Energy Analysis of a Contention Tree-based Access Protocol for Machine-to-Machine Networks with Idle-to-Saturation Traffic Transitions

F. Vázquez-Gallego*, J. Alonso-Zarate*, P. Tuset-Peiró†, and L. Alonso‡
*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Castelldefels, Barcelona, Spain
{francisco.vazquez, jesus.alonso}@cttc.es
†Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), Barcelona, Spain
peretuset@uoc.edu
‡Universitat Politècnica de Catalunya (UPC), Castelldefels, Barcelona, Spain
luisg@tsc.upc.edu

***Abstract*—Machine-to-Machine (M2M) area networks must provide connectivity between an M2M gateway and a large number of energy-constrained M2M devices. Attaining high energy efficiency is essential in order to prolong devices lifetime. In this paper, we consider a wireless M2M area network composed of hundreds or even thousands of dormant devices that wake up periodically to transmit data upon request from a gateway. We theoretically analyze the energy efficiency of a Medium Access Control (MAC) protocol that uses a tree-splitting algorithm to resolve the collisions among devices: the Distributed Queuing (DQ) access. Computer-based simulations have been carried out to validate the accuracy of the analytical model and to evaluate and compare the energy consumption of devices using also a basic Contention Tree Algorithm (CTA) and Frame Slotted-ALOHA (FSA). Results show that DQ can reduce energy consumption in more than 35% with respect to CTA and in more than 80% with respect to FSA in dense M2M area networks with devices in compliance with the IEEE 802.15.4 physical layer.**

***Index Terms*—collision resolution, tree splitting, Distributed Queuing, energy analysis, performance evaluation.**

I. Introduction

Energy-efficiency is one of the major challenges in the deployment of densely populated Machine-to-Machine (M2M) networks. These networks aim at connecting a huge number of M2M devices that must operate autonomously for years and, in many cases, with none or very limited access to energy sources. This paper is focused on data collection applications, such as Automatic Meter Reading (AMR) or Asset Tracking, using wireless M2M area networks [1]. These networks are composed of hundreds or even thousands of devices that periodically transmit data upon request from a coordinator, e.g., an M2M gateway. Although the amount of data traffic generated by every device is low, the number of devices attempting to get access to the channel simultaneously can be potentially larger than the one manageable by traditional Medium Access Control (MAC) protocols.

This work was supported in part by the Research Projects CO2GREEN (TEC2010-20823), GREEN-T (TSI-020400-2011-16), NEWCOM# (FP7-318306), and by the Catalan Government under grant 2009SGR1046.

The high density of M2M networks makes it difficult to maintain knowledge of the network topology and to apply a deterministic centralized scheduling that allows every device to transmit without collisions. Therefore, devices must compete for the channel using random access protocols such as Carrier Sense Multiple Access (CSMA) or ALOHA [2]. These protocols do not require knowledge of the network topology *a priori* and their simplicity makes them ideal for low-cost and low-complexity devices. Unfortunately, they suffer from degraded performance [3] in terms of delay, throughput, and energy consumption when the traffic load is high or when the number of devices is large, due to the high probability of collision.

A strategy to improve the performance of random access consists in using a Collision Resolution Algorithm (CRA). CRAs resolve collisions by organizing the retransmission of colliding packets in such a way that every packet is always transmitted successfully with finite delay and energy consumption. The basic CRA is the tree-splitting algorithm [4], also referred to as Contention Tree Algorithm (CTA). It iteratively splits large groups of contending devices into sub-groups in order to reduce collisions in an efficient manner. The delay and energy performance of CTA has been studied in different types of applications, e.g., Radio Frequency Identification (RFID) [5] or sensor networks [6], among others. These works consider that devices are synchronized to a common time frame pattern where every frame is divided into a number of slots in which devices contend to transmit data. Results show that there is a frame length, i.e., a particular number of slots per frame, that minimizes delay and energy consumption and which is independent of the number of contending devices. This makes CTA very appealing when the number of devices is high and unknown, which may be the case in many M2M networks.

While CTA uses data slots to resolve contention, a tree-splitting approach can be also implemented to make reservations or requests for data transmission. This mechanism is called Distributed Queuing (DQ) access. DQ was introduced in the DQ Random Access protocol [7] (DQRAP) for cable TV distribution, and later was applied to Wireless Local Area Networks (DQCA) [8]. In DQ access, the collision resolution is separated from the transmission of data. Since the duration of a request is shorter than the transmission time of a data packet, DQ may achieve even higher energy efficiency than CTA.

