Analysis of IEEE 802.15.4 Sensor Networks for Event Detection

Marco Gribaudo, Daniele Manini
Università degli Studi di Torino, Italy
Email: {marcog,manini}@di.unito.it

Alessandro Nordio, Carla-Fabiana Chiasserini
Politecnico di Torino, Italy
Email: {nordio,chiasserini}@polito.it

Abstract—We study the delay performance of a sensor network, whose nodes access the medium by using the unslotted MAC protocol specified by the IEEE 802.15.4 standard. Unlike previous works which focus on the average throughput and delay analysis, we develop a detailed model that allows us to obtain the delivery delay distribution of messages sent by concurrently contending sensors toward a central controller. We carry out a transient analysis, which is of particular interest when sensor networks are deployed to provide k-coverage for real-time applications, and validate our analytical results against simulation results obtained through ns2.

I. INTRODUCTION

Wireless sensor networks (WSN) are often used for real-time applications, such as environment surveillance, medical care, and vehicle traffic control [1]. In these contexts, in spite of the severe resource limitations that characterize the sensor nodes, WSNs have to provide a reliable coverage of the area of interest as well as to meet severe timing constraints.

In this work, we focus on WSNs for event detection, where each spatial point of the controlled area is covered by k sensors. Upon an event occurrence, each of the k nodes generate a detection report and try to transmit it to a central controller. At the MAC layer, we assume that the sensor nodes implement the IEEE 802.15.4 contention-based scheme, i.e., the emerging standard for low-rate, low-power wireless personal area networks. Furthermore, we consider that the central controller detects an event if it receives m positive detection reports from different sensors (with m ≤ k), within a time interval of t_d seconds from the arrival of the first positive report. Note that, when m > 1, this corresponds to considering group-based detection – a widely used technique in real WSN applications to avoid false positive alarms [2], [4].

Under this scenario, we pose the following questions:

i) What is the delay distribution of the detection reports transmitted by the sensors, conditioned on their order of arrival at the central controller?

ii) Given that the central controller needs to receive m ≤ k reports to perform reliable detection, what is a good value for k, considering that the larger the value of k, the higher the contention on the radio channel?

iii) Given k and the value of t_d imposed by the application, how many reports one can hope to receive at the central controller by the given time constraint?

While the problem of providing k-coverage in a connected communication topology has been widely addressed, fewer works have dealt with the delay performance of IEEE 802.15.4-based WSNs. In particular, the existing studies carry out a stationary analysis of IEEE 802.15.4-based WSNs [5]–[10], which is not suitable for the study of event detection.

To cope with the aforementioned questions, we proceed as follows. Motivated by the typical network structure used in group-based detection systems [2], we consider a star network topology such that all sensors can communicate with the central controller, which may be either the detection group leader or the final processing unit. Also, all nodes operate according to the unslotted mode. Note, however, that our model can be extended to the case of slotted channel access as well as multihop topologies.

We then carry out a transient analysis [3] where, upon an event occurrence, k sensors start contending for the channel to transmit their report. Note that our system significantly differs from the ones analyzed in [11] and in other works inspired by Bianchi’s pioneer study. Indeed, to derive the delay distribution of the m-th report, we cannot model just the behavior of a tagged sensor but we have to take into account the inter-dependencies existing between different sensors that simultaneously contend for the channel. In particular, the fact that collision events are strongly correlated prevents us from assuming independent sensors and makes the use of a fixed-point technique unsuitable for this study. We therefore resort to a different approach and model each sensor and the WSN as a Hybrid Automata [12]. This technique allows us to consider every possible evolution of the system, taking into account also rare events which are difficult to be captured by simulation. Then, by exploiting the properties of WSNs, we derive an efficient solution algorithm that significantly reduces the complexity of the model and its solution time. The model is validated, and shown to be extremely accurate, through simulations carried out with ns2.

To the best of our knowledge, this is the first work that presents a transient analytical model of the IEEE 802.15.4 contention-based scheme and derives the message delay distribution, thus providing important insights for the design of real event detection systems.

II. BACKGROUND ON IEEE 802.15.4

We focus on the IEEE 802.15.4 contention-based access and, in particular, on the unslotted mode.
According to this scheme, represented in Fig. 1, a sensor wishing to transmit a message first extracts a random backoff time. This value is computed by multiplying the duration of the so-called BackoffPeriod by an integer number uniformly distributed between 0 and $2^{BE} - 1$. The backoff exponent, $BE$, is initially set to $MinBE$ and can achieve the maximum value of $MaxBE$. As the backoff time expires, the node senses the channel for a Clear Channel Assessment (CCA) time: if the channel is idle, the sensor switches from listening to transmitting mode (transceiver turnaround) and accesses the channel, otherwise it increases $BE$ and repeats the procedure. The variable recording the number of backoff extractions performed by the sensor is denoted with $NB$.

