Chapter 3

RFID Anti-collision Protocols

This chapter presents the most common anti-collisions algorithms for passive RFID systems, either by using the physical layer or the MAC layer. More emphasis will be upon the anti-collision algorithms which are compatible with the EPCglobal C1 G2 standard, as this is the focus of this thesis.

This chapter is organized as follows: section 1 gives an overview about the physical layer anti-collision algorithms. In section 2, different MAC-layer anti-collision algorithms are presented, thus clarifying the main differences between them. Afterwards, a brief description for the EPCglobal C1 G2 reading process will be presented in section 3. Section 4 will give an introduction about the collision recovery and the slots duration in modern RFID readers and the effect of these parameters in MAC-layer anti-collision algorithm, which will be focused on, in the remaining part of this thesis.

3.1 PHY-Layer Anti-collision Protocols

Different physical layer anti-collision protocols have been developed to separate colliding signals on the physical layer. Figure 3.1 shows the most common physical layer anti-collision protocols, which are: FDMA, SDMA, CDMA and TDMA. These algorithms will be briefly discussed in the following paragraphs:

Frequency Division Multiple Access (FDMA): In this protocol, the frequency band is divided into different sub-frequency bands and tags are distributed among them [42]. However, this technique adds complexity to the

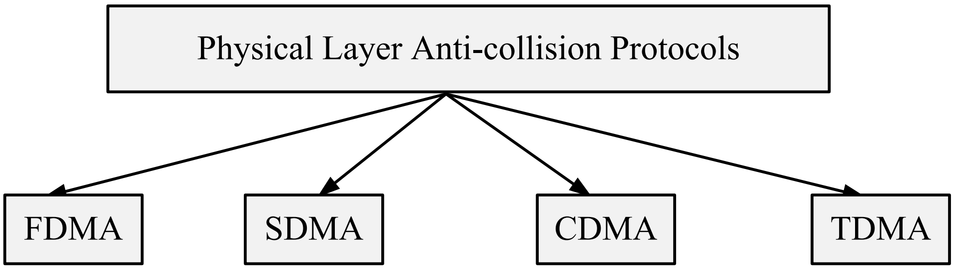


Figure 3.1: Common existing PHY-Layer anti-collision protocols

system. Readers should be able to decode different frequencies at the same time. Moreover, tags should be able to select its desirable sub-channel, which is incompatible with EPCglobal C1 G2. Only active tags can do such functionality.

Space Division Multiple Access (SDMA): This technique makes use of spreading tags over the reading area. It provides a high increase in the reading efficiency [43]. The main drawback is the cost of implementing the RFID reader with multiple antennas. Moreover, in dense applications, the distances between the tags is very small to be distributed on the reading area.

Code Division Multiple Access (CDMA): This protocol uses spread-spectrum modulation techniques to transmit the data over the entire spectrum [44]. CDMA is the ideal procedure for many applications, e.g. navigation systems. However in case of RFID systems, it is not compatible with EPCglobal C1 G2, as the cost of the tags will be dramatically increased. Thereby, it is not a sufficient protocol within the scope of this thesis.

Time Division Multiple Access (TDMA): In this protocol, a single frequency band is divided to time slots and is assigned to tags. One of the most important features of this technique is that each tag must be synchronized to the time slots and send its information at the beginning of the selected slot [45]. This technique can be directly applied to passive RFID systems. In such systems, the simplicity of tags transfers the complexity to the readers, where the reader has to control the time synchronization. However, in active RFID systems, synchronization can be either centralized or distributed to the tags.

In UHF passive dense RFID system, both TDMA and SDMA are the most commonly used PHY-layer anti-collision protocols. In these systems, there is only a single reader versus a large number of tags. Thus, there is no problem to increase the complexity of the reader. However, the tags should be as cheap and simple as possible.

3.2 MAC-Layer Anti-collision Protocols

Unfortunately, the physical layer anti-collision proposals are not cost effective for the market challenges of the passive RFID technologies. Therefore, collision solutions are commonly implemented at the MAC-layer. This section will discuss the most common MAC-layer anti-collision protocols.

Figure 3.2 presents the main classification of the most common MAC-layer anti-collision protocols. According to figure 3.2, MAC-layer anti-collision protocols is classified into two main categories: deterministic protocols and probabilistic protocols. Deterministic protocols are used in systems with known number of tags to be identified in the reading area. These types of protocols are mainly based on tree algorithms for the identification processes. Probabilistic protocols are used in systems with an unknown number of tags. Probabilistic protocols are mainly based on ALOHA algorithm.

The following sections will discuss in details the most commonly used anti-collision algorithms either deterministic or probabilistic based algorithm.

