

Hindawi Publishing Corporation  
Journal of Sensors  
Volume 2015, Article ID 685164, 12 pages  
<http://dx.doi.org/10.1155/2015/685164>



## Research Article

# Energy Analysis of Contention Tree-Based Access Protocols in Dense Machine-to-Machine Area Networks

Francisco Vázquez-Gallego,<sup>1</sup> Luis Alonso,<sup>2</sup> and Jesus Alonso-Zarate<sup>1</sup>

<sup>1</sup>Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Parc Mediterrani de la Tecnologia (PMT),  
Avenida Carl Friedrich Gauss 7, Castelldefels, 08860 Barcelona, Spain

<sup>2</sup>Department of Signal Theory and Communications, EETAC, Universitat Politècnica de Catalunya (UPC) (BarcelonaTECH),  
Castelldefels, 08860 Barcelona, Spain

Correspondence should be addressed to Francisco Vázquez-Gallego; [francisco.vazquez@cttc.es](mailto:francisco.vazquez@cttc.es)

Received 6 November 2014; Accepted 20 March 2015

Academic Editor: Banshi D. Gupta

Copyright © 2015 Francisco Vázquez-Gallego et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine-to-Machine (M2M) area networks aim at connecting an M2M gateway with a large number of energy-constrained devices that must operate autonomously for years. Therefore, attaining high energy efficiency is essential in the deployment of M2M networks. In this paper, we consider a dense M2M area network composed of hundreds or thousands of devices that periodically transmit data upon request from a gateway or coordinator. We theoretically analyse the devices' energy consumption using two Medium Access Control (MAC) protocols which are based on a tree-splitting algorithm to resolve collisions among devices: the Contention Tree Algorithm (CTA) and the Distributed Queuing (DQ) access. We have carried out computer-based simulations to validate the accuracy of the theoretical models and to compare the energy performance using DQ, CTA, and Frame Slotted-ALOHA (FSA) in M2M area networks with devices in compliance with the IEEE 802.15.4 physical layer. Results show that the performance of DQ is totally independent of the number of contending devices, and it can reduce the energy consumed per device in more than 35% with respect to CTA and in more than 80% with respect to FSA.

## 1. Introduction

Machine-to-Machine (M2M) area networks provide connectivity between M2M gateways and a large number of physical devices which interact with the environment (e.g., sensors, actuators). M2M gateways provide physical devices with connection to the Internet and other networks, so that monitoring and control applications can be remotely run [1]. The M2M gateways perform protocol translation, traffic aggregation, data processing, and authentication functionalities. The devices can be connected to the M2M gateways through a wide range of wireless technologies such as Wireless Personal Area Networks (WPAN), Body Area Networks (BAN), Radio Frequency Identification (RFID), Wireless Local Area Networks (WLAN), and Wireless Sensor Networks (WSN). Once deployed, the devices must operate autonomously for years with none or very limited access to energy sources. Therefore, energy efficiency is crucial in the design of M2M area

networks that must operate autonomously for years without human intervention.

In this paper, we focus on applications where a group of hundreds or thousands of devices periodically transmit data to a gateway upon request. Some examples of such applications are, among others, asset tracking using RFID, where a reader sends requests to the tags and they respond with their ID, and Automatic Meter Reading (AMR), where a gateway collects electricity, gas, or water consumption readings from a network of sensing devices in smart grids. The traffic load generated by every device can be very low; however, the number of devices that may respond simultaneously to the gateway can be larger than the one manageable by traditional Medium Access Control (MAC) protocols. Therefore, an efficient MAC protocol is required to manage the access to the wireless channel and avoid collisions in dense M2M area networks.

The mobility and huge number of devices in dense M2M area networks make it difficult to know the network topology and maintain a centralized schedule [2] that allows every device to transmit data without collisions. Contrarily, random access protocols (e.g., ALOHA [3] or Carrier Sense Multiple Access) do not require knowledge of the network topology, and their simplicity of operation makes them ideal for simple energy-constrained devices. For this reason, many standards for short-range communications rely on random access protocols such as the IEEE 802.15.4 [4] for WPAN and WSN; the IEEE 802.11 [5] for WLAN; and the ISO/IEC 18000-6 [6] for RFID systems based on frame slotted-ALOHA (FSA). Unfortunately, random access protocols suffer from degraded performance in terms of throughput, delay, and devices' energy consumption, when the traffic load increases or when the number of contending devices is large and thus the probability of collisions is high [7].

A strategy to improve the maximum stable throughput of random access protocols is to use a Collision Resolution Algorithm (CRA) as it was demonstrated in [8, 9]. CRAs resolve collisions by organizing the retransmission of colliding packets in such a way that all packets are always successfully transmitted with finite delay. The basic CRA is the tree-splitting algorithm that was introduced in [8], which iteratively splits a group of contending devices into smaller subgroups to reduce collisions in an efficient manner. The delay and energy performance of the tree-splitting algorithm have been evaluated in different applications such as RFID [10–13] or sensor networks [14], among others. All these works use the tree-splitting algorithm to resolve collisions among devices that contend in a sequence of time frames divided into a number of time slots where they transmit data. Results show that there is an optimal frame length that minimizes delay and energy consumption regardless of the number of contending devices. This makes the tree-splitting algorithm very appealing in dense M2M networks where the number of devices is large and is not known a priori.

While the tree-splitting algorithm implemented in [10–12, 14] uses data slots to resolve contention, it can be also used to handle channel access requests using minislots, which are actually shorter than data packets. This approach is called Distributed Queuing (DQ) access and was introduced in the DQ Random Access Protocol (DQRAP) [15] and DQ Collision Avoidance (DQCA) [16] protocols. In DQ access, the collision of data packets is avoided by separating access requests from the transmission of data. Therefore, since the duration of data packets is longer than an access request, DQ achieves greater energy efficiency.

As discussed later in Section 2, previous works related to DQ have analysed the network performance in steady-state conditions, assuming that all devices generate data traffic according to a random Poisson distribution. Under these conditions, DQ achieves maximum throughput when just 3 slots are used for access requests regardless of the number of devices. In our recent work [17], we experimentally demonstrate the operation of DQ and compare the performance of DQ and FSA in terms of packet delivery and collision rates on RFID systems at 433 MHz. However, to the best of our knowledge, the energy performance of DQ has never been

analytically evaluated in M2M area networks with abrupt idle-to-saturation traffic transitions (delta traffic), that is, when a huge amount of devices have data ready to transmit in a given time and attempt to get access simultaneously. This is the main motivation for the work presented in this paper, where we focus on analysing the conditions when DQ can reduce the devices' energy consumption with regard to other approaches based on the conventional tree-splitting algorithm and FSA.

Therefore, the main contributions of this paper are as follows.

(1) We formulate accurate analytical energy models of two MAC protocols based on a tree-splitting algorithm for networks with abrupt idle-to-saturation traffic transitions: (i) the Contention Tree Algorithm (CTA) and (ii) DQ. The proposed energy models are based on a generic mathematical analysis of tree-splitting algorithms presented in [18], which derives the number of frames required to resolve the contention, and the number of levels of the contention tree required for a contender to have successful access.

(2) We evaluate and compare the performance of DQ, CTA, and FSA, in terms of average energy consumed per device, in dense M2M area networks. The energy model of FSA can be found in [14].

(3) We determine the configuration parameters of the MAC layer to maximize the energy efficiency when using either DQ or CTA. For this last purpose, we consider devices equipped with radio transceivers in compliance with the IEEE 802.15.4 standard.

The remainder of this paper is organized as follows. In Section 2, we describe the related work to motivate the contribution presented in this paper. In Sections 3 and 4, we describe the system model and the MAC protocols that we have considered, respectively. In Section 5, we present the theoretical analysis of CTA and DQ and formulate the energy models of the two protocols. Section 6 is devoted to validating the models and to evaluating the performance of both mechanisms through comprehensive computer-based simulations. The comparison with the performance of FSA is also presented in this section. Finally, Section 7 concludes the paper.

