SafeRFID Project: A Complete Framework for the Improvement of UHF RFID System Dependability

Vincent Beroulle, Oum-El-Kheir Aktouf, David Hély
Univ. Grenoble Alpes, Grenoble INP*, LCIS, F-26000 Valence, France
* Institute of Engineering Univ. Grenoble Alpes
e-mails: vincent.beroulle@grenoble-inp.fr;oum-el-kheir.aktouf@grenoble-inp.fr;david.hely@grenoble-inp.fr

Abstract— **The SafeRFID project targets the improvement of Ultra High Frequency Radio Frequency Identification (UHF RFID) system dependability using system level simulation and emulation. RFID systems are based on low cost components (tags) more and more often used in critical applications and running in harsh environments (railway, aeronautic, food production, product manufacturing). Defects can have different origins (1) hardware failures, (2) medium perturbations (electromagnetic interferences), or (3) software bugs. The main goals of this project are (1) to develop hardware and software validation environments to validate and evaluate new methods for detecting and diagnosing defects within RFID systems, (2) to develop new middleware services to improve the performances of RFID systems in presence of defects and (3) to develop robust tag architectures. This paper sums up all these complementary solutions, which have been validated thanks to system level simulation and emulation and, which have been integrated in a global dependable UHF RFID system. The results of this work are (1) the design of a robust middleware, (2) the design of a robust hardware tag and (3) the evaluation of the dependability of such global RFID systems thanks to system level simulation and emulation.**

Keywords*— *RFID; system level simulation; fault injection and simulation; on-line test; diagnosis.

I. Introduction

In critical domains, RFID system errors can have catastrophic consequences in terms of human safety whereas in high quality applications, they can have economic consequences for product quality, manufacturing costs, etc. Monitoring RFID systems, which are based on low cost and uncertain components, is thus a must in order to perform on-line detection of failures. These failures can result from hardware malfunctions (aging effects are particularly sensitive to harsh environments), medium disturbances (for example, electromagnetic bursts), or software bugs. These failures can be due to a broken or a misplaced antenna, RF interferences, low signal strength, hardware defect in the tag chip, middleware dysfunctions, etc. Therefore, the main goal of the SafeRFID project is to propose a global strategy for the simulation of RFID system in order to develop and evaluate the on-line detection and diagnosis of defects in UHF RFID systems in order to enhance the RFID systems dependability. This paper is an extended version of [1], and gathers all the most important results of the SafeRFID project.

The objectives of existing RFID middlewares are especially to manage various data sources in RFID systems and process large amounts of raw data. Some of them also provide error fixing mechanisms, mainly by using basic on-line monitoring approaches, such as WinRFID [2]. Other RFID middlewares focus on a reliable integration of RFID technology into existing applications (SunRFID [3], FlexRFID [4]). Fault-tolerance is taken into account in the RFID middleware RF2ID [5] by detecting abnormal behavior of the system and introducing the concept of Virtual Reader, that is a group of physical readers determined for fault-tolerance purposes. However in this middleware no low level information (physical information) coming from each reader measurements are mixed with the high level information gathered by the numerous readers in the system.

The classical RFID system on-line monitoring methods are based on reader performance monitoring. In fact, to detect component or environment failures and defects, many performance parameters of the reader can be observed. The classical performance parameters observed are the Average Tag Traffic Volume (ATTV) and the Read Errors to Total Reads (RETR) [6]. ATTV allows determining unusual tag traffic, which is a symptom of a faulty system. For instance, if between 8:00am and 11:00am a reader usually reads 100 tags/hour every day and if one day, during the same period, the same reader reads only 50 tags/hour, then it can be assumed that a failure or a disturbance has occurred. The second parameter RETR consists of counting erroneous reads over the total read attempts (correct and faulty) of a specific reader. High RETR means there is probably a problem. The evolution of this RETR can also be analyzed. These methods can also be used as final optimization approaches during RFID system deployment.

In order to validate RFID systems during design phases, several RFID simulators have been proposed in the literature [7]-9], but none of them focuses on the RFID system dependability evaluation. These simulators allow simulating the communication protocol between the tags and readers or the interactions between the readers and middleware. Thus, designers generally use these simulators to perform a functional verification of their systems. For instance, Rifidi [1] only tackles RFID system deployment issues; fault simulation with Rifidi would be unrealistic. RFIDSim [8] is a complete RFID simulator; nevertheless its main goal is to evaluate RFID protocols and tag hardware characteristics are not modelled.