To the best of our knowledge, previous works related to DQ have analyzed throughput in steady-state conditions and assuming that all devices generate traffic according to a random Poison distribution. Under this type of traffic, DQ achieves maximum performance when only 3 access request slots per frame are used, regardless of the total number of contending devices. Nevertheless, the energy performance of DQ has never received attention in applications with abrupt idle-to-saturation transitions, i.e., when a huge number of devices have data ready to transmit in a given time and attempt to get access to the channel simultaneously. This is the main motivation for the work presented in this paper, where we focus on analyzing the conditions when DQ can reduce the energy consumption of dense M2M networks with respect to traditional approaches based on CTA or Frame Slotted-ALOHA (FSA) [6].

Therefore, the main contributions of this paper are:

1) We formulate an accurate energy model of a MAC protocol based on DQ for M2M applications with abrupt idle-to-saturation transitions in the data traffic.

2) We evaluate and compare the performance of DQ with respect to CTA and FSA in terms of average energy consumption. For this purpose, we consider devices equipped with radio transceivers in compliance with the IEEE 802.15.4 Standard [9], typically used in M2M networks.

The remainder of this paper is organized as follows. In Section II and Section III, we describe the system model and the DQ access protocol, respectively. In Section IV, we present the analysis and formulate the energy model of DQ. Section V is devoted to validate the model and to evaluate the performance through comprehensive computer-based simulations. Finally, Section VI concludes the paper.

II. System Model

We consider a single-hop wireless network with a coordinator surrounded by n devices in star topology. Devices can be in five different modes of operation: i) transmitting a packet, ii) receiving, iii) idle listening, iv) standby, or v) sleeping. The associated power consumptions are ρ_{tx}, ρ_{rx}, ρ_{σ}, ρ_{stby}, and ρ_{sleep}, respectively. We assume that $\rho_{\sigma} = \rho_{rx}$, and the energy required by a device to switch between inactive (i.e., standby, sleep) and active modes (i.e., transmitting, receiving, idle listening) is negligible. In sleep mode, devices' radio transceivers are fully disabled, thus providing the lowest power consumption. In contrast, $\rho_{stby} > \rho_{sleep}$ and the transition time from standby to active mode is shorter than from sleep mode. Therefore, we consider that devices use the sleep mode when they spend more time inactive than switching back and forth between inactive and active modes.

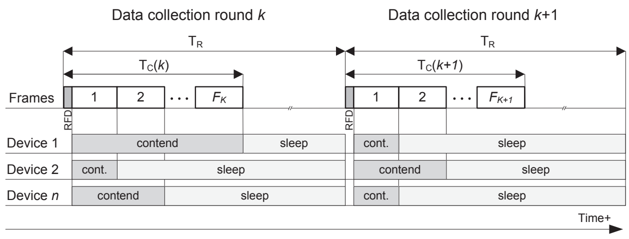

Figure 1. Sequence of data collection rounds.

Periodically, every T_R seconds, the coordinator broadcasts a Request for Data (RFD) packet to initiate a data collection round, as depicted in Figure 1. We assume that each device has one data packet ready to transmit to the coordinator at the beginning of every data collection round. The k-th data collection round is organized into a sequence of F_k time frames. After decoding a RFD, all devices are synchronized to a common time frame pattern and transmit their data packet according to the rules of the adopted MAC protocol, which defines the structure of the frames. When the data packet is successfully received by the coordinator in any of the F_k frames, then an acknowledgement is piggy-backed by the coordinator at the end of the frame so that the device can switch to sleep and save energy until the next data collection round. All devices wake up again to listen to the channel when the coordinator sends the RFD of every data collection round. 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 in transmitting their data packet.

In order to focus the analysis on the contention process, we assume that all data and control packets are always transmitted without transmission errors due to the wireless channel. In addition, when two or more packets collide, none of them can be decoded by the coordinator (i.e., there is no capture effect). The inclusion of channel errors and capture effect is left for future work.

