IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 10, 2019, accepted September 27, 2019, date of publication October 3, 2019, date of current version October 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945396

Stability of Synchronous Queued RFID Networks

JAVIER VALES-ALONSO[1], PABLO LÓPEZ-MATENCIO[1], JUAN J. ALCARAZ[1], AND FRANCISCO JAVIER GONZÁLEZ-CASTAÑO[2]

[1]Department of Information and Communications Technologies, Technical University of Cartagena, 30202 Cartagena, Spain
[2]Department of Telematics Engineering, Universidade de Vigo, 36310 Vigo, Spain

Corresponding author: Javier Vales-Alonso (javier.vales@upct.es)

This work was supported by the Project AIM, (AEI/FEDER, EU) under Grant TEC2016-76465-C2-1-R.

ABSTRACT Queued Radio Frequency Identification (RFID) networks arise naturally in many applications, where tags are grouped into batches, and each batch must be processed before the next reading job starts. In these cases, the system must be able to handle all incoming jobs, keeping the queue backlogs bounded. This property is called stability. Besides, in RFID networks, it is common that some readers cannot operate at the same time, due to mutual interferences. This fact reduces the maximum traffic that readers can process since they have to share the channel. Synchronous networks share the channel using a TDMA approach. The goal of this work is to analytically determine whether a synchronous queued RFID network attains stable operation under a given incoming traffic. Stability depends on the service rate, which is characterized in this paper using an exact numerical method based on a recursive analytical approach, overcoming the limitations of previous works, which were based on simplifications. We also address different flow optimization problems, such as computing the maximum joint traffic that a network can process stably, selecting the minimal number of readers to process a given total load, or determining the optimal timeslot duration, which are novel in the RFID literature.

INDEX TERMS Queueing systems, RFID networks, stability.

I. INTRODUCTION

The expansion of RFID technology is evident in the areas of supply chain management [1]–[5], manufacturing [6]–[9], health [10], [11] and daily life applications [12], [13], posing new design issues related to the coexistence of interfering RFID readers. In many applications, the tags arrive at the reader grouped in *batches*, and these batches form a queue where works are processed one by one, by inventorying all tags in each batch. A typical example of a queued RFID system could be a grocery store (e.g. [12]) with automatic billing at checkout portals. Another case could be a logistic center where incoming and outgoing boxes are inventoried using RFID. In other applications, tag batches do not form physical queues, but virtual ones, since the same reading job has to be repeated at a given pace. For example, in location and tracking systems [14], [15] and health monitoring systems [10], [11] enhanced performance and accuracy have been demonstrated using groups of tags, and reading operation is periodically repeated.

The previous examples usually convey several readers working closely, where one reader's transmission might easily jam the inventory process in other readers, even in distant ones (see [16]–[19]), given the weakness of the tags' backscattered signals. These configurations are called *queued RFID networks* (QRNs).

Several ways for mitigating the interference problem have been considered, including physical separation between readers, shielding, and splitting the power uplink and the backscattered channel using the EPC dense and multiple interrogation modes [20]. Besides, in high-density scenarios, arbitration algorithms among readers may be necessary. These algorithms can be contention-based like ETSI 302 208-1 [21] or operate synchronously. In the second case, readers use time-division multiple access (TDMA), either controlled in a distributed way or via a central coordinator. Centralized ones are called *schedulers*, and they can achieve a collision-free operation. Some recent examples of distributed algorithms are [22]–[24], while some schedulers have been discussed in [17], [25]–[28] and references therein.

The central question in schedulers' design is how to assign timeslots (the *scheduling policy*) so that a given performance metric is fulfilled or optimized. Some alternatives

The associate editor coordinating the review of this manuscript and approving it for publication was Kai Yang.

already explored in the previously cited works are related to throughput maximization. Other schedulers aim at minimizing energy usage and maximizing network lifetime [29], [30]. However, in QRNs, the goal is to know if the system can process all the jobs, without the queue backlogs increasing indefinitely. Queueing systems with this property are said to be *stable*. Single reader systems are stable if the arrival rate (i.e., the time-average incoming batches per timeslot) is less than the service rate (i.e., the time-average batches per timeslot that the reader can process). But, in TDMA QRNs, the actual service rate is reduced since the channel is shared. Therefore, stability is also affected by interferences between readers.

Besides, the scheduling policies can depend on some state variable (e.g., on the backlogs) or be stateless. Stateless policies are randomized and use a predefined probability distribution to select which readers become activated, and can be either stationary or non-stationary, depending on whether the policy changes over time or not. In [31], it was shown that all classes are equivalent (even stationary and non-stationary ones) in terms of the *capacity region*, i.e., the maximal set of incoming traffics that can be stabilized. In QRNs we can adopt any of these policy classes but in many cases determining the backlogs will be challenging (e.g., the number of customers waiting for checkout in a shop, or the number of boxes to offload from a truck) and it will be easier to implement a stationary stateless policy. Therefore we assume this type of policies in this work.

In order to compute these type of policies, basic QRN information is necessary. Namely, the arrival rates, the interference patterns, and the service rates. The service rates depend on the tag anti-collision protocol used, the distribution of the number of tags per batch (*batch size*), and the channel characteristics, that we describe through a probabilistic model including packet errors and capture effect.

In addition, we assume that at a given timeslot one batch can be processed at most since a new batch will need some preparation before identification can start (e.g., moving to the designated area). Thus, under stationary conditions, the service rate is equivalent to the *batch identification probability*. Computing this probability is cumbersome since it depends on how many interrogation frames are allocated within the TDMA timeslot duration, and on the particular tag anti-collision protocol used. For the Frame Slotted Aloha (FSA) tag anti-collision protocol, the interrogation frames have a random duration, which depends on how many tags contend and the outcome of their contention (see Section V-A). Hence, the number of frames that can be allocated in a TDMA timeslot is also random, and it indeed depends on how the identification evolves and on the FSA-policy (the algorithm selecting the number of contending slots per frame). Obtaining the batch identification probability subject to this stochastic process is an open question that we tackle in this work using an exact numerical method based on a recursive analytical formulation. Earlier, Vogt [32] gave an approximation for this probability by obtaining the average interrogation frame duration experimentally. Other works (see [28], [33], and references therein) have proposed Markov chains to obtain this probability, where the stages correspond to interrogation frames/slots and assuming that the TDMA timeslot has a fixed number of these frames/slots. This method can be precise if they have similar durations, but in RFID durations depend on the event (singleton, idle or collision slot) and therefore Markov-based methods are inaccurate. As far as we know, our work is the first proposal of an exact computation method.

A. CONTRIBUTIONS AND WORK DISTRIBUTION

To finish this introduction let us summarize the main contributions of this work:

- We develop an exact recursive procedure to compute the batch identification probability (service rates). The analysis is addressed in Section V. Different examples exploring the effects of the timeslot duration t, the FSA tag anti-collision protocol, the probabilistic channel model and the (random) batch size are provided in Section V-E.
- We compute the capacity regions of a QRN for a variety of network, reader and traffic configurations. Section IV describes the computation methodology, and Section VI provides examples of the capacity region associated with different setups.
- Based on these results we study different flow control problems in Section VII.
- In particular, in Section VII-A, we describe how the timeslot duration t has to be configured to attain QRN stability.
- In Section VII-B, we show that QRNs may be unstable due to incompatibilities of the timeslot duration t among different readers. This effect may even appear in QRNs without interferences between readers.
- In Section VII-C we address the question of determining the optimal load distribution and timeslot duration t for a general QRN. This procedure also allows us to compute the maximum joint traffic that a QRN is able to process stably.
- Finally, in Section VII-D we present a simple dimensioning example that determines the minimum number of readers to serve a given total traffic, depending on the interference patterns.

In addition, Section II presents the related work, Section III describes the QRN model, Section VIII analyzes the proposed algorithms' performance, and Section IX concludes this work.

II. RELATED WORK

Different stability issues related to RFID systems have been discussed in the literature. Most works deal with the stability of the tag anti-collision protocol and are thus oriented to modeling and optimizing the operation of a *single cell/RFID reader* system. FSA systems have stability issues when the tags contending and the number of slots in the FSA interrogation frame is unbalanced, causing either many collisions or many empty slots and thus inefficiency in the interrogation process. Some examples of works on this topic are [33]–[36] (see also references therein). Their goal is proposing and analyzing variations of the tag anti-collision protocol to achieve a constant (stable) performance. For example, Yu and Chen [33] discussed multiple and single packet reception p-persistent FSA protocols, Qian *et al.* [34] proposed a scheme named ASAP to adaptively split the tag set into multiple subsets, estimating the cardinalities during the splitting to adapt the frame length. Jia *et al.* [35] introduced a new anti-collision protocol named collision tree protocol (CT) and showed that it achieves constant time delay and power consumption for one-tag identification. Reference [36] presents another tree-search based approach, Tree-based Tag Search (TTS), which achieves a high efficiency. In our work, we focus on the standard FSA tag anti-collision protocol defined by EPCglobal Class-1 Gen-2 - ISO 18000-6C standard [20] allowing any arbitrary FSA-policy, including Dynamic-FSA (DFSA) policies, which permit to balance the number of tags and the FSA frame-length during operation.

Besides, some works (e.g., [33], [37]–[39] have also studied the stability regions and schedulers for different types of RF-backscattering systems based on slotted aloha protocols. For example, the authors of [33] developed a model based on a discrete-time Markov chain, and through Lyapunov's drift analysis they proved the ergodicity of the chain. In [37], the authors analyzed the properties of slotted aloha with capture for random access over fading channels with infinitely-many users. In [38], authors modeled a power and probability control mechanism that is only based on decentralized channel state information. The authors of [39] used a stochastical dominance method to prove the stability of a network with opportunistic RF energy harvesting nodes. Unfortunately, these models are only useful to study the equivalent of a single cell/RFID reader system. Indeed, a QRN fits in the model of constrained queued networks that have been studied in [31]. As in some of the works cited, the authors of [31] carried out a drift analysis to study the stability of the network, and this result can be directly applied to QRNs (see Section III).

