

Article

Delay and Energy Consumption Analysis of Frame Slotted ALOHA variants for Massive Data Collection in Internet-of-Things Scenarios

Francisco Vázquez-Gallego ^{1,†,*} , Pere Tuset-Peiró ² , Luis Alonso ³  and Jesus Alonso-Zarate ¹ 

¹ M2M Communications Department, Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), 08060 Castelldefels, Barcelona, Spain; fvazquez@cttc.es, jesus.alonso@cttc.es

² Wireless Networks (WiNe) Research Laboratory, Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), 08060 Castelldefels, Barcelona, Spain; peretuset@uoc.edu

³ Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels (EETAC), Universitat Politècnica de Catalunya (UPC), 08060 Castelldefels, Barcelona, Spain; luisg@tsc.upc.edu

* Correspondence: fvazquez@cttc.es

† Corresponding address: Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA). Av. Carl Friedrich Gauss 7, 08860 Castelldefels, Barcelona, Spain.

Received: 21 November 2019; Accepted: 23 December 2019; Published: 1 January 2020

Abstract: This paper models and evaluates three FSA-based (Frame Slotted ALOHA) MAC (Medium Access Control) protocols, namely, FSA-ACK (FSA with ACKnowledgements), FSA-FBP (FSA with FeedBack Packets) and DFSA (Dynamic FSA). The protocols are modeled using an AMC (Absorbing Markov Chain), which allows to derive analytic expressions for the average packet delay, as well as the energy consumption of both the network coordinator and the end-devices. The results, based on computer simulations, show that the analytic model is accurate and outline the benefits of DFSA. In terms of delay, DFSA provides a reduction of 17% (FSA-FBP) and 32% (FSA-ACK), whereas in terms of energy consumption DFSA provides savings of 23% (FSA-FBP) and 28% (FSA-ACK) for the coordinator and savings of 50% (FSA-FBP) and 24% (FSA-ACK) for end-devices. Finally, the paper provides insights on how to configure each FSA variant depending on the network parameters, i.e., depending on the number of end-devices, to minimize delay and energy expenditure. This is specially interesting for massive data collection in IoT (Internet-of-Things) scenarios, which typically rely on FSA-based protocols and where the operation has to be optimized to support a large number of devices with stringent energy consumption requirements.

Keywords: Internet-of-Things; massive communication; data collection; Medium Access Control; Frame Slotted ALOHA; delay; energy consumption

1. Introduction

IoT (Internet-of-Things) [1] is a paradigm where constrained computing devices with embedded sensing and wireless communication capabilities are used to collect data from physical processes in a distributed fashion. The data from multiple end-devices is then aggregated by an edge network coordinator and forwarded to a cloud-based back-end to be further stored, processed and analyzed [2]. The results of the aggregated data analysis provide valuable insight regarding the physical processes and allow to optimize its operation [3]. Hence, given its potential value, the IoT paradigm is currently being applied to homes [4], cities [5], health care [6] and industries [7], among others. However, as the adoption of the IoT paradigm increases, there are several aspects of the data chain that need to be properly engineered to ensure that the solution is both scalable, reliable and of low power.

One of the weakest links in the IoT data chain is the communication between end-devices and the network coordinator. As end-devices are physically distributed, power-constrained and cost sensitive, communications are typically based on long-range and low-power wireless technologies (i.e., IEEE 802.15.4, SigFox or LoRa) [8]. Despite being standardized [9,10], these technologies operate in unlicensed bands (i.e., 868 MHz in Europe, 915 MHz in USA and 2.4 GHz world-wide) and rely on simple radio modulation techniques. Hence, despite the fact that end-devices transmit seldomly (i.e., one packet per minute) and packets are small (i.e., tens of bytes per packet), and that most use cases tolerate certain latency and packet loss, as IoT deployments become denser (i.e., thousands of devices) the scalability and reliability of communications between the end-device and the network coordinator will become a limiting factor for its success.

Given the idiosyncrasies of low-power wireless communications (i.e., propagation and interference), Medium Access Control (MAC) protocols [11] are deployed on top of the physical layer to coordinate access to the wireless channel among end-devices. MAC protocols can be broadly classified into deterministic and random access depending on how they organize transmissions among end-devices that form a network [12]. In deterministic MAC protocols, the network coordinator assigns resources (i.e., time, frequency or code) to each end-device, thus ensuring that they do not collide when transmitting their data packets. Contrarily, random access MAC protocols are based on contention, meaning that end-devices can transmit without central coordination. This approach simplifies implementation and relaxes requirements, but collisions between packets transmitted by end-devices simultaneously may occur. Ultimately, collisions lead to re-transmissions, which may cause congestion and become a hindrance for network scalability, reliability and low-power.

Over the years, there have been various works that have studied the limits of low-power wireless communication technologies operating in unlicensed bands and have made proposals to improve their performance and robustness, as well as to reduce the energy consumption of both the end-devices and the network coordinator. Regarding LoRa, the authors in [13] study the limits of the technology, whereas in [14] the authors propose a lightweight scheduling mechanism to improve its performance. Regarding IEEE 802.15.4, in [15] the authors study the effects of multi-path propagation and how to mitigate it using channel hopping, whereas in [16] the authors propose the use of path diversity to further enhance the reliability of these networks. Finally, in [17] the authors study the robustness of three new modulations (i.e., SUN-FSK, SUN-OQPSK and SUN-OFDM) introduced in IEEE 802.15.4g for industrial settings.

Considering the challenges of massive data collection in IoT scenarios described earlier, in this article, we focus on modeling and evaluating the performance of three Frame Slotted ALOHA (FSA) variants named FSA-ACK (FSA with ACKnowledgements), FSA-FBP (FSA with FeedBack Packets) and DFSA (Dynamic FSA). We have selected these FSA variants because they are typically used in such scenarios due to their simplicity, which eases implementation in constrained environments. In particular, we model each protocol variant using an AMC (Absorbing Markov Chain) and derive closed expressions to determine the average transmission delay and energy consumption of both the network gateway and end-devices. We then use computer simulations to check the validity of the models and compare the average delay and energy consumption of each protocol variant depending on the network size. Results show that the mathematical model is accurate and outline the benefits of using DFSA with respect to FSA-FBP and FSA-ACK in this scenarios.

Last but not least, it is important to mention that the work presented in this article is an extension of the work presented in [18,19]. The mathematical model to derive the delay and energy consumption of the described protocols, as well as the methodology used to validate the obtained results, are similar. However, in [18] we compare FSA-FBP with CTA (Contention Tree Algorithm) and in [19] we compare the FSA-FBP with RSFA (Reservation FSA). In contrast, in this paper we compare the three FSA variants with respect to each other and provide metrics to optimize their performance. Hence, we believe that this work is novel and can be useful for researchers and practitioners aiming to select and optimize

the operation of a FSA-based protocol for massive data collection in IoT scenarios, where scalability, reliability and low-power requirements are key to success.