III. Medium Access Control Protocol

The basic idea of the DQ access protocol is to concentrate access requests in a short contention window while data transmission is kept collision-free. When a device succeeds in transmitting its access request, it waits for a collision-free data slot to transmit its data packet. The frame structure of DQ is divided in three parts as shown in Figure 2: (i) m contention slots devoted to the transmission of access requests, (ii) one collision-free data slot, and (iii) a feedback packet (FBP). A guard time called Inter Frame 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.

At the beginning of every data collection round, devices randomly select one of the m contention slots to transmit an Access Request Sequence (ARS) packet. A given contention slot can be in one of three states: empty, i.e., no ARS has been transmitted, success, i.e., only one ARS has been transmitted,

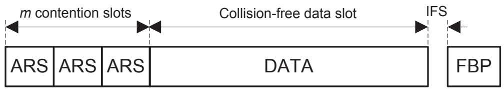

Figure 2. Frame structure of DQ.

or collision, i.e., more than one device has transmitted in that slot. Depending on whether the ARS packet collides or is successfully decoded by the coordinator, every device is queued into one of two logical and distributed queues:

1) Devices that collide transmitting their ARS are queued into the Collision Resolution Queue (CRQ). The length of CRQ and the position of devices in CRQ are updated by executing the tree-splitting algorithm represented in the example of Figure 3.a. Each node of the tree represents a frame of 3 contention slots, and the number in each slot denotes the number of devices that transmit an ARS in that slot. At frame 1, all devices contend. If two or more devices collide in a slot, a new frame is assigned only to devices that caused the collision in order to reattempt access, and they are queued into CRQ. Therefore, if there are k slots with collision, then k new frames are scheduled after the current frame, and k new sub-groups of devices are queued in CRQ. Devices in the first position of CRQ always contend in the next frame by selecting an access slot at random. The process is repeated leading to the formation of a tree whose expansion stops at frames which contain only empty and/or successful slots.

2) Devices that succeed in transmitting their ARS are queued into the Data Transmission Queue (DTQ). Even though any queue management strategy could be applied, we 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. When a device occupies the first position of DTQ, it transmits its data packet in the collision-free data slot of the next frame.

CRQ and DTQ are characterized at every device by 2 integer numbers: $i)$ the position of the device in the queue, and $ii)$ the length of the queue, i.e., total number of elements in the queue. The length of CRQ represents the number of sub-groups of devices waiting to retransmit an ARS. The length of DTQ represents the number of devices that have succeeded in transmitting an ARS and wait for a data transmission slot. The coordinator updates the length of CRQ and DTQ at the end of every frame according to the following rules: $1)$ the length of CRQ is incremented by the number of contention slots with collision in the previous frame; $2)$ if the length of CRQ>0, then it is decremented by one after the current frame; $3)$ the length of DTQ is incremented by the number of contention slots with success in the previous frame; and $4)$ if the length of DTQ>0, it is decremented by one if there was a data packet transmitted in the previous frame. In the example of Figure 3, the contents of the slots and the lengths of CRQ and DTQ in each frame are shown in Figure 3.a. The contents of CRQ and DTQ are shown in Figure 3.b.

The coordinator broadcasts in every FBP the length of the two queues and the state of the m contention slots. With this information, a device which transmitted an ARS can compute its position in CRQ when it collided, or its position in DTQ when succeeded. The position of a device in CRQ and DTQ is always decremented by one at the end of each frame. Therefore, devices receive the FBP in those frames where they transmit either ARS or data, and they switch to sleep mode in those frames where they do not transmit either ARS or data.

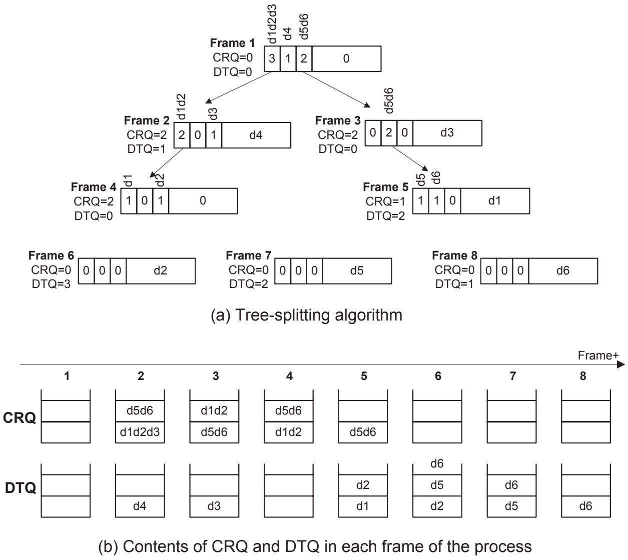

Figure 3. Example of DQ with 6 devices (d1 to d6) and 3 contention slots: (a) tree-splitting algorithm, and (b) contents of CRQ and DTQ in each frame.