The stability of RFID networks has been analyzed from different perspectives depending on the traffic model. In [40], [41], the authors studied networks with continuous non-stopping traffic, where readers compete for the interrogation time of flows of non-grouped tags. In particular, time-assignment policies seeking for the lowest tag-loss probability are proposed. The dynamic analyses developed in these works show that some configurations reach stable equilibrium points, whereas in other cases the tag-loss probability fluctuates between different points, causing network performance instability.

The stability of queued RFID networks has been considered previously by Tang et al. in [42]. They discussed the stability conditions for RFID networks and developed a distributed stabilizing scheduling based on the max-weight policy described in [31], which assigns slots to the servers with the longest queue backlogs that can transmit correctly. In [42] the stability problem was addressed from a different perspective than our work. It focused on algorithm design, while we focus on modeling by determining the service rates explicitly, and thus the capacity region.

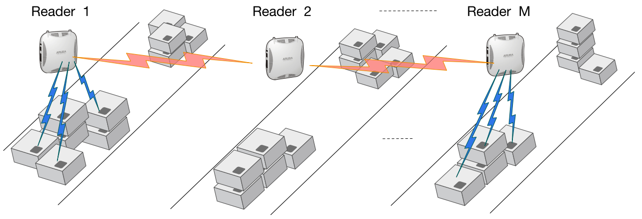


FIGURE 1. QRN model. Readers read tag batches until all tags are identified, then they proceed to the next batch. The central reader cannot operate at the same time as the other two due to interferences.

Besides those scenarios, the development of IoT systems (especially with the advent of 5G networks [43]) will likely boost the use and density of RFID readers. As discussed in the introduction, many of them may have virtual queues of reading jobs that have to be repeated continuously. In this case, the results obtained in this work can be directly applied to dimension and study these systems. Besides, other scenarios with concurrent operation of several RFID readers have been proposed in the literature; e.g., Dong *et al.* [44] analyzed how a given set of tags should be assigned to readers so that the cost of tag monitoring across the different readers is balanced, while guaranteeing that at least one reader monitors each tag. These analyses can require the precise determination of the service rates (batch identification probabilities) that we develop in this work.

III. QUEUED RFID NETWORK MODEL

Fig. 1 depicts the main characteristics of our model and Table 1 summarizes the nomenclature used. The QRN is composed of m readers that operate synchronized in a time-slotted fashion controlled by some scheduling algorithm, which prevents interfering readers from operating simultaneously. The interferences in the QRN are characterized by an $m \times m$ edge matrix E whose element ($e_{jj'}$) is 1 if reader j blocks reader j' operation due to interferences, and 0 otherwise. Fig. 2 shows some examples. At assigned timeslots, the readers perform inventories using the EPCglobal Class-1 Gen-2 ISO 18000-6C standard [20], which uses Frame Slotted Aloha (FSA) as the anti-collision protocol. FSA operation is described in detail in Section V-A.

Tags arrive in batches (e.g., modeling the usual case where a bunch of items is inside a box, a shopping cart or the like), and at each timeslot at most one batch can be read per reader. The timeslot length t is a configurable parameter that determines the maximum duration of the identification process for each batch. At timeslot i, a random number A_j^i of

TABLE 1. Main variables and parameters of the QRN model.

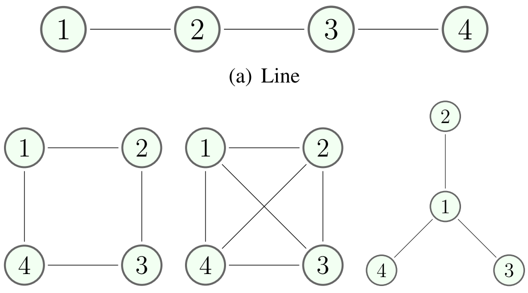
Variable	Description
m	Number of readers in the QRN
t	Timeslot duration (seconds)
r	Number of valid reader activation sets
E	$m \times m$ matrix whose element $(e_{jj'})$ is 1 if during operation reader j interferes with reader j', 0 otherwise
S	$r \times m$ matrix containing all possible valid reader activation sets (one per row, setting active readers as '1' and non-active readers as '0')
A_j^i	Number of batches arriving at reader j at timeslot i (random)
$a_j(t)$	Mean number of batches arriving at reader j (batches/timeslot with duration t)
$\boldsymbol{a}(t)$	Vector $(a_1(t), \ldots, a_m(t))'$
N_j	Batch size (tags) of incoming batches to reader j (random)
N	Similar as above, but omitting dependency on a specific reader
n	Deterministic batch size (tags)
B_j^i	Number of batches that reader j can read at timeslot i (random, takes value 0 or 1)
$b_j(t)$	Mean number of batches read by reader j per timeslot with duration t ($0 \leq b_j \leq 1$). It can also be interpreted as the batch identification probability
$b(t)$	Similar as above, but omitting dependency on a specific reader
$b(n, t)$	Similar as above, but assuming dependency on a deterministic batch size n
$\boldsymbol{b(t)}$	Vector $(b_1(t), \ldots, b_m(t))'$
Q_j^i	Backlog at reader j at the beginning of timeslot i, i.e., number of batches waiting for interrogation (including the one that is being read)
$\boldsymbol{Q}$	Discrete-time Markov chain whose state at stage i is vector $(Q_1^i, \ldots, Q_m^i)'$
ρ_j	Activity ratio of reader j, i.e., the percentage of time that reader j is activated by the scheduler
$\boldsymbol{\rho}$	Vector $(\rho_1, \ldots, \rho_m)'$
$\boldsymbol{p}$	Vector $(p_1, \ldots, p_r)'$. It represents a randomized policy, where $p_{r'}$ is the probability that the scheduler selects the r'th row of matrix S as activation set
ε_j	Slack variable for the linear program (4) representing the unused capacity in reader j
λ	Joint load of the QRN (batches/second)
$\lambda_{\mathbf{MAX}}$	Maximum joint load of the QRN (batches/second)
γ_j	Load of reader j (batches/second). If it is homogeneous for all readers, it is denoted simply as γ
p_s	Probability that the identification transmission can be correctly decoded
p_c	Capture probability, i.e., probability that the strongest signal of colliding tags can be correctly decoded
$\kappa(n)$	FSA-policy, i.e., a function that, depending on the tag count n, determines the number of slots in the next FSA-frame
K	Number of slots in an FSA-frame
ν_e	Number of empty FSA-frames at the end of the interrogation process required to confirm that all tags have been read
Δ_Q	Extra duration of the first slot (*Query* packet) in an FSA-frame (seconds)
δ	Number of tags read during an FSA-frame (random)
τ	Time elapsed during an FSA-frame (random)
s	Number of slots chosen by a single tag in an FSA-frame (random)
s^*	Number of tags out of s that correctly transmit their identification (random)
c	Number of slots chosen by more than one tag in an FSA-frame (random)
c'	Number of slots where the capture effect permits to solve collision (random)
c^*	Number of tags out of c' that correctly transmit their identification (random)
e	Number of empty slots in an FSA-frame (random). Note that n=s+c+e
t_s	Duration of a slot with a single tag transmission (seconds)
t_c	Duration of a slot with a collision (seconds)
t_e	Duration of an empty slot (seconds)

new batches arrive at the queue of reader j. When a batch is identified, the next batch in the queue will be processed in the next timeslot (like in a shopping queue, where a customer must wait until the previous one has been served). If a batch is not completely identified at the end of a timeslot, its identification process *restarts* in the next available timeslot. This characteristic is introduced since the persistence time (i.e., the time while identified tags avoid participating in a new identification process) is in the order of the timeslot duration, and therefore we assumed the worst-case scenario, that all the tags in the batch (even those previously identified) would participate again in the next identification. Thus, the number batches that a non-empty reader j can read at timeslot i is a Bernoulli random variable B_j^i (either 1 or 0 batches are read per slot). B_j^i depends on the channel conditions and the FSA configuration for the reader j at timeslot i, and on the number of tags per batch (*batch size*), which we assumed to be random and depending only on the particular reader j. Let variables N_j, for $j = 1, \ldots, m$ denote them.

In the remainder, we assume that variables A_j^i and B_j^i are independent, i.i.d. over timeslots (i.e., their distributions do not change over time) and independent over readers. Besides, we will assume stationary channel conditions and readers configurations. Therefore, their means will not depend on the particular timeslot i. Henceforth, let $a_j(t)$ and $b_j(t)$ denote the mean arrival rate and the mean service rate (which can be regarded as the *batch identification probability*), respectively. Let us remark that both depend on the timeslot duration t, and so we stress this dependency on the notation.

IV. QUEUED RFID NETWORK STABILITY

Queueing systems' stability is a major concern in operations research, and it is related to the long-term time average behavior of the system. Single-queue systems are rate stable when the time-average traffic rate is less than or equal to the time-average service rate [45] (e.g., by the law of the large numbers in our model this condition is equivalent to $a_j(t) \leq b_j(t)$ for reader j). However, in multi-queue systems, the equivalence is



(a) Line

(b) Square (c) Full (d) Star

FIGURE 2. Example network graphs for $m = 4$. Connected readers cannot operate at the same time due to interferences. Even though it is not shown in the examples above, links may be directed or undirected, since readers may have decoupled reception and transmission antennas, and therefore a reader j may block reader j', but not *vice versa*.

not straightforward. Different networks in subcritical regime (i.e., when $a_j(t) \leq b_j(t)$ for all $j = 1, \ldots, m$) are known to be unstable (in terms of a queue length growing indefinitely). For example, in the Lu-Kumar networks [46], due to the queue discipline used, some nodes cannot work simultaneously, resulting in operational *constraints*. In order to determine which traffics can be processed stably, it is necessary to account for these constraints and adjust the service rates.

The same occurs in a QRN: depending on which readers are active, others may operate or not. Intuitively, it requires to adjust the services rates to the actual ones by applying an *activity ratio* to each reader. This activity ratio would represent the percentage of time that each reader j is active.

