(Towards) Model Checking Medium Access Control for Sensor Networks

Paolo Ballarini and Alice Miller

Department of Computing Science
University of Glasgow
Glasgow, Scotland.

{paolo,alice}@dcs.gla.ac.uk

Abstract. We describe initial attempts to verify S-MAC, a medium access control protocol designed for wireless sensor networks, by means of the PRISM model checker. The S-MAC protocol is built on top of the IEEE 802.11 standard for wireless ad hoc networks and, as such, it uses the same randomised backoff procedure as a means to avoid collision. In order to minimise energy consumption, in S-MAC, nodes are periodically put into a sleep state, Synchronisation of the sleeping schedules is necessary for the nodes to be able to communicate. Intuitively, energy saving obtained through a periodic sleep mechanism will be at the expense of performance. In previous work on S-MAC verification [25], a combination of analytical techniques and simulation has been used to confirm the correctness of this intuition for a simplified (abstract) version of the protocol in which the initial schedules coordination phase is assumed correct. We discuss initial steps towards applying model checking to the S-MAC protocol. Our aim is twofold: to confirm the results that refer to the abstract version of the protocol and to study the effect of coordinated sleeping on the protocol performance.

Keywords

Verification; sensor networks; PRISM; medium access control

1 Introduction

The authors of this paper form part of the research team of a multi-site UK-based collaborative project, DIAS \(^1\) (Design, Implementation and Adaptation of Sensor Networks). The major goal of this project is to develop methods and tools for the construction of entire environmental sensor network systems. One of the intentions of the project is to ensure that networks are optimal with respect to a global cost function specified by the network designer. Part of this global cost function will be energy consumption. Within the DIAS project a wireless

---

\(^1\) funded under the Engineering and Physical Sciences Research Council WINES programme
sensor network (WSN) design is characterised with respect to four dimensions, namely: application, management, networking and operating system. The final goal is to adopt a co-design approach through which design solutions should be evaluated (with respect to the inherent global cost), by considering combinations of different possibilities for each of the four dimensions.

A fundamental objective of the DIAS project is to study how/what formal methods can be applied to the specification and verification of the various dimensions that characterise a WSN design.

Validation of a system can take the form of verification (proving that properties of a system hold) or simulation (observation that a system behaves as expected). Mathematical analysis can, in some cases, be used to study a system. The major advantage of this approach is that it allows one to generate general properties of a system (that are independent of the system execution). However, it can only be applied to relatively simple systems – the difficulty of finding an analytical result is proportional to the system complexity.

Modelling techniques, on the other hand, require a model of the system to be built in terms of a specific formalism (e.g. Process Algebras, Petri Nets, Queuing Networks) which is subsequently analysed through suitable methods. The complexity of model analysis is proportional to the size of the associated state-space. Automatic verification methods, such as, for example model checking, are based on an exhaustive search of the associated state-space to establish whether a given property holds for the modelled system. Whenever the model is too large to model check, simulation, by which estimates are derived through a (possibly large) non-exhaustive, state-space exploration, can be used.

In this paper we illustrate a modelling approach tailored to the automatic verification of communication protocols for wireless networks. Specifically we concentrate on a comparative analysis of the IEEE 802.11 protocol for wireless ad-hoc network versus the S-MAC protocol, an 802.11 variant designed to preserve energy in WSN. We show how models can be incrementally built in the Reactive Module language of the PRISM tool, starting from the (simpler) 802.11 and extending it to three versions of the S-MAC protocol corresponding to different levels of detail. We then describe how model analysis can be performed through probabilistic model checking, by verifying specific temporal formulae against each of the models.

The paper is organised in the following way. In the next section we report on related work; in Section 3 an informal description of both IEEE 802.11 and S-MAC is provided; a brief introduction to probabilistic model checking is given in Section 4; in Section 5 the modelling approach is described and the different versions of the model considered. A summary of our contribution is contained in Section 6 together with a discussion on future work.