The SafeRFID project integrates in the same RFID system complementary and multi-level solutions for improving the overall system dependability. These solutions target the improvement (1) of the tags hardware architecture, (2) of the readers fault detection capability and (3) of the middleware for multi-readers RFID systems fault diagnosis. In this context, our three main results are: (1) two new validation environments, a simulator and a FPGA-based emulation platform allowing hardware and software RFID systems co-design and fault simulation; (2) new on-line test and diagnostic services for RFID middleware, and (3) a new tag robust architecture.

The next sections of this article are organized as described in the following. In Section II, two new RFID validation environments are described. The first one is a system level simulator, which is capable of performing fault injection and simulation. The second one is an emulation platform (based on FPGA), which is also capable of both performing hardware fault injections and monitoring its internal signals. Section III presents two test and diagnosis methods, which have been implemented and validated thanks to these simulators or emulators. This section also describes the robust tag architecture developed within the SafeRFID project as well as the proposed RFID middleware. Section IV concludes the article.

II. Validation Environments

This section describes the two validation environments, which have been developed for the purposes of the SafeRFID project. These two environments allow (1) the validation of software and hardware RFID components and (2) the evaluation and the improvement of RFID system robustness using fault injection. The first validation environment, called SERFID, is a complete RFID system level simulator. The second one, called RFIM, is a RFID emulation platform allowing modelling and evaluating tag Integrated Circuit (IC) digital architectures into actual RFID systems. These two validation environments are compliant with the RFID UHF EPC C1 Gen2 standard [10].

A. *SERFID Simulator: a virtual validation environment*

SERFID is a UHF RFID system simulator. It permits to evaluate RFID systems robustness by means of fault injection and simulation. It models the whole RFID system including the numerous hardware tags and readers and their electromagnetic environment. SERFID can be interfaced with an RFID middleware. SERFID allows validating and optimizing middleware implementation. Figure 1 illustrates a SERFID high level view containing several readers and tags.

SERFID has been developed using the C++ SystemC library, which is adapted to both hardware and software component modeling. SERFID consists in 20,000 lines of C++ code. A RFID system modeling is made possible using the configurable tag and reader C++ components. For example, each tag identification number and location can be easily modified. The number of readers and their locations can also be easily modified. The C++ code of SERFID is an open source code. Thus, each component model can be improved.

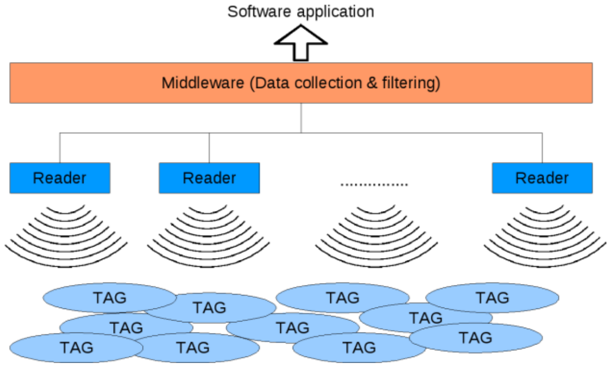

Figure 1. SERFID high level view architecture including several readers and tags,with a connected middleware managing the reader data for the final software application

SERFID allows the middleware co-design and co-verification using realistic data coming from simulated tags and readers. Figure 2 illustrates how a middleware can be connected to SERFID, which models a real RFID system with numerous tags and readers including some perturbations. SERFID can simulate numerous test cases including ones with faulty tags or readers. The middleware is placed between SERFID and the final software business application. It manages the high number of data coming from the RFID system to simplify the work of the application.