## 2. Related Work

*2.1. Contention Tree Algorithm.* CTA is based on the tree-splitting algorithm that was originally proposed in [8] and later analysed in [9, 18, 19]. The contention process is composed of a sequence of time frames further divided into slots. In the first frame, every device randomly selects one of the slots to transmit. If two or more devices collide, that is, they transmit in the same slot, a new frame is assigned only to the subgroup of devices that caused the collision in that slot. Therefore, new frames become available after each frame with one or more slots with collision. The process terminates when all devices have been able to successfully transmit their data packet.

Several works in the field of RFID have analysed the energy performance of CTA [10] and specific implementations of CTA based on binary tree-splitting: the Binary

Tree (BT) [13] and the Query Tree (QT) protocols [11, 12]. Both protocols work by splitting the RFID tag identifications bitwise using queries from the reader until all the tags are identified. For example, in each request of QT, the reader or coordinator sends a query including a binary prefix. The tags with this binary prefix in their identification (ID) respond back in a single slot. In the case of a collision, the reader adds 1 bit (0 or 1) to the prefix and sends a new query with the longer prefix. The process is repeated until all tags successfully transmit their ID.

The work in [11] compares the energy consumed by the reader and the tags using QT and FSA. Similarly, the work in [12] analyses the energy consumption of the reader and tags using QT. For the purpose of the analysis, authors derive the average number of frames required to identify all tags and the average number of tag responses, and they consider only the energy consumed in transmission and reception modes, but not in sleep. Results show that the energy consumed by the reader using QT is lower than using FSA. For the energy consumed by the tags, QT achieves better performance than FSA when the tags contend in all the frames. However, if the identified tags withdraw from contention in subsequent frames, FSA outperforms QT in terms of tags energy consumption.

The work in [10] presents an analysis of the energy consumed by the tags using CTA. The analysis neglects the energy consumed by the tags in reception and sleep mode. Results show that the energy consumed by one tag decreases when the frame length increases, and it increases logarithmically when the number of tags increases.

In our recent work [14], we consider an M2M network formed by a group of devices that periodically transmit data upon request to a coordinator. We evaluate the energy consumption of devices using CTA and FSA. Results show that there exists an optimum frame length which maximizes energy efficiency. This optimum frame length is constant in CTA regardless of the number of devices, but it needs to be optimized in FSA according to the number of devices [14, 20]. Using the optimum frame lengths, FSA and CTA perform almost identical energy consumption when the number of devices is lower than 50. However, this work does not compare the energy performance of CTA and FSA in dense networks with hundreds of devices.

**2.2. Distributed Queuing (DQ).** DQ can be seen as an evolution of CTA where contention for the channel is separated from data transmissions. The basic idea of DQ is that devices request for channel access in a reserved time interval, thus confining collisions to a part of the frame. Then, collisions are resolved by using the tree-splitting algorithm proposed in [8]. When a device succeeds in transmitting its access request, it enters into a queue and waits for its turn to transmit data in a collision-free data slot.

Even though DQ was originally designed for cable TV distribution (DQRAP [15]), it has been adapted to different communication networks over the last years; to wired centralized networks (Extended DQRAP [21], Prioritized DQRAP [22]), satellite communications with long propagation delays

(Interleaved DQRAP [23]), and Wireless Local Area Networks (DQCA [16]).

Existing analyses of DQ [15, 16, 21–23] consider that the devices generate packets following a random Poisson distribution and study the steady-state performance of the protocol. In these conditions, DQ shows near-optimum performance in terms of throughput and delay, and its performance is independent of the number of contending devices and the total offered data traffic. DQ behaves as a random access scheme under low traffic and switches to a reservation-based access scheme when the traffic load increases. It has been demonstrated that DQ achieves maximum performance using 3 request slots regardless of the number of devices.

Up to our knowledge, none of the previous works in DQ have studied the energy performance of DQ in the case of delta traffic conditions in dense M2M networks, that is, when a huge number of devices have data ready to transmit simultaneously to a coordinator in a given time. This is the main motivation for the contribution presented in this paper.

### 3. System Model

We consider a single-hop wireless network with one coordinator surrounded by a number  $n$  of devices. They can be in five modes of operation: (i) transmitting a packet, (ii) receiving, (iii) idle listening, (iv) standby, or (v) sleeping. The power consumptions associated with each mode are  $\rho_{tx}$ ,  $\rho_{rx}$ ,  $\rho_{\sigma}$ ,  $\rho_{stby}$ , and  $\rho_{sleep}$ , respectively. The model is general for any power saving mechanism and any value of the power consumptions in different modes of operation. We assume that the energy consumed when switching between inactive (i.e., standby, sleep) and active modes (i.e., transmitting, receiving, and idle listening) is negligible. In sleep mode, the radio interface is fully disabled, thus consuming the lowest power consumption. In contrast,  $\rho_{stby} > \rho_{sleep}$  and the time of transition from standby mode to an active mode is much shorter than from sleep mode. Therefore, we consider that devices only switch to sleep mode when they can remain more time inactive than switching between inactive and active modes.

The coordinator broadcasts a Request for Data (RFD) packet periodically, every  $T_R$  seconds, in order to initiate a data collection round, as depicted in Figure 1. The  $k$ th data collection round is composed of a sequence of  $F_k$  time frames. We consider that at the beginning of each data collection round every device has exactly one data packet ready to transmit to the coordinator. We assume that all devices are listening to the channel when the coordinator transmits an RFD. After decoding an RFD, all devices get synchronized and transmit their data packet according to the rules of the adopted MAC protocol. The coordinator transmits a feedback packet (FBP) at the end of each frame in order to inform about the data packets successfully received. When a device succeeds in transmitting its data packet in any of the  $F_k$  frames, it switches to sleep mode and saves energy until the next data collection round starts. We assume that  $T_R$  is greater than the maximum time  $T_C(k)$  elapsed since the  $k$ th data collection round starts until all devices succeed.

Since this work focuses on the contention process, we assume that all data and control packets are always

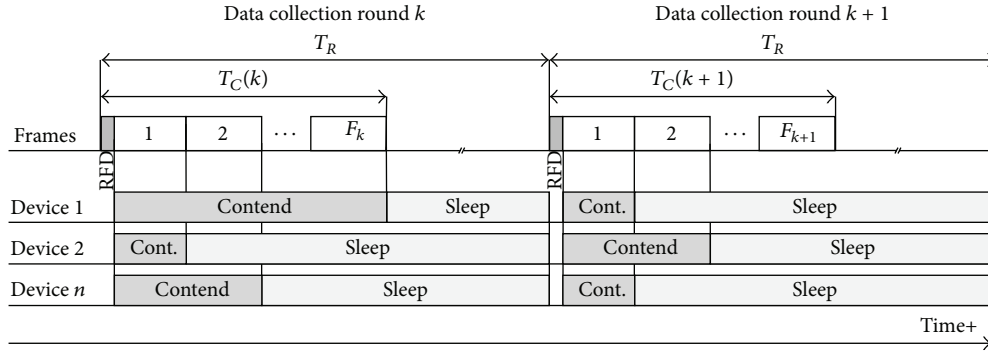


FIGURE 1: Sequence of data collection rounds.

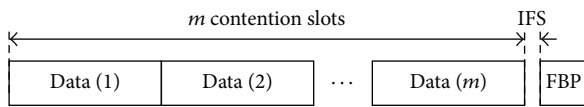


FIGURE 2: Frame structure of FSA and CTA.

transmitted without transmission errors induced by the wireless channel impairments. In addition, we assume that there is no capture effect; that is, when two or more packets collide none of them can be decoded. The inclusion of transmission errors and capture effect and their impact on the overall network performance is left for future work.