Formally, the QRN dynamics can be described as follows. Let 1_j^i denote an indicator variable, whose value is 1 if the scheduler activates reader j at timeslot i, or 0 otherwise. Associated to each reader j, there is a queue backlog Q_j counting the jobs (batches of tags) waiting for identification. The evolution of the backlogs is given by:

$$Q_j^{i+1} = \max\{Q_j^i - B_j^i 1_j^i, 0\} + A_j^i \tag{1}$$

for timeslot $i = 1, 2, \ldots$ and for all queue $j = 1, \ldots, m$. Let $\boldsymbol{Q}^i$ denote the vector $(Q_j^{i+1}; j = 1, \ldots; m)$. Then $\boldsymbol{Q} = \{Q^0, Q^1, Q^2, \ldots\}$ is a discrete-time Markov chain. $\boldsymbol{Q}$ is rate stable if the actual service rate, i.e. $\mathbb{E}\{B_j 1_j^i\}$, is greater than or equal to the arrival rate $a_j(t)$ for each j.

Since the scheduler is based on a stationary randomized policy independent of the queue backlog, the stability condition is given by $\mathbb{E}\{B_j^i\}\mathbb{E}\{1_j^i\} = b_j(t)\rho_j \geq a_j(t)$, where ρ_j represents the *activity ratio* of the reader j. We can write this expression more compactly by using vector notation (see Table 1) as:

$$\boldsymbol{b}(t) \odot \boldsymbol{\rho} \geq \boldsymbol{a}(t), \tag{2}$$

where $\odot$ denotes the element-wise multiplication.

The activity ratio depends on the activation sets that are applied by the scheduler, and how often they are selected. An activation set is the set of readers that the scheduler selects at a given timeslot. The only condition that an activation set must fulfill is that it should not contain any incompatible readers; that is, a reader in the activation set cannot block any other reader in the set. Given the network edge matrix E, it is straightforward to obtain a list of all possible valid activation sets as a matrix S. Each row in S contains m binary elements, indicating whether reader j becomes active (cell content 1) or waits (cell content 0). A general approach to build S is given in algorithm 1. As an example, Fig. 4 shows the valid activation sets for the networks of Fig. 2. Henceforth, let r denote the number of valid activation sets, that is, the number of rows in S.

$$E = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

(a) Line

$$E = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

(b) Square

$$E = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

(c) Full

$$E = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

(d) Star

FIGURE 3. Edge matrices for the network graphs in the examples of Fig. 2. Matrix E element ($e_{jj'}$) is 1 if reader j blocks reader j' operation due to interferences and 0 otherwise, that is, if there is an edge between readers j and j' in the network graph.

Algorithm 1 Activation Matrix Calculation

```
# Parameters:
# $E$ (matrix $m \times m$), network graph

for cont = 0:$2^m$-1 do
    $v \leftarrow$ Binary representation of cont
    if $v'Ev = 0$ then
        Add row $v'$ to matrix $S$
    end if
end for

return $S$
```

To compute the activity ratios let $\boldsymbol{p} = (p_1, p_2, \ldots, p_r)$ be the probability mass characterizing the scheduler's policy, i.e. a vector assigning to each activation set in S the probability of being selected by the scheduler. Then, $\boldsymbol{\rho} = S'\boldsymbol{p}$, and from eq. (2) the network is rate stable if

$$\boldsymbol{b}(t) \odot S'\boldsymbol{p} \geq \boldsymbol{a}(t), \tag{3}$$

and the probability mass $\boldsymbol{p}$ defines a stabilizing policy (not necessarily unique). The *capacity region* (CR) is the set of non-negative arrival traffic vectors $\boldsymbol{a}(t)$ such that a stabilizing policy $\boldsymbol{p}$ exists.

Different equivalent ways to determine when a given rate vector $\boldsymbol{a}(t)$ is in the CR can be used (see [45]). For example,

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

(a) Line

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

(b) Square

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(c) Full

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

(d) Star

FIGURE 4. **Activation sets for the networks graphs of Fig. 2 and the corresponding edge matrices of Fig.3. Matrix *S* provides an exhaustive enumeration of all valid activation sets for the network. Each row of *S* contains a configuration that does not violate the constraints indicated in the corresponding edge matrix *E*.**

by solving the linear program (LP):

$$\begin{aligned} &\max_{\boldsymbol{p}} \sum_{j=1}^{m} \varepsilon_i \\ &\text{subject to: } p_{r'} \geq 0, \ \text{ for all } r' = 1, \ldots, r \\ &\qquad\qquad \varepsilon_j \geq 0, \quad \text{for all } j = 1, \ldots, m \\ &\qquad\qquad \sum_{j=1}^{m} p_j \leq 1, \\ &\qquad\qquad \boldsymbol{a}(t) + \boldsymbol{\varepsilon} = (S'\boldsymbol{p}) \odot \boldsymbol{b}(t) \end{aligned} \tag{4}$$

This LP has $m + r$ variables. Slack variables ε_i indicate the exceeding unused capacity or reader j (if any). The LP can be solved efficiently by using, for example, interior-point algorithms like Karmarkar's [47]. Moreover, solving this LP provides an explicit computation of a stabilizing policy $\boldsymbol{p}$. The only unknown parameters are the batch identification probabilities $\boldsymbol{b}(t)$, whose calculation is addressed in Section V.

Finally, in [31] the authors have demonstrated, using Lyapunov's drift analysis, that if the second moment of the number of arrivals per timeslot is finite, then, whenever the arrival rate vector $\boldsymbol{a}(t)$ is *interior* to the CR, the Markov chain $\boldsymbol{Q}$ is strongly stable. This form of stability guarantees finite average backlog and by Little's theorem finite average interrogation delay [45].

V. BATCH IDENTIFICATION PROBABILITY

In this section we compute the batch identification probability $b(t)$ for a given batch size N,[1] assuming an identification time t. First, we describe the FSA multi-frame operation. Then, we obtain the batch identification probability, $b(n, t)$, assuming a fixed number of contenders, n. Then, given the distribution of the batch size of N, $b(t)$ is obtained. Let us remark that other tag anti-collision mechanisms different than FSA are possible in RFID, such as tree-based searching protocols variants (see [34]–[36] and the references therein). For these protocols, specific analyses targeting the batch identification probability will be needed in order to analyze the QRN stability.

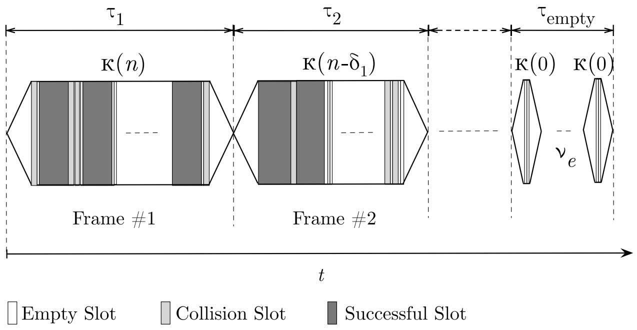


FIGURE 5. **RFID identification process. It comprises a sequence of FSA-frames whose length is defined by FSA-policy $\kappa(n)$. The process finishes when a number ν_e of ending frames is empty.**

A. FSA MULTI-FRAME OPERATION

The main characteristics and parameters of the identification process are summarized in Fig. 5. The process comprises a sequence of FSA reading frames that ends when no unidentified tags remain in the batch. The reader is aware of this condition if no tags reply to the interrogation queries for a given number of ending frames, ν_e. The reader starts an FSA frame by sending a *Query* packet, which states the length of the interrogation frame (K slots[2]). The tags randomly select one of these slots for identification. Correctly identified tags withdraw from the process, while the rest keep trying in the next FSA frame. The detailed configuration of the timing parameters is provided in [48]. Note that different configurations would result in different command durations. In particular, since the number of successful identifications (s), empty slots (e) and collisions (c) are random, the frame duration is also random. Additionally, the first slot in a frame has a longer duration, since the *Query* packets are longer than the *Query Rep* packets used in subsequent slots. This extra duration is given by Δ_Q.

Besides, an FSA-policy κ is applied, where $\kappa(n)$ determines the FSA-frame length given the number of contending tags n ([3]). Examples of these policies could be a simple fixed FSA-policy $\kappa = 2^Q$ with a fixed Q, or the selection of the closest possible frame length to the number of contenders (see Table 2 and reference [49] for details).

[1] Without loose of generality in this section we omit the explicit reference to the reader j in $b_j(t)$ and N_j notation.

[2] K must be a power of 2.

[3] In practice, the tag count should be obtained by some estimator. Here we are assuming the usual simplification of perfect tag count estimation.

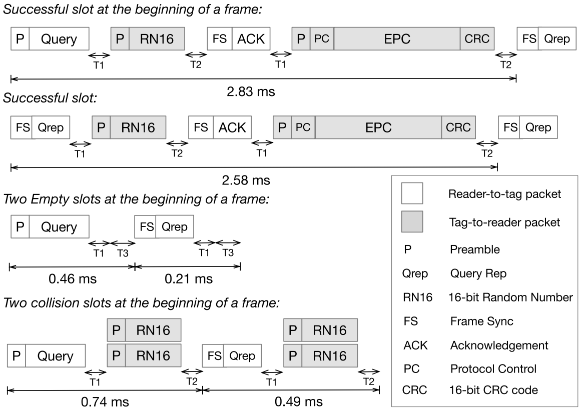


FIGURE 6. Durations of different command sequences in an ISO 18000-6C frame computed from the timing parameters specified in [48].

TABLE 2. Optimal DFSA-policy (see [49]).

$\kappa(n)$	n range
2^0	$n = 1$
2^1	$1 < n \leq 3$
2^2	$3 < n \leq 6$
2^3	$6 < n \leq 11$
2^4	$11 < n \leq 22$
2^5	$22 < n \leq 44$
2^6	$44 < n \leq 89$
2^7	$89 < n \leq 177$
2^8	$177 < n \leq 355$
2^9	$355 < n \leq 710$
2^{10}	$710 < n \leq 1420$
2^{11}	$1420 < n \leq 2839$
2^{12}	$2839 < n \leq 5678$
2^{13}	$5678 < n \leq 11357$
2^{14}	$11357 < n \leq 22713$
2^{15}	$22713 < n$

To model the tag contention outcome we assume a multi-tag probabilistic interference model with capture probability, p_c, which represents the the probability that the strongest signal of the colliding tags can be successfully decoded; p_c can be determined from a Signal-to-Interference model considering some spatial distribution for the tags as in [50] or [51]. Once the tag sends the (short) preamble and the reader acknowledges it, the tag transmits the (much longer) identification packet. The identification packet is always transmitted without collisions, but transmission errors might still happen. In our probabilistic model, we assume that identification is correctly received with probability $p_s \leq 1$.