3.2.1 Deterministic Anti-collision Protocols

These protocols are commonly named tree-based anti-collision protocols. Using these algorithms, the reader aims to identify a set of tags in the coverage area in subsequent time slots [46]. Each time slot contains a query command, transmitted from the reader, and the response of tags in the reading area. If there is more than one tag reply in one slot, a collision occurs and the reader tries to split the tags into two subgroups. The reader repeats the splitting procedure until it receives a single tag reply. Tree based anti-collision protocols can be classified into two groups:

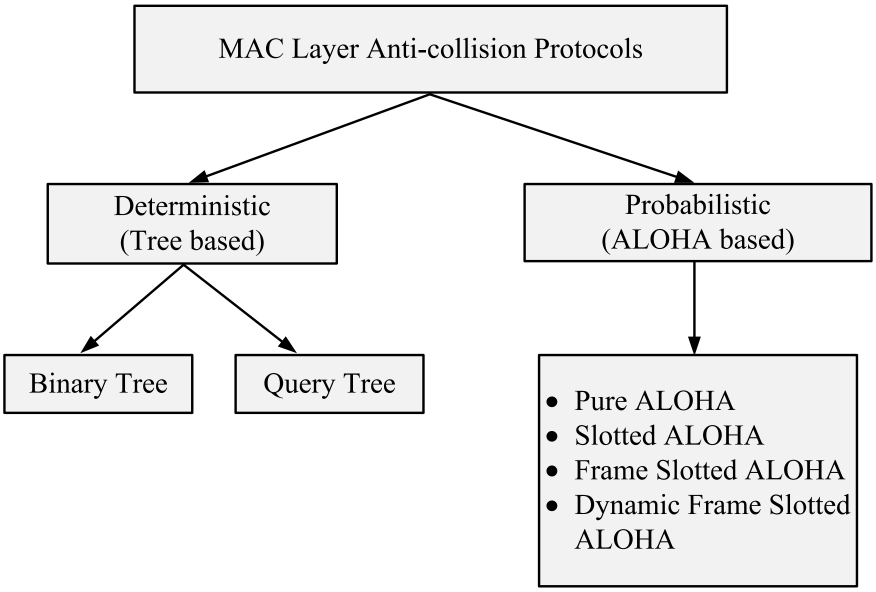


Figure 3.2: Common existing MAC-Layer anti-collision protocols

Binary tree

The binary tree algorithm [47] is commonly used in tree-based anti-collision protocols. Using this algorithm, if a collision occurs in a time slot, each collided tag selects randomly '0' or '1'. Thus, the colliding tags will be separated into two subgroups. Tags, which have selected '0' always transmit their IDs to the reader first. If a collision re-occurs, collided tags are splitted again by selecting '0' or '1'. Tags, which have selected '1' have to wait until all other tags which have selected '0' are successfully identified by the reader. This procedure continues recursively until the subset is reduced to one tag, that is identified without collisions.

Figure 3.3 shows an example of the binary tree algorithm resolving the collision of four tags in a reading area. Thus, we have a collided time slot at the beginning. At time slot 2, each collided tag has to choose '0' or '1' randomly. In our example, tags 1 and 2 have selected '0'. However, tags 3 and 4 have selected '1'. According to binary tree algorithm, tags 3 and 4 have to wait until tags 1 and 2 are successfully identified. Therefore, time slot 2 is a collided slot due to the collision between tags 1 and 2. Due to collision, both tags 1 and 2 have to choose either '0' or '1'. In this example, both tags 1 and 2 have selected '1'. This resulted and empty slot in time slot 3 and collided

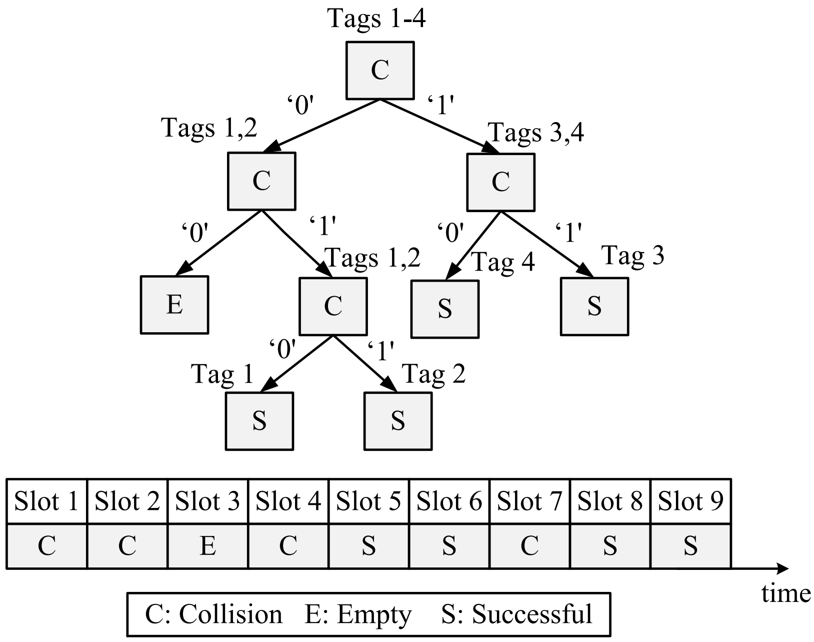


Figure 3.3: Binary tree anti-collision algorithm example

slot in time slot 4. Afterwards, tag 1 has selected '0' and tag 2 has selected '1'. This random selection made them separated and results in two successive successful slots in time slots 5 and 6. At this moment, tags 3 and 4 started their identification process. The reader repeats the previous process until identifying all tags in the reading area.

Query tree

Another category from tree algorithm is the query tree algorithm [48]. It is also commonly used in tree-based anti-collision algorithm. Using this algorithm, the broadcast is a query signal asking the tags for a reply. If there is a collision, it starts splitting the collided tags into two groups by sending a new query signal with a single bit 0 or 1 randomly. Tags in the reading area receive this signal and match this bit with their ID. If this bit matches their ID, they transmit their ID. If a collision happened again, the reader adds another random bit 0 or 1 to its next query signal. This process is repeated until the reader receives a successful single tag reply.