The remaining parts of this article are organized as follows. Section 2 provides an overview of FSA-ACK, FSA-FBP and DFSA frame structure and protocol operation. Section 3 presents the analytic model of the FSA-ACK, FSA-FBP and DFSA protocol variants using an AMC, which allows to calculate the average number of contention frames, as well as to obtain the average delay and energy consumption for all network device types. Section 4 presents the protocol evaluation based on computer simulations that we have conducted to validate the analytic models and compare its average delay and energy consumption. Finally, Section 5 concludes the article.

2. Protocol Overview

This section presents the frame structure and the protocol operation of the three FSA variants that are modeled and evaluated in later sections. In particular, the frame structure of FSA-ACK and DFSA is displayed in Figure 1a, whereas the FSA-FBP frame structure is displayed in Figure 1b.

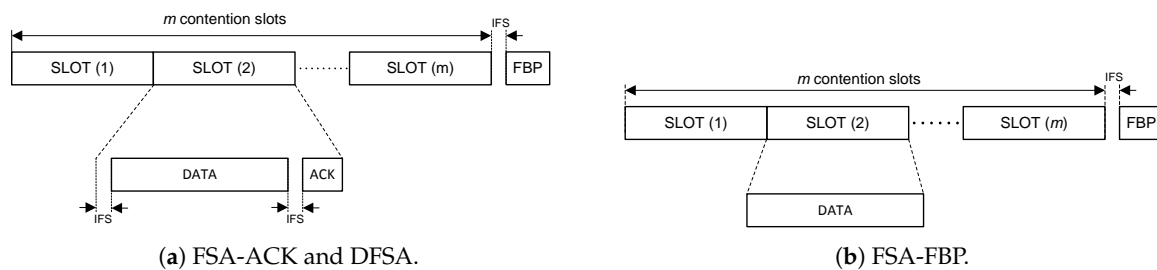


Figure 1. Frame structure of FSA (Frame Slotted ALOHA)-ACK (ACKnowledgements), FSA-FBP (FSA with FeedBack Packets) and DFSA (Dynamic FSA).

Regarding operation, for all FSA variants, we consider that communication between the end-devices and the network coordinator is structured in DCRs (Data Collection Round) composed of frames with m contention slots. In each DCR, end-devices have exactly one data packet and randomly select one of the slots available in each frame to transmit the data packet to the network coordinator. For FSA-ACK and FSA-FBP each frame F_i is sub-divided into a fixed number of contention slots, whereas for DFSA the number of contention slots m_i of each frame F_i is variable. Notice that for DFSA the number of contention slots per frame is adjusted to be proportional to the estimated number \hat{n}_i of end-devices that contend in every frame as $m_i = \lceil \rho \cdot \hat{n}_i \rceil$, where ρ is a positive real number.

With respect to slot duration, for FSA-ACK and DFSA it matches the duration of a data packet and an acknowledgement (ACK) packet, and includes the guard times (called IFS, Inter-Frame Space) between transmit and receive modes to compensate for signal propagation, processing delays and radio turn-around time between modes. Conversely, for FSA-FBP the duration of each slot is adjusted to allow a node to transmit a single data packet, as the coordinator transmits a FBP (FeedBack Packet) at the end of the frame to indicate the state of every slot and allow end-devices to maintain synchronization.

It is important to mention that, regardless of the protocol variant, packets are transmitted without CCA (Clear Channel Assessment). This means that end-devices do not check the channel status (i.e., busy or idle) before transmission. As a consequence, a given slot in any frame can be in the following states: empty (when no packet is transmitted), success (when a single packet is transmitted) or collision (when two or more end-devices have transmitted their respective packet). Based on the outcome of each slot, end-devices that have succeeded to transmit their data packet in a given frame will enter into sleep mode and stop contending in subsequent frames. Contrarily, end-devices that have collided in frame F_i will transmit again in frame F_{i+1} .

Finally, in Figure 2 we provide an example of the FSA-FBP protocol operation. As it can be observed, all end-devices try to transmit a data packet in Frame 1. The outcome is that end-device 1 and end-device 2 succeed, whereas end-device 3 and end-device 4 collide. Hence, end-device 3 and

end-device 4 will transmit a packet again in Frame 2. As both end-devices collide again, both will transmit again in Frame 3, where they succeed and the process ends.

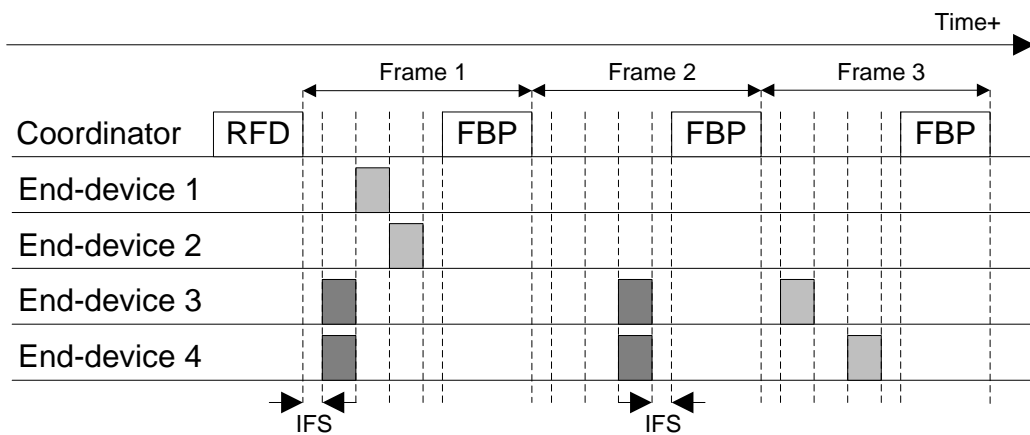


Figure 2. FSA-FBP operation example with four end-devices and three contention slots per frame.

3. Delay and Energy Consumption Analysis

Based on the frame structure and protocol operation described earlier, in this section we use an AMC (Absorbing Markov Chain) model and formulate its transition matrix for the different FSA protocol variants, which allows to derive two important performance parameters:

- The average number of frames in which the process keeps in the same transient state.
- The average number of frames until the process is absorbed.

Ultimately, we use these two parameters to formulate the analytical expressions needed to compute the average delay and energy consumption of the FSA-ACK, FSA-FBP and DFSA protocols.

3.1. Absorbing Markov Chain Model

In the AMC model depicted in Figure 3, each state represents the number of end-devices that have successfully transmitted their data packet to the network coordinator since the contention process started. The AMC is characterized by a transition matrix $P = [p_{ij}]$, where each element p_{ij} represents the probability that the number of successful end-devices changes from i to j after one frame, with $i, j = \{0, 1, \dots, n\}$. The process will finish when all the end-devices have succeeded transmitting their data packet, i.e., $i = n$. Once this absorbing state is reached, the probability that the process changes to any other state is zero.