B. COMPUTATION OF $B(N, T)$

Our goal is to compute the batch identification probability $b(n, t)$ for a given deterministic batch size n and identification time t. Let us recall that in a frame with n contenders the frame length is given by $\kappa(n)$, and let us denote:

- s as the number of slots chosen by a single tag (it is limited by the number of tags n and by the frame length $\kappa(n)$),
- s^* as the number of tags out of s that correctly transmit their identification,
- c as the number of slots chosen by more than one tag (it is limited by the maximum number of groups with two tags that can be formed with $n - s$ tags, $\lfloor\frac{n-s}{2}\rfloor$, or by the number of slots that are not occupied by a single tag $\kappa(n) - s$),
- c' as the number of slots where the capture effect permits to solve the collision, and
- c^* as the number of slots out of c' where the tags correctly transmit their identification.

Then, the outcome of the interrogation frame can be described as a random vector $(s, s^*, c, c', c^*|n, \kappa(n))$.

Moreover, since the events in different frames are independent, if the current frame lasts τ seconds and δ tags are identified, then the batch identification probability is just the probability that in the remaining frames $n - \delta$ tags are identified in $t - \tau$ seconds. That is, for a given interrogation outcome, it holds that $b(n, t) = b(n - \delta, t - \tau)$, where the number of identifications $\delta = s^* + c^*$, and the frame duration $\tau = \Delta_Q + (s + c')t_s + (c - c')t_c + (\kappa(n) - s - c)t_e$, being t_s, t_c, and t_e the duration of each type of slot, and Δ_Q the extra duration of the first slot in the frame, as described in Section V-A.

Then, by summing over all the possible outcomes of the interrogation frame, $b(n, t)$ can be expressed as the recursive formula (5), as shown at the bottom of the next page. In order to compute it, the probability of each outcome has to be determined. This probability can be expressed with the following chain of conditional probabilities:

1) Given s, since the identification events are independent on different slots and happen with probability p_s, s^* has binomial distribution:
$$p(s^*|s) = \begin{cases} \binom{s^*}{s} p_s^{s^*}(1 - p_s)^{(s-s^*)}, & \text{if } 0 \geq s^* \geq s \\ 0, & \text{otherwise} \end{cases} \tag{8}$$
Note that writing $p(s^*|s)$ is equivalent to $p(s^*|s, c, n, \kappa(n))$, since, once s is fixed, s^* does not depend on the other variables.

2) Similarly, the distribution of tags that were correctly identified from slots with solved-collisions is:
$$p(c^*|c') = \begin{cases} \binom{c^*}{c'}(p_s)^{c^*}(1 - p_s)^{(c'-c^*)}, & \text{if } 0 \geq c^* \geq c' \\ 0, & \text{otherwise} \end{cases} \tag{9}$$
and $p(c^*|c')$ is equivalent to $p(c^*|c', s, c, n, \kappa(n))$

3) Given c, the distribution of slots with solved-collision is also binomially distributed:
$$p(c'|c) = \begin{cases} \binom{c'}{c}(p_c)^{c'}(1 - p_c)^{(c-c')}, & \text{if } 0 \geq c' \geq c \\ 0, & \text{otherwise} \end{cases} \tag{10}$$
and $p(c'|c)$ is equivalent to $p(c'|s, c, n, \kappa(n))$

4) Finally, $p(s, c|n, k)$ is provided by Chung *et al.* [52] and shown in formula (6), as shown at the bottom of the next page, using the notation followed in this work.

Now, it can be checked that

$$p(s, s^*, c, c', c^*|n, \kappa(n)) \\ = p(s^*|s)p(c^*|c')p(c'|c)p(s, c|n, \kappa(n)) \quad (11)$$

holds, since s^* and c^* are independent once the pair s, c is fixed. Therefore, using previous formulas we can rewrite eq. (5) as eq. (7), as shown at the bottom of this page.

C. COMPUTATION OF B(T)

Given the batch load distribution $p(\mathrm{N} = \mathrm{n})$ the identification probability can be determined directly as:

$$b(t) = \sum_{n} b(n, t)p(N = n) \quad (12)$$

D. ENDING FRAMES

As stated at the beginning of this section, the identification process includes ν_e ending frames, whose goal is to guarantee that no tags remain unidentified. If all tags have already been identified, these frames will be empty, and therefore their duration will be $\kappa(0)$. Thus, the total ending time is given by $\nu_e\kappa(0)$. It is straightforward to include this effect in the computation of the batch identification probability by shifting the function in time, i.e. $b(t - \nu_e\kappa(0))$.

E. EXAMPLES

Fig. 7 shows the batch identification probabilities obtained for two kinds of batch sizes: (i) Poisson-distributed N with rate $\overline{N} = 100$ tags per batch, and (ii) Fixed $N = 100$ per batch, and for two kinds of channel conditions: (i) ideal channel: $p_c = 0$ and $p_s = 1$, and (ii) realistic channel: $p_c = 0.1$ and $p_s = 0.9$. In all cases, the reader FSA-policy is the same, $\kappa(n) = 16$ for all n. This is the default setup for most off-the-shelf RFID readers. The number of empty frames used to declare the end of the reading procedure was set to $\nu_e = 1$. Overall, the results reveal that the load distribution and the channel conditions shape the batch identification probability and thus the capacity region of a network. Moreover, the batch identification probability computed under ideal channel conditions overestimates the time required to identify a batch by large. This result stresses the need for realistic channel models.

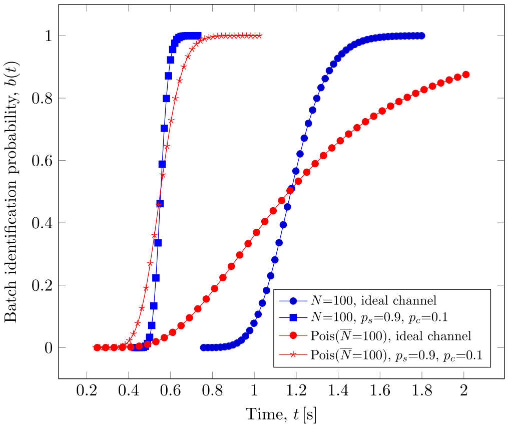


FIGURE 7. Batch identification probabilities versus timeslot duration for different load distributions and channel conditions, FSA-policy $\kappa(n) = 16$.

Besides, Fig. 8 shows the batch identification probabilities for different size distributions. The higher the load, the longer the identification time, as is it could be expected. Another feature that is also shown in Fig. 7 is the slower growth rate of $b(t)$ when the dispersion of the batch size is higher, as happens with the Poisson distribution compared to the fixed one, and for Poisson loads with higher means.

The effect of the FSA-policy can be observed in Fig. 9. It depicts the batch identification probability for three common policies: (1) $\kappa(n) = 16$, (2) $\kappa(n) = 64$, and (3) $\kappa(n)$ given by the DFSA selection table published in [49], reproduced in table 2 for convenience. As expected, the performance is better for the DFSA-policy, although the policy $\kappa(n) = 64$ performs quite close in this example.

$$b(n, t) = \begin{cases} 1, & \text{if } n = 0 \\ 0, & \text{if } n > 0 \text{ and } t \le 0 \\ \displaystyle\sum_{s=0}^{\min\{n,\kappa(n)\}} \sum_{c=0}^{\min\{\lfloor \frac{n-s}{2}, \kappa(n)-s\}} \sum_{c'=0}^{c} \sum_{c^*=0}^{c'} b(n - \delta, t - \tau)p(s, s^*, c, c', c^*|n, \kappa(n)), & \text{otherwise.} \end{cases} \quad (5)$$

$$p(s, c|\kappa(n), n) = \frac{\kappa(n)!n!}{(\kappa(n) - s - c)!s!\kappa(n)^n} \sum_{a=0}^{\min\{n-s,c\}} \sum_{b=0}^{c-a} (-1)^{a+b} \frac{1}{a!b!(n - a - s)!} \frac{(c - a - b)^{n-a-s}}{(c - a - b)!} \quad (6)$$

$$b(n, t) = \begin{cases} 1, & \text{if } n = 0 \\ 0, & \text{if } n \neq 0 \text{ and } t \le 0 \\ \displaystyle\sum_{s=0}^{\min\{n,\kappa(n)\}} \sum_{c=0}^{\min\{\lfloor \frac{n-s}{2} \rfloor, \kappa(n)-s\}} \sum_{c'=0}^{c} \sum_{c^*=0}^{c'} b(n - s^* - c^*, t - (\Delta_Q + (s + c')t_s + (c - c')t_c + (\kappa(n) - s - c)t_e)) \\ \times p(s^*|s)p(c^*|c')p(c'|c)p(s, c|n, \kappa(n)), & \text{otherwise.} \end{cases} \quad (7)$$

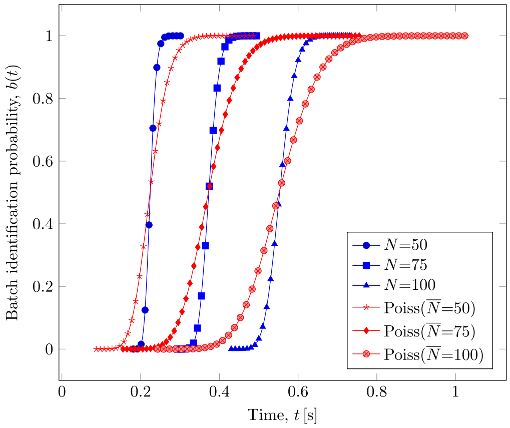


FIGURE 8. Batch identification probabilities versus timeslot duration for different load distributions, realistic channel conditions, $p_s = 0.9$, $p_c = 0.1$, FSA-policy $\kappa(n) = 16$.