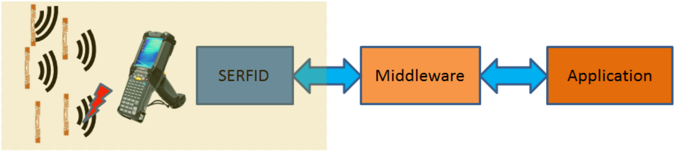

Figure 2. SERFID connection to a middleware for the middleware co-design and co-verification

SERFID also allows Failure Mode Effect Analysis (FMEA) of RFID systems. FMEA permits to evaluate the robustness of a RFID system in presence of perturbations. This analysis is automated by SERFID using fault injection and fault simulation. SERFID models the communication links between each tag and reader using high level functional models (Timed Transaction Level Model). Figure 3 illustrates a simple RFID system consisting in one tag and one reader only (of course more tags and readers could be added). As we have previously said, SERFID component models are high level models. For example, the delay of each computation is modeled with a fixed duration depending on the operation (the minimal and maximal times of each operation are given in the EPC C1 Gen2 standard). In addition, SERFID models the most important RFID physical effects, which are: message collisions, tag remote powering, and tag masking. Message collisions happen when two tags simultaneously emit a message. Both of these messages cannot generally be read by the reader. However, if one of these two messages is highly more powerful than the other, then it can be read by the reader, and the other tag message is masked. This is called the masking effect and SERFID takes it into account.

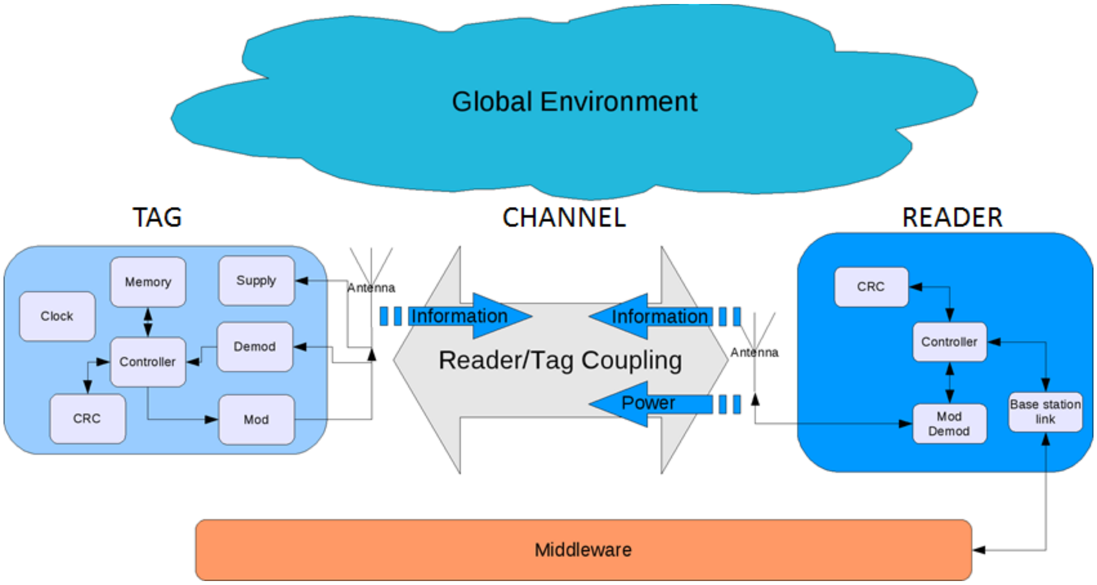

Figure 3. SERFID architecture of a simple RFID system including one tag, one reader, one channel and a global environment for the storage of global parameters

The power of each message depends only on the tag to reader distance and is computed with the Friss transmission equation. The remote powering modeling consists of adding all electromagnetic power emitted by all readers depending on their distance to the tags. The fault injection and simulation functionalities consider three different fault models: channel inactivation, no communication, and Bit-Error-Rate (BER) variation. Channel inactivation means that no power and no information are exchanged into a given channel during a specific period. No communication model means that no information is exchanged into a given channel during a specific period (but power is still emitted). BER involves the injection of error in the exchanged bits. These bit error injections can be done with different random models (uniform, burst, etc).

In order to illustrate the use of SERFID, we describe in the following a real case study. This case study is inspired from a classical RFID application in a warehouse context. In this context, the goal of the RFID system is to identify the boxes (more than 100 boxes) arranged within a pallet. As illustrated in Figure 4, this pallet is rotating between two RFID reader antennas. This environment is highly disturbed due to the numerous reflections of the electromagnetic waves on the products into the boxes. The rotation of the pallet helps for the tag detections.