#### 4. Medium Access Control Protocols

In this section, we describe the three MAC protocols that we have considered in this paper: FSA, CTA, and DQ. Although we focus on the energy analysis of CTA and DQ, the description of FSA is also included to make the paper self-contained. The energy analysis of FSA can be found in [14].

**4.1. Frame Slotted-ALOHA.** The frame structure of FSA is divided into  $m$  contention slots as shown in Figure 2. Each device randomly selects one of the slots of a frame to transmit its data packet with no Clear Channel Assessment (CCA). Consequently, a given slot can be in one of three states: empty, that is, no data packet has been transmitted, success, that is, only one data packet has been transmitted, or collision, that is, more than one device has transmitted in that slot. At the end of each frame, the coordinator transmits a FBP to inform of the state of every slot of the frame. Devices that have not succeeded yet in frame  $F_i$  will transmit again in frame  $F_{i+1}$ .

As shown in Figure 2, a guard time called Interframe Space (IFS) is left between reception and transmission modes to compensate propagation and processing delays and the time required to switch the radio transceivers between reception and transmission modes.

**4.2. Contention Tree Algorithm.** The frame of CTA has the same structure of FSA shown in Figure 2. At the beginning of each frame, devices choose randomly one of the slots of the frame to transmit their data packet. However, the way that

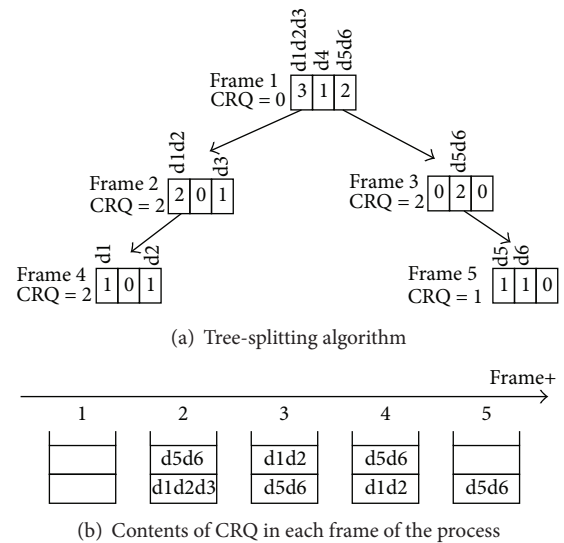


FIGURE 3: Example of CTA with 6 devices (d1 to d6) and 3 slots: (a) tree-splitting algorithm and (b) time diagram with the contents of CRQ.

collisions are resolved in CTA is different to the approach in FSA. In CTA, devices are organized into subgroups using a tree-splitting algorithm [8]. An example of the contention tree formed by the algorithm is represented in Figure 3(a). Each node of the tree represents a frame of 3 slots, and the number in each slot denotes the number of devices that transmit in that slot. At the first frame, all devices transmit their data packet. When two or more devices collide in a specific slot, a new frame is assigned only to the devices that caused the collision in order to reattempt access, and they are queued into a logical queue referred to as Collision Resolution Queue (CRQ). Therefore, if there are  $k$  slots with collision, then  $k$  new frames are scheduled after the current frame, and  $k$  new subgroups of devices are queued into CRQ. The process is repeated, frame after frame, leading to the formation of a tree whose expansion stops at frames which contain only empty and/or successful slots.

CRQ is represented at each device by two integer numbers: (i) the position of the device in CRQ and (ii) the length of CRQ, that is, number of subgroups of devices waiting to



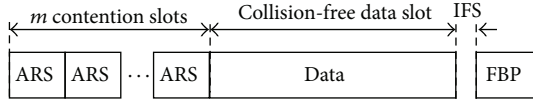


FIGURE 4: Frame structure of DQ.

retransmit in another frame. The length of CRQ is updated by the coordinator at the end of each frame according to the following rules: (1) it is incremented by the number of slots with collision in the current frame and (2) if the length of CRQ  $> 0$  (i.e., CRQ is not empty), it is decremented by one after the current frame. The coordinator broadcasts the length of CRQ and the state of the  $m$  contention slots (success, empty, or collision) in every FBP. With this information, a device that collided in a given frame can compute its new position in CRQ. The position of a device in CRQ is always decremented by one at the end of every frame. When a device occupies the first position, it transmits again in the next frame. Therefore, devices must receive the FBP in those frames where they contend in order to know whether they succeed (to leave CRQ) or collide (to enter again in CRQ). However, devices can sleep during those frames where they do not contend.

The contents of CRQ in every frame are shown in the example illustrated in Figure 3(b). At frame 1, all devices contend: d1, d2, and d3 collide in slot 1; d4 succeeds in slot 2; d5 and d6 collide in slot 3. Thus, d1, d2, and d3 enter in the first position of CRQ; d5 and d6 enter in the second position. At frame 2, only d1, d2, and d3 contend (because they occupy the first position in CRQ): d1 and d2 collide and enter in CRQ again; d3 succeeds and leaves CRQ; d5 and d6 move to the first position of CRQ. At frame 3, d5 and d6 contend, collide, and enter in the second position of CRQ again; d1 and d2 move to the first position of CRQ. At frame 4, d1 and d2 contend and succeed; d5 and d6 move to the first position of CRQ. At frame 5, d5 and d6 contend and succeed.

**4.3. Distributed Queuing.** The frame of DQ is divided in three parts as shown in Figure 4: (i)  $m$  contention slots devoted to the transmission of access requests, (ii) one collision-free data slot for the transmission of data packets, and (iii) a feedback packet. At the first frame, all devices randomly select one of the  $m$  contention slots to transmit an Access Request Sequence (ARS) packet. Depending on whether the ARS packet collides or is successfully decoded by the coordinator, each device is queued into one of two logical queues as follows.

(1) Devices that collided in a given contention slot are queued into CRQ. The length of CRQ and the position of each device in CRQ are updated by executing the rules described for CTA; that is, a given position of CRQ is occupied only by those devices that collided in that specific access slot, thus splitting devices into subgroups according to the access slot where the collision occurred.

(2) Devices that succeeded in transmitting their ARS are queued into the Data Transmission Queue (DTQ). Even though any queue management strategy could be applied,

for the remainder of the paper we will consider that devices transmit their data packet in the collision-free data slot of subsequent frames according to a first-in first-out (FIFO) mechanism.

Analogously to the CRQ, the DTQ is represented at each device by two integer numbers: (i) the position of the device in DTQ and (ii) the length of DTQ, that is, number of devices that have succeeded in transmitting their ARS and wait for their data slot. The length of DTQ is updated by the coordinator at the end of every frame according to the following rules: (1) it is incremented by the number of contention slots with success in the current frame and (2) if the length of DTQ  $> 0$  (i.e., DTQ is not empty), it is decremented by one if one device transmitted data in the previous frame.

The coordinator broadcasts in every FBP the length of DTQ, the length of CRQ, and the state of the  $m$  contention slots. With this information, a device that transmitted an ARS can compute its position in CRQ when it collided or its position in DTQ when it succeeded. The position of a device in CRQ and DTQ is always decremented by one at the end of every frame. Therefore, devices must receive the FBP in those frames where they transmit either ARS or data and switch to sleep mode during those frames where they do not transmit either ARS or data. When a device is in the first position of CRQ, it transmits a new ARS in the next frame. When a device occupies the first position of DTQ, it transmits its data packet in the collision-free data slot of the next frame.

Figure 5 shows an example of operation of DQ. The contents of the slots and the lengths of CRQ and DTQ in every frame are shown in Figure 5(a). The contents of both queues are shown in Figure 5(b). Since the evolution of CRQ is identical to the one in the example of CTA, we only describe here the evolution of DTQ. At frame 1, all devices contend: d4 succeeds in transmitting its ARS and enters in the first position of DTQ. At frame 2: d4 transmits data, and d3 succeeds in transmitting its ARS and enters in DTQ. At frame 3: d3 transmits data. At frame 4: d1 and d2 succeed and enter in DTQ; since DTQ was empty, no device transmits data. At frame 5: d5 and d6 succeed and enter in DTQ and d1 transmits data. At frames 6, 7, and 8: d2, d5, and d6 transmit data, respectively.