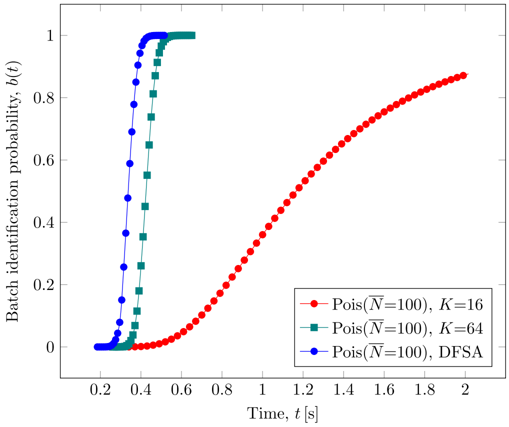


FIGURE 9. Batch identification probabilities versus timeslot duration for different FSA policies, ideal channel conditions, Poisson batch size with $\overline{N} = 100$ tags/batch.

VI. CAPACITY REGION EXAMPLES

In this section, we provide numeric examples of the capacity region (CR) for several different network graphs and traffic configurations. In order to provide useful insight, and also as a practical tool for configuring QRNs, we have considered traffic subject to a maximum of 3 independent variables $(a_1(t), a_2(t), a_3(t))$ defined by an affine transformation. For example, for a QRN with 4 readers examples of the target traffic could be $(a_1(t), a_2(t), a_3(t), a_1(t) + a_2(t) + a_3(t))$ or $(a_1(t), a_2(t), a_3(t), 1/4)$.

Fig. 10 shows the results. In all cases the CR grows as t increases, since so does $b(t)$. Any factor altering $b(t)$, such as changing the FSA-policy, the channel conditions or the batch size, determines the CR final size. CR shape is mainly related to the network topology and the dependencies between the incoming traffic rates. Although it is not shown in the examples, when the FSA-policy or the channel conditions are heterogeneous, the CRs tend to be asymmetric because the service rates vector $\boldsymbol{b}(t)$ is unbalanced.

Fig. 11 shows the sum of queue backlogs versus time for different traffic configurations. For the points inside the CR, the stationary stabilizing policies computed using program (4) have been applied, whereas for traffics outside the CR the max-weight policy has been used. The results clearly show how that the backlog grows uncontrollably for points far from the CR border, and that it remains bounded for very inner points. For points close to the border, the trend is not evident, but it can be intuited.

VII. OPTIMAL FLOW CONTROL

In this section, we address several flow control problems related to QRNs:

1) Determining whether a given total load of λ batches per second can be processed by a QRN, the suitable timeslot durations t that can be used, and how to distribute the load among readers.
2) Computing the maximum joint load λ that can be processed per second by a QRN.
3) Obtaining the number of readers required to process a given joint load λ given the QRN interference patterns.

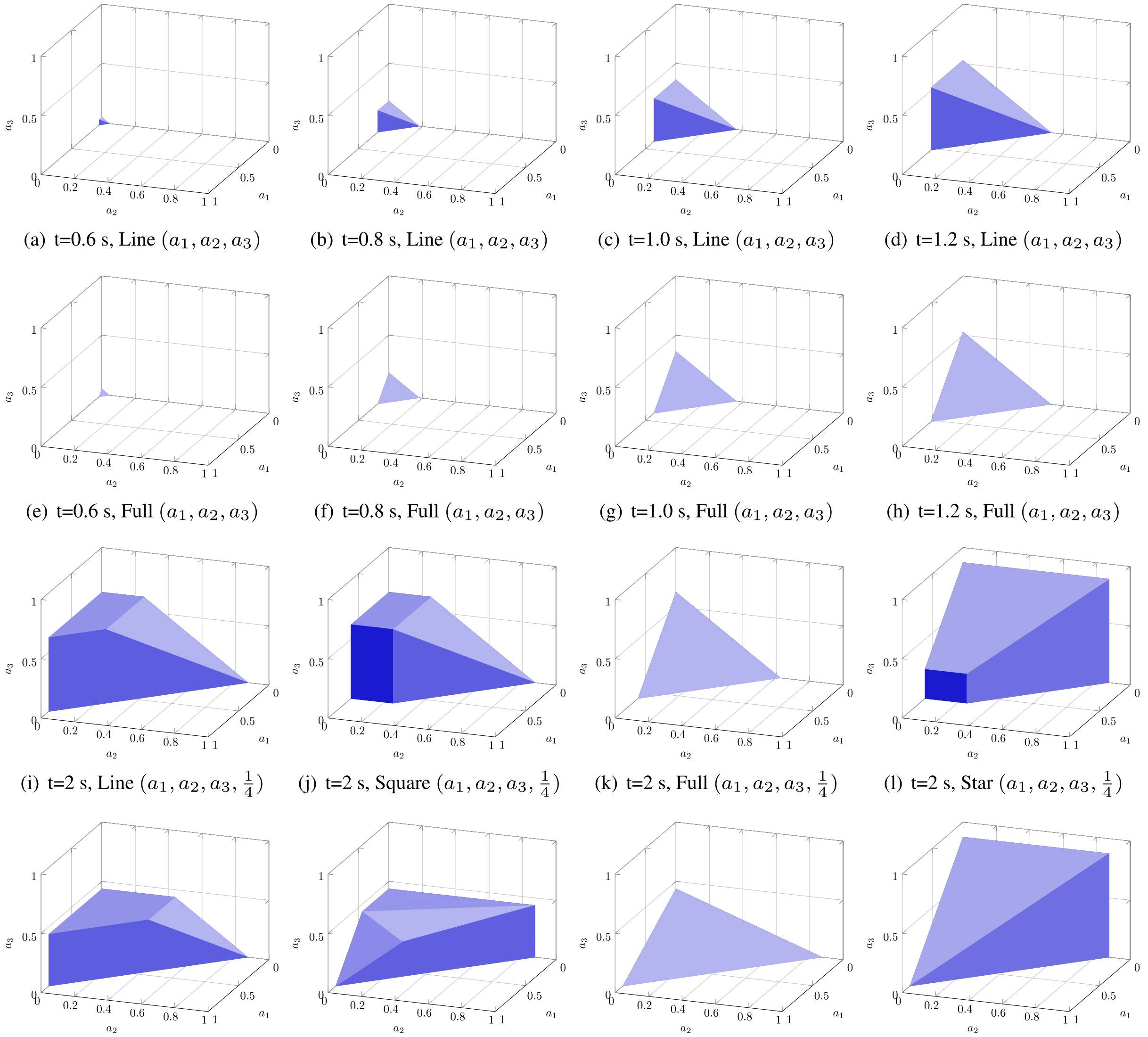
A. SINGLE READER CASE

In this case, the load per second at the single reader, γ, matches the total network load per second λ. Stability can be trivially determined by comparing the arrival to the service rate. The unique variable to configure is the timeslot duration t, yielding an arrival rate $a(t) = \lambda t$ and a service rate given by $b(t)$. The system is stable when $\lambda t \leq b(t)$, i.e. when $\lambda \leq \frac{b(t)}{t}$. Fig. 12 shows examples for this curve and represents the intervals of t leading to stable QRN operation (henceforth t-intervals). For example, in Fig. 12, for $\gamma = \lambda = 3.75$ batches/s the system is only stable for batches of fixed size $N = 50$ tags/batch and only if $t \in (0.23, 0.28)$ s. When the arrival rate is lower, other traffic regimes can be served as well, for example, when $\gamma = \lambda = 0.5$ batches/s, all traffic and batch size configurations can be stabilized, although the t-interval varies for each of them.

B. QRN WITHOUT READER TO READER INTERFERENCES

Although this case could be considered naive, the previous result allows us to arrive at a perhaps surprising conclusion: since the stability of each reader depends on the timeslot duration; the network is stable only if *all* the t-intervals overlap in a common interval, which might not occur in the general case.

In the flow control problem the distributions of the batch sizes will be similar for all readers, but there could be changes in the FSA-policy set at each reader, or the channel conditions could vary. For example, consider the case shown in Fig. 13, where the readers receive batches of fixed load $N = 50$ but use different FSA policies. Assuming an homogeneous traffic per second $\gamma = \frac{\lambda}{m}$ for each reader, when $\gamma < 4.15$ there exists an overlapping stable t-interval. However, for a higher γ the t-intervals are disjoint. The figure remarks the case for $\gamma = 4.26$, which can always be served but not in the same t-interval. Moreover, this deadlock condition might happen when channel conditions vary, even for homogeneous loads and using the same reader configurations, as shown in Fig. 14.



(a) t=0.6 s, Line (a_1, a_2, a_3) (b) t=0.8 s, Line (a_1, a_2, a_3) (c) t=1.0 s, Line (a_1, a_2, a_3) (d) t=1.2 s, Line (a_1, a_2, a_3)

(e) t=0.6 s, Full (a_1, a_2, a_3) (f) t=0.8 s, Full (a_1, a_2, a_3) (g) t=1.0 s, Full (a_1, a_2, a_3) (h) t=1.2 s, Full (a_1, a_2, a_3)

(i) t=2 s, Line $(a_1, a_2, a_3, \frac{1}{4})$ (j) t=2 s, Square $(a_1, a_2, a_3, \frac{1}{4})$ (k) t=2 s, Full $(a_1, a_2, a_3, \frac{1}{4})$ (l) t=2 s, Star $(a_1, a_2, a_3, \frac{1}{4})$

(m) t=2 s, Line (a_1, a_2, a_3, a_3) (n) t=2 s, Square (a_1, a_2, a_3, a_3) (o) t=2 s, Full (a_1, a_2, a_3, a_3) (p) t=2 s, Star (a_1, a_2, a_3, a_3)

FIGURE 10. Capacity regions for the scenarios of Fig. 2, FSA-policy $\kappa(n) = 16$, ideal channel, Poisson batch size with $\overline{N} = 100$ tags/batch.

Hence, even a QRN without interferences between readers might be unstable. Let us stress that, in this case, the time-slotted operation would be unnecessary and should be avoided, preventing this issue. Nevertheless, in QRNs with interferences, this same issue would irrevocably lead to deadlocks.