IV. Energy Consumption Analysis

In this section, we first derive the average number of frames required for a device to contend until it succeeds in transmitting an ARS. Then, we compute the energy consumed by one device in a data collection round using DQ.

A. Average Number of Contention Frames per Device

The number of frames where a device has to contend until it succeeds in transmitting an access request is equivalent to the number d of tree levels required by a device to transmit an ARS until it succeeds. In the example of Figure 3, d is 1, 2, or 3 levels 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

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

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 no transmission errors, and is given by

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

where $1/m^d$ is the probability that a device selects one of the slots in level d. The difference between $p_s(d)$ and $p_s(d-1)$ is the probability that the random device requires precisely d levels to be the only occupant of a contention slot.

The average number $\overline{d_n}$ of levels (or frames) required for a device to have a successful transmission can be expressed as

$$\overline{d_n} = \sum_{d=1}^{\infty} d \Pr(d|n), \tag{3}$$

which is derived in [10] and can be formulated as

$$\overline{d_n} \simeq \log_m(n-1) + \left(\frac{1}{2} + \frac{\gamma}{\log m}\right) + \frac{1}{2n \log m}, \tag{4}$$

where the Euler's constant $\gamma \approx 0.5772$. From expression (4), it can be observed that $\overline{d_n}$ increases logarithmically with n for a given value of m. In addition, it is worth noting that the value of $\overline{d_n}$ is finite when m is very low, and the values of $\overline{d_n}$ are very similar when the number n of devices is either low or high, regardless of the number m of slots.

B. Energy Analysis

The duration of a data collection round can be expressed as

$$T_R = \overline{T}_{DQ} + \overline{T}_{sleep}, \tag{5}$$

where $\overline{T}_{DQ}$ is the average time elapsed since the data collection round starts until one device is capable of successfully transmitting a single data packet to the coordinator, which can be defined as

$$\begin{aligned}\overline{T}_{DQ} = &\left(\overline{CRQ}_{ARS} + \overline{CRQ}_{sleep}\right) T^{DQ}_{frame} + \\ &+ \left(\overline{DTQ}_{sleep} + \overline{DTQ}_{listen} + \overline{DTQ}_{data}\right) T^{DQ}_{frame},\end{aligned} \tag{6}$$

where $\overline{CRQ}_{ARS}$ is the average number of frames where a device contends (i.e., it transmits an ARS); $\overline{CRQ}_{sleep}$ is the average number of frames where a device is in CRQ without contending (i.e., it sleeps until it has to contend); $\overline{DTQ}_{sleep}$ is the average number of frames where a device is in DTQ waiting for its collision-free data slot (i.e., it 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 in order 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 data; and T^{DQ}_{frame} is the duration of a frame, which is given by

$$T^{DQ}_{frame} = mT_{ARS} + T_{data} + 2T_{IFS} + T_{FBP}, \tag{7}$$

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 data slot, an IFS, and the time of transmission of a FBP packet, respectively.

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

$$\overline{E}^{DQ}_R = \overline{E}_{DQ} + \rho_{sleep}\overline{T}_{sleep}, \tag{8}$$

where $\overline{E}_{DQ}$ is the average energy consumed by a device in the DQ contention resolution process, which can be expressed as

$$\begin{aligned}\overline{E}_{DQ} = &\overline{CRQ}_{ARS} E_{ARS} + \\ &+ \left(\overline{CRQ}_{sleep} + \overline{DTQ}_{sleep}\right) \rho_{sleep} T^{DQ}_{frame} + \\ &+ \overline{DTQ}_{listen} E_{listen} + \overline{DTQ}_{data} E_{data}.\end{aligned} \tag{9}$$