Note that, at the time of submission, verification is still in process and so no numerical results are included here. A more complete version of this work, including numerical analysis, will be presented at the ISOLA symposium.
2 Related Work

There have been several attempts to verify communication protocols for wireless networks. In [12], simulation of Stochastic Petri Nets models is used to compare the effect of two different Distributed Coordination Functions (DCF), namely Basic Access (BA) and Request-To-Send/Cancel-To-Send (RTS/CTS), on system performance, in terms of throughput and waiting time. The models are very detailed and capture most aspects of the coordination protocols including the timing synchronisation procedure (used to maintain synchronisation amongst local clocks), as well as the basic backoff procedure for collision avoidance.

In [17], automatic verification through model-checking is used for the verification of the 802.11 standard. The focus here is on assessing the performance of the BA, two-way handshake, coordination function, which is achieved through verification of soft-deadline properties expressed in terms of the PCTL temporal logic. For that purpose the authors construct a probabilistic model (a Markov Decision Process) referring to a specific, fixed topology consisting of two sending and two destination stations.

The S-MAC protocol, introduced in [25] uses the same basic collision avoidance mechanism as IEEE 802.11, but involves a sleep state in which a station radio is switched off to preserve energy. In [25] protocol validation is achieved by a combination of analytical results and results obtained through simulation of a protocol implementation under TOSSIM [19], a software framework used to develop and simulate applications for WSN based on the TinyOS [13] operating system. The authors base their experiments on two different topologies (a two-hop and a multi-hop network) and show that the use of S-MAC results in a trade-off between energy saving and latency. However their models do not account for the S-MAC schedule coordination phase (a single sleep/listen schedule is assumed to be shared by all stations and the agreement/maintenance of such a schedule is not modelled). As a result it is not possible to evaluate the impact that the (likely) existence of different schedules has on the system performance.

The goal of our work to build an accurate (probabilistic) model of S-MAC which includes the schedule coordination/maintenance behaviour. The results illustrated in [25] show that with S-MAC performance deteriorates, when compared to 802.11, even when the most optimistic scenario (a single global schedule exists) is considered. We aim to extend such analysis by allowing different schedules to co-exist throughout the network. Intuitively this co-existence should result in improving the latency for packet delivery at the cost of higher energy consumption for boundary stations (i.e. nodes belonging to several neighborhoods).

Our work has strong links to both [25] and [17]. We compare 802.11 and S-MAC performance referring to a fixed multi-hop topology (as in [25]), but rather than using a simulative approach for protocol analysis, we apply (probabilistic) model checking verification, (as in [17]).
3 Medium Access Control for Wireless Networks

A computer network consists of a number of nodes that exchange information over a shared medium. A communication protocol regulates the behaviour of communicating nodes within a concurrent environment. The Open System Interconnection (OSI) model [10] 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 simultaneous transmissions (i.e., collisions) 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.

Criteria such as the type of medium, the communication range, the communication form (e.g., radio, infrared), the number of possible nodes in the network, the required performance/reliability/security are used to classify computer networks. Generally speaking, we refer to networks for which the communication medium is the ether as wireless networks. 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. Below we present the IEEE 802.11 MAC scheme for ad-hoc wireless networks.

3.1 MAC for Wireless LAN (802.11)

The IEEE 802.11 [11] 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 coordinate for the use of a (shared) medium in an attempt to avoid collision. The DCF is a variant of Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC [2] scheme developed for collision avoidance over a shared wired medium using a randomised backoff procedure. Two variants of the DCF have been defined in the standard: the Basic Access (BA) and the Request-To-Send/Clear-To-Send (RTS/CTS) and three time periods considered to characterise them: the DCF InterFrame Space (DIFS), the Short InterFrame Space (SIFS) and the Extended InterFrame Space (EIFS), where $SIFS < DIFS << EIFS$.

BA: A station can start a transmission of data packets only after sensing the medium free for either a DIFS, if the previous transmission was successful, or for an EIFS otherwise. On reception of a data packet, the destination station, after sensing the channel free for SIFS, sends an acknowledgment packet (ACK) back to the sender. A collision is recognised by the sending station if either: on termination of the transmission the channel is sensed occupied by another station, or if an ACK packet is not received within a given time. The main advantage of the BA scheme (also known as two-way handshake), is its simplicity.
However, in this case, collision involves (large) data packets and can result in significantly deteriorated performance.