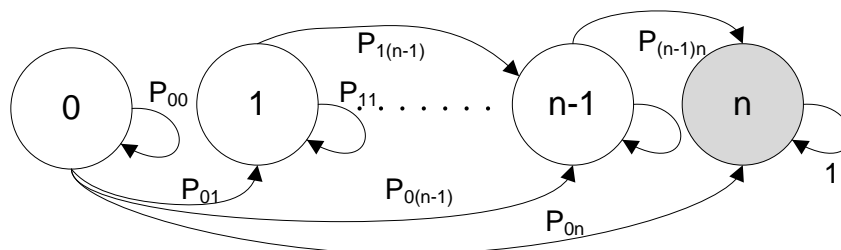


Figure 3. Generalized state transition diagram of the absorbing Markov chain that models the evolution of the numbers of end-devices that succeed in transmitting data using FSA-ACK, FSA-FBP and DFSA (Dynamic FSA)

The transition probabilities between states depend on the probability $P_s(m_c, m_1, n_c)$ that there is a number m_1 of successful slots in a given frame with m_c contention slots having n_c contending end-devices, as derived in [20]. According to [21], the probability $P_s(m_c, m_1, n_c)$ can be expressed as

$$P_s(m_c, m_1, n_c) = \frac{\binom{m_c}{m_1} \prod_{k=0}^{m_1-1} (n_c - k) G(m_c - m_1, n_c - m_1)}{m_c^{n_c}}, \tag{1}$$

where

$$G(T, t) = T^t + \sum_{k=1}^t \left\{ (-1)^k \cdot \prod_{j=0}^{k-1} \{(t-j)(T-j)\} (T-k)^{t-k} \frac{1}{k!} \right\}, \tag{2}$$

with $T = m_c - m_1$ and $t = n_c - m_1$.

Hence, the transition probabilities p_{ij} for FSA-ACK and FSA-FBP can be defined as

$$p_{ij} = \begin{cases} P_s(m, j - i, n - i), & \text{if } (i < j \leq i + m) \\ 0, & \text{if } (j < i) \text{ or } (j > i + m) \\ 1 - \sum_{k=i+1}^{i+m} p_{ik}, & \text{if } (i = j) \text{ and } (i < n) \\ 1, & \text{if } (i = j) \text{ and } (i = n) \end{cases} \tag{3}$$

The rationale for the FSA-ACK and FSA-FBP transition probabilities is that in the first condition, $j - i$ end-devices succeed and the number of contending end-devices is $n - i$, where i the number of end-devices that have succeeded in previous frames. In the second condition, the number of successful end-devices decreases, or is greater than the number of slots, which are both difficult to achieve. In the third condition, no new end-devices succeed and the process is not yet absorbed. Finally, in the fourth condition the process is absorbed.

In contrast, the transition probabilities p_{ij} for DFSA can be defined as

$$p_{ij} = \begin{cases} P_s(m_i, j - i, n - i), & \text{if } (i < j \leq i + m_i) \\ 0, & \text{if } (j < i) \text{ or } (j > i + m_i) \\ 1 - \sum_{k=i+1}^{i+m_i} p_{ik}, & \text{if } (i = j) \text{ and } (i < n) \\ 1, & \text{if } (i = j) \text{ and } (i = n) \end{cases} \tag{4}$$

The rationale for the DFSA transition probabilities is similar to FSA-ACK and FSA-FBP, but the number m_i of contention slots per frame changes dynamically as the number i of successful end-devices increases, i.e., $m_i = \lceil \rho(n - i) \rceil$.

Based on the AMC model, we define Q as a sub-matrix of the transition matrix P that contains the probabilities in the transient states (i.e., $i < n$ and $j < n$). Then, the fundamental matrix N of the process can be defined as

$$N = (I - Q)^{-1}, \tag{5}$$

where each element n_{ij} of N is the average number of frames that the process remains in the transient state X_j . This means that no new end-devices succeed for n_{ij} consecutive frames, given that it started in the transient state X_i .

Hence, we can define the absorption time t_0 as the average number of frames until the process is absorbed, which can be expressed as

$$t_0 = n_{01} + n_{02} + \dots + n_{0n-1} = \sum_{j=0}^{n-1} n_{0j}, \tag{6}$$

where n is the total number of end-devices in the network.

3.2. Delay Analysis

Based on the AMC model presented in the previous section, we can now calculate the *average delay*, expressed in seconds, required to resolve the contention process.

For FSA-ACK and FSA-FBP the *average delay* can be formulated as

$$\bar{T}_C^{\text{FSA-ACK}} = t_0 \cdot T_{\text{frame}}^{\text{FSA-ACK}}, \tag{7}$$

$$\bar{T}_C^{\text{FSA-FBP}} = t_0 \cdot T_{\text{frame}}^{\text{FSA-FBP}}, \tag{8}$$

where t_0 is the average number of frames required to resolve the contention process, and $T_{\text{frame}}^{\text{FSA-ACK}}$ and $T_{\text{frame}}^{\text{FSA-FBP}}$ are the duration of a frame in FSA-ACK and FSA-FBP.

Also, $T_{\text{frame}}^{\text{FSA-ACK}}$ and $T_{\text{frame}}^{\text{FSA-FBP}}$ can be expressed as

$$T_{\text{frame}}^{\text{FSA-ACK}} = m(T_{\text{data}} + T_{\text{ack}} + 2T_{\text{IFS}}) + T_{\text{IFS}} + T_{\text{FBP}}^{\text{FSA-ACK}}, \tag{9}$$

$$T_{\text{frame}}^{\text{FSA-FBP}} = mT_{\text{data}} + 2T_{\text{IFS}} + T_{\text{FBP}}^{\text{FSA-FBP}}, \tag{10}$$

where m is the number of contention slots per frame, T_{data} , T_{ack} , T_{IFS} , $T_{\text{FBP}}^{\text{FSA-ACK}}$ and $T_{\text{FBP}}^{\text{FSA-FBP}}$ are the time of transmission of a data packet, an ACK, the duration of an IFS, and the time of transmission of a FBP.

In contrast, for DFSA the *average delay* can be formulated as

$$\bar{T}_C^{\text{DFSA}} = \sum_{j=0}^{n-1} n_{0j} \cdot T_{\text{frame}}^{\text{DFSA}}(j), \tag{11}$$

where j is the number of end-devices that have succeeded in a given point in time, n_{0j} is the average number of frames for which the contention process remains with j end-devices with success, and $T_{\text{frame}}^{\text{DFSA}}(j)$ is the duration of a frame.

Also, $T_{\text{frame}}^{\text{DFSA}}(j)$ can be expressed as

$$T_{\text{frame}}^{\text{DFSA}}(j) = m_j(T_{\text{data}} + T_{\text{ack}} + 2T_{\text{IFS}}) + T_{\text{IFS}} + T_{\text{FBP}}^{\text{DFSA}}, \tag{12}$$

where $m_j = \lceil \rho(n - j) \rceil$ is the number of contention slots in a frame where $n - j$ end-devices have not succeeded yet, and $T_{\text{FBP}}^{\text{DFSA}}$ is the time of transmission of a FBP.