This harsh environment requires the use of a robust inventory approach in order to detect all the tags in a limited amount of time. Optimizing the parameters of this robust inventory can be done with SERFID.

In Figure 5, we compare the inventory results achieved by a real RFID system with the inventory results obtained with the SERFID simulation. Of course the two inventory read rate curves are not exactly the same. Indeed, an accurate modeling of a so complex electromagnetic environment is not possible (or would be very time consuming). However, the shapes of the two inventory read rate curves are nearly the same and the inventory duration estimation is quite good (160s in the real system versus 176s in the simulated one). The SERFID model is enough accurate to allow optimizing the inventory parameters and all the middleware design parameters.

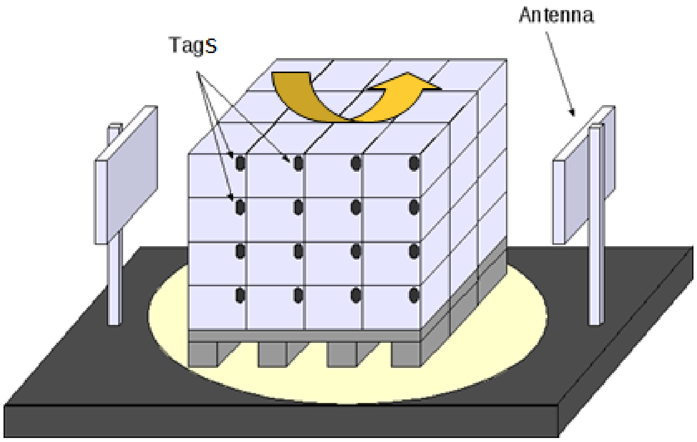

Figure 4. Example of the inventory of boxes under a rotating pallet thanks to an UHF RFID system

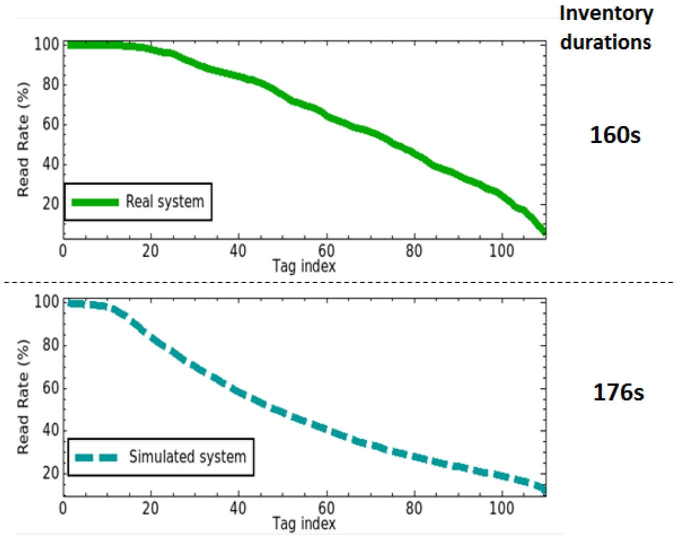

Figure 5. Real RFID system inventory read rate curve (top) vs. SERFID simulated inventory read rate curve (bottom)

More details on this simulator are given in [11].

B. RFIM: an emulation platform

The digital baseband of the tag itself is a very important element concerning the safety and security chain for the whole RFID system. It imports then to study the tag itself in order to (1) explore its best architecture compliant with the standard in terms of safety and security and (2) to analyze the effect of a faulty tag on the rest of the system. A deep study of different digital baseband architectures considering all the possible interactions with the complete RFID system is not a trivial task due to the complexity and the heterogeneity of this system. Nevertheless, the validation of the tag itself should be done considering all the interactions of the tag and the RFID system. While digital design requires cycle accurate simulation it becomes unpractical for large systems involving hardware and software levels and a multitude of devices. Also, it is necessary to provide IC designer a tool which allows a quick validation of the circuit under design in order to avoid costly design respins. It has then been decided to develop a hardware emulation platform dedicated to RFID transponder dependability and security study.