Figure 3.4 presents an example for the query tree algorithm resolving the collision of 4 tags in a reading area. At time slot 1, the reader broadcasts a query

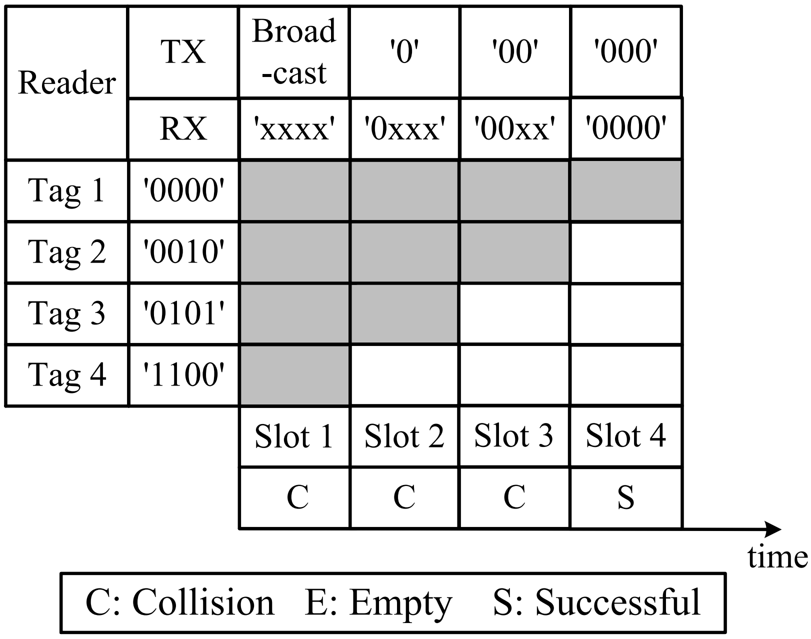


Figure 3.4: Query tree anti-collision algorithm example

signal asking them for a reply. A collision between the four tags is happening. The reader sends a ’0’ in a new query signal at time slot 2. However, there are three tags sharing this bit, so a new collision between the three tags is happening. Thereby, the reader has sent a ’00’ in its next query signal in time slot 3. At this time, there is a collision between two tags, leading the reader to send ’000’ in time slot 4. At last, the reader has received a single successful reply from tag 1. This process is repeated until the reader identify all the tags in the reading area.

3.2.2 Probabilistic Anti-collision Protocols

The main problem of using tree-based protocols is that these protocols are not efficient in dense networks (large number of tags), due to the increase in identification time [49]. Therefore, in dense network, ALOHA anti-collision protocols are more suitable. ALOHA anti-collision protocols are the most commonly used in UHF active and passive RFID. In these protocols, the readers do not know exactly the number of tags in the reading area to be identified. ALOHA anti-collision protocols are classified into the following four groups:

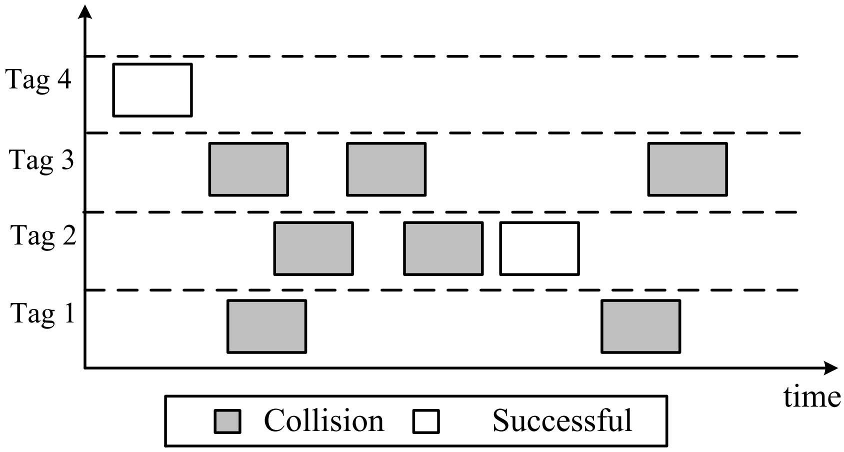


Figure 3.5: Example of pure ALOHA protocol

Basic ALOHA

The first one is the widely known basic ALOHA [50] anti-collision protocol. Basic ALOHA is the simplest anti-collision protocol for passive read-only-memory RFID tags. This protocol works as follows: The reader sends a query signal to power on tags. Then, tags send their ID randomly in time. The reader can only recognize the single tag reply case, without any ability to handle the collision. According to [50], the maximum reading efficiency is 18.4 %. Figure 3.5 shows an example of basic ALOHA for a single reader identifying four tags.

Slotted ALOHA

The second ALOHA anti-collision protocol is the slotted ALOHA (SA) protocol [51]. As shown in figure 3.6, slotted ALOHA is based on basic ALOHA. However, the time is divided into slots. In this protocol, the reader broadcasts a query signal which includes the beginning of each slot. Each tag chooses randomly if it will transmit in this slot or wait for a coming slot. The main advantage of this technique compared to basic ALOHA is: In slotted ALOHA, tag replies are completely synchronized. Therefore, collided slots are completely timing overlapping tags. However, in the basic ALOHA protocol, the partial timing overlapping exists. Thus, the maximum reading efficiency using slotted ALOHA is 36.8 % [51].