A successful transmission is acknowledged by the receiver after a time $T_{a}$; if unsuccessful, the sensor waits for a timeout, $T_{w}$, before detecting the failure. A message is discarded after a $MaxCSMABackoffs$ number of backoff extractions have been performed by the sensor while attempting to transmit it. Finally, two successive transmissions from a sensor are separated by an Inter-Frame Spacing (IFS) time period.

III. MODEL DESCRIPTION

We develop our model based on the following assumptions:

- the WSN has a star topology and all nodes are in radio proximity of each other;
- upon an event occurrence, $k$ sensors generate their detection report (of fixed length) and start contending for the channel at the same time;
- the transmission of report messages and ACKs over the wireless channel may fail due to bad propagation conditions, with probability $p_{m}$ and $p_{a}$, respectively.

Next, we detail the model for a single sensor node and describe how the model of the whole network is obtained. We then outline the solution method and the techniques that we apply to reduce the model complexity.

A. Single sensor model

We describe the behavior of a generic sensor by using a Hybrid Automata, i.e., a formal model to handle systems with both discrete and continuous components. The model of a single sensor is shown in Fig. 2, where each box represents a state of a Finite State Machine (FSM).

![Fig. 2. Finite State Machine representing the single sensor behavior](image-url)
is represented by letting the sensor wait for a time corresponding to the ACK reception, plus an IFS period, after which the sensor can actually consider its transmission as successful.

As shown in Fig. 2, a sensor wishing to transmit its detection report enters the state START, where the variable NB is set to 0 and BE is set to MinBE. The sojourn in this state does not take time since $T$ is set to 0 and, thus, the condition $T = 0$? on the output arc is immediately fulfilled.

Before entering the state BOT (BackOffTime), the sensor extracts a backoff value and sets the variable $T$ so as it has to wait for $T$ seconds before moving into the next state.

When the condition $T = 0$? holds true, it means that the backoff time is expired. Then, the FSM sets $T = T_s$, i.e., the time interval during which channel sensing has to be performed, and enters the CCA state.

If, while sensing the channel, the node detects a busy medium (is_busy? condition is true), its attempt at accessing the channel fails. The FSM first enters the state CCA_busy, where the time left is consumed, and then it moves into state MAX_BO (MAX BackOff). Here, it is checked whether the variable NB has reached the MaxCSMABackoffs limit. If so, the transmission fails and the sensor enters the absorbing state FAIL. Otherwise, NB and BE are updated, a new backoff value is extracted, and the FSM returns into the BOT state.

On the contrary, if while being in state CCA the node senses the channel as idle for $T_s$ seconds, it switches from listening to transmitting mode and sends its report message. The transceiver turnaround operation is identified by the state TA and requires $T_s$ seconds. The message transmission takes $D_{msg}$ seconds and is performed in state MT.

If during the message transmission a collision or a channel error occurs, an ACK will not be sent back by the central controller. Clearly, the unsuccessful outcome of the transmission cannot be detected by the sensor while being in transmitting mode. However, the model accounts for these events through the is_busy? condition in case of collision, and through the msg_err? condition in case of channel error. Both these conditions trigger the FSM into the state MT_fail. Here, the sensor waits for the time variable $T$ to decrease to 0 and then enters state $\Delta$, where it waits for additional $T_w$ seconds before detecting the transmission failure. Instead, if the transmission of the report message is successful, $T$ is set to $T_s$ and the sensor enters state TA_ACK. Here, the sensor switches to receiving mode and waits for the ACK to be sent by the central controller.

The state ACK models the reception of the acknowledgment, which takes $D_{ack}$ seconds. Similarly to the message transmission dynamics, collisions or channel errors during the ACK transmission are taken into account through the is_busy? and ack_err? conditions, respectively. When either of these conditions holds true, the FSM moves to the state ACK_fail. In this case the ACK is not correctly received, and the sensor detects a message transmission failure after $T_w - T_s - D_{ack}$ seconds. If, instead, the ACK transmission is successful, the state IFS accounts for the time $T_{ifs}$, required by the MAC layer of the central controller to process the data received from the physical layer. Eventually, the sensor reaches the absorbing state SUCCESS.

B. Network model

We consider that all sensors are identical and we model the $k$ nodes that simultaneously access the medium by replicating $k$ times the FSM representing the single sensor.

Interactions among sensors, which have to be taken into account in case of collision or during the CCA operation, are handled by using the is_busy? event (see Fig. 2). When a sensor is in the states MT, MT_fail, ACK or ACK_fail (represented by a box with a round border), the event is_busy? is generated continuously in time. The generation of this event will cause other sensors that are in the states characterized by an output arc labeled by is_busy? (i.e., CCA, MT, and ACK) to perform that transition and change their state accordingly.