**RTS/CTS:** The same sensing/randomised-backoff procedure of the BA scheme is used but an additional handshake is involved using RTS and CTS control packets (as a result this scheme is referred to as a four-way handshake). After sensing that the medium is free, a station wishing to send data packets over the medium sends an RTS packet, which includes information on the duration of subsequent transmissions. On reception of an RTS, the destination (after sensing the medium free), replies with a CTS. The sender will start the transmission of actual data packets on reception of the CTS confirmation. Every neighbouring node overhearing the RTS/CTS exchange is aware of the future communication duration hence refrains from attempting to access the medium for the whole duration of the communication. Collisions can only happen between control packets, hence the cost of collision is usually smaller than with BA. Furthermore the RTS/CTS handshake eliminates collisions that are due to the so-called hidden terminal problem [5], which BA does not. On the other hand the additional handshake introduces a further latency delay and so the RTS/CTS scheme should only be used be in application whose average communication duration is large enough to justify the additional overhead.

A transmitting station goes into backoff if the medium is not free for a sufficient length of time (either $DIFS$ or $EIFS$), no $ACK$ arrives in time, or the frame to be transmitted is consecutive to a previous transmission.

**Randomised Backoff procedure:** As soon as a backoff condition becomes true, the deferring station selects a $BackoffTime$ composed of a random number ($backoffvalue$) of slot times, where each slot has size $aSlotTime$.

$$BackoffTime = backoffvalue \cdot aSlotTime$$

The value of $Backoffvalue$ is a pseudo-random integer drawn from a uniform distribution over the interval $[0, CW]$, where $CW$ is the Contention Window which has initial value $aCWmin$ (provided by the PHY) and takes values of ascending powers of 2 minus 1. Thus

$$CW = (aCWmin + 1) \cdot 2^k - 1$$

where $ke$ (the $BackoffCounter$) increases with the number of unsuccessful transmissions. Note that the likelihood of a longer backoff delay for repeatedly detected collisions (where $ke$ is large) is increased.

The value of $CW$ has an upper bound of $aCWmax$. Once this value has been reached, $CW$ will remain at this value until it is reset (following a successful transmission attempt).

With no medium activity for a duration of $aSlotTime$ time units, $Backoffvalue$ is reduced by one. However, if the medium becomes busy the reduction of $Backoffvalue$ is frozen. The value of $Backoffvalue$ will only continue to be decremented when the channel has been sensed free for either a $DIFS$ or $EIFS$ period. The transmission restarts only when $Backoffvalue$ reaches zero.
3.2 MAC for Wireless Sensor Network (S-MAC)

Wireless Sensor Networks (WSNs) consist of battery powered, usually static, sensing devices. Applications include: weather conditions monitoring; moving objects detection/recognition; pollution measuring and military surveillance. Since replacing the battery of sensing devices is usually highly costly (often accessing the sensors after deployment is difficult), energy preservation is paramount for WSNs. Thus in considering aspects of a WSN, e.g. the communication protocol, routing algorithms or query processing methods, energy saving optimisation must be considered. In a typical WSN framework, data packets generated at the (power-constrained) sensing nodes flow towards the (less power-constrained) sink node whereas control packets, generated by the sink follow the opposite direction. WSNs belong to the wireless ad-hoc family and as such are characterised by peer-to-peer communication, self-discovery and de-centralised co-ordination. However a communication protocol, and more specifically a MAC scheme, for WSNs must be tailored for energy saving. In [25] the so-called S-MAC scheme, designed to reduce energy consumption of communicating nodes, is presented. It is based on the simple observation that for most WSN applications the sensed data streams are generated at low frequency (most of the time there is nothing to be sensed). Hence sleeping nodes’ radios switched on in an idle listening mode wastes energy. Four main sources of energy waste are identified.

1. Collision: re-transmission of corrupted transmitted packets.
2. Overhearing: listening to packets destined to other nodes.
3. Control packet overhead: sending/receiving control packets.
4. Idle listening: listening for possible traffic that is not sent.