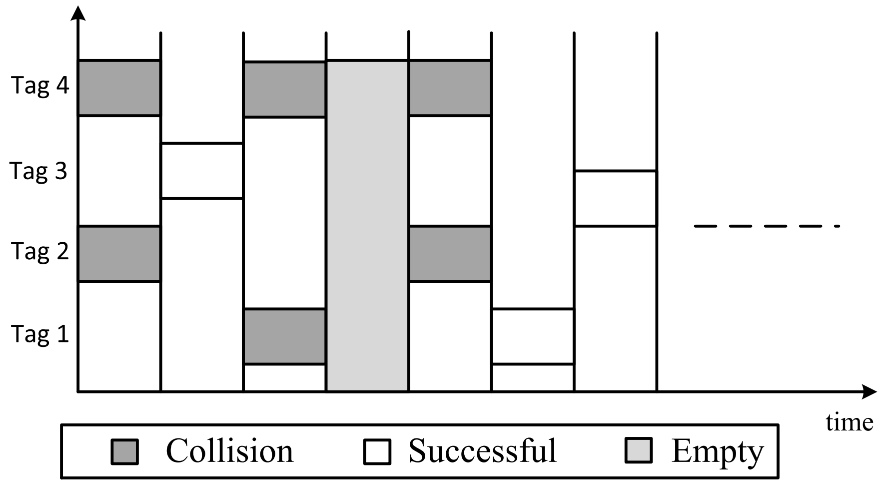


Figure 3.6: Example of slotted ALOHA

Framed Slotted ALOHA (FSA)

The third group is Framed Slotted ALOHA (FSA) [49]. The FSA anti-collision protocol uses a fixed frame length. Thus, the frame length is fixed during the complete tag identification process. At the beginning of each frame, the reader broadcasts a query signal to all tags. This signal includes the frame size. Each tag has to choose random number between 0 and $L-1$, where L is the frame length. If a collision happened, the colliding tags have to wait for the next frame.

Figure 3.7 presents an example for the identification process of four tags using FSA. In this example, the frame length is selected to four slots. According to figure 3.7, tag 4 transmits in the first slot alone. Thus, it is a successful slot. In the second slot, no tag has replied. So, it is an empty slot. In the third slot, tags 2 and 3 have replied together, which results in a collided slot. According to the FSA rules, tags 3 and 4 are not allowed to resubmit their IDs again during the same frame. Thus in slot 4, tag 1 only is allowed to reply to have another successful slot. In the next frame, the same procedure is repeated until all tags are identified.

Dynamic Frame Slotted ALOHA (DFSA)

The final type of ALOHA anti-collision protocols is Dynamic Framed Slotted ALOHA (DFSA) [52]. Using this algorithm, the number of slots per frame is

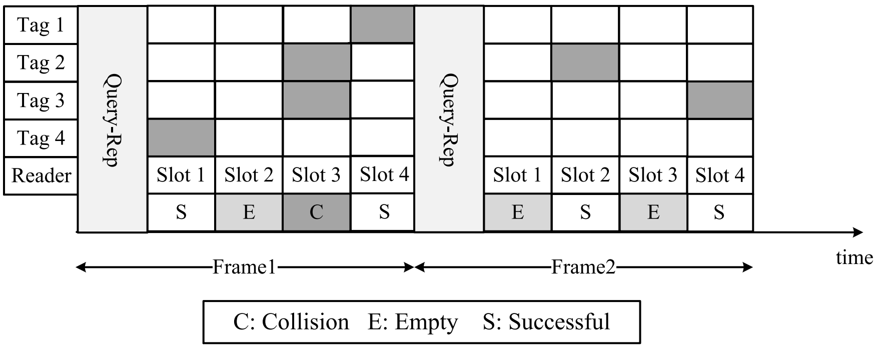


Figure 3.7: Example for Frame Slotted ALOHA

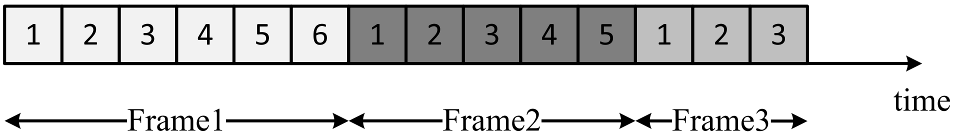


Figure 3.8: Slots of Dynamic Frame Slotted ALOHA

variable as shown in figure 3.8. According to the previously published RFID work, DFSA [53] is the most widely used anti-collision protocol for RFID systems owing to its simplicity and robustness. In DFSA, the reading process is divided into successive frames, in which each frame includes a specific number of slots. During the reading process, each active tag randomly assigns itself to one of the available slots in the frame. Therefore, each slot can take one of the following three variable states: 1) Successful Slot: One tag only chooses this slot, is fully identified, and then is deactivated by the reader within the successive frames. 2) Collided Slot: Multiple tags reply, resulting in a collision. The collided tags normally remain in their active state and retry their transmission in the next frame. 3) Empty Slot: No tag responds and the slot remains unused.