Emulation permits to evaluate the UHF RFID tag within its real environment considering interferences between tags themselves and interaction with the upper layer of the system from reader to middleware. Indeed, a minor tag modification can have multiple incidences on system parameters such as inventory time or other. Moreover the emulator is very flexible to explore different digital architectures while ensuring compliance with the UHF standard. As depicted in Figure 6 below, the RFID emulator can be used within an RFID environment including reader and other transponders or even other emulators. This way, the emulator is placed within a real RFID environment which allows accurately analyzing many hard to simulate system effects.

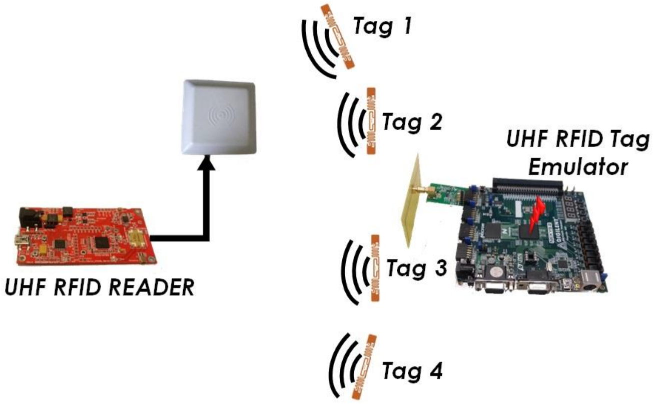

Figure 6. Emulation based digital baseband validation

Thanks to the in-system validation capabilities inherent to the emulation, the proposed platform offers many opportunities. The emulation platform has been enhanced in order to be able to monitor and to control in real time internal states of the digital baseband. It is thus possible to perform fault injection within the digital baseband. Emulation allows bit level fault injection such as single event upset (SEU) or multi event upset (MEU). It has been experimentally shown in [12] that this fault model is realistic with the failure types of RFID tag IC. While in-system validation allows identifying the most critical faults from a system point of view, observing capabilities helps to understand fault propagation in order to finely tune mitigation techniques reducing the cost of the hardening. The emulation based platform which has been developed is depicted in Figure 7. This platform embeds a digital baseband fully compliant with the UHF EPC C1 Gen2 protocol. As shown in Figure 7, the RFIM platform is divided into eight modules: monitoring interface, fault injector, activation of injection, event detector, golden and instrumented faulty tags, register comparator and embedded microprocessor. The embedded microprocessor controls all the platform modules and then permits to perform on-line tag monitoring and to play on-line fault attacks. The processor allows the on-line capture of data in the two tag basebands for analyzing the RFID communication. The interface monitoring is a mechanism that transports the internal register values from the tag basebands to the microprocessor. This monitoring interface block uses a First-In-First-Out (FIFO) memory in order to compensate the latency of the microprocessor for outputting register values. Faults are only injected in the faulty tag. The golden tag, which is always fault free served as a reference. The register comparator compares all the internal registers of the golden and the faulty tags. This comparison helps the embedded processor to detect and to localize faults and errors in the faulty tags.

RFIM allows quick and accurate validation taking into account all the complex physical effects involved into RFID systems. Also, RFIM can be used in order to tackle security issues of RFID systems.