3.3. Energy Consumption Analysis

Based on the delay models for each FSA protocol variant presented in the previous subsection, we can now derive the energy consumption of the network coordinator and the end-devices by determining their operation states in each slot of each frame in the contention process.

In general, these devices have four different operation states (transmitting a packet, receiving a packet, idle-listening or sleeping) that is associated with a power consumption (i.e., ρ_{tx} , ρ_{rx} , ρ_{σ} and ρ_{sleep} , respectively). Notice that in this model we assume that the energy required to switch between sleep and active modes (i.e., transmitting, receiving and idle-listening) is negligible. Hence, in the following subsections we derive the energy consumption of the network coordinator and the end-devices.

3.3.1. Network Coordinator

Using FSA-FBP, the network coordinator executes the following operations repeatedly in every frame: listens to the channel for the m slots of each frame in order to receive incoming data packets, keeps in idle state for the duration of two IFS, and transmits a FBP packet at the end of the frame. Therefore, we denote by $\bar{E}_{coord}^{FSA-FBP}$ the average energy consumed by the coordinator in a contention process, which can be formulated as

$$\bar{E}_{coord}^{FSA-FBP} = t_0 \left(m\rho_{rx}T_{data} + 2\rho_{\sigma}T_{IFS} + \rho_{tx}T_{FBP}^{FSA-FBP} \right). \quad (13)$$

For FSA-ACK and DFSA, the network coordinator executes the following operations repeatedly in every frame: listens to the channel for the slots of each frame in order to receive incoming data packets, responds with an ACK to every data packet decoded or sleeps for the time of transmission of an ACK, keeps in idle state for the duration of an IFS, and transmits a FBP packet at the end of the frame. Therefore, we denote by $\bar{E}_{coord}^{FSA-ACK}$ and \bar{E}_{coord}^{DFSA} the average energy consumption of the coordinator in a contention process using FSA-ACK and DFSA, respectively, which can be formulated as

$$\begin{aligned} \bar{E}_{coord}^{FSA-ACK} = t_0 \left(m \left(\rho_{rx}T_{data} + 2\rho_{sleep}T_{IFS} + \rho_{sleep}T_{ack} \right) + \rho_{\sigma}T_{IFS} + \rho_{tx}T_{FBP}^{FSA-ACK} \right) + \\ n \left(\left(\rho_{tx} - \rho_{sleep} \right) T_{ack} + 2 \left(\rho_{\sigma} - \rho_{sleep} \right) T_{IFS} \right), \end{aligned} \quad (14)$$

$$\begin{aligned} \bar{E}_{coord}^{DFSA} = \sum_{j=0}^{n-1} n_{0j} \left(m_j \left(\rho_{rx}T_{data} + 2\rho_{sleep}T_{IFS} + \rho_{sleep}T_{ack} \right) + \rho_{\sigma}T_{IFS} + \rho_{tx}T_{FBP}^{DFSA} \right) + \\ n \left(\left(\rho_{tx} - \rho_{sleep} \right) T_{ack} + 2 \left(\rho_{\sigma} - \rho_{sleep} \right) T_{IFS} \right). \end{aligned} \quad (15)$$

3.3.2. End-Devices

Regarding end-devices, for each FSA protocol variant we define $\bar{E}_{devices}^{FSA-FBP}$, $\bar{E}_{devices}^{FSA-ACK}$ and $\bar{E}_{devices}^{DFSA}$, respectively, which can be expressed as

$$\bar{E}_{devices}^{FSA-FBP} = \sum_{j=0}^{n-1} n_{0j} \left((n-j)E_{data}^{FSA-FBP} + jE_{sleep}^{FSA-FBP} \right), \quad (16)$$

$$\bar{E}_{devices}^{FSA-ACK} = \sum_{j=0}^{n-1} n_{0j} \left((n-j)E_{data}^{FSA-ACK} + jE_{sleep}^{FSA-ACK} \right), \quad (17)$$

$$\bar{E}_{devices}^{DFSA} = \sum_{j=0}^{n-1} n_{0j} \left((n-j)E_{data}^{DFSA} + jE_{sleep}^{DFSA}(j) \right), \quad (18)$$

where j represents the number of end-devices that have succeeded in a given point in time, n_{0j} represents the average number of frames for which the process remains with j end-devices with success, and $n - j$ is the number of end-devices that have not succeeded yet.

For all FSA protocol variants, we can also define $E_{data}^{FSA-FBP}$, $E_{data}^{FSA-ACK}$ and E_{data}^{DFSA} as the energy consumed by an end-device in a frame where it transmits its data packet, and $E_{sleep}^{FSA-FBP}$, $E_{sleep}^{FSA-ACK}$ and $E_{sleep}^{DFSA}(j)$ as the energy consumed by an end-device in a frame where it sleeps.

In FSA-FBP, an end-device that has not yet succeeded to transmit its data packet performs the following tasks in every frame: randomly selects a slot in the m slots of the frame to transmit its data packet, sleeps for the other $m - 1$ slots of the frame, keeps in idle state for two IFS, and receives a FBP. Therefore, the value of $E_{data}^{FSA-FBP}$ can be obtained as

$$E_{data}^{FSA-FBP} = \rho_{tx}T_{data} + (m - 1)\rho_{sleep}T_{data} + 2\rho_{\sigma}T_{IFS} + \rho_{rx}T_{FBP}^{FSA-FBP}. \quad (19)$$

In contrast, for FSA-ACK and DFSA, an end-device that has not yet succeeded to transmit its data packet performs the following tasks in every frame: randomly selects one slot the m slots of the frame to transmit its data packet, receives an ACK packet, sleeps for the remaining $m - 1$ slots of the frame, keeps in idle state for two IFS, and receives the FBP. Therefore, the $E_{data}^{FSA-ACK}$ and $E_{data}^{DFSA}(j)$ values can be obtained as

$$E_{data}^{FSA-ACK} = \rho_{tx}T_{data} + \rho_{rx}T_{ack} + 2\rho_{\sigma}T_{IFS} + (m - 1)\rho_{sleep}(T_{data} + T_{ack} + 2T_{IFS}) + \rho_{\sigma}T_{IFS} + \rho_{rx}T_{FBP}^{FSA-ACK}, \quad (20)$$

$$E_{data}^{DFSA}(j) = \rho_{tx}T_{data} + \rho_{rx}T_{ack} + 2\rho_{\sigma}T_{IFS} + (m_j - 1)\rho_{sleep}(T_{data} + T_{ack} + 2T_{IFS}) + \rho_{\sigma}T_{IFS} + \rho_{rx}T_{FBP}^{DFSA}. \quad (21)$$

Finally, regardless of the FSA protocol variant, end-devices that succeed in transmitting their data in a given frame do not contend in subsequent frames, remaining in sleep mode for the rest of the frame. Hence, during that period, an end-device will consume an energy that can be expressed as

$$E_{sleep}^{FSA-FBP} = \rho_{sleep} \cdot T_{frame}^{FSA-FBP}, \quad (22)$$

$$E_{sleep}^{FSA-ACK} = \rho_{sleep} \cdot T_{frame}^{FSA-ACK}, \quad (23)$$

$$E_{sleep}^{DFSA}(j) = \rho_{sleep} \cdot T_{frame}^{DFSA}(j). \quad (24)$$

4. Delay and Energy Consumption Evaluation

In this section, we verify the models presented in Section 3 using computer-based simulations. In particular, we evaluate how the delay and the energy consumption are influenced by the number n of end-devices in the network. For FSA-ACK and FSA-FBP we take into account the number m of slots per frame, whereas for DFSA we consider the value of ρ . Based on the results, we discuss how to minimize the delay and energy consumption depending on the protocol variant.