## 5. Energy Consumption Analysis

In this section, we analyse the energy consumption of both CTA and DQ. Towards this end, we first formulate the average number of frames required for a device to contend until it succeeds in transmitting a data packet in CTA, or an access request in DQ. Note that this particular expression was derived in [18] and is the same for the two protocols because the underlying mechanism is exactly the same in both cases. Then, using this parameter, we compute the energy consumed per device in a data collection round using either CTA or DQ.

**5.1. Number of Contention Frames per Device.** The number of frames in which a device contends until it succeeds in transmitting a data packet in CTA, or an access request in DQ, is equivalent to the number  $d$  of tree levels required by

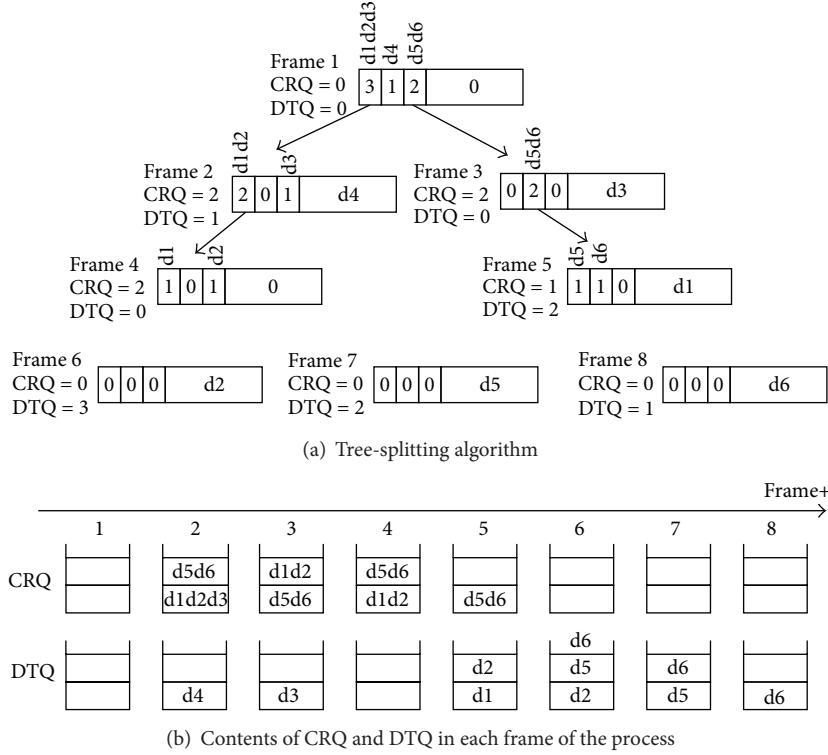


FIGURE 5: Example of DQ with 6 devices (d1 to d6) and 3 contention slots: (a) tree-splitting algorithm and (b) time diagram with the contents of CRQ and DTQ.

a device to succeed. In the examples of Figures 3 and 5,  $d$  is 1, 2, or 3, depending on the device. The probability distribution of  $d$ , when the number of devices is  $n$  and the number of slots per frame is  $m$ , can be formulated as

$$\Pr(d | n) = \begin{cases} 0, & \text{if } (n = 1) \text{ and } (d \neq 1) \\ 1, & \text{if } (n = 1) \text{ and } (d = 1) \\ p_s(d) - p_s(d-1), & \text{if } (n \geq 2), \end{cases} \quad (1)$$

where  $p_s(d)$  is the probability that a slot in level  $d$  selected by a random device is not selected by any of the other  $n-1$  devices, assuming that there are neither transmission errors nor capture effect, and is given by

$$p_s(d) = m^d \frac{1}{m^d} \left(1 - \frac{1}{m^d}\right)^{n-1} = \left(1 - \frac{1}{m^d}\right)^{n-1}, \quad (2)$$

where  $1/m^d$  is the probability that one device selects one of the slots in level  $d$ .

The difference  $p_s(d) - p_s(d-1)$  is the probability that a device requires exactly  $d$  levels to be the only device transmitting in a slot. Then, the average number  $\bar{d}_n$  of levels (frames) required for a device to succeed in transmitting a data packet in CTA, or an access request in DQ, can be expressed as

$$\bar{d}_n = \sum_{d=1}^{\infty} d \Pr(d | n), \quad (3)$$

which is derived in [18] and can be formulated as

$$\bar{d}_n \approx \log_m(n-1) + \left(\frac{1}{2} + \frac{\gamma}{\log m}\right) + \frac{1}{2n \log m}, \quad (4)$$

where Euler's constant  $\gamma \approx 0.5772$  and the logarithm base  $m$  is given by  $\log_m n \equiv \log n / \log m$ , with  $\log n \equiv \ln n$ .

Figure 6(a) shows the value of  $\bar{d}_n$  as a function of the number  $n$  of devices (from 10 to 1000), considering  $m = 5, 10,$  and  $20$  contention slots per frame. As it could be expected, the value of  $\bar{d}_n$  increases logarithmically with  $n$  for a given value of  $m$ . Figure 6(b) shows the value of  $\bar{d}_n$  as a function of the number  $m$  of slots (from 2 to 50), considering  $n = 100, 500,$  and  $1000$  devices. Indeed, the value of  $\bar{d}_n$  decreases exponentially when  $m$  increases. It is worth noting that the value of  $\bar{d}_n$  is finite and less than 12 frames when the number of slots is minimum ( $m = 2$ ) and  $n \leq 1000$  devices. In addition, the values of  $\bar{d}_n$  are very similar when the number of devices is either low or high regardless of the number of contention slots.

**5.2. Energy Analysis of CTA.** The duration of a data collection round using CTA can be expressed as

$$T_R = \bar{T}_{\text{CTA}} + \bar{T}_{\text{sleep}}, \quad (5)$$

where  $\bar{T}_{\text{CTA}}$  is the average time elapsed since the data collection round starts until one device succeeds in transmitting its single data packet to the coordinator and  $\bar{T}_{\text{sleep}}$  is

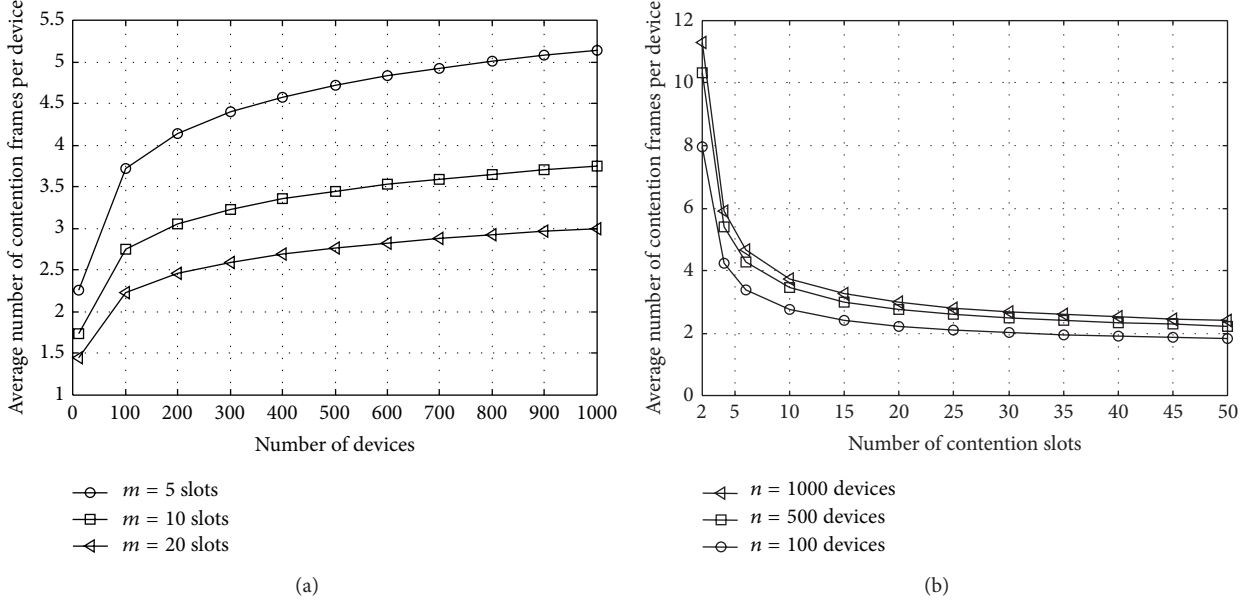


FIGURE 6: Average number of contention frames per device as a function of (a) the number of contending devices and (b) the number of contention slots per frame.

the remaining time of the data collection round in which the device sleeps until the next round starts.