C. MAXIMUM JOINT LOAD OF A QRN

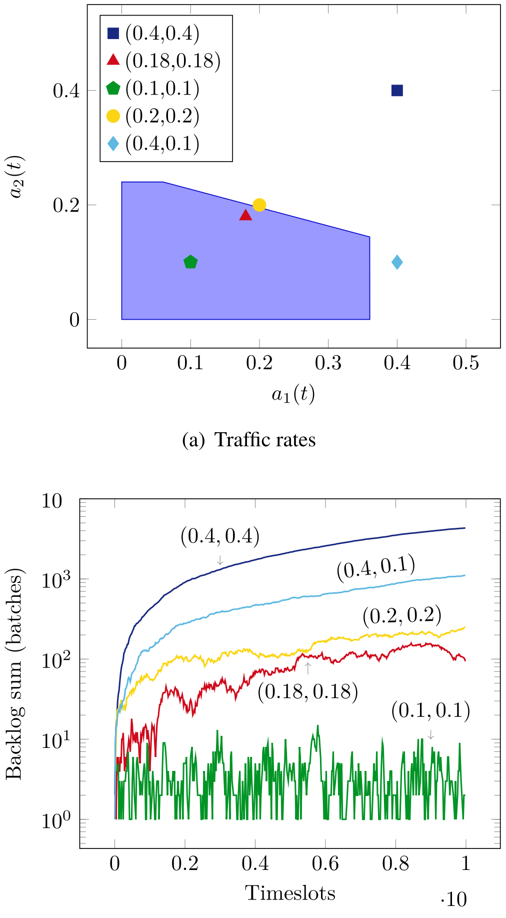
To compute the maximum joint load λ_{MAX} that a QRN can process stably, let us recall that in the LP (4) the slack variables ε_j are associated to the "unused" capacity of a reader. Since the joint arrival rate has to be maximum, ε_j must vanish in the solution. This is equivalent to $\boldsymbol{a}(t) = \boldsymbol{\rho} \odot \boldsymbol{b}(t) = (S'\boldsymbol{p}) \odot \boldsymbol{b}(t)$, and then the joint load per second is

$$\lambda = \sum_{j=1}^{m} \frac{a_j(t)}{t} = \frac{(S'\boldsymbol{p})'\boldsymbol{b}(t)}{t} = \frac{\boldsymbol{p}'S\boldsymbol{b}(t)}{t}. \tag{13}$$

Hence, the maximum joint load per second can be computed as:

$$\begin{aligned} \lambda_{\text{MAX}} = \quad & \max_{\boldsymbol{p},t} \frac{\boldsymbol{p}'S\boldsymbol{b}(t)}{t} \\ & \text{subject to: } t > 0 \\ & \qquad p_i \geq 0, \quad \text{for all } i = 1, \ldots, r \\ & \qquad \sum_{i=1}^{r} p_i \leq 1, \end{aligned} \tag{14}$$

This program is non-convex since functions $b_j(t)/t$ are non-convex as can be seen in Figs. 12-14, so neither is their weighted sum. However, once t is fixed, the problem is linear and efficient algorithms can be applied to solve it. Therefore, an approach for solving (14) is to exhaustively search over a given t range, and get the best solution.



(a) Traffic rates

(b) Backlog evolution

FIGURE 11. **Backlog evolution for different traffic input rates, t = 1 s, FSA policy $\kappa(n) = 16$, ideal channel, Poisson batch size with $\overline{N} = 100$ tags/batch.**

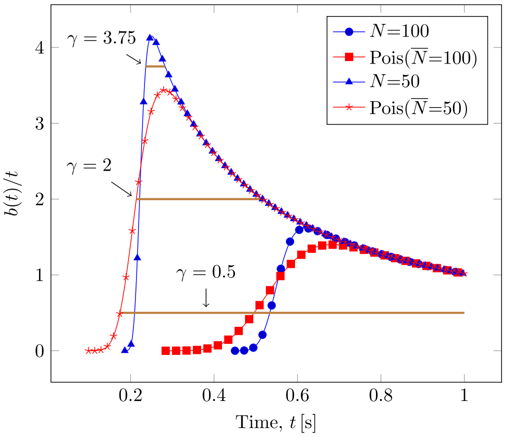


FIGURE 12. ***t*-intervals with stable operation, single reader case, realistic channel conditions, $p_s = 0.9$, $p_c = 0.1$, FSA-policy $\kappa(n) = 16$.**

Determining whether a given load λ can be processed in the case of a general QRN is done trivially by checking if $\lambda \leq \lambda_{MAX}$. If so, the solution of program (14) provides a suitable configuration for t and a stabilizing policy $\boldsymbol{p}$. The load per second to be assigned to j, γ_j, can be obtained

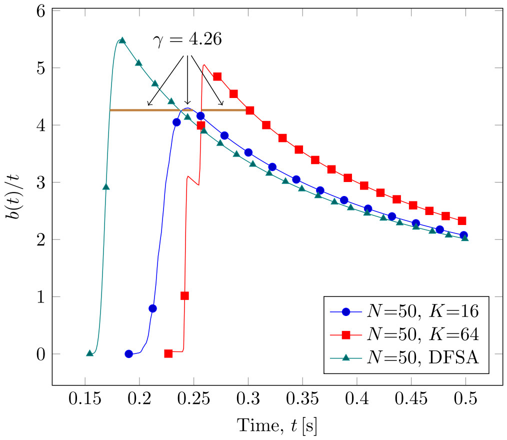


FIGURE 13. **Incompatible *t*-intervals for stable operation, ideal channel conditions, fixed batch size $N = 50$.**

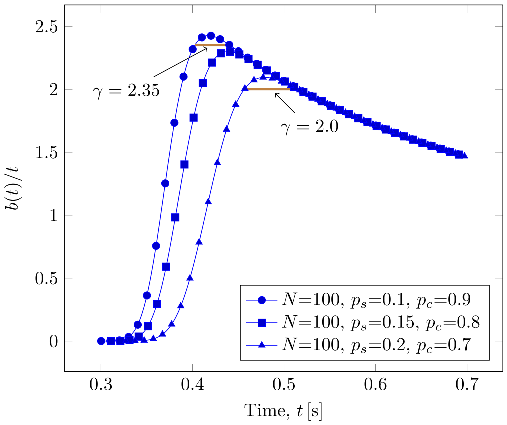


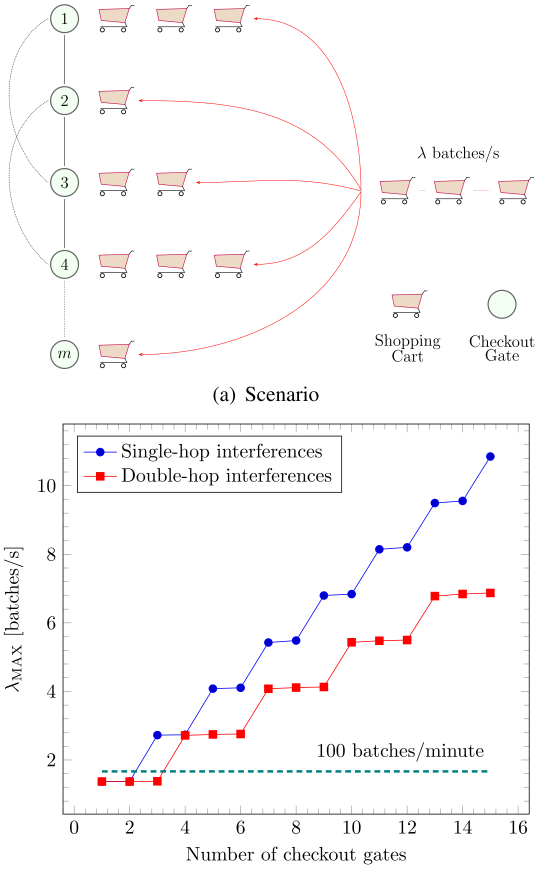
FIGURE 14. **Incompatible *t*-intervals for stable operation, fixed batch size $N = 100$, FSA-policy $\kappa(n) = 16$.**

(assuming a fair load distribution) as:

$$\gamma_j = \lambda \frac{b_j(t)\rho_j}{\lambda_{MAX}} = \lambda \frac{b_j(t)(S'\boldsymbol{p})_j}{\boldsymbol{p}'S\boldsymbol{b}(t)} \tag{15}$$

D. PRACTICAL DIMENSIONING EXAMPLE

Finally, we provide an example where we determine the number of readers required to process a total incoming load λ. The scenario is depicted in Fig. 15(a). It consists of a sequence of readers arranged in a line, representing the checkout gates in a shop. Depending on the distance between them, there will be interferences. We consider two cases: (i) single-hop interferences and (ii) double-hop interferences. The results are displayed in Fig. 15(b) and show the maximum possible rate versus the number of readers (checkout gates). As expected, with double-hop interferences the load that can be handled by the network is reduced. The landmark of 100 batches per minute can be handled by 3 readers for single-hop interferences but demands an additional reader when interferences also occur at two hops. Note that this result also provides insight into how the countermeasures described in the introduction (e.g., shielding) must be designed to support a given traffic load.



(a) Scenario

(b) Results

FIGURE 15. Dimensioning example.

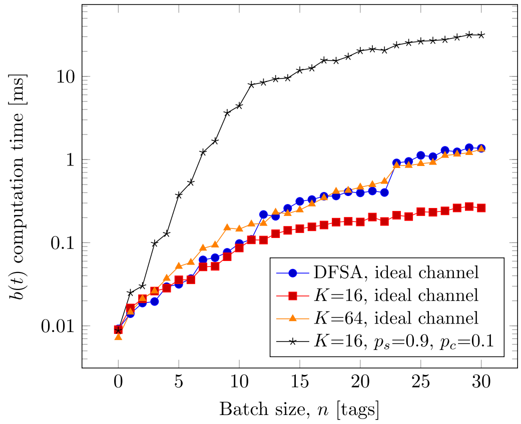


FIGURE 16. Average computation time for the batch identification probabilities.