4.1. Simulation Parameters

To validate the models and evaluate the delay and energy consumption of the FSA variants we use the parameters summarized in Table 1. These values have been selected according to the IEEE 802.15.4 standard [22] and the Texas Instruments CC2520 radio transceiver [23] specifications. In particular, all the packets are composed of a physical layer preamble, a MAC header, a payload (114 bytes) and a

2-byte CRC (Cyclic Redundancy Code). However, in FSA-FBP the length of the FBP payload is set to attach 2 bits per slot, allowing to inform the status of each slot. In contrast, for DFSA and FSA-ACK the coordinator responds to each data packet decoded successfully in each slot with an ACK.

4.2. FSA-ACK and FSA-FBP Delay and Energy Evaluation

For FSA-ACK and FSA-FBP, the delay and energy consumption of the coordinator are depicted in Figures 4a and 5a, respectively, whereas the energy consumption of the end-devices is depicted in Figure 5b. In both cases the results are reported according to m (number of slots per frame) and for $n \in \{25, 50, 100\}$ nodes.

Table 1. System Parameters for FSA-ACK, FSA-FBP and DFSA.

Parameter	Value	Parameter	Value
MAC header	8 bytes	Data-rate	250 kbps
Data payload	114 bytes	T_{data}	4.1 ms
$T_{preamble}$	160 μ s	T_{IFS}	192 μ s
ρ_{tx}	100.8 mW	T_{ACK}	512 μ s
$\rho_{rx} = \rho_{\sigma}$	66.9 mW	ρ_{sleep}	60 nW

For the network coordinator, the results show that the delay and energy consumption are minimal when the number of slots per frame is $m = n/2$. In contrast, when m is below the optimum (i.e., $m < n/2$), the delay and the energy consumption tend to a very large number if $m < n/4$. This is expected, as the shorter the frame length the higher the probability of collision, and the more frames are required to complete contention process. Otherwise, when the number of slots per frame is above the optimum (i.e., $m > n/2$), the delay and the energy consumption of the coordinator increase following a linear fashion and being proportional to m . Here, a higher number of slots per frame leads the coordinator to listen to the channel during longer time periods, yielding greater delay and energy consumption.

Thus, the number of slots in FSA-ACK and FSA-FBP has to be adjusted according to the number of end-devices in order to minimize the delay and the energy consumption of the coordinator. However, since the delay and the energy consumption of the coordinator increase with a small slope after the minimum, it is better to over-dimension the number of slots when the number of end-devices is unknown in order to ensure that the condition $m > n/4$ holds.

For end-devices, results show that the energy consumption is minimal when $m \geq n$. As expected, a higher frame length leads to a lower average number of frames where an end-device has to contend and, thus, a minimum in the energy consumption. In addition, although a higher number of slots leads to longer frames and longer periods in sleep, the use of a very low power sleep mode yields reduced energy consumption when the value of m increases. Of course, there is a trade-off between the coordinator and the end-device delay and energy consumption. When the number of slots per frame increases above $n/2$, the end-devices consume less energy in a DCR, at the cost of increasing the delay and the energy consumption of the coordinator.

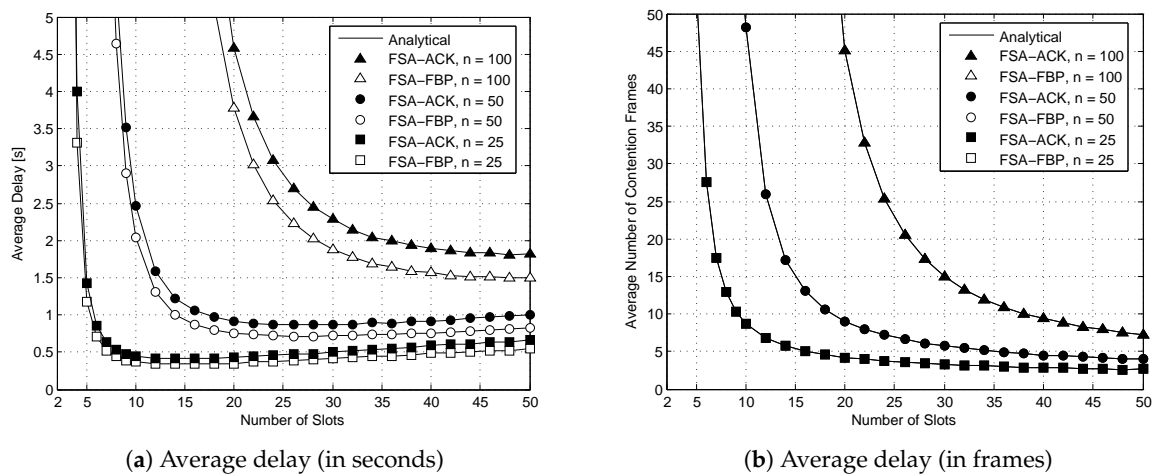


Figure 4. Coordinator and end-device average delay using FSA-ACK and FSA-FBP.

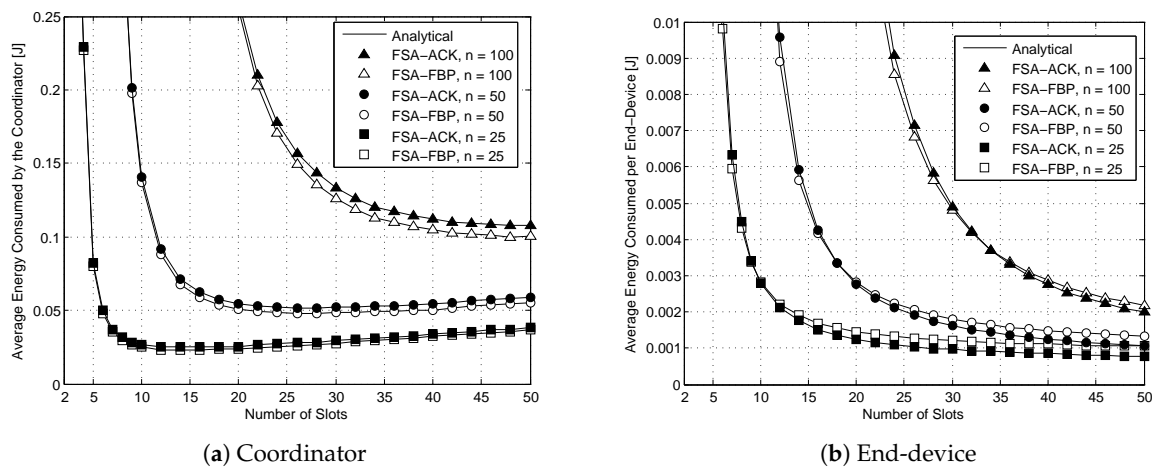


Figure 5. Coordinator and end-device average energy consumption using FSA-ACK and FSA-FBP.