The average time  $\bar{T}_{CTA}$  can be defined as

$$\bar{T}_{CTA} = (\overline{CRQ}_{data} + \overline{CRQ}_{sleep}) T_{frame}^{CTA}, \quad (6)$$

where  $\overline{CRQ}_{data}$  is the average number of frames where a device contends (i.e., it transmits the data packet) until it succeeds;  $\overline{CRQ}_{sleep}$  is the average number of frames where a device is in CRQ without contending (i.e., it sleeps until it has to contend in a future frame);  $T_{frame}^{CTA}$  is the duration of a CTA frame, which is given by

$$T_{frame}^{CTA} = mT_{data} + 2T_{IFS} + T_{FBP}, \quad (7)$$

where  $m$  is the number of slots and  $T_{data}$ ,  $T_{IFS}$ , and  $T_{FBP}$  are the duration of a slot, an IFS, and a FBP packet, respectively.

The average energy consumed by one device in a data collection round using CTA, denoted by  $\bar{E}_R^{CTA}$ , can be expressed as

$$\bar{E}_R^{CTA} = \bar{E}_{CTA} + \rho_{sleep} \bar{T}_{sleep}, \quad (8)$$

where  $\bar{E}_{CTA}$  is the average energy consumed by a device in the CTA contention resolution process and can be expressed as

$$\bar{E}_{CTA} = \overline{CRQ}_{data} E_{data} + \overline{CRQ}_{sleep} \rho_{sleep} T_{frame}^{CTA}, \quad (9)$$

where  $E_{data}$  is the energy consumed by the device in a frame where it contends. The device executes the following operations: (i) transmitting data in 1 slot selected randomly, (ii) remaining in standby mode in the other  $m - 1$  slots, and

(iii) listening to the channel to receive the FBP. Then,  $E_{data}$  can be formulated as

$$E_{data} = \rho_{tx} T_{data} + (m - 1) \rho_{stby} T_{data} + 2\rho_o T_{IFS} + \rho_{rx} T_{FBP}. \quad (10)$$

By substituting  $\bar{T}_{sleep}$  from (5) and (9) in (8), the average energy consumed per device in a CTA data collection round can be expressed as

$$\bar{E}_R^{CTA} = \overline{CRQ}_{data} E_{data} + \overline{CRQ}_{sleep} \rho_{sleep} T_{frame}^{CTA} + \rho_{sleep} (T_R - \bar{T}_{CTA}). \quad (11)$$

Substituting (6) in (11), and after some basic algebra,  $\bar{E}_R^{CTA}$  can be formulated as

$$\bar{E}_R^{CTA} = \overline{CRQ}_{data} E_{data} + \rho_{sleep} (T_R - \overline{CRQ}_{data} T_{frame}^{CTA}). \quad (12)$$

Since we assume that there are neither transmission errors nor capture effect, then  $\overline{CRQ}_{data} = \bar{d}_n$  and  $\bar{E}_R^{CTA}$  can be expressed as

$$\bar{E}_R^{CTA} = \bar{d}_n (E_{data} - \rho_{sleep} T_{frame}^{CTA}) + \rho_{sleep} T_R. \quad (13)$$

**5.3. Energy Analysis of DQ.** The duration of a data collection round using DQ can be expressed as

$$T_R = \bar{T}_{DQ} + \bar{T}_{sleep}, \quad (14)$$

where  $\bar{T}_{DQ}$  is the average time elapsed since the data collection round starts until one device succeeds in transmitting its

single data packet to the coordinator. The average time  $\bar{T}_{DQ}$  can be defined as

$$\bar{T}_{DQ} = \left( \overline{CRQ}_{ARS} + \overline{CRQ}_{sleep} + \overline{DTQ}_{sleep} + \overline{DTQ}_{listen} + \overline{DTQ}_{data} \right) T_{frame}^{DQ}, \quad (15)$$

where  $\overline{CRQ}_{ARS}$  is the average number of frames where a device transmits an access request (i.e., it is in CRQ and contends);  $\overline{CRQ}_{sleep}$  is the average number of frames where a device is in CRQ without contending (i.e., it sleeps until it has to contend in a future frame);  $\overline{DTQ}_{sleep}$  is the average number of frames where a device is in DTQ waiting for its reserved data transmission slot (i.e., the device sleeps);  $\overline{DTQ}_{listen}$  is the average number of frames where a device is in DTQ and listens to the feedback packet of the frame before the one where it has to transmit data to check whether the device in the previous position of DTQ has transmitted data successfully;  $\overline{DTQ}_{data}$  is the average number of frames where a device is in DTQ and has to transmit a data packet;  $T_{frame}^{DQ}$  is the duration of a DQ frame, which is given by

$$T_{frame}^{DQ} = mT_{ARS} + T_{data} + 2T_{IFS} + T_{FBP}, \quad (16)$$

where  $m$  is the number of contention slots and  $T_{ARS}$ ,  $T_{data}$ ,  $T_{IFS}$ , and  $T_{FBP}$  are the duration of a contention slot (for access requests), a collision-free slot (for data), an IFS, and a FBP packet, respectively.

The average energy consumed by one device in a data collection round using DQ, denoted by  $\bar{E}_R^{DQ}$ , can be expressed as

$$\bar{E}_R^{DQ} = \bar{E}_{DQ} + \rho_{sleep} \bar{T}_{sleep}, \quad (17)$$

where  $\bar{E}_{DQ}$  is the average energy consumed by the device in the DQ contention resolution process and can be expressed as

$$\begin{aligned} \bar{E}_{DQ} = & \overline{CRQ}_{ARS} E_{ARS} + \left( \overline{CRQ}_{sleep} + \overline{DTQ}_{sleep} \right) \rho_{sleep} T_{frame}^{DQ} \\ & + \overline{DTQ}_{listen} E_{listen} + \overline{DTQ}_{data} E_{data}. \end{aligned} \quad (18)$$

$E_{ARS}$  is the energy consumed by the device in a frame where it transmits an access request. The device executes the following operations: (i) transmitting an ARS in 1 contention slot selected randomly, (ii) remaining in standby mode in the other  $m - 1$  contention slots and in the data slot, and (iii) listening to the channel to receive a FBP. Then,  $E_{ARS}$  can be formulated as

$$\begin{aligned} E_{ARS} = & \left( \rho_{tx} + (m - 1) \rho_{stby} \right) T_{ARS} \\ & + \rho_{stby} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}. \end{aligned} \quad (19)$$

$E_{listen}$  is the energy consumed by the device in a frame where it only listens to the feedback packet of the frame before it has to transmit. The device executes the following operations: (i) sleeping in the  $m$  contention slots and in

the data slot and (ii) listening to the channel to receive the FBP. Then,  $E_{listen}$  can be formulated as

$$E_{listen} = m\rho_{sleep} T_{ARS} + \rho_{sleep} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}. \quad (20)$$