VIII. COMPUTATIONAL PERFORMANCE ANALYSIS

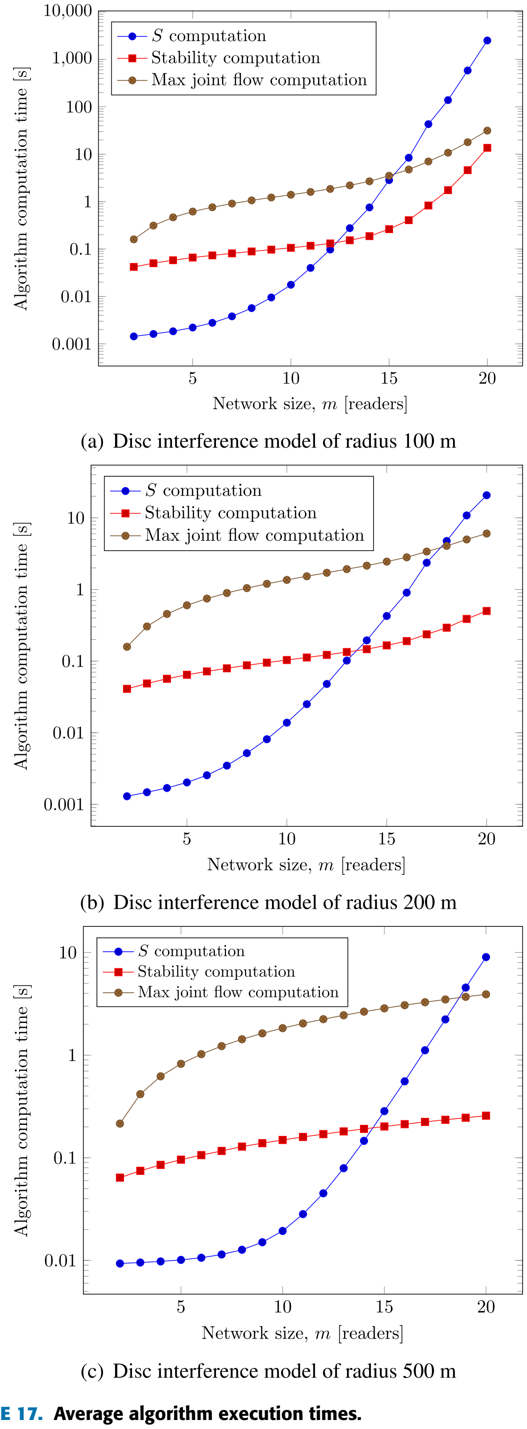
The main algorithm developed in this work is the computation of the batch identification probability, in Section V. Computing $b(n, t)$ requires to exhaustively explore all possible frame allocation combinations for a given time t and batch size n. Thus, the number of combinations to explore grows exponentially with these parameters, and is $O(e^{tn})$). Besides, depending on the FSA-policy and the channel conditions, the number of combinations may vary substantially. For example, using FSA $\kappa(n) = 16$ requires analyzing the behavior of a *single* frame-length, but if DFSA is used, then many different frame-lengths have to be considered, exponentially increasing the frame allocation combinations. As another example, when the channel is ideal (neither transmission errors nor capture effects), the possible outcomes from an interrogation frame are less than in the case of a realistic channel and, therefore, there are more combinations to explore in the second case.

In practice, $b(n, t)$ can be efficiently computed by caching results in the recursive formula (7). With this strategy, it is possible to analyze the practical range of the parameters for a realistic RFID system in a time-span of several hours. As an example, Fig. 16 shows the time required to perform the following experiment: For a starting time $t = 0$ and $n = 0$, $b(n, t)$ is computed. Then, t is increased (in steps of 0.01 ms) up to 0.2 s. After computing the case for $t = 0.2$ s, n is increased and t is reset back to $t = 0$. This process was repeated until $n = 30$ on a single-core CPU i7-3930K @ 3.20GHz and the figure shows the average execution time per $b(n, t)$ computation (which is in the order of 30 ms for the worst-case experiment).

Also, as a comparative example, we have measured the time required to compute the batch identification probability with Monte Carlo simulation (all results in the paper have been validated using this second procedure). Simulations run until the confidence level is at least 99% for a confidence interval of ± 1% centered on the average. Even if computation time is compared with a dry-run of our analytical-numerical method (that is, starting without cached values), the simulation times are much higher. For example, the computation of $b(n = 9$ tags, $t = 35$ ms) for a realistic-channel ($p_s = 0.55$, $p_c = 0.12$) and DFSA-policy took 22 s with the analytical method and 1200 s with the Monte Carlo simulation. As another example, $b(n = 11$ tags, $t = 40$ ms) for an ideal channel ($p_s = 1$, $p_c = 0$) with DFSA-policy took 3.2 s with the analytical method and 7.6 s with the Monte Carlo simulation. Let us stress that in the Monte Carlo simulation any variation on the input parameters requires a new run of the simulator. Therefore, the analytical procedure, which allows an incremental computation by caching previous results and performs individually better than the Monte Carlo method, has a notable performance advantage.

Besides the batch identification probability, this work describes other two main algorithms, based on the solution of linear programs:

1) Linear program (4), described in Section IV, to determine stabilizing policies given the QRN configuration.
2) Program (14), described in Section VII-C, to compute the maximum joint stable traffic in a QRN. As described, it exhaustively analyzes a range of t values, solving for each of them program (14), which becomes linear in this case, and selecting the best solution.



(a) Disc interference model of radius 100 m

(b) Disc interference model of radius 200 m

(c) Disc interference model of radius 500 m

FIGURE 17. Average algorithm execution times.

Efficient methods (e.g., Karmarkar's algorithm [47]) to solve LPs are known, whose number of iterations is a polynomial function of the number of variables in the LP. In the two algorithms above, the number of variables is determined by the number of valid activation sets, r, which depends on the QRN graph characteristics. For fully connected graphs, the number of activation sets is small. As the network becomes sparse, more activation sets are possible, and algorithm execution times grow. Fig. 17 shows the average execution time for both algorithms for 20 randomly generated networks with a disc interference model of radius 100, 200 and 500 meters. In program (14), t has been analyzed from 0 to 1 s in steps of 50 ms. In the worst case, for networks of 20 readers and a disc interference model of radius 100 meters, execution times are in the order of a few seconds.

Finally, the number of iterations of auxiliary Algorithm 1, which determines the activation sets, is $O(2^m)$, where each iteration execution time is $O(m^2)$. Execution times for this algorithm are also shown in Fig. 17, becoming dominant as the network grows.

IX. CONCLUSION

We have computed the capacity regions for QRNs by developing an exact procedure to obtain the service rate under arbitrary reader setup and channel conditions. Also, our formulation can be used to address several complementary problems, like finding the minimal number of readers to serve a given incoming traffic or determining the maximum joint flow that a QRN can process stably. The results show that QRNs are prone to unbalanced CRs when readers have heterogeneous configurations. Besides, we have also shown that QRNs may suffer from deadlock issues, even in networks without interferences. This happens when readers do not have overlapping t-intervals, which is ultimately caused by the non-linearity of the service rate curve that we have accurately characterized in this work.

REFERENCES

[1] A. Pal and K. Kant, "Internet of perishable logistics: Building smart fresh food supply chain networks," *IEEE Access*, vol. 7, pp. 17675–17695, 2019.
[2] C. Shousong, W. Xiaoguang, and Z. Yuanjun, "Revenue model of supply chain by Internet of Things technology," *IEEE Access*, vol. 7, pp. 4091–4100, 2019.
[3] M. Sidorov, M. T. Ong, R. V. Sridharan, J. Nakamura, R. Ohmura, and J. H. Khor, "Ultralightweight mutual authentication RFID protocol for blockchain enabled supply chains," *IEEE Access*, vol. 7, pp. 7273–7285, 2019.
[4] Q. Tao, C. Gu, Z. Wang, J. Rocchio, W. Hu, and X. Yu, "Big data driven agricultural products supply chain management: A trustworthy scheduling optimization approach," *IEEE Access*, vol. 6, pp. 49990–50002, 2018.
[5] W. Jiang, "An intelligent supply chain information collaboration model based on Internet of Things and big data," *IEEE Access*, vol. 7, pp. 58324–58335, 2019.
[6] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
[7] Z. Meng, Z. Wu, and J. Gray, "RFID-based object-centric data management framework for smart manufacturing applications," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2706–2716, Apr. 2019.
[8] Z. Luo, Q. Zhang, Y. Ma, M. Singh, and F. Adib, "3D backscatter localization for fine-grained robotics," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2019, pp. 765–782.
[9] F. Deng, P. Zuo, K. Wen, X. Wu, and Y. He, "Low delay technology research of transmission line tower monitoring network integrating WSN and RFID," *IEEE Access*, vol. 7, pp. 111065–111073, 2019.
[10] Y. Hou, Y. Wang, and Y. Zheng, "TagBreathe: Monitor breathing with commodity RFID systems," *IEEE Trans. Mobile Comput.*, to be published.
[11] C. Yang, X. Wang, and S. Mao, "Unsupervised detection of apnea using commodity RFID tags with a recurrent variational autoencoder," *IEEE Access*, vol. 7, pp. 67526–67538, 2019.
[12] R. Li, T. Song, N. Capurso, J. Yu, J. Couture, and X. Cheng, "IoT applications on secure smart shopping system," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1945–1954, Dec. 2017.

[13] L. Zheng, C. Song, N. Cao, Z. Li, W. Zhou, J. Chen, and L. Meng, "A new mutual authentication protocol in mobile RFID for smart campus," *IEEE Access*, vol. 6, pp. 60996–61005, 2018.

[14] F. Xiao, Z. Wang, N. Ye, R. Wang, and X.-Y. Li, "One more tag enables fine-grained RFID localization and tracking," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 161–174, Feb. 2018.

[15] M. A. Al-Jarrah, A. Al-Dweik, E. Alsusa, and E. Damiani, "RFID reader localization using hard decisions with error concealment," *IEEE Sensors J.*, vol. 19, no. 17, pp. 7534–7542, Sep. 2019.

[16] P. V. Nikitin and K. V. S. Rao, "Antennas and propagation in UHF RFID systems," in *Proc. IEEE Int. Conf. RFID*, vol. 22, Apr. 2008, pp. 277–288.