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

$$\begin{aligned}E_{ARS} = &\left(\rho_{tx} + (m-1)\rho_{stby}\right) T_{ARS} + \rho_{stby} T_{data} + \\ &+ 2\rho_\sigma T_{IFS} + \rho_{rx} T_{FBP}.\end{aligned} \tag{10}$$

E_{listen} is the energy consumed by a 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) remains in sleep mode in the m contention slots and in the collision-free data slot, and (ii) listens to the channel to receive the FBP. Then, E_{listen} can be formulated as

$$\begin{aligned}E_{listen} = &m\rho_{sleep} T_{ARS} + \rho_{sleep} T_{data} + \\ &+ 2\rho_\sigma T_{IFS} + \rho_{rx} T_{FBP}.\end{aligned} \tag{11}$$

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) remains in standby mode in the m contention slots, (ii) transmits data in the collision-free data slot, and (iii) listens to the channel to receive the FBP. Then, E_{data} can be formulated as

$$\begin{aligned}E_{data} = &m\rho_{stby} T_{ARS} + \rho_{tx} T_{data} + \\ &+ 2\rho_\sigma T_{IFS} + \rho_{rx} T_{FBP}.\end{aligned} \tag{12}$$

By substituting $\overline{T}_{sleep}$ from (5), $\overline{T}_{DQ}$ (6), and $\overline{E}_{DQ}$ (9) in (8), and after some basic algebra, the average energy consumed by a device in a data collection round can be expressed as

$$\begin{aligned}\overline{E}^{DQ}_R = &\overline{CRQ}_{ARS} E_{ARS} + \overline{DTQ}_{listen} E_{listen} + \\ &+ \overline{DTQ}_{data} E_{data} + \rho_{sleep} T_R - \\ &\rho_{sleep}\left(\overline{CRQ}_{ARS} + \overline{DTQ}_{listen} + \overline{DTQ}_{data}\right) T^{DQ}_{frame}.\end{aligned} \tag{13}$$

Since we assume that there are no transmission errors, then $\overline{CRQ}_{ARS} = \overline{d_n}$ (4), $\overline{DTQ}_{listen} = 1$, and $\overline{DTQ}_{data} = 1$. Therefore, $\overline{E}^{DQ}_R$ can be expressed as

$$\begin{aligned}\overline{E}^{DQ}_R = &\overline{d_n}\left(E_{ARS} - \rho_{sleep} T^{DQ}_{frame}\right) + E_{listen} + \\ &+ E_{data} + \rho_{sleep}\left(T_R - 2T^{DQ}_{frame}\right).\end{aligned} \tag{14}$$

In the next section, we validate the analysis and evaluate the energy performance under different network configurations.

V. Model Validation and Performance Evaluation

The energy model of DQ formulated in Section IV has been validated by means of computer simulations using MATLAB. We have averaged the results of 1000 simulation samples for each test case. The analytical results have been compared to the simulation. The deviation is lower than 1.5% in all tested cases, thus validating the correctness of the analysis.

The system parameters used to run the simulations are summarized in Table I. They have been selected according to the IEEE 802.15.4 [9] standard and from the specifications of the CC2520 [11] radio transceiver. The coordinator initiates one data collection round per hour, i.e., $T_R = 3600$s. The payload of the FBP includes 2 bits per slot to inform about the status of the contention slots (i.e., empty, success, or collision), 2 bytes for the length of CRQ, and 2 bytes for the length of DTQ. We have considered an ARS packet of 10 bytes composed of a physical layer preamble, a MAC header, and a Cyclic Redundancy Code (CRC) of 2 bytes for error control.

In the next subsections, we first evaluate the energy performance of DQ in order to determine the criteria to minimize energy consumption. Secondly, we compare the energy performance of DQ with that of CTA and FSA in dense M2M networks. Finally, we compare the energy consumption of devices using DQ, CTA, and FSA over the data packet length. Results for CTA and FSA were obtained through computer-based simulations. The operations of DQ, CTA and FSA [6] were implemented without any simplification.

A. Frame Length (Value of m)

The average energy consumed by one device in a data collection round using DQ and CTA is represented in Figure 4 as a function of the number m of contention slots, and considering n=100, 500, and 1000 devices. Results show that the energy consumption using DQ and CTA tends to a minimum value that is approximately constant when $m \geq 10$ and $m \geq 20$, respectively, regardless of the number n of contending devices. The independency of the results with n relaxes the need to know the size of the network in DQ and CTA. Contrarily, the number of contention slots in FSA has to be optimized according to the expected number of devices (using $m = n$) in order to minimize energy consumption [6].