Increasing the reading speed can directly be translated into the maximization of the number of successful slots wrt. the number of idle or collided slots. Based on the Random Access Theory [54], for a given number of n tags, the expected number of empty E, successful S, and collided C slots in each frame with a length of L slots can be expressed by the following equations:

$$E = L\left(1 - \frac{1}{L}\right)^{n}, \; S = n\left(1 - \frac{1}{L}\right)^{n-1}, \; C = L - E - S \qquad (3.1)$$

The conventional definition of the expected reading efficiency η_{conv} is given by the ratio between the expected number of successful slots S in a frame and the frame length L [55]:

$$\eta_{conv} = \frac{S}{L} \tag{3.2}$$

Based on (3.1) and (3.2), this results in the conventional definition of the efficiency:

$$\eta_{conv} = \frac{n}{L}\left(1 - \frac{1}{L}\right)^{n-1} \tag{3.3}$$

Figure 3.9 shows the FSA reading efficiency η_{conv} for a constant frame length $L = 64$ and different numbers of tags. The main goal of optimizing the DFSA algorithm is finding the optimal frame length L, which maximizes the reading efficiency η_{conv}. Based on (3.3), the reading efficiency η_{conv} is maximized when:

$$L_{opt} = n \tag{3.4}$$

In practical applications, the number of tags n in the interrogation region is unknown. Furthermore, the number of tags may even vary, e.g. when the tags are mounted on moving goods, and because successfully read tags get inactive in the following frames. Therefore, such applications employ DFSA [56]. First, DFSA has to estimate the number of tags in the interrogation area, and then has to calculate the optimal frame size L for the next reading frame. Figure 3.10 presents a summary for DFSA. As shown in the chart, the reader starts with an initial frame length. Then, it broadcasts this frame length to the tags in the reading area. Afterwards, it performs a normal FSA. At the end of the frame, the reader checks if there are any successful or collided slots. If yes, the reader estimates the remaining number of tags in the reading area, and then optimizes the next frame length and starts again normal FSA. If not, the reading cycle will be terminated.

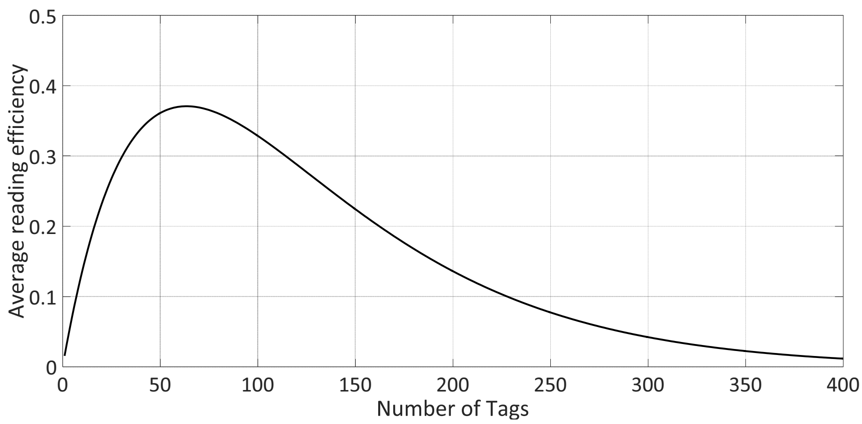


Figure 3.9: Frame Slotted ALOHA Reading efficiency, maximum reached for $L = n = 64$ tags of $\eta_{conv} = 0.36$

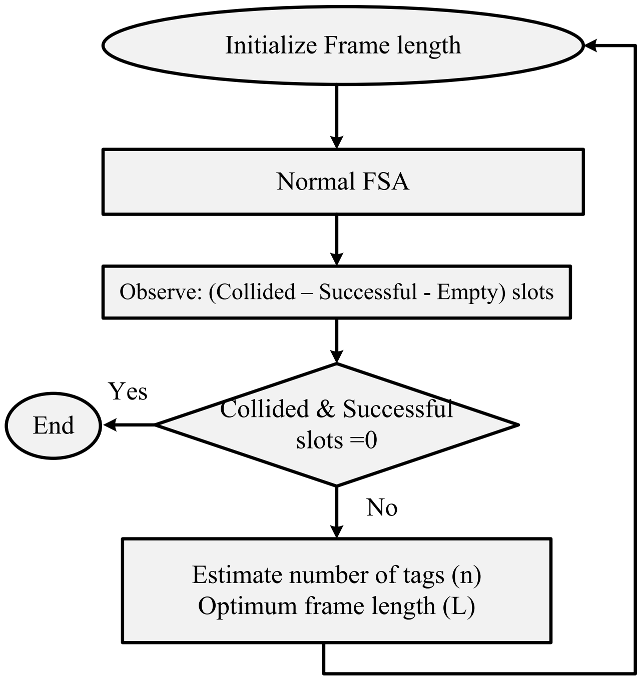


Figure 3.10: Flow chart of Dynamic Framed Slotted ALOHA (DFSA)