[17] H. Seo and C. Lee, "A new GA-based resource allocation scheme for a reader-to-reader interference problem in RFID systems," in *Proc. IEEE Int. Conf. Commun.(ICC)*, May 2010, pp. 1–5.

[18] D.-Y. Kim, B.-J. Jang, H.-G. Yoon, J.-S. Park, and J.-G. Yook, "Effects of reader interference on the RFID interrogation range," in *Proc. Eur. Microw. Conf.*, Oct. 2007, pp. 728–731.

[19] K. S. Leong, M. L. Ng, A. R. Grasso, and P. H. Cole, "Synchronization of RFID readers for dense RFID reader environments," in *Proc. Int. Symp. Appl. Internet Workshops*, Jan. 2006, pp. 4–51.

[20] EPCglobal, *EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID Standard: Specification for RFID Air Interface Protocol for Communications at 860 MHz 960 MHz, Version 2.0.0*, Standard ISO 18000-6C, 2013.

[21] *Electromagnetic Compatibilityand Radio Spectrum Matters (ERM); Radio Frequency Identification Equipment Operating in the Band 865 MHz to 868 MHz With Power Levels up to 2 W; Part 1: Technical Requirements and Methods of Measurement*, document ETSI EN 302 208-1 Version 1.4.1, 2011. [Online]. Available: http://www.etsi.org

[22] F. Nawaz and V. Jeoti, "NFRA-C, neighbor friendly reader to reader anti-collision protocol with counters for dense reader environments," *J. Netw. Comput. Appl.*, vol. 49, pp. 60–67, Mar. 2015.

[23] H. Rezaie and M. Golsorkhtabaramiri, "A fair reader collision avoidance protocol for RFID dense reader environments," *Wireless Netw.*, vol. 24, no. 6, pp. 1953–1964, Aug. 2018.

[24] Z. Li, G. He, and S. Wang, "NFRA-AIC: A RFID reader anti-collision protocol with adaptive interrogation capacity," *IEEE Access*, vol. 7, pp. 86493–86509, 2019.

[25] J. Ho, D. W. Engels, and S. E. Sarma, "HiQ: A hierarchical Q-learning algorithm to solve the reader collision problem," in *Proc. Int. Symp. Appl. Internet Workshops (SAINT)*, Jan. 2006, p. 4.

[26] A.-H. Mohsenian-Rad, V. Shah-Mansouri, V. W. S. Wong, and R. Schober, "Distributed channel selection and randomized interrogation algorithms for large-scale and dense RFID systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1402–1413, Apr. 2010.

[27] V. Deolalikar, J. Recker, M. Mesarina, and S. Pradhan, "Optimal scheduling for networks of RFID readers," in *Embedded and Ubiquitous Computing—EUC Workshops* (Lecture Notes in Computer Science), vol. 3823, T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai, and L. T. Yang, Eds. Berlin, Germany: Springer, 2005, pp. 1025–1035.

[28] J. Vales-Alonso, F. J. Parrado-García, and J. J. Alcaraz, "OSL: An optimization-based scheduler for RFID dense-reader environments," *Ad Hoc Netw.*, vol. 37, pp. 512–525, Feb. 2016.

[29] F. Campioni, S. Choudhury, and F. Al-Turjman, "Scheduling RFID networks in the IoT and smart health era," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 10, pp. 4043–4057, Oct. 2019. doi: 10.1007/s12652-019-01221-5.

[30] M. Golsorkhtabaramiri, N. Issazadehkojidi, N. Pouresfehani, M. Mohammadialamoti, and S. M. Hosseinzadehsadati, "Comparison of energy consumption for reader anti-collision protocols in dense RFID networks," *Wireless Netw.*, vol. 25, no. 5, pp. 2393–2406, Jul. 2019. doi: 10.1007/s11276-018-1670-y.

[31] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[32] H. Vogt, "Efficient object identification with passive RFID tags," in *Pervasive Computing* (Lecture Notes in Computer Science), vol. 2414. Berlin, Germany: Springer, 2002, pp. 98–113.

[33] J. Yu and L. Chen, "Stability analysis of frame slotted aloha protocol," *IEEE Trans. Mobile Comput.*, vol. 16, no. 5, pp. 1462–1474, May 2017.

[34] C. Qian, Y. Liu, R. H. Ngan, and L. Ni, "ASAP: Scalable collision arbitration for large RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1277–1288, Jul. 2013.

[35] X. Jia, Q. Feng, and L. Yu, "Stability analysis of an efficient anti-collision protocol for RFID tag identification," *IEEE Trans. Commun.*, vol. 60, no. 8, pp. 2285–2294, Aug. 2012.

[36] J. Yu, W. Gong, J. Liu, L. Chen, and K. Wang, "On efficient tree-based tag search in large-scale RFID systems," *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 42–55, Feb. 2019.

[37] Y. Yu, X. Cai, and G. B. Giannakis, "On the instability of slotted aloha with capture," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 257–261, Feb. 2006.

[38] X. Gao, P. Wang, D. Niyato, K. Yang, and J. An, "Auction-based time scheduling for backscatter-aided RF-powered cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1684–1697, Mar. 2019.

[39] A. M. Ibrahim, O. Ercetin, and T. ElBatt, "Stability analysis of slotted aloha with opportunistic RF energy harvesting," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1477–1490, May 2016.

[40] J. J. Alcaraz, E. Egea-López, J. Vales-Alonso, and J. García-Haro, "Dynamic system model for optimal configuration of mobile RFID systems," *Comput. Netw.*, vol. 55, no. 1, pp. 74–83, Jan. 2011.

[41] J. J. Alcaraz, J. Vales-Alonso, and J. García-Haro, "RFID reader scheduling for reliable identification of moving tags," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 816–828, Jul. 2013.

[42] S. Tang, J. Yuan, X. Li, G. Chen, Y. Liu, and J. Zhao, "RASPberry: A stable reader activation scheduling protocol in multi-reader RFID systems," in *Proc. 17th IEEE Int. Conf. Netw. Protocols*, Oct. 2009, pp. 304–313.

[43] J. An, K. Yang, J. Wu, N. Ye, S. Guo, and Z. Liao, "Achieving sustainable ultra-dense heterogeneous networks for 5G," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 84–90, Dec. 2017.

[44] Q. Dong, A. Shukla, V. Shrivastava, D. Agrawal, S. Banerjee, and K. Kar, "Load balancing in large-scale RFID systems," *Comput. Netw.*, vol. 52, no. 9, pp. 1782–1796, Jun. 2008.

[45] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures Commun. Netw.*, vol. 3, no. 1, pp. 1–211, 2010.

[46] P. R. Kumar and S. P. Meyn, "Stability of queueing networks and scheduling policies," *IEEE Trans. Autom. Control*, vol. 40, no. 2, pp. 251–260, Feb. 1995.

[47] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proc. 16th Annu. ACM Symp. Theory Comput.*, 1984, pp. 302–311.

[48] C. Floerkemeier and S. Sarma, "RFIDSim—A physical and logical layer simulation engine for passive RFID," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 33–43, Jan. 2009.

[49] J. Vales-Alonso, V. Bueno-Delgado, E. Egea-Lopez, F. J. Gonzalez-Castaño, and J. Alcaraz, "Multiframe maximum-likelihood tag estimation for RFID anticollision protocols," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 487–496, Aug. 2011.

[50] P. Šolic, J. Maras, J. Radić, and Z. Blažević, "Comparing theoretical and experimental results in Gen2 RFID throughput," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 349–357, Jan. 2017.

[51] B. Li, Y. Yang, and J. Wang, "Anti-collision issue analysis in Gen2 protocol: Anti-collision issue analysis considering capture effect," Auto-ID Labs White Paper WP-HARDWARE-047, Mar. 2009. [Online]. Available: http://www.autoidlabs.org/single-view/dir/article/6/320/page.html

[52] I.-H. Chung, M.-C. Yen, and C.-K. Hwang, "An accurate analytical formula for the essential joint probability of framed slotted aloha protocols," *J. Franklin Inst.*, vol. 350, no. 10, pp. 3432–3440, Dec. 2013.

JAVIER VALES-ALONSO received the degree in telecommunication engineering from the Universidad de Vigo, Spain, in 2000, the M.Sc. degree in mathematics from the Universidad Nacional de Educación a Distancia, Spain, in 2005, and the Ph.D. degree in computer science from the Universidad Politécnica de Cartagena (UPCT), Spain, in 2015, where he is currently a Full Professor with the Department of Information and Communication Technologies. He is involved in different research topics related to modeling and optimization.

PABLO LÓPEZ-MATENCIO received the degree in telecommunication engineering from the Polytechnic University of Madrid, in 1994, and the Ph.D. degree from the Polytechnic University of Cartagena (UPCT), Spain, in 2012. He has been involved in relevant projects at Vodafone and Ono companies, related to network infrastructure, call-centers, and management of information systems. He is currently an Associate Professor with UPCT, where he has participated in several TIC projects. His research interests include in the areas of wireless sensor networks and their applications to develop ambient intelligence environments.

JUAN J. ALCARAZ received the engineering degree from the Polytechnical University of Valencia, in 1999, and the Ph.D. degree from the Technical University of Cartagena (UPCT), in 2007. After working for several telecommunication companies, he joined UPCT, in 2004. He was a Fulbright Visiting Scholar with the Electrical Engineering Department, UCLA, in 2013, and a Visiting Professor with the Department of Information Engineering, University of Padova, in 2017. He is currently an Associate Professor with the Department of Information and Communication Technologies, UPCT. His current research interest includes learning algorithms for wireless network management.

FRANCISCO JAVIER GONZÁLEZ-CASTAÑO received the Ingeniero de Telecomunicación degree from the University of Santiago de Compostela, Spain, in 1990, and the Doctor Ingeniero de Telecomunicación degree from the University of Vigo, Spain, in 1998. He is currently a Full Professor with the Telematics Engineering Department, University of Vigo, where he leads the Information Technologies Group. He has authored more than 100 articles in international journals, in the fields of telecommunications and computer science, and has participated in several relevant national and international projects. He holds two U.S. patents.