As it can be observed in Figure 4, when the number of contention slots decreases, the energy consumed by the devices increases exponentially up to a finite value. This is due to the fact that when m is low, the probability of collision becomes higher and the number of frames required for a device to successfully transmit a data packet in CTA, or an ARS in

Table I
System Parameters

Parameter	Value	Parameter	Value
MAC header	8 bytes	Data-rate	250 kbps
Data payload	114 bytes	FBP payload	m·2 bits + 4 bytes
$T_{preamble}$	160 μs	T_{IFS}	192 μs
ρ_{tx}	100.8 mW	ρ_{stby}	525 μW
$\rho_{rx} = \rho_{\sigma}$	66.9 mW	ρ_{sleep}	60 nW

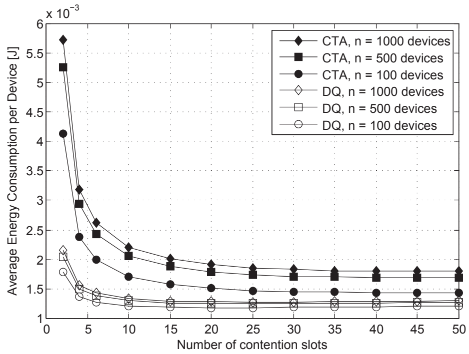

Figure 4. Energy consumed per device in a data collection round using DQ and CTA over the number m of contention slots.

DQ, also increases exponentially. When $m = 2$, the energy consumption increases up to 1.5 and 3 times with respect to the minimum value in DQ and CTA, respectively. Contrarily, as it was demonstrated in [6], the energy consumed by the devices using FSA tends to infinite for low values of m. Indeed, since the tree splitting algorithm organizes devices into sub-groups to reduce the probability of collision, the number of frames is much lower in DQ and CTA than in FSA when m is low.

When the number of contention slots increases, the average energy consumed by one device tends to the minimum value. Indeed, although a higher number of contention slots lead to longer periods in sleep, the use of a very low power sleep mode yields reduced energy consumption.

B. Number of Contending Devices

The average energy consumed by one device in a data collection round using DQ, CTA, and FSA is represented in Figure 5 (left vertical axis) as a function of the number n of devices (from 10 to 5,000 devices). In all cases, we have used the values of m that minimize energy consumption: $m = 10$ in DQ, $m = 20$ in CTA, and $m = n$ in FSA [6].

As it can be observed in Figure 5, the average energy consumption using DQ and CTA increases logarithmically with the value of n. Indeed, according to (14), this is due to the logarithmic nature of $\overline{d_n}$ (4), and also to the use of a very

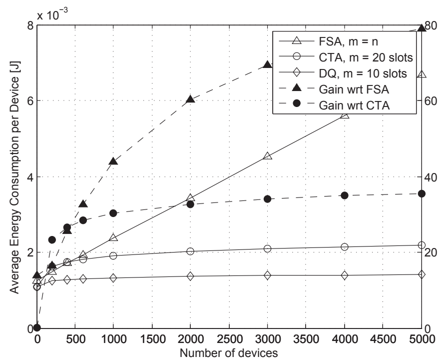

Figure 5. Energy consumed (left axis) and energy saving (right axis) per device in a data collection round.

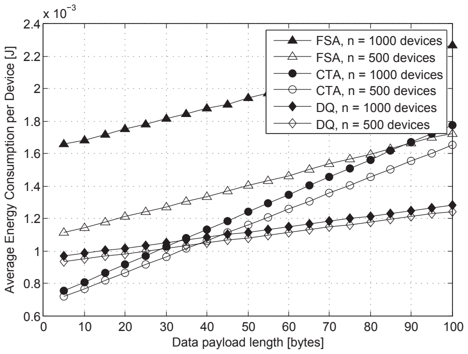

Figure 6. Energy consumed per device in a data collection round over the data packet length.

low power sleep mode in those frames where the device does not transmit data or ARS packets. On the contrary, the average energy consumption using FSA increases linearly with n and is much higher than in DQ and CTA. DQ provides energy savings (shown on the right vertical axis of Figure 5) of more than 35% with respect to CTA and more than 80% with respect to FSA. These energy savings increase with the value of n. In addition, it is worth noting that when $n < 500$ the energy consumption of a device is very similar with either FSA or CTA. Therefore, the use of contention tree-based access can improve considerably the energy efficiency in dense M2M networks, compared to that of FSA, when the number of devices is very high.