$E_{data}$  is the energy consumed by a device in a frame where it transmits a data packet. The device executes the following operations: (i) remaining in standby mode in the  $m$  contention slots, (ii) transmitting data in the collision-free data slot, and (iii) listening to the channel to receive the FBP at the end of the frame. Then,  $E_{data}$  can be formulated as

$$E_{data} = m\rho_{stby} T_{ARS} + \rho_{tx} T_{data} + 2\rho_{\sigma} T_{IFS} + \rho_{rx} T_{FBP}. \quad (21)$$

By substituting  $\bar{T}_{sleep}$  from (14) and (18) in (17), the average energy consumed per device in a DQ data collection round can be expressed as

$$\begin{aligned} \bar{E}_R^{DQ} = & \overline{CRQ}_{ARS} E_{ARS} + \left( \overline{CRQ}_{sleep} + \overline{DTQ}_{sleep} \right) \rho_{sleep} T_{frame}^{DQ} \\ & + \overline{DTQ}_{listen} E_{listen} + \overline{DTQ}_{data} E_{data} + \rho_{sleep} \left( T_R - \bar{T}_{DQ} \right). \end{aligned} \quad (22)$$

Substituting  $\bar{T}_{DQ}$  (15) in (22) and after some basic algebra,  $\bar{E}_R^{DQ}$  can be formulated as

$$\begin{aligned} \bar{E}_R^{DQ} = & \overline{CRQ}_{ARS} E_{ARS} + \overline{DTQ}_{listen} E_{listen} + \overline{DTQ}_{data} E_{data} \\ & + \rho_{sleep} \left( T_R - \left( \overline{CRQ}_{ARS} + \overline{DTQ}_{listen} + \overline{DTQ}_{data} \right) T_{frame}^{DQ} \right). \end{aligned} \quad (23)$$

Since we assume that there are neither transmission errors nor capture effect, then  $\overline{CRQ}_{ARS} = \bar{d}_n$  (4),  $\overline{DTQ}_{listen} = 1$ , and  $\overline{DTQ}_{data} = 1$ . Therefore,  $\bar{E}_R^{DQ}$  can be expressed as

$$\begin{aligned} \bar{E}_R^{DQ} = & \bar{d}_n \left( E_{ARS} - \rho_{sleep} T_{frame}^{DQ} \right) + E_{listen} + E_{data} \\ & + \rho_{sleep} \left( T_R - 2T_{frame}^{DQ} \right). \end{aligned} \quad (24)$$

In the next section, we validate the energy analysis for both CTA and DQ and evaluate the devices' energy consumption for different network configurations.

## 6. Model Validation and Performance Evaluation

The energy models of DQ and CTA formulated in Section 5 have been validated by means of computer simulations using MATLAB. The results of 1000 simulation samples have been averaged for each test case. Results show a tight match between analysis and simulation for both protocols in all tested cases, thus validating the correctness of the theoretical analyses. Table 1 summarizes the system parameters used to validate the analytical models and evaluate the performance.



TABLE 1: System parameters.

Parameter	Value	Parameter	Value	Parameter	Value
MAC header	8 bytes	Data-rate	250 kbps	Data payload	114 bytes
FBP payload	$m \cdot 2 \text{ bits/slot} + 4 \text{ bytes}$	CRC	2 bytes	$T_R$	3600 s
$T_{\text{preamble}}$	160 $\mu\text{s}$	$T_{\text{IFS}}$	192 $\mu\text{s}$	$\rho_{\text{tx}}$	100.8 mW
$\rho_{\text{stby}}$	525 $\mu\text{W}$	$\rho_{\text{tx}} = \rho_{\sigma}$	66.9 mW	$\rho_{\text{sleep}}$	90 nW

They have been selected from the specifications of the CC2520 [24] transceiver and from the IEEE 802.15.4 standard. Thus, the performance evaluation has been particularized for IEEE 802.15.4 technology. We assume that  $\rho_{\sigma} = \rho_{rx}$  during Interframe Spaces, and we have considered that the coordinator initiates one data collection round every hour; that is,  $T_R = 3600 \text{ s}$ ; the FBP packet includes 2 bits to inform about the status of each contention slot (i.e., empty, success, or collision), 2 bytes for the length of CRQ, 2 bytes for the length of DTQ, and ARS packets of 10 bytes composed of a physical layer preamble, a MAC header, and Cyclic Redundancy Code (CRC) for error control.

In the next sections, we first evaluate the energy performance of CTA and DQ to determine the criteria to minimize the energy consumption of the devices. Secondly, we compare the energy performance of DQ, CTA, and FSA over the number of contending devices in dense M2M networks and compute the amount of energy consumed in each mode of operation. Finally, we compare the energy consumed per device using DQ, CTA, and FSA over the data packet length. The operation of DQ, CTA, and FSA has been implemented without any simplification. Results for FSA have been obtained through computer-based simulations also with MATLAB.

**6.1. Number of Contention Slots per Frame.** The average energy consumption of one device in a data collection round using DQ and CTA is represented in Figure 7(a) over the number  $m$  of contention slots (from 2 to 50 slots). We have considered  $n = 100, 500, \text{ and } 1000$  devices.

Results show that the average energy consumed per device tends to a minimum value when  $m \geq 10$  and  $m \geq 20$  in DQ and CTA, respectively, regardless of the number  $n$  of devices. The independency of the results with  $n$  relaxes the need to know the size of the network. Contrarily, as it was demonstrated in [10, 11], the number of contention slots in FSA has to be adjusted according to the number of devices (using  $m = n$ ) to minimize the energy consumed per device. As it could be expected, the average energy consumed per device in a data collection round using CTA and DQ increases exponentially, up to a finite value, when the number of contention slots decreases. This is due to the fact that the probability of collision among devices becomes higher when  $m$  decreases, thus leading the value of  $\bar{d}_n$  to increase exponentially. It can be also observed that the energy consumed per device when  $m = 2$  is 1.5 and 3 times bigger than the respective minimum values of DQ and CTA for higher values of  $m$ . On the contrary, as it was demonstrated in [11], the energy consumed per device using FSA tends to infinity when

$m$  is very low. Indeed, since the tree-splitting algorithm splits the devices into subgroups in order to reduce the probability of collision, the number of contending frames in DQ and CTA is much lower than in FSA when  $m$  is small. Finally, it is worth mentioning that the average energy consumed per device in a data collection round using DQ and CTA tends to the minimum value when the value of  $m$  increases. Indeed, although a higher value of  $m$  leads to longer sleep periods, the use of an ultralow power sleep mode (90 nW [24]) yields reduced energy consumption.

**6.2. Number of Devices.** The average energy consumed per device in a data collection round using DQ, CTA, and FSA is represented in the left vertical axis of Figure 7(b) over the number  $n$  of devices (from 10 to 5000 devices). We have considered the number of contention slots that minimizes the energy consumption of one device for each protocol:  $m = 10$  in DQ,  $m = 20$  in CTA, and  $m = n$  in FSA [10, 11, 20]. Therefore, we are comparing the optimal performance of all of them.

As it can be observed, the average energy consumed per device using DQ and CTA increases logarithmically with the number of devices. Indeed, according to (13) and (24), this behavior is due to the logarithmic nature of the value of  $\bar{d}_n$  as expressed in (4). In contrast, the average energy consumed per device using FSA increases linearly with the number of devices and is much higher than in DQ and CTA.