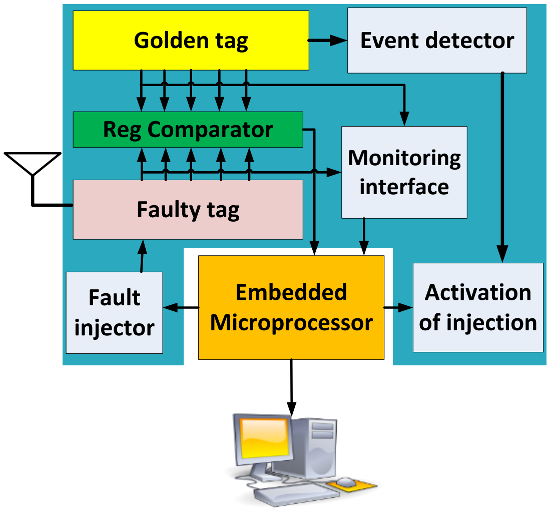

Figure 7. RFIM platform for faults injection and monitoring

First the emulator can be configured to process attacks coming from the tag against the RFID system. Then system level countermeasure can be developed and validated against real case attacks. As an example, RFIM can be used to evaluate the Hardware Trojan (HT) threats against RFID system and to validate appropriate system countermeasure. HT threats are malicious modifications of the circuit (such as backdoor) which can later be used when the circuit is in mission mode. RFIM allows then to emulate HT to attack the RFID system using off the shelf reader and middleware in order to identify weakest points which can be further secure. Moreover, the EPC protocol offers room for cryptographic based security. Nevertheless, one main limitation of such security is the inherent cost in time. So, using the emulator, cryptographic add-on of the EPC protocol can be validated considering the whole chain and then finely evaluate the associated cost such as the time overhead for a given inventory.

III. Test and Diagnosis Methods and Tag Robustness Enhancement

This section describes the three main approaches which have been proposed by the SafeRFID project in order to improve UHF RFID system dependability. Each approach is embedded on a specific part of the RFID system: the reader, the middleware and the tag digital architecture. These approaches have been validated by simulation or emulation using the two previously described platforms and validated by experimentations.

A. *Profile test method*

The Profile test (PT) method is inspired by classical monitoring techniques (ATTV, RETR), which are based on reader performance monitoring. This method, as the classical monitoring methods are, is nonintrusive. In this method, we propose to measure and compare individual tag performance indicators rather than a single global average parameter. To this end, we define a new performance metric - called read rate profile – individually involving all the tags of the population rather than an average value computed for the same population.

The initialization of our monitoring method requires computing the statistical parameters of the fault free inventory read rate profiles. Let us first explain what these inventory read rate profiles are. Each tag inventory leads to a specific inventory read rate profile, which is the ordered read rate curve of the entire tag population. The '-'curve in Figure 8 represents the inventory profile of a fault free inventory occurrence. Then, with numerous inventory profiles, an average read rate profile is computed. This average profile is represented by the bold curve in Figure 8.

The second step for the initialization of our approach consists in computing a threshold for the failure detection. This threshold, called limit profile, is represented by the '+'curve in Figure 8.

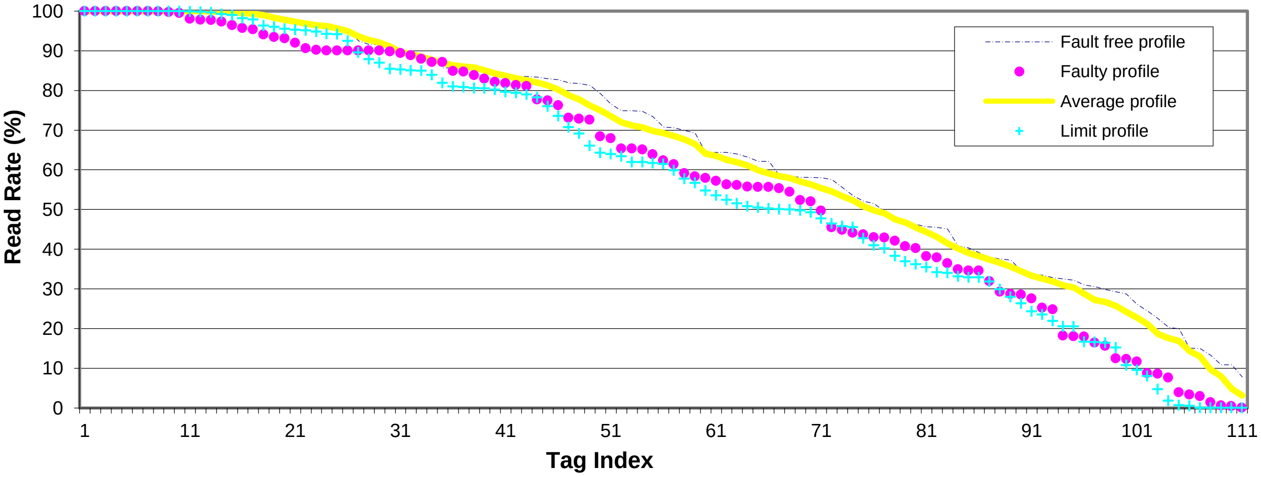

Figure 8. Average, limit, fault free and faulty inventory profiles

An inventory profile with one or more tag read rates under this limit implies that the RFID system is considered faulty. The '•'curve in Figure 8 illustrates a faulty inventory profile with several points under the limit. The limit profile is computed using the average profile and the standard deviation of each ordered tags. The evaluation of this approach has been realized experimentally and by simulation. In both cases the detection results have been compared with the detection results of the RETR and ATTV classical approaches (described in the introduction).

