	1	0.5817	0.3169	0.1668	0.0862	1.2823	75.89 ms
slotted	6	2	0.9735	1	0.3819	0.1276	0.0397	0.0119	0.5712	65.24 ms
slotted	6	3	0.9959	1	0.1887	0.0346	0.0063	0.0010	0.2313	61.73 ms
unslotted	54	0	0	1	1	1	1	1	∞	∞
unslotted	54	1	0	1	0.5003	0.2502	0.1251	0.0625	1.0706	212.6 ms
unslotted	54	2	0.0954	1	0.2653	0.0667	0.0168	0.0042	0.4018	172.5 ms
unslotted	54	3	0.3495	1	0.1601	0.0217	0.0029	0.0004	0.2115	162.5 ms
slotted	54	0	0	1	1	1	1	1	∞	∞
slotted	54	1	0.4207	1	0.5	0.5	0.5	0.5	∞	∞
slotted	54	2	0.7288	1	0.25	0.25	0.25	0.25	∞	∞
slotted	54	3	0.8710	1	0.125	0.125	0.125	0.125	∞	∞

approach to real-world scenarios.

ACKNOWLEDGEMENTS

The author would like to thank Marta Kwiatkowska, Gethin Norman, Dave Parker, and the anonymous referees for many valuable comments and suggestions. This work was partially supported by the EPSRC grants EP/D076625/1 and EP/D07956X/1.

REFERENCES

[1] ZigBee Alliance, "ZigBee Specification 1.0," 2005.

[2] IEEE Computer Society, "IEEE Standard 802.15.4-2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," 2003.

[3] M. Kwiatkowska, G. Norman, J. Sproston, and F. Wang, "Symbolic Model Checking for Probabilistic Timed Automata," in *Proc. Joint Conf. on Formal Modelling and Analysis of Timed Systems and Formal Techniques in Real-Time and Fault Tolerant Systems (FORMATS-FTRTFT 2004)*, ser. LNCS, vol. 3253. Springer, 2004, pp. 293–308.

[4] M. Duflot, L. Fribourg, T. Herault, R. Lassaigne, F. Mangiette, S. Messika, S. Peyronnet, and C. Picaronny, "Probabilistic Model Checking of the CSMA/CD Protocol Using PRISM and APMC," *Electronic Notes in Theoretical Computer Science*, vol. 128, no. 6, pp. 195–214, 2005.

[5] M. Kwiatkowska, G. Norman, and J. Sproston, "Probabilistic Model Checking of the IEEE 802.11 Wireless Local Area Network Protocol," in *Proc. 2nd Joint International Workshop on Process Algebra and Probabilistic Methods and Performance Modeling in Verification (PAPM-PROBMIV 2002)*, ser. Lecture Notes in Computer Science, vol. 2399. Springer, 2002, pp. 169–187.

[6] M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston, "Performance Analysis of Probabilistic Timed Automata using Digital Clocks," *Formal Methods in System Design*, vol. 29, no. 1, pp. 33–78, 2006.

[7] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: A Tool for Automatic Verification of Probabilistic Systems," in *Proc. 12th International Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2006)*, ser. Lecture Notes in Computer Science, vol. 3920. Springer, 2006, pp. 441–444.

[8] M. Kwiatkowska, G. Norman, and D. Parker, "Probabilistic model checking in practice: Case studies with PRISM," *ACM Performance Evaluation Review*, vol. 32, no. 4, pp. 16–21, 2005.

[9] PRISM website. [Online]. Available: http://www.cs.bham.ac.uk/~dxp/prism/

[10] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston, "Automatic verification of real-time systems with discrete probability distributions," *Theoretical Computer Science*, vol. 282, no. 1, pp. 101–150, 2002.

[11] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[12] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "A User Guide to HyTech," in *Proc. 1st International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, ser. Lecture Notes in Computer Science, vol. 1019. Springer, 1995, pp. 41–71.

[13] C. Daws and S. Yovine, "Two examples of verification of multirate timed automata with Kronos," in *Proc. 16th IEEE Real-Time Systems Symp. (RTSS 1995)*. IEEE Computer Society Press, 1995, pp. 66–75.

[14] S. Tripakis, "Timed Diagnostics for Reachability Properties," in *Proc. 5th International Conf. on Tools and Algorithms for Construction and Analysis of Systems (TACAS 1999)*, ser. Lecture Notes in Computer Science, vol. 1579. Springer, 1999, pp. 59–73.

[15] ——, "The formal analysis of timed systems in practice," PhD thesis, Université Joseph Fourier, 1998.

[16] D. Beyer, "Improvements in BDD-Based Reachability Analysis of Timed Automata," in *Proc. 10th International Symp. of Formal Methods Europe (FME 2001)*, ser. Lecture Notes in Computer Science, vol. 2021. Springer, 2001, pp. 318–343.

[17] R. Alur, A. Itai, R. P. Kurshan, and M. Yannakakis, "Timing Verification by Successive Approximation," *Information and Computation*, vol. 118, no. 1, pp. 142–157, 1995.

[18] C. Derman, *Finite state Markovian decision processes*. Academic Press, 1970.

[19] A. Heindl and R. German, "Performance modeling of IEEE 802.11 wireless LANs with stochastic Petri nets," *Performance Evaluation*, vol. 44, no. 1–4, pp. 139–164, 2001.

[20] PRISM manual. [Online]. Available: http://www.cs.bham.ac.uk/~dxp/prism/manual/

[21] H. Hansson and B. Jonsson, "A Logic for Reasoning about Time and Reliability," *Formal Aspects of Computing*, vol. 6, no. 5, pp. 512–535, 1994.

[22] M. Kwiatkowska, G. Norman, and A. Pacheco, "Model Checking Expected Time and Expected Reward Formulae with Random Time Bounds," *Computers and Mathematics with Applications*, vol. 51, no. 2, pp. 305–316, 2006.

[23] N. F. Timmons and W. G. Scanlon, "Analysis of the Performance of IEEE 802.15.4 for Medical Sensor Body Area Networking," in *Proc. 1st IEEE Communications Society Conf. on Sensor and Ad Hoc Communications and Networks (SECON 2004)*. IEEE Computer Society Press, 2004, pp. 16–24.

[24] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene, "Energy Efficiency of the IEEE 802.15.4 Standard in Dense Wireless Microsensor Networks: Modeling and Improvement Perspectives," in *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE 2005)*. IEEE Computer Society Press, 2005, pp. 196–201.

[25] C. K. Singh and A. Kumar, "Performance Evaluation of an IEEE 802.15.4 Sensor Network with a Star Topology," to be published.