4.3. DFSA Delay and Energy Consumption Evaluation

The average delay and energy consumption of the coordinator using DFSA are depicted in Figures 6 and 7a, respectively, whereas for end-devices the average energy consumption is depicted in Figure 7b. In all cases, the average delay and energy consumption are presented as a function of ρ and considering $n \in \{25, 50, 100\}$. Notice that, in DFSA, the number of slots per frame (m_i) is proportional to the number of end-devices that contend in every frame (\hat{n}_i). That is, $m_i = \lceil \rho \cdot \hat{n}_i \rceil$, where ρ is a positive real number.

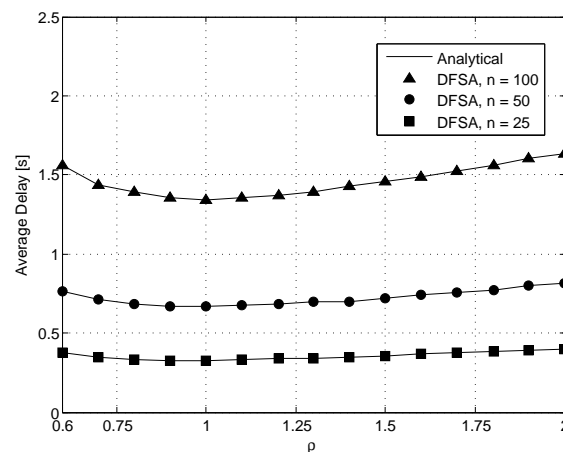


Figure 6. DFSA average delay (seconds) depending on ρ .

As it can be observed, the results show that the delays and the energy consumption of the coordinator have a minimum at $\rho = 1$. In contrast, when $\rho < 1$, the number of slots per frame becomes lower and the probability of collision in a given slot increases, leading to an increase in the number of frames required to complete a DCR. Otherwise, if $\rho > 1$, the number of slots per frame increases and the probability of collision decreases, leading to a decrease in the the number of frames required to complete a DCR.

In both cases, the network delay and the energy consumption of the coordinator increase. In the first scenario ($\rho < 1$), the coordinator has to transmit more FBP packets, whereas in the second scenario ($\rho > 1$), the number of slots per frame is higher, thus requiring the coordinator to waste more energy in idle-listening. Taking that into consideration, the number of slots per frame in DFSA has to be adjusted to be equal to the number of end-devices that contend in every frame in order to minimize the delay and the energy consumption of the coordinator.

Regarding end-devices, results show that energy consumption is minimized as ρ increases. This can be explained by the fact that an increase in the frame length leads to a lower collision probability among nodes and, hence, a lower energy consumption. It is also important to notice that end-device energy consumption is almost insensitive to the number of end-devices. Here, the average number of re-transmissions of an end-device in DFSA for a given ρ does not change with the number of end-devices. This holds because the probability that an end-device succeeds to transmit in a given frame is constant regardless of the number of end-devices, as the protocol is able to re-configure the frame length after each frame of a DCR.

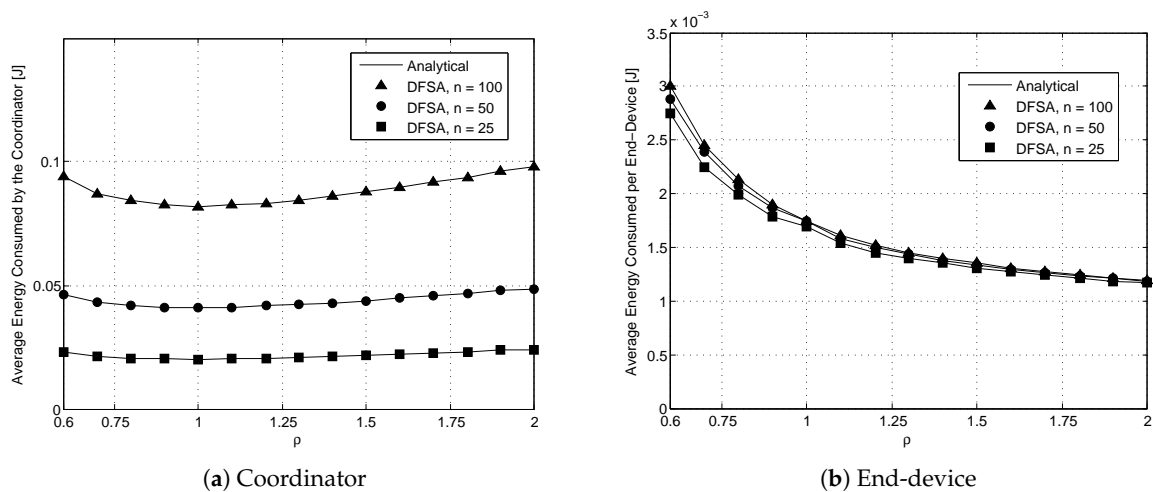


Figure 7. DFSA coordinator and end-device average energy consumption depending on ρ .

4.4. FSA-ACK, FSA-FBP and DFSA Delay and Energy Consumption Comparison Under Optimal Configuration

In this subsection, we compare the delay and energy consumption of both the network coordinator and end-devices for all FSA protocol variants as a function of n (number of end-devices), where $n \in \{25, \dots, 1000\}$. For each protocol, we have selected m (number of slots per frame) to minimize: (a) the delay and the energy consumption of the coordinator (i.e., $m = n/2$ in FSA-ACK and FSA-FBP, and $\rho = 1$ in DFSA), and, (b) the energy consumption per end-device (i.e., $m = n$ in FSA-ACK and FSA-FBP, and $\rho = 1.25$ in DFSA). This allows to compare all the protocols in their ideal conditions, allowing to draw conclusions regarding their suitability for the target use case.