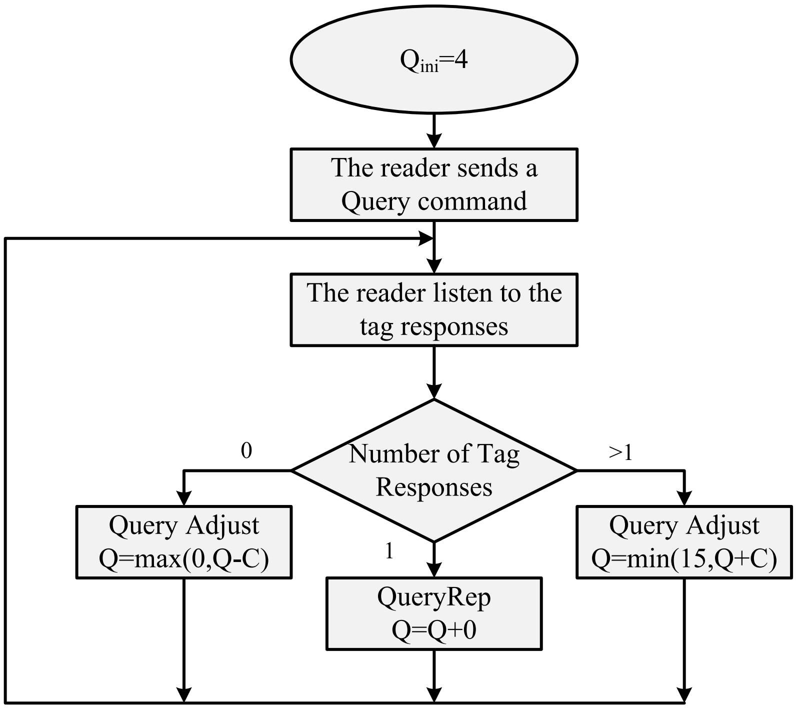


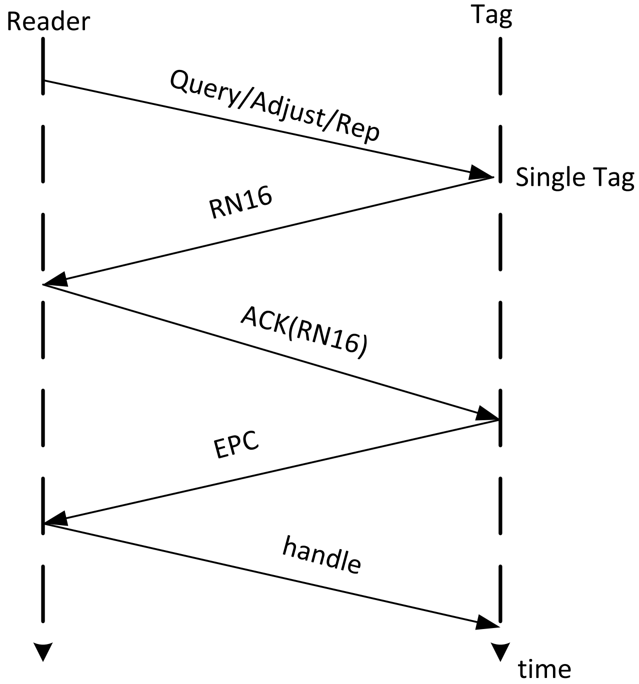
Figure 3.11: Conventional variable frame length procedure EPCglobal C1 G2 [11]

3.3 DFSA with EPCglobal C1G2

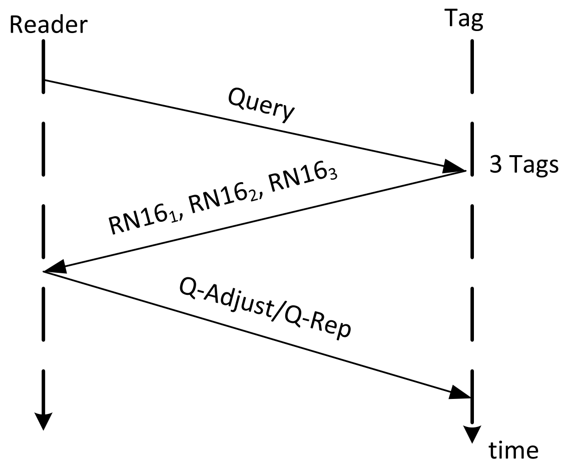
In this section, a brief introduction about DFSA with EPCglobal C1 G2 [11] will be given. The reading process consists of multiple inventory rounds. Each inventory round has a different frame length. Figure 3.11 shows an example for the frame length adaptation in EPCglobal C1 G2. According to figure 3.11, the initial frame length is $2^{Q_{ini}}$, where $Q_{ini} = 4$. Then each slot is checked. If there is no tag reply, the frame length should be decreased. If it is an collided slot, the frame length is increased. Finally, if the slot is a successful slot, the frame length will remain as it is.

Figure 3.12 shows an example for timing diagram of different inventory rounds between a single RFID reader and different tag reply situations. The reader starts with a query command. In this command, the reader broadcast the current frame length for all the tags in the reading area. Each tag has to choose a random slot between 0 and $2^Q - 1$.

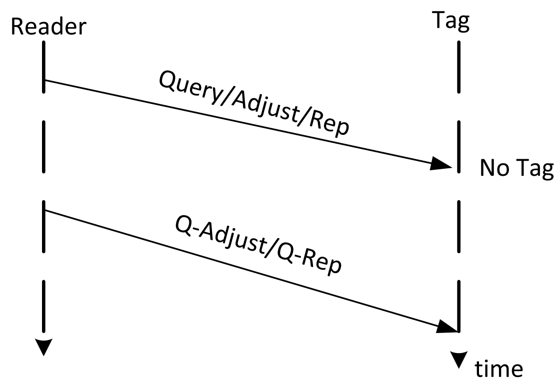
Figure 3.12a shows the case of single tag reply. In this case, the tag replies