1) *Experimental validation.* It is highly difficult to validate the PT approach on real RFID systems. Indeed, it's not trivial to control the fault injection into these real systems and in particular to be sure that each fault has been correclty injected. Thus, we only did this experimental validation on a few different scenarios. We use the same RFID application than the one previously described in Section II.A. Some faults are injected in this application to generate system faulty behaviors. These faults are injected in the communication channel only (no fault has been injected into the tag or the reader hardware nor into the software components). The 3 different fault injection techniques are:

- The rotation of 5 random tags on 5 different boxes
- The displacement of 5, 15, and 20 random tags on the surface of the boxes
- The pallet rotation stop during 15s and 20s

Using these fault injection techniques, a total of 9 faulty system behaviors are generated. The RETR approach does not detect these faulty system behaviors. The ATTV approach detects 3 faulty system behaviors over the 9 faulty behaviors. The PT approach detects 4 faulty behaviors, and among these 4 faulty behaviors 3 were not detected by the previous approaches. By conjointly using the PT and the ATTV approaches, it is then possible to detect 6 faulty behaviors over the 9 possible faulty behaviors. Finally the PT approach detects more faults than the classical approaches but this approach must be used with classical approaches to detect the maximum number of faults.

2) *Evaluation with SERFID simulation.* More scenarios can be evaluated thanks to SERFID simulation. In the following this evaluation is done using the 2 simple following fault models:

- 40% Read Rate decrease of 5 random tags
- 10% Read Rate decrease of 20 random tags

Each of these faults are injected and simulated 100 times to obtain statistical representative results. Table I gives the detection results achieved by the classical approaches and by the PT approach.

TABLE I. EVALUATION OF CLASSICAL ON-LINE TEST APPROACHES AND OF PROFILE TEST (PT) APPROACH BY SERFID SIMULATION

	5 random faulty tags with Read Rate decreased of 40%	**5 random faulty tags with Read Rate decreased of 10%**
ATTV	35%	11%
RETR	4%	5%
PT	**63%**	**92%**

Table I shows that the PT approach detects more faults than the classical approaches ATTV and RETR. As in the experimental validation achieved on an actual system, SERFID simulation of the ATTV approach shows that more faults are detected than with the RETR approach. Once again the results show that the RETR and PT approaches are two complementary approaches. They have to be combined to achieve the best fault detection. In addition, the read rates of the tags which are impacted by the fault injection impact the performance of the PT approach. If the impacted tags have high read rates then the modification of the profile curve is more important than if the impacted tags have low read rates. Both Figures 9 and 10 show how the read rate values of the impacted tags modify the profile curve. In Figure 9 the impacted read rates have high values (the lowest one is 60%), and the fault injection drives to modify the profile curve to achieve a fault detection (at 2 different locations corresponding to the 2 red circles).

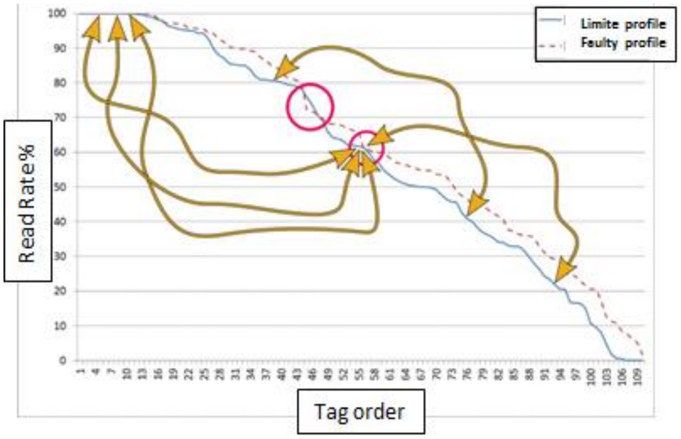

Figure 9. Simulation of the profile curve modifications when faulty tags have high read rates; the fault injection is detected at two different locations (two red circles)

Then, the next figure shows the case when the impacted read rates have low values. In this case the profile curve is not highly modified and no detection is achieved.