S-MAC is designed to reduce energy consumption from each of these sources of energy waste. Below we give a brief description of the main functionality of S-MAC. For a more detailed treatment the reader is referred to [25].

S-MAC is based on two distinct operative states for the sensing nodes, an energy expensive LISTEN mode where the radio of a node is switched on, and an energy saving SLEEP mode in which the radio is turned off. Each node uses a periodic listen-sleep schedule to switch between LISTEN/SLEEP operative modes. A complete LISTEN/SLEEP cycle is referred to as a frame and the duty cycle is the ratio of the listen interval to frame length which can be adapted according to the application requirements. The S-MAC scheme is concerned with two different aspects: choosing and maintaining of sleeping schedules for each node (usually referred to as coordinated sleeping) and collision avoidance.

**Coordinated sleeping**: Each node maintains a schedule table where the LISTEN/SLEEP period of each of its neighbours is recorded. When a node wants to send some data to one of its neighbours it will start the RTS/CTS protocol during the LISTEN phase of the destination node, whose details are retrieved from the schedule table. The schedule table is built in a distributed fashion, through broadcasting of SYNCH packets between neighbouring nodes. A SYNCH packet
contains the sender's chosen schedule. Each node establishes its own schedule by either choosing its own schedule or by following a schedule received by one of its neighbours. In the former case a node is referred to as a synchroniser, and in the latter a follower. As soon as a node picks a schedule, it broadcasts it so that all neighbours can update their table.

Co-existence of different schedules: Although co-ordinated sleeping attempts to synchronise neighbouring nodes on a single, shared, schedule, it is possible for nodes in a neighbourhood to have different schedules. Such an eventuality happens whenever a node that has announced its own schedule receives a different schedule from one of its neighbours. This may result, for example, if a neighbouring node, possibly because of collision, does not receive a previously broadcasted SYNCH in time. A node receiving two different schedules (referred to as a boundary node), records it in its schedule table and will adopt both schedules (i.e. it will switch to LISTEN state according to both schedules).

Collision avoidance: This is achieved through the 802.11 MAC. The RTS/CTS protocol is used to avoid collision for unicast packets, whereas a randomised carrier sense is used to prevent simultaneous transmission of broadcast packets (i.e. SYNCH). A unicast data packet follows the sequence RTC/CTS/DATA/ACK. After a successful RTS/CTS exchange, the sender and destination nodes will stop following their sleeping schedule and use the sleeping time (and more if necessary) for the actual data transmission. The sleeping period will resume on completion of the transmission.

The synchronisation phases of the S-MAC protocol is illustrated in Fig 1, and the Listen period in Figs 2 and 3.

With the coordinated sleeping procedure of S-MAC, a recursive broadcast of SYNCH packets takes place. The first node choosing its own schedule will broadcast it to all of its neighbours (in form of a SYNCH packet) and each one of them will (potentially) re-broadcast it to its respective neighbours (assume that in the meantime a SYNCH packet has not been received from elsewhere). This approach may result into a flooding of a SYNCH packet throughout the network.2

Since all nodes shall be assumed to start the coordinated sleeping process simultaneously, no conclusion can be drawn as to which node will be first to chose its schedule and start broadcasting its SYNCH packet. Similarly it is not possible to know, a priori, how far from its originator the first SYNCH packet will reach, or equivalently, how many different schedules will co-exist in the network. The performance of S-MAC, both in terms of the induced latency and energy saving, is affected by the resulting topology of sleeping schedules. From the point of view of energy consumption, a greater number of sleep schedules should result in a reduction in the energy saved as boundary nodes have to wake up with respect to several schedules. From the point of view of latency, however, more

2 The flooding of SYNCH packets stops as soon as each node in the network has completed its own schedule table.
Fig. 1. Phase 1: Establishing a SYNCH table
Fig 2. Phase 2: Listen period, Upper section
schedules should improve performances, as the latency for a packet to traverse a boundary node is shorter if the node is awake more often. It seems likely that a correlation would exist between the network topology and the resulting set of sleeping schedules. Nodes that are many hops away from each other are likely to adopt different schedules.