The average delay is depicted in Figure 8, whereas the network coordinator and end-device average energy are depicted in Figure 9a,b, respectively. Results show that the delay and energy consumption of the coordinator increase almost linearly with the number of devices (n) regardless of the protocol variant. Also, the delay and energy consumption of the coordinator are minimal when $m = n/2$ in FSA-ACK and FSA-FBP, and when $\rho = 1$ in DFSA. In addition, there are five aspects to be further discussed, as described in the following bullets:

- FSA-FBP outperforms FSA-ACK in terms of delay. This is because the frame duration is longer in FSA-ACK than it is in FSA-FBP. That is, despite the duration of the FBP packets is shorter in FSA-ACK, the duration of the slots in FSA-ACK is longer because they are adjusted to fit a data and an acknowledgement packet.
- FSA-FBP outperforms FSA-ACK in terms of coordinator energy consumption. This is caused by the fact that in FSA-ACK the coordinator consumes more energy in transmission mode due to the transmission of acknowledgment packets. The differences in delay and energy consumption of the coordinator between the two protocols increase as the number of end-devices increases. In that regard, it is worth noting that DFSA outperforms FSA-ACK and FSA-FBP in terms of delay and energy consumption of the coordinator as it is capable of adapting the frame length to the number contenders in every frame.
- The end-device energy consumption in FSA-FBP increases linearly with the number of end-devices. Since the payload of the FBP packet in FSA-FBP is proportional to the number of slots, the energy consumed by an end-device in reception mode also increases with the number of slots. Contrarily, the end-device energy consumption using DFSA and FSA-ACK is almost insensitive to the number of nodes. Since end-devices receive a very short FBP packet in every frame regardless of the number of slots and listen for the duration of a short acknowledgement packet in every transmission attempt, the energy consumption is almost constant.

- If the protocols are configured with their optimal number of slots that minimize the delay and the energy consumption of the coordinator, DFSA provides 17% delay reduction with respect to FSA-FBP and 32% with respect to FSA-ACK; and FSA-FBP provides delay reductions of a 18% with respect to FSA-ACK. Regarding coordinator energy consumption, DFSA provides 23% energy savings with respect to FSA-FBP and 28% with respect to FSA-ACK, whereas FSA-FBP provides 6% energy savings with respect to FSA-ACK.
- The end-device energy consumption is minimal when $m = n$ in FSA-ACK and FSA-FBP, and $\rho = 1.25$ in DFSA. With these values an for $n = 1000$ devices, FSA-ACK provides energy savings of more than 28% with respect to DFSA and 54% with respect to FSA-FBP. However, if the frame length of FSA-ACK is not well configured, e.g., $m = n/2$, then DFSA outperforms FSA-ACK in terms of end-device energy consumption. In particular, DFSA provides energy savings of more than a 50% with respect to FSA-FBP and 24% with respect to FSA-ACK.

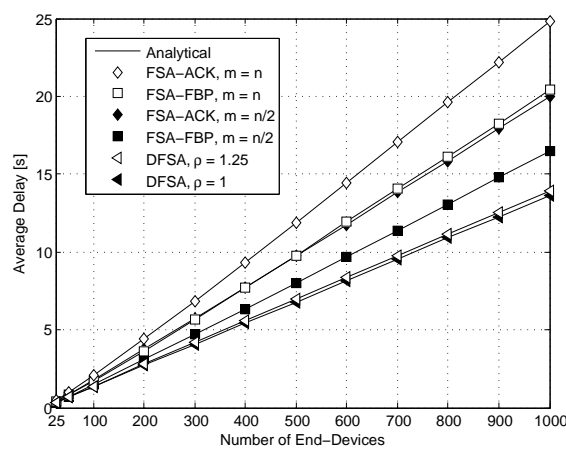


Figure 8. Network coordinator average delay (seconds) depending on the number of end-devices when using the optimal FSA-ACK, FSA-FBP and DFSA configuration.

4.5. Results Summary

Based on the results presented in the previous sub-sections, we now provide two insights regarding the optimal configuration of each FSA protocol variant.

First, the optimum frame length (m) in FSA-ACK and FSA-FBP to minimize the delay and coordinator energy consumption depends on the number of end-devices (n) and is reached when $m = n/2$. Results also show that the end-device energy consumption is minimal when $m \geq n$. Hence, the number of slots in FSA-ACK and FSA-FBP has to be pre-adjusted according to the number of end-devices expected in the network. Also, there is a trade-off between the delay and network coordinator energy consumption, and the end-device energy consumption. If the number of slots per frame m increases above the number of end-devices n , then each end-device consumes less energy, but the delay and network coordinator energy consumption increases slightly.

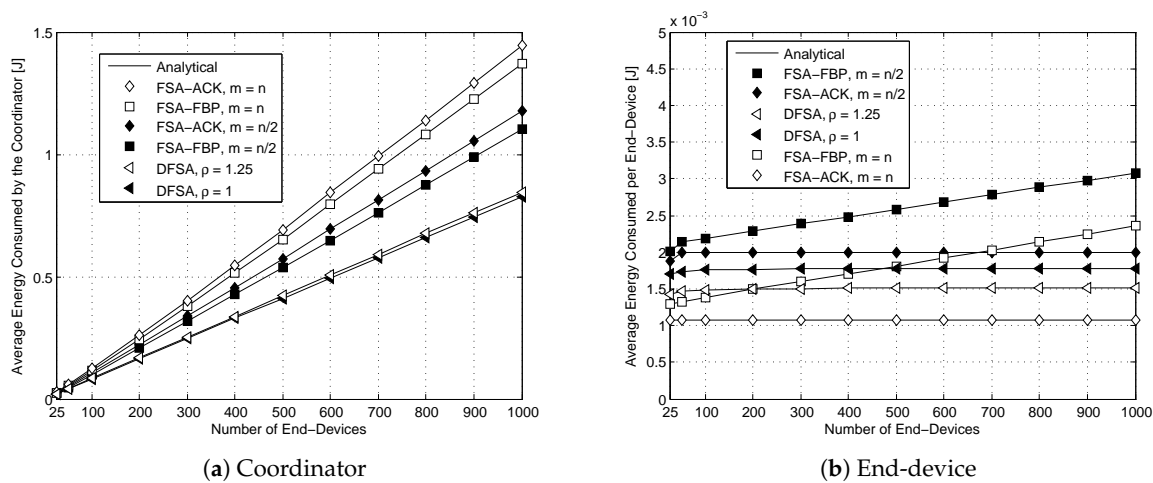


Figure 9. Network coordinator and end-device average energy consumption (Joules) depending on the number of end-devices when using the optimal FSA-ACK, FSA-FBP and DFSA configuration.

Second, DFSA results show that, to minimize the delay and the energy consumption of the coordinator, the number of slots per frame (m) has to be adjusted to be equal to the number of end-devices (n) that contend in every frame, i.e., $\rho = 1$. In addition, the end-device energy consumption tends to a minimum value as ρ increases. However, there is a trade-off between the delay and coordinator energy consumption, and the end-device energy consumption. When the value of ρ increases, end-devices consume less energy, but the delay and coordinator energy consumption increase slightly.

Hence, considering that in most real-world deployments, the end-devices are battery-powered but the network coordinator is mains-powered, the optimal configuration for FSA-ACK and FSA-FBP is $m > n/2$, and for DFSA is $m = n$.

5. Conclusions

This article has modeled and evaluated three MAC protocols based on FSA (namely, FSA-ACK, FSA-FBP and DFSA) that are targeted at massive data collection in IoT scenarios. The analytic models are based on an AMC, which allows to derive the average delay and energy consumption for both the network coordinator and the end-devices. We then have used computer simulations to validate the model and compare each protocol variant in terms of these parameters, and depending on the number of end-devices present in the network.