As it could be expected, the energy consumption is further reduced with DQ with respect to that of both CTA and FSA. 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.

C. Data Packet Length

The average energy consumed per device in a data collection round is represented in Figure 6 as a function of the data packet length with $n = 500$ and 1000 devices. We have considered that $m = 10$ in DQ, $m = 20$ in CTA, and $m = n$ in FSA. As it could be expected, the average energy consumption with CTA and FSA increases linearly with the data packet length. Indeed, the longer the contention slots, the higher the energy wastage in case of collision and in standby and sleep modes. In all cases, CTA attains better energy efficiency than FSA due to the fact that the number of collisions is reduced.

In its turn, the energy consumption using DQ is less sensitive to variations in the data packet length because data is transmitted in a collision-free slot and the contention slots have fixed duration. As it can be observed in Figure 6, while DQ outperforms FSA in all cases, it shows a worse performance than CTA when the data packet length is very low (below 35 bytes for the considered simulation layout). Indeed, 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, then CTA outperforms DQ.

VI. Conclusion

In this paper, we have theoretically analyzed the energy consumed by the devices of dense M2M area networks that use the contention tree-based Distributed Queuing (DQ) access protocol to periodically transmit data to a coordinator. We have compared the energy performance of DQ with respect to that of the Contention Tree Algorithm (CTA) and Frame Slotted-ALOHA (FSA). Results show that there is an optimal frame length for CTA and DQ, i.e. number of contention slots, which minimizes the energy consumption regardless of the number of devices. This is not the case of FSA, where the number of slots must be optimized as a function of the number of devices. In the case of DQ and CTA, the average energy consumed by the devices increases exponentially when the frame length decreases below its optimal value. However, it tends to a finite value in both cases. In addition, the energy consumption using DQ and CTA increases logarithmically with the number of devices, while it increases linearly with FSA. In particular, DQ provides energy savings of more than 35% with respect to CTA, and of more than 80% with respect to FSA. These savings increase further with the number of devices. Finally, we have also shown that the energy savings provided by DQ with regard to CTA and FSA increase with the length of the data packets. Therefore, the use of DQ can improve considerably the energy efficiency of dense M2M networks when the number of devices is huge. Future work aims at including transmission errors and capture effect in the analysis and at evaluating the performance of DQ experimentally.

References

[1] *Technical Specification, Machine-to-Machine communications (M2M); Functional Architecture. TS 102 690, v1.1.1*, ETSI, October 2011.

[2] L. Roberts, "Aloha packet system with and without slots and capture," in *ACM SIGCOMM Comput. Commun. Rev.*, vol. 5, no. 2, 1975.

[3] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels:Part 1: CSMA modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. 23, pp. 1400–1416, 1975.

[4] J. I. Capetanakis, "Tree Algorithms for Packet Broadcast Channels," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 505–515, Sept. 1979.

[5] Y. Xinqing and L. Xuemei, "Evaluating the energy consumption of RFID tag collision resolution protocols," in *International Conference on Mobile Ad-hoc and Sensor Networks*, February 2010, pp. 215–221.

[6] F. Vazquez-Gallego, L. Alonso, and J. Alonso-Zarate, "Energy and Delay Analysis of Contention Resolution Mechanisms for Machine-to-Machine Networks based on Low-Power WiFi," in *IEEE International Conference on Communications (ICC)*, June 2013.

[7] W. Xu and G. Campbell, "A Near Perfect Stable Random Access Protocol for a Broadcast Channel," in *IEEE International Conference on Communications (ICC)*, vol. 1, June 1992, pp. 370–374.

[8] J. Alonso-Zarate, C. Verikoukis, E. Kartsakli, A. Cateura, 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.

[9] *IEEE Std 802.15.4-2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std., September 2006.

[10] A. Janssen and M. Jong, "Analysis of Contention Tree Algorithms," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2163–2172, September 2000.

[11] CC2520 datasheet. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc22520.pdf