For a given network topology, the following questions are relevant:

- what is the average/maximum length (in terms of number of hops) a SYNCH packet will traverse?
- what is the average/maximum number of different schedules a boundary node may have to follow?
- what is the average/maximum number of boundary nodes?

We believe that formal verification will allow us to address these questions and provide valuable information toward improved sensor network design.

4 Verification through Model-Checking

Model checking [9, 21, 22] is a technique whereby properties of a system can be checked by building a model of the system and checking whether the model satisfies the properties. The model is constructed using a specification language,
and checked using specific model checking algorithms. Examination of counter-examples provided by the model checker enable the user to either refine the model or, more importantly, to debug the original system.

Traditional (non-probabilistic) modelling languages like Promela [15] and SMV [20], allow one to express the behaviour of a system in terms of states and transitions between states. The associated model checkers, SPIN and SMV can then be used to check qualitative properties, described using a temporal logic.

Systems that exhibit probabilistic behaviour (i.e. unpredictable processes such as, for example, fault tolerant systems or computer networks) can be verified using probabilistic model checking. A probabilistic (Markovian) model is first built and then analysed using a probabilistic model checker like PRISM [24]. In a Discrete Time Markov Chain (DTMC) transitions are labelled with probabilistic values in the range \([0,1]\) and each transition is assumed to “consume” precisely one time unit. In a Continuous Time Markov Chain (CTMC) transitions are labelled with non-negative real values, time is continuous and the delay of a transition is governed by an exponentially distributed random variable with rate given by the associated real-valued label. Finally Markov Decision Processes (MDPs) are discrete-time probabilistic models that combine probabilistic behaviour with non-determinism.

Specific temporal logics have been introduced to verify probabilistic models. The logics PCTL [11] and CSL [3] are probabilistic extensions to CTL [8], and are dedicated to the verification of DTMC/MDP models and CTMC models respectively.

Both PCTL and CSL contain the probabilistic operator \(\mathcal{P}\) which allows one to express properties (probabilistic reachability properties) concerning the likelihood of an event occurring. For example, the property: \(SEND \land \mathcal{P}_{\geq 0.95}(true \cup \leq 100\text{ACK})\) states that there is at least a 95% chance that an acknowledgment (ACK) is received within 100 time units after the transmission of a packet (SEND).

Recently Markov Reward Models (MRMs) [23] have been introduced to PRISM to enhance system verification capability. Here reward/cost values are attached to states/transition of the models. This allows one to express more detailed quantitative properties referred as expected reachability properties (see Section 5.1 for an example).

5 Modelling 802.11 and S-MAC in PRISM

We have developed PRISM specifications based on a specific three-hop topology, for which we have used PRISM to generate associated DTMC models. Time is considered as a discrete quantity representing actual clock ticks. In the considered topology (Fig 4) a single source of data (node A) generates data packets according to a geometric distribution with rate \(p\). These packets reach the destination sink node (node D), through two relaying nodes (node B and node C respectively).

Note that, in order to assess the effect of coordinated sleeping on performance, a multi-hop topology must be considered. With a single-hop there are
no boundary nodes and hence no nodes with multiple schedules. With three-hop there is the possibility for (at most) two different schedules to co-exist, while keeping the model size within limits. It should be noted that with respect to the model size a two-hop topology would be preferred to a three-hop one (two-hop is the minimal length with which at most two different schedules may be observed). However, the inclusion of the additional hop allows one to observe the effect of different schedules on the propagation of packets over a multi-hop route.

With this topology we have modelled both the 802.11 and the S-MAC scheme. For the sake of simplicity, in the current version of each model the BA coordination function (only) has been considered for both schemes (the four-way RTS/CTS handshake significantly increases the size of the model). In the 802.11 model all nodes (apart from the source) perform idle-listening until they detect a package on the medium, at which point they start to process communication. Observe that, since a single source node is considered, collisions between data packets can only take place at relaying nodes, whereas collisions at the source/sink node occur between data and control packet only. In order to compare the performance of 802.11 and S-MAC we have developed three different versions of the model referring to the S-MAC scheme, each one corresponding to a different level of complexity.