(a) Single tag reply



(b) Multiple tags reply



(c) No tag reply

Figure 3.12: Example of an inventory between reader and different tag reply situations [11]

with its Random Number 16 (RN16), which is a 16 bits random number. When the reader receives this RN16 it will acknowledge this tag with an ACK command including this RN16. As soon as the tag receives a valid Acknowledgment (ACK) with its RN16, it will reply with its unique Electronic Product Code (EPC). Finally, the reader will send a handle command to the tag to mute it until the end of the complete reading process. In figure 3.12b, the reader starts broadcasting a query command for tags in the reading area. In the proposed example, three tags select this slot. Therefore, the reader receives simultaneously three different RN16s. In this case, the conventional RFID reader will not be able to decode any of these RN16s. The tags will wait for the next frame to be identified. Afterwards, the reader broadcasts a query-rep command to inform all the remaining tags that the next slot will start. Figure 3.12c shows the behavior of an empty slot in EPCglobal C1 G2. The reader starts with a query command, and waits for a tag reply during a certain time out period. If there is no tag reply during this time, it would terminate the the slot by sending a new query-rep command.

3.4 Cross Layer Anti-Collision Protocol

Recently, RFID receivers have been developed to be able to convert a part of collided slots into successful slots. Moreover, new RFID readers can even identify the type of the slots and terminate the empty and collided slots earlier. In this section, a brief discussion about these two parameters will be presented. Afterwards, a motivation to reconfigure the MAC layer to make use of the PHY-layer parameters is presented.

3.4.1 Collision Recovery in UHF RFID

Collision recovery in RFID systems is the capability of the reader to convert a part of the collided slots into successful slots. According to EPCglobal C1 G2, the reader can only acknowledge one single tag per slot. According to [57], the collision recovery capability of the RFID system depends on different factors: The capabilities of the receiver e.g. number of antennas, the distance between the collided tags, and the type of the channel.

Recently, some research groups (e.g. [58]) have concentrated on collision recovery using the spatial diversity of the received signal. They have proposed the following reading efficiency equation:

$$\eta = P(1) + \alpha \cdot \sum_{i=2}^{n} P(i), \tag{3.5}$$

where $\sum_{i=2}^{n} P(i)$ is the probability of collision, α is the average collision resolving probability coefficient. In this efficiency equation, the RFID reader can convert α part of the collided slots into successful slots. The authors have assumed unlimited and equal collision resolving probabilities coefficients. For example, the probability to resolve two collided tags is identical to the probability to resolve ten collided tags, which is a strong simplification. Another research groups [59–62] considered the limited capability of a RFID reader to resolve collisions. They have proposed a limited reading efficiency expressed as:

$$\eta = \sum_{i=1}^{M} P(i), \tag{3.6}$$

where $P(i) = \binom{n}{i} \left(\frac{1}{L}\right)^{i} \left(1 - \frac{1}{L}\right)^{n-i}$, and M represents the number of collided tags that the reader is capable to recover. The authors assumed that the probability to recover one tag from i collided tags equals to 100%, independently of the actual i.

According to (3.5) and (3.6), the reading efficiency strongly depends on the capability of the physical layer to resolve the collision. Thus, in this thesis, the effect of the collision recovery capability on the MAC layer optimization will be addressed in more details.

3.4.2 Slots Durations in EPCglobal C1 G2

Conventional RFID systems cannot identify the type of the slots in FSA [56]. Therefore, such systems consider that the slot duration of FSA is constant, neglecting the type of the slot. Modern RFID readers can quickly identify the type of a slot (i.e. idle, successful, or collided). Hence, the durations of the different slot types are not identical, which reduces the overall reading time. Figure 3.13 shows two frames, each one with a frame length of $L = 6$ slots. The first frame in (a) presents the conventional view of the frame with equal slots