Results show the model accuracy and outline the benefits of DFSA with respect to FSA-FBP and FSA-ACK. In terms of delay, DFSA provides a reduction of 17% and 32% compared to FSA-FBP and FSA-ACK, respectively. In terms of energy consumption, for the network coordinator, DFSA provides savings of 23% and 28% compared to FSA-FBP and FSA-ACK, respectively. In contrast, for end-devices, DFSA provides savings of 50% and 24% compared to FSA-FBP and FSA-ACK, respectively.

Finally, based on the analytic model and simulation results, the article has provided insights to configure each FSA variant optimally. Specifically, the FSA-ACK and FSA-FBP optimum value is $m = n/2$, that is, when the number of slots per frame is twice the number of nodes in the network. In contrast, for DFSA the number of slots per frame has to be adjusted to $\rho = 1$, that is, be equal to the number of end-devices that contend in every frame. These optimizations are key for massive data collection in IoT scenarios, as these networks rely on low-power wireless communication technologies and have to support a large number of devices with stringent energy consumption requirements.

Author Contributions: F.V.-G. has created the analytic model and performed the computer simulations to generate the results. P.T.-P. has prepared and reviewed the manuscript. L.A. and J.A.-Z. have validated the results and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially supported by Generalitat de Catalunya (SGR-60-2017 and SGR-891-2017) and the Spanish Ministry of Science, Innovation and Universities (SPOT5G <TEC2017-87456-P> and SPOTS <RTI2018-095438-A-I00>).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2347–2376. doi:10.1109/COMST.2015.2444095.
- Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. doi:10.1109/JIOT.2016.2579198.
- Lin, J.; Yu, W.; Zhang, N.; Yang, X.; Zhang, H.; Zhao, W. A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet Things J.* **2017**, *4*, 1125–1142. doi:10.1109/JIOT.2017.2683200.
- Kelly, S.D.T.; Suryadevara, N.K.; Mukhopadhyay, S.C. Towards the Implementation of IoT for Environmental Condition Monitoring in Homes. *IEEE Sensors J.* **2013**, *13*, 3846–3853. doi:10.1109/JSEN.2013.2263379.
- Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet Things J.* **2014**, *1*, 22–32. doi:10.1109/JIOT.2014.2306328.
- Islam, S.M.R.; Kwak, D.; Kabir, M.H.; Hossain, M.; Kwak, K. The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access* **2015**, *3*, 678–708. doi:10.1109/ACCESS.2015.2437951.
- Xu, L.D.; He, W.; Li, S. Internet of Things in Industries: A Survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243. doi:10.1109/TII.2014.2300753.
- Raza, U.; Kulkarni, P.; Sooriyabandara, M. Low Power Wide Area Networks: An Overview. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 855–873. doi:10.1109/COMST.2017.2652320.
- Vilajosana, X.; Tuset-Peiro, P.; Vazquez-Gallego, F.; Alonso-Zarate, J.; Alonso, L. Standardized Low-Power Wireless Communication Technologies for Distributed Sensing Applications. *Sensors* **2014**, *14*, 2663–2682. doi:10.3390/s140202663.
- Palattella, M.R.; Accettura, N.; Vilajosana, X.; Watteyne, T.; Grieco, L.A.; Boggia, G.; Dohler, M. Standardized Protocol Stack for the Internet of (Important) Things. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1389–1406. doi:10.1109/SURV.2012.111412.00158.
- Bachir, A.; Dohler, M.; Watteyne, T.; Leung, K.K. MAC Essentials for Wireless Sensor Networks. *IEEE Commun. Surv. Tutor.* **2010**, *12*, 222–248. doi:10.1109/SURV.2010.020510.00058.
- Rajandekar, A.; Sikdar, B. A Survey of MAC Layer Issues and Protocols for Machine-to-Machine Communications. *IEEE Internet Things J.* **2015**, *2*, 175–186. doi:10.1109/JIOT.2015.2394438.
- Adelantado, F.; Vilajosana, X.; Tuset-Peiro, P.; Martinez, B.; Melia-Segui, J.; Watteyne, T. Understanding the Limits of LoRaWAN. *IEEE Commun. Mag.* **2017**, *55*, 34–40.
- Reynders, B.; Wang, Q.; Tuset-Peiro, P.; Vilajosana, X.; Pollin, S. Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling. *IEEE Internet Things J.* **2018**, *5*, 1830–1842. doi:10.1109/JIOT.2018.2815150.
- Watteyne, T.; Lanzisera, S.; Mehta, A.; Pister, K.S.J. Mitigating Multipath Fading through Channel Hopping in Wireless Sensor Networks. In Proceedings of the 2010 IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; pp. 1–5. doi:10.1109/ICC.2010.5502548.
- De Armas, J.; Tuset, P.; Chang, T.; Adelantado, F.; Watteyne, T.; Vilajosana, X. Determinism through Path Diversity: Why Packet Replication Makes Sense. In Proceedings of the 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), Ostrava, Czech Republic, 7–9 September 2016; pp. 150–154. doi:10.1109/INCoS.2016.105.
- Tuset-Peiró, P.; Vázquez-Gallego, F.; Muñoz, J.; Watteyne, T.; Alonso-Zarate, J.; Vilajosana, X. Experimental Interference Robustness Evaluation of IEEE 802.15.4-2015 OQPSK-DSSS and SUN-OFDM Physical Layers for Industrial Communications. *Electronics* **2019**, *8*, 1045. doi:10.3390/electronics8091045.

18. Vázquez-Gallego, F.; Alonso-Zarate, J.; Alonso, L. Energy and delay analysis of contention resolution mechanisms for machine-to-machine networks based on low-power WiFi. In Proceedings of the 2013 IEEE International Conference on Communications (ICC), Budapest, Hungary, 9–13 June 2013; pp. 2235–2240. doi:10.1109/ICC.2013.6654860.
19. Vázquez-Gallego, F.; Alonso, L.; Alonso-Zarate, J. Modeling and Analysis of Reservation Frame Slotted-ALOHA in Wireless Machine-to-Machine Area Networks for Data Collection. *Sensors* **2015**, *15*, 3911–3931. doi:10.3390/s150203911.
20. Schoute, F. Dynamic Frame Length ALOHA. *IEEE Trans. Commun.* **1983**, *31*, 565–568. doi:10.1109/TCOM.1983.1095854.
21. Vogt, H. Efficient Object Identification with Passive RFID Tags. In *Proceedings of the First International Conference on Pervasive Computing*; Springer-Verlag: London, UK, 2002; Pervasive '02; pp. 98–113.
22. 802.15.4-2015 - IEEE Standard for Low-Rate Wireless Networks; Revision of IEEE Std 802.15.4-2011; IEEE: Piscataway, NJ, USA, 2016; pp. 1–709. doi:10.1109/IEEESTD.2016.7460875.
23. Texas Instruments. *CC2520 Datasheet: 2.4 GHz IEEE 802.15.4/Zigbee RF Transceiver (SWRS068)*; Texas Instruments: Dallas, TX, USA, 2007.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).