The energy savings provided by DQ with respect to FSA and CTA are shown on the right vertical axis of Figure 7(b). As it can be observed, the energy savings increase with the number  $n$  of devices. When  $n > 5000$  devices, DQ provides energy savings of more than 35% with respect to CTA and more than 80% with respect to FSA. In addition, it is worth noting that the average energy consumed per device is very similar with either FSA or CTA when  $n < 500$ . Therefore, the use of CTA improves the energy efficiency of FSA only when the number of devices is very high. However, DQ provides energy savings of up to 35% with respect to FSA and CTA even when  $n < 500$ . As it could be expected, the energy consumption is further reduced with DQ with respect to that of both CTA and FSA for any number of devices. Indeed, while the contention process in CTA and FSA is done through the transmission of data packets, DQ uses shorter contention slots to transmit access requests, and thus the energy efficiency is improved.

**6.3. Energy Distribution per Radio Operating Mode.** Figure 8(a) shows the distribution of the average energy consumed by one device in the different modes of operation

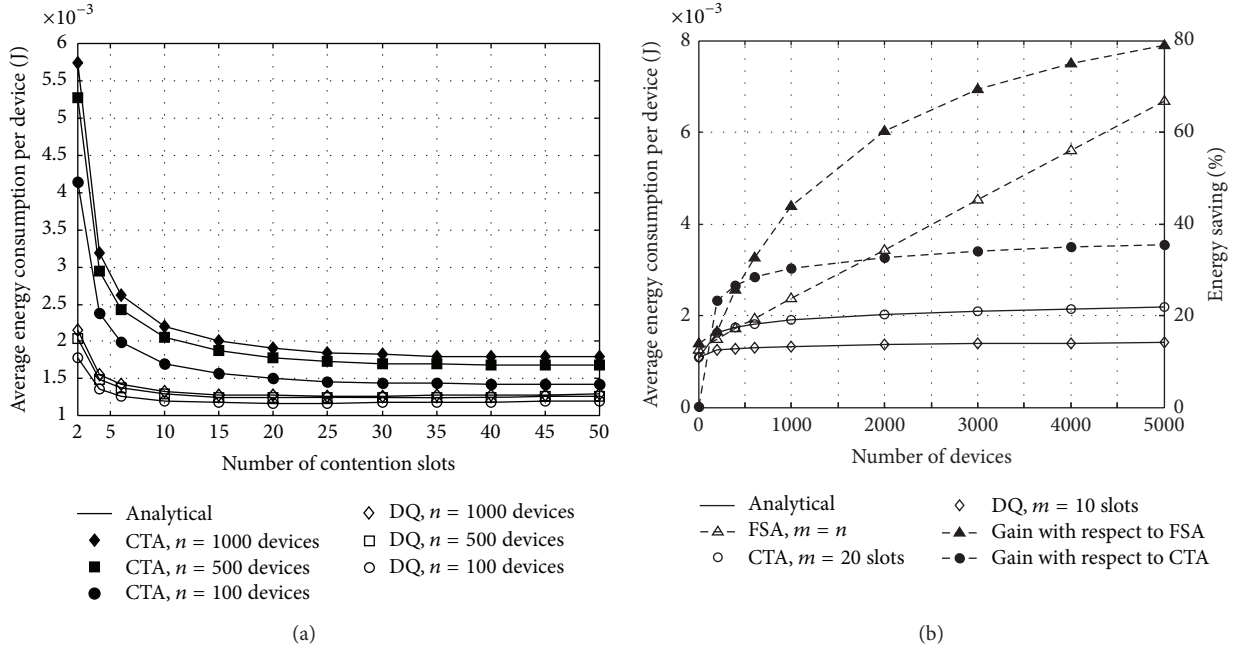


FIGURE 7: (a) Average energy consumed per device in a data collection round using DQ and CTA over the number  $m$  of contention slots and (b) average energy consumed (left axis) and energy saving (right axis) per device in a data collection round using  $m = 10$  contention slots in DQ,  $m = 20$  in CTA, and  $m = n$  in FSA.

of the device's radio transceiver (transmission, reception, standby, and sleep) using DQ, CTA, and FSA in a data collection round. We have considered  $n = 100, 500$ , and  $1000$  devices,  $m = 10$  in DQ,  $m = 20$  in CTA, and  $m = n$  in FSA. Again, we compare the performance of the protocols when operating at their respective optimal configuration.

In transmission mode, the energy consumption using DQ increases very slightly with the number  $n$  of devices. On the contrary, the energy consumption increases with  $n$  using CTA, and it is considerably higher than in DQ. Indeed, this is due to the fact that the contention process in DQ is done using shorter contention slots for access requests. In FSA, the energy consumption associated with the transmission mode is not sensitive to the number of devices. This behavior in FSA indicates that the average number of frames where each device has to contend in FSA is almost constant with the number of devices when using  $m = n$ .

In reception mode, as it could be expected, the energy consumption using DQ and CTA increases slightly with the number of devices because the feedback packet has the same length for all  $n$ . Contrarily, the energy consumption in FSA increases linearly with  $n$  because the length of the feedback packet also increases since it informs about the state of all the slots in every frame.

**6.4. Data Packet Length.** The average energy consumed per device in a data collection round using DQ, CTA, and FSA is represented in Figure 8(b) over the data packet length. We have considered  $n = 500$  and  $1000$  devices,  $m = 10$  in DQ,  $m = 20$  in CTA, and  $m = n$  in FSA.

As it can be observed, in all cases the average energy consumed per device increases linearly with the data packet length. Indeed, the longer the data packet, the higher the energy to transmit the packet and the higher the wastage in case of collision, in standby and sleep. It is worth noting that the slope of the curves associated with DQ is much lower than in CTA or FSA, thus being less sensitive to the data packet length. This is due to the lack of data packet collisions in DQ.

It is also worth mentioning that while DQ outperforms FSA in all cases, it shows worse energy consumption than CTA when the data packet length is very low (below 35 bytes for the considered simulation layout). Indeed, CTA outperforms DQ when the ratio between the number of useful data bits and the overhead of DQ (i.e., contention slots and feedback) is reduced below a certain threshold. The gain of DQ comes from the avoidance of data collisions and the concentration of collisions in the shorter access requests. Therefore, the gains of DQ become apparent when the data packet length is greater than the length of the access requests.

## 7. Conclusions

Future communication networks must deal with an unprecedented number of simultaneous connections. This will become a great challenge in dense M2M networks where every single object may be connected to the network. For such kind of networks with massive number of connections, it is essential to design and analyze the performance of MAC protocols which can deal with a huge number of simultaneous users and still ensure high energy efficiency. While it is well