C. Solution technique

The solution technique we adopted consists of three phases: 1) time discretization, 2) transient solution and 3) performance metrics computation.

Time discretization. Recall that the Hybrid Automata, describing a single sensor behavior, is characterized by a continuous variable $T$, which tracks the time spent by the sensor in a state, and by the discrete variable NB, which records the current number of backoff values extracted by the sensor. Moreover, the is_busy? condition models the dependencies among the automata that represent different sensors.

As a first step, we discretize the time evolution of the system, by taking as discrete time unit the greatest common divisor of all time durations $\Delta T$ (namely, 32 $\mu$s). By doing so, we convert the Hybrid Automata of a single sensor into a discrete time Markov chain (DTMC): given $NB = 0$ (i.e., the first backoff stage), we transform every state of the original automata into a sequence of DTMC states, each of them representing the elapsing of $\Delta T$ seconds (see Fig. 3). We then replicate the time discretization of the states for all possible values of NB, thus accounting for both variables $T$ and $NB$.

It follows that the state of the DTMC representing the single sensor behavior is defined by: $NB$, the identifier of the state in which the automata is, and the residual sojourn time in the automata state (expressed as number of time units $\Delta T$).

Then, to account for the condition is_busy?, for each sensor we consider two different DTMCs that are characterized by the same state space but different state transition probabilities matrices. We denote the transition probabilities matrices of these DTMCs by $F$ and $B$, respectively. The DTMC generated by $F$ models the sensor behavior when the channel is idle, while the other when the channel is busy. This means that matrix $F$ does not include the transitions controlled by the condition is_busy?, while in matrix $B$ these transitions are considered and have priority over the others. Portions of the two DTMCs are depicted in Figs. 3 and 4, respectively.

Next, we write the DTMC generator $C$ of the model of the $k$-sensor network, as: $C = \otimes_{i=1}^{k} F + \chi(F, B)$, where $\otimes_{i=1}^{k} F$ is the Kronecker product of $k$ copies of the DTMC.
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.
that, since the probability values derive from the extraction of different backoff times, in virtue of the property at point ii) many states are characterized by the same probability value.

Finally, note that, at the beginning of the solution algorithm, a very small number of states have to be considered. As time evolves, the states number gradually increases due to the different combinations of backoff values that can be extracted. However, as time further increases, the number of states tends to reduce again, since some of the sensors reach either the SUCCESS or FAIL state. In the end, we obtain the vector $\pi_N$ that includes exactly $k + 1$ components, with $\pi_N(j)$ being the probability that $j$ ($j = 0, \ldots, k$) transmissions have failed.

**Performance metrics computation.** Once we obtain the probability vectors $\pi_n$ ($n = 1, \ldots, N$), we can compute the delay probability distribution of the $m$-th arrival, $f_m(t)$. Let us define the probability $P(m, n)$ that exactly $m$ sensors have reached the SUCCESS state at time step $n$, as: $P(m, n) = \pi_n \cdot v^T_m$, where $v^T_m$ is a column vector whose components are all equal to zero, except the one corresponding to the network states where there are exactly $m$ sensors in the SUCCESS state. Note that $\sum_{n=0}^{N} P(k, n) \leq 1$, indeed some of the reports delivery may fail. From the expression of $P(m, n)$ and by denoting with $\delta(\cdot)$ the Dirac’s delta, we compute $f_m(t)$ as:

$$f_m(t) = \sum_{n=1}^{N} \sum_{j=m}^{k} \left[ P(j, n) - P(j, n - 1) \right] \delta(t - n\Delta T)$$

**IV. MODEL VALIDATION AND EXPLOITATION RESULTS**

We first validate our model against simulation results obtained through ns2. Then, we show how our model can be used to derive performance metrics of interest for the design of event detection systems.

While deriving the results, we use the parameter setting reported in Table I; in addition, we set $\text{MaxCSMABackoffs}$ to 4 and $\sigma_m = \sigma_a = 0$, unless otherwise specified.

The tight matching between simulation results (labeled by “Simulation”) and analytical results (labeled by “Analysis”) is shown in Figs. 6 and 7.

The plot in Fig. 6 presents the delivery delay distribution $f_m(t)$, with $m = 1, 2, 3, 4$, when $k = 4$ sensors detect the event and generate their report. Note that, as $m$ increases, the delay distribution spreads out, thus increasing its variance.