- **Single (global) sleeping schedule.** All four nodes are assumed to be synchronised on a single sleeping schedule. As a result they all go to sleep and wake up at the same time. This model allows us to measure the overhead of the coordinated sleeping of S-MAC under the best (in terms of energy saving) possible conditions (i.e. with a unique global schedule). Since we use the same configuration as that of [25], we will be able to compare results.

- **Fixed multiple sleeping schedule.** In this version, two different schedules are assumed to co-exist for the nodes in the topology, namely $schAB$ and $schCD$ where $schAB \neq schCD$. Nodes A and B follow schedule $schAB$, whereas nodes C and D go to sleep according to schedule $schCD$. For simplicity, schedules $schAB$ and $schCD$ are statically chosen. This model allows us to verify the effect of co-existence of different schedules on both the latency for a data packet to reach the sink node and the energy consumption at boundary nodes.

- **Random generation of sleeping schedule.** This version of the model is designed to reflect the more realistic behaviour of a network on which the S-MAC is implemented. No assumption is made on the sleeping schedules, so each node in the network can potentially become either a synchroniser.
or a follower. As such the complete coordinated sleeping function (which includes the initial listening-for-SYNCH packets phase) must be modelled for each node. Nodes start the coordinated sleeping phase simultaneously and each one of them will try to choose its own schedule if it does not receive a SYNCH packet from elsewhere. Verification, in this case, is aimed to assess the overhead of acquiring a coordinated sleeping schedule over system performance (which is not possible to achieve with either of the other two versions of the model).

5.1 Model Verification

The models are to be verified against PCTL properties. Following the approach used in [17] the relevant properties can be grouped as follows: probabilistic reachability properties; time-bounded probabilistic reachability properties; and expected reachability properties (see Section 4).

An example probabilistic reachability property in this context is “eventually (with probability at least 1) a packet generated at the source node will reach the sink node”. Time-bounded probabilistic reachability properties allow for the computation of the probability of some event happening within a (soft) deadline (they are also referred to as soft-deadline properties). For example: “the minimum probability for a data packet to reach the sink node within a time deadline $T$”. Finally expected reachability properties are verified with respect to Reward Markov Models, an extension of classical Markovian models in which reward/cost values are associated with transitions and/or states of a DTMC/MDP/CTMC model. This allows us to verify useful properties of our models such as “is it true that the average energy consumption for a given node on the long run is less than a given value $X$?” or “is it true that the energy needed to deliver a message does not exceed a given threshold $X$?”

As stated in Section 1, verification of the different versions of the model is still under process. Full verification results will be reported at the ISOLA symposium.

6 Conclusion and Future Directions

We have described models for probabilistic model checking of a communication protocol designed to reduce energy consumption in Wireless Sensor Networks. We have developed several Discrete Time Markov Chain models, representing both the most popular MAC scheme for generic ad-hoc networks (IEEE 802.11) and the (recently introduced) S-MAC protocol for WSNs. To keep our models tractable, we have used a simple three-hop topology where a single source node periodically generates data destined for the sink three hops away. Our models will be used to verify different aspects of the S-MAC behaviour and to compare the more energy-expensive 802.11 with the less performant S-MAC protocol. An initial, simplified model of S-MAC, where each node is assumed to be synchronised on a single global schedule, will be used to assess the “pure” overhead
introduced by periodic sleeping. The more complex model, where two different sleeping schedules co-exist, will be used to evaluate the effect of multiple schedules on both energy saving and latency. The final model reflects more realistically the complex behaviour of coordinated sleeping procedure.

Model verification will be achieved through a combination of specific probabilistic reachability PCTL properties and rewards properties. The introduction of rewards in the defined models, allows for evaluating the average energy cost of communication in each protocol.

Future development of this work includes achieving full verification results for the presented models and the our modelling approach to other formalisms/tools including ProBMLA/LiQuor[4,7], UPPAAL [18], and the GREATSPN tool [6] for stochastic petri nets modelling.

References