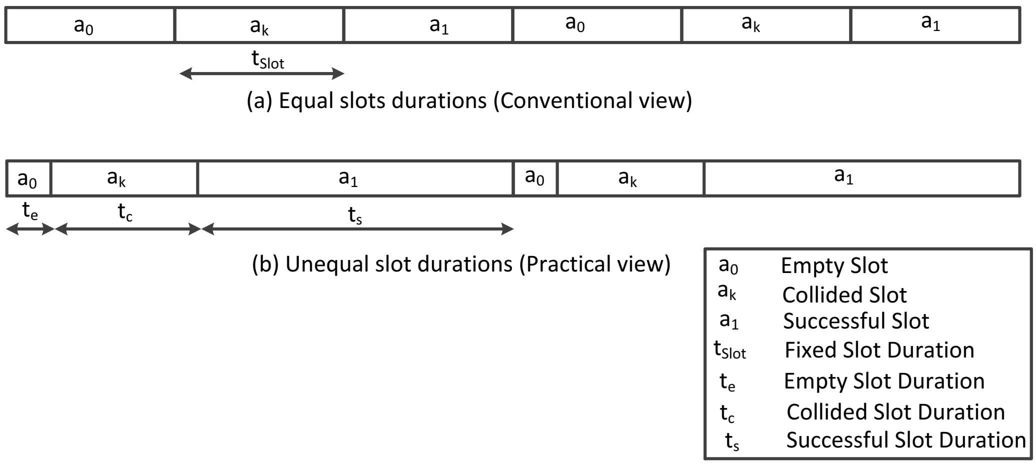


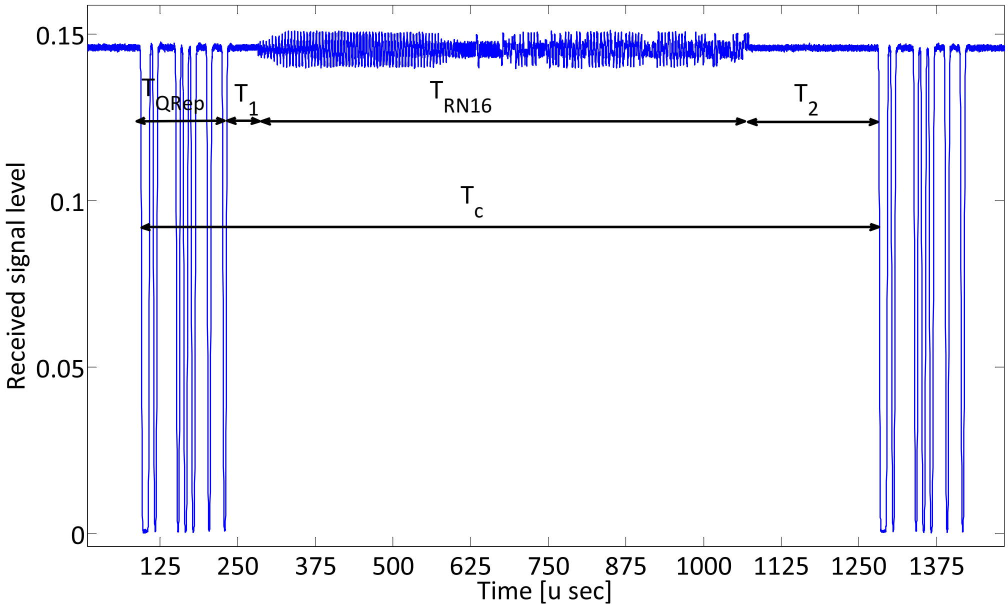
Figure 3.13: Equal and unequal views of slots in Frame Slotted ALOHA with frame length $L = 6$.

durations t_{Slot} for all slot types. The second frame in (b) presents the behavior of the real RFID slots behavior. Here, the slot duration depends on the slot type.

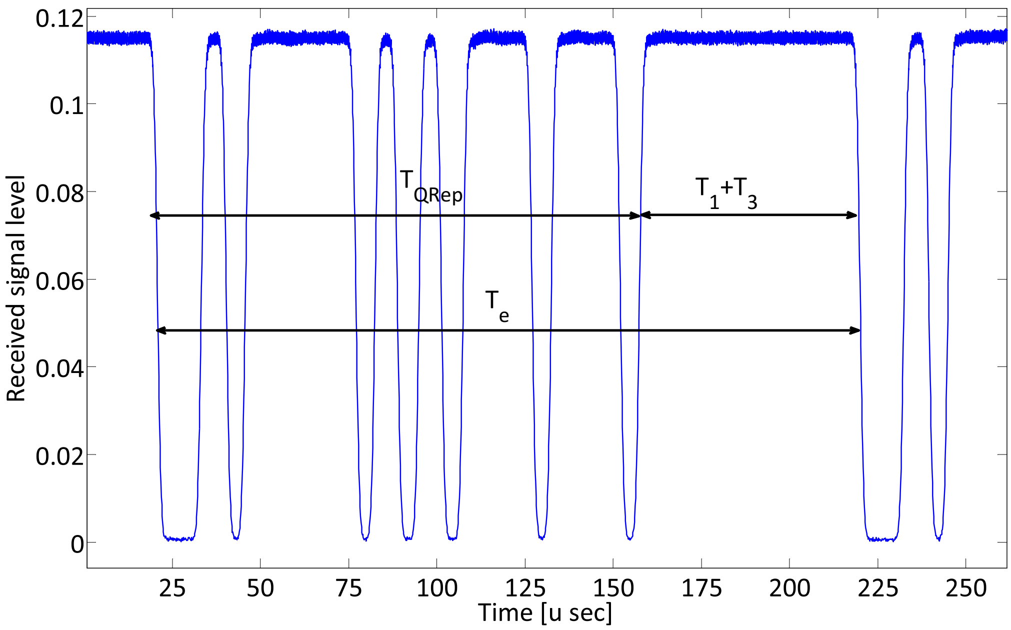
Figure 3.14 shows an example of a real measurements for slots durations using the Universal Software Radio Peripheral SDR receiver (USRP B210) [63]. In these measurements, we used a sampling frequency of $f_s = 8\,\text{MHz}$, because the total RFID bandwidth in the European system is 4 MHz, the tags data rate is 160 kbps. For the given parameters and as shown in figure 3.14, the collided slot duration is $\simeq 1200\,\mu$sec, and the empty slot duration is $\simeq 200\,\mu$sec.

According to the above discussion, the reading efficiency equation will be affected by the differences in slots duration, hence, the MAC layer optimization. In this thesis, the differences in slot duration length will be addressed in more details in the MAC layer optimization.

Summarizing, the main lack in the previous RFID research is that the MAC layer is optimized independently on the PHY layer layer. However, the PHY-layer properties affect the optimization parameters of the MAC layer, e.g. the number of tags estimation and the optimum frame length. In this thesis, I will concentrate on optimizing the DFSA anti-collision protocol. In the proposed algorithm, the physical layer parameters will be taken into consideration, which are presented on the physical collision recovery capability of the RFID reader and the differences in slot durations.



(a) Collided slot timing



(b) Empty slot timing

Figure 3.14: Slots durations measurements, $f_s = 8\,\mathrm{MHz}$, tag Backscatter Link Frequency $BLF = 160\,\mathrm{kbps}$