Another interesting effect can be observed by looking at $f_1(t)$ in Fig. 7, where the cases $k = 1, 3, 6$ are compared: as $k$ increases from 1 to 3, the probability that the first message arrives at the central controller with a delay comprised between 2 ms and 3 ms (i.e., the probability of an early arrival) increases. The reason for this behavior is that, as the number of sensors extracting the first backoff grows, the probability that smaller values are selected increases, thus making an early transmission of the first report more likely. However, a further look at Fig. 7 reveals that when the number of sensors grows from 3 to 6, early arrival times for the first message ($m = 1$) become less likely due to the increased collision probability. This can be noted by computing the probability that the first message is delivered between 2 ms and 3 ms, for $k = 3$ and $k = 6$, which are equal to 0.83 and 0.79, respectively. We therefore conclude that the probability to have small delivery delays increases as the number of sensors grows up to a certain value of $k$, after which an opposite behavior emerges.

Next, we look at the probability to receive $m$ reports at the central controller, given the time deadline on the detection decision specified by the application. Fig. 8 presents the cumulative distribution function $F_m(t)$ of the $m$-th report delivery delay, obtained by using our model for $k = 3$ and $m = 1, 2, 3$. Given the time constraint $t_d$, from the plot we can derive the probability that $m$ detection reports are delivered within the deadline. Interestingly, for $k = 3$ it is evident that the cumulative distribution function $F_3(t)$ does not saturate to 1. This means that the probability that one or more sensors drop their detection report because they have reached the maximum number of possible backoff extractions ($\text{MaxCSMABackoffs} = 4$) is not negligible.

In Fig. 9, we fix the time constraint at $t_d = 20$ ms and present the probability that only $m$ out of $k$ detection reports are successfully delivered within $t_d$. The results are shown for $m = 1, 2$ and different values of $k$. Here, we also consider
different values of $p_m$ and $p_a$, i.e., the probability that the transmissions of, respectively, a report message and an ACK fail due to channel errors.

As expected, the larger $p_m$ and $p_a$, the higher the probability that a message needs to be retransmitted and, thus, that only a fraction of the $k$ reports are delivered within $t_d$. In particular, while the probability that only one out of $k$ reports reaches the central controller within 20 ms is quite small, the probability that only 2 reports are delivered by the deadline is about 0.04 for $p_m = 0$ and 0.1 for $p_m = 0.1$. Furthermore, it is important to note that, as the number of sensors detecting the event grows, the probability that only $m$ out of $k$ detection reports arrive within the time constraint has a minimum. Indeed, initially the larger the number of generated reports, the less likely the event that only 1 or 2 of them are delivered. However, as the number of sensors further increases, the collision probability grows significantly. As a consequence, the message drop probability due to the reached $MaxCSMABackoffs$ limit increases, and so does the probability that only $m$ reports arrive at the central controller.

We conclude that, by using our model, the number of reporting sensors can be determined so that, depending on the number of detection reports required at the central controller and on the time deadline imposed by the application, both the arrival time of the delivered reports and the probability that $m$ reports arrive within the deadline can be optimized.

V. CONCLUSION

We studied the performance of a IEEE 802.15.4-based WSN for event detection. We carried out a transient analysis of the system when $k$ sensors detect the event and attempt to send their report to a central controller. We validated our model using $ns2$ simulations. Through the proposed model, we derived the delay distribution of each detection report delivery, and the probability that only $m$ out of $k$ reports reach the central controller within a given time constraint. Our work current focuses on extending the proposed model to include the slotted access mode and the use of multihop topologies.

**ACKNOWLEDGMENT**

This work was supported by the Italian project PRIN No. 2007J4SKYP 003.

**REFERENCES**


**TABLE I**

<table>
<thead>
<tr>
<th>Data rate</th>
<th>Message payload</th>
<th>MAC header</th>
<th>PHY header</th>
<th>$T_a$</th>
<th>$T_w$</th>
<th>$D_{ack}$</th>
<th>BackoffPeriod</th>
<th>MinBE</th>
<th>MaxBE</th>
<th>Turnaround</th>
<th>$T_a$</th>
<th>$T_{ifs}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>250 kbit/s</td>
<td>22 byte</td>
<td>7 byte</td>
<td>6 byte</td>
<td>192 µs</td>
<td>864 µs</td>
<td>352 µs</td>
<td>320 µs</td>
<td>3</td>
<td>5</td>
<td>192µs</td>
<td>128µs</td>
<td>192µs</td>
</tr>
</tbody>
</table>

Fig. 8. Cumulative distribution function of the reports delivery delay, for $k = 3$ and $m = 1, 2, 3$. Fig. 9. Probability that only $m$ reports out of $k$ are delivered within $t_d = 20$ ms, for $k = 3, 4, 5, 6$, $m = 1, 2$ and different values of $p_m$ and $p_a$.