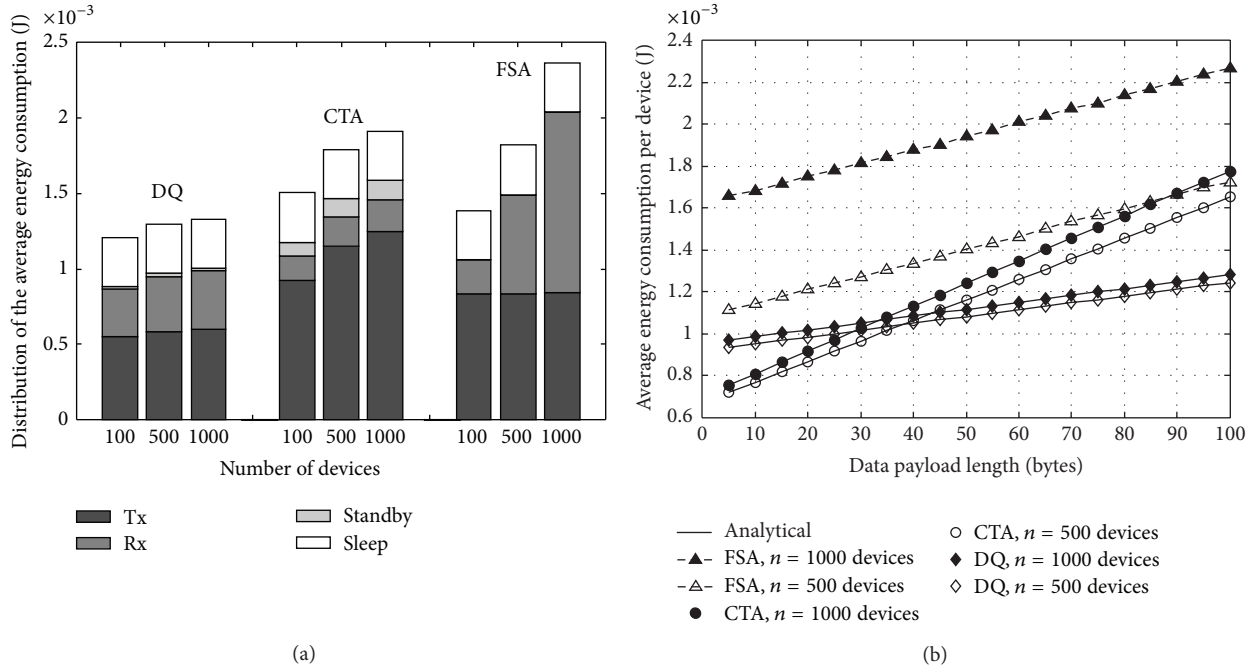


FIGURE 8: (a) Distribution of the average energy consumed per device in each radio operating mode using  $m = 10$  contention slots in DQ,  $m = 20$  in CTA, and  $m = n$  in FSA and (b) average energy consumed per device in a data collection round over the data packet length.

known that tree-based algorithms can deal with great number of devices, their energy efficiency in data collection M2M networks has remained unknown to date. Motivated by this, in this paper we have theoretically analyzed the energy consumption of two types of contention tree-based access protocols to periodically transmit data to a coordinator in M2M networks: the Contention Tree Algorithm (CTA) and Distributed Queuing (DQ) access protocol. In addition, we have also compared their performance with that of Frame Slotted-ALOHA (FSA) in terms of energy efficiency, since FSA is one of the most common MAC protocols used today to solve the contention among vast number of devices. Results show that there is a frame length for CTA and DQ, that is, number of contention slots per frame, which minimizes the energy consumption of the devices regardless of the total number of contending devices. This is not the case of FSA, where the frame length must be optimized according to the number of devices. In the case of DQ and CTA, the average energy consumed per device increases exponentially when the frame length decreases. However, it tends to a finite value in DQ and CTA, thus ensuring the stability of the network. In addition, the energy consumption using DQ and CTA increases logarithmically with the number of devices, while it increases linearly with FSA. This leads to an energy saving as the network becomes bigger. In particular, when the number of devices is above 5000, DQ provides energy savings of more than 35% with respect to CTA and 80% with respect to FSA. Finally, the analysis shows that the energy savings provided by DQ with regard to CTA increase with the length of the data packets. Therefore, the use of DQ and CTA can improve considerably the energy efficiency of dense M2M

networks when the number of devices is large and unpredictable. Future work aims at including transmission errors and the capture effect in the analysis and modelling presented in this paper as well as at experimentally evaluating the energy models of DQ and CTA in a real M2M testbed with hundreds of end-devices. Preliminary results can be found in [17]. In addition, since the energy efficiency provided by DQ increases with shorter contention slots, novel collision detection techniques will be investigated in order to enable the use of very short access request sequences such as physical preambles. This approach would make DQ a quasioptimal MAC protocol for M2M networks with a huge, unknown, and dynamic number of end-devices.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work has been supported by the Research Projects ADVANTAGE (FP7-MCA-607774), P2P-SmarTest (H2020-646469) and by the Generalitat de Catalunya under Grant 2014 SGR 1551.

## References

- [1] Technical Specification ETSI, "Machine-to-machine communications (M2M); functional architecture," TS 102 690, v1.1.1, 2011.

- [2] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.
- [3] L. Roberts, "ALOHA packet system with and without slots and capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.
- [4] IEEE Standard, "Part 15.4: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications for low-rate Wireless Personal Area Networks (WPANs)," IEEE Standard 802.15.4-2006, 2006.
- [5] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," IEEE Standard 802.11, 1999.
- [6] ISO/IEC 18000-6:2010, "Information technology—radio frequency identification for item management—part 6: parameters for air interface communications at 860 MHz to 960 MHz".
- [7] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: part 1—CSMA modes and their throughput-delay characteristics," *IEEE Transactions on Communications*, vol. 23, no. 12, pp. 1400–1416, 1975.
- [8] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 505–515, 1979.
- [9] J. L. Massey, "Collision resolution algorithm and random access communications," in *Multiuser Communication Systems*, G. Longo, Ed., pp. 73–137, Springer, New York, NY, USA, 1981.
- [10] X. Su and Y. Xiao, "Analysis of energy consumption for multiple object identification system with active RFID tags," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 4019–4023, March 2007.
- [11] Y. Xinqing and L. Xuemei, "Evaluating the energy consumption of RFID tag collision resolution protocols," in *Proceedings of the 6th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '10)*, pp. 215–221, February 2010.
- [12] V. Namboodiri and L. Gao, "Energy-aware tag anticollision protocols for RFID systems," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 44–59, 2010.
- [13] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted ALOHA for passive RFID tag anticollision," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 19–31, 2013.
- [14] F. Vázquez-Gallego, J. Alonso-Zarate, and L. Alonso, "Energy and delay analysis of contention resolution mechanisms for machine-to-machine networks based on low-power WiFi," in *Proceedings of the IEEE International Conference on Communications (ICC '13)*, pp. 2235–2240, IEEE, Budapest, Hungary, June 2013.
- [15] W. Xu and G. Campbell, "A near perfect stable random access protocol for a broadcast channel," in *Proceedings of the IEEE International Conference on Communications (ICC '92)*, vol. 1, pp. 370–374, Chicago, Ill, USA, June 1992.
- [16] J. Alonso-Zarate, E. Kartsakli, A. Cateura, C. Verikoukis, and L. Alonso, "A near-optimum cross-layered distributed queuing protocol for wireless LAN," *IEEE Wireless Communications*, vol. 15, no. 1, pp. 48–55, 2008.
- [17] P. Tuset-Peiro, L. Alonso, F. Vázquez-Gallego, J. Alonso-Zarate, and X. Vilajosana-Guillen, "Demonstrating low-power distributed queuing for active RFID communications at 433 MHz," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS '14)*, pp. 157–158, Toronto, Canada, April-May 2014.
- [18] A. J. E. M. Janssen and M. J. M. de Jong, "Analysis of contention tree algorithms," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2163–2172, 2000.
- [19] M. A. Kaplan and E. Gulko, "Analytic properties of multiple-access trees," *IEEE Transactions on Information Theory*, vol. 31, no. 2, pp. 255–263, 1985.
- [20] V. Namboodiri, M. Desilva, K. Deegala, and S. Ramamoorthy, "An extensive study of slotted Aloha-based RFID anti-collision protocols," *Computer Communications*, vol. 35, no. 16, pp. 1955–1966, 2012.
- [21] C. T. Wu and G. Campbell, "Extended DQRAP (XDQRAP). A cable TV protocol functioning as a distributed switch," in *Proceedings of the 1st IEEE International Workshop on Community Networking Integrated Multimedia Services to the Home*, pp. 191–198, San Francisco, Calif, USA, 1994.
- [22] H. J. Lin and G. Campbell, "PDQRAP—prioritized distributed queueing random access protocol," in *Proceedings of the 19th Conference on Local Computer Networks*, pp. 82–91, Minneapolis, Minn, USA, October 1994.
- [23] C. T. Wu and G. Campbell, "Interleaved DQRAP with global TQ," Internal Rep., Department of Computer Science, Illinois Institute of Technology, 1994.
- [24] CC2520 datasheet, <http://www.ti.com/lit/ds/symlink/cc2520.pdf>.





**Hindawi**  
Submit your manuscripts at  
<http://www.hindawi.com>