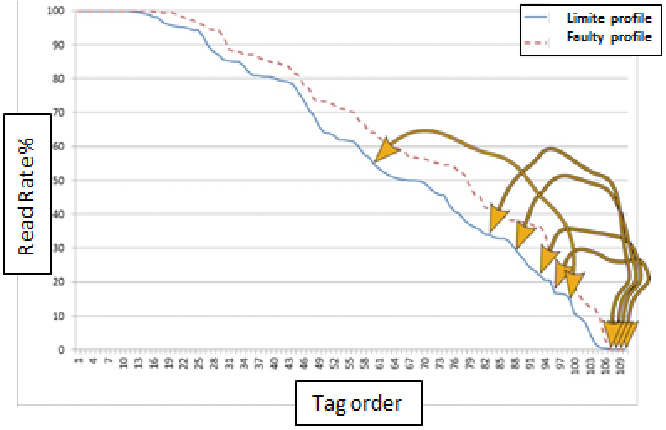

Figure 10. Simulation of the profile curve modification when faulty tags have low read rates; no detection of the fault injection

Details on this test approach are given in [13][14] [15].

B. *SafeRFID-MW: a Middleware for On-Line Testing and Diagnosis*

The proposed PT approach, as well as most existing test approaches (RETR, ATTV, etc.), operate mainly at the reader level. Obtained local results are not capitalized for global processing of errors at the whole system level. Consequently, in case of distributed RFID systems involving several readers, there are no means to determine the whole system state. Furthermore, in case of a faulty behavior, it is not easy to locate the origin of the observed failure: does it originate from the readers, the groups of read tags or the communications involved in the system?

In our work, this issue has been tackled by developing a dedicated RFID middleware that integrates not only testing operations at the level of each reader, but also a diagnosis process at the middleware level. By positioning this part of our study of RFID systems at the middleware level, the simultaneous observations of many reading results are made possible. Consequently, analysis of these results can help producing a sharpened diagnosis and more accurately locate the likely causes of a failure. To this end, we applied the comparison of inventory results of several readers as it is done in the RF2ID middleware [8]. However, in our approach, comparisons are carried out among physical readers that read the same tag groups, providing inherent redundancy.

The random nature of the tag-reader interaction has directed our research towards the probabilistic diagnosis approach [16] whose basic idea is to associate a probability of failure to each element in the system as well as a fault coverage for each performed test. The user can then put a justified confidence in the obtained system diagnosis results. Our middleware called SafeRFID-MW implements a diagnosis algorithm called RFID_Diag_Algo. This algorithm uses the basic idea of probabilistic diagnosis developed in the work of Fussel and Rangarajan on multiprocessor systems [16]. Nevertheless, the fault models, as well as the diagnosis operations, have been largely adapted to the RFID features. As a result, the diagnosis process we developed, takes place in two main phases. The first phase consists of running the RFID_Diag_Algo algorithm. This one performs its operations in three steps: i) reader partitioning in groups according to some criteria issued by the application (i.e., which readers, read the same groups of tags), ii) read rate results comparison in a way that ensures a consensus on faulty components, whether readers or tags, iii) evaluation of the diagnosis accuracy by applying a new probabilistic model suitable to such systems. The second phase is executed for each identified faulty reader. It is based on the analysis of the communication logs between the faulty readers and the middleware to identify the precise cause of the observed failure. Such information is not provided by the LLRP protocol and is therefore an innovative aspect of our work. The rest of this section provides more details on these two phases.

1) Description of the first phase (RFID_Diag_Algo): global probabilistic diagnosis. During this phase, RFID readers are partitioned into groups according to the actual paths of the tags through the various readers of the system. Thus, obtained groups include readers that process the same groups of tags (see Figure 11). For the example of Figure 11, we can observe that tags belonging to groups g_1 and g_2 are read by readers in the set (R_1, R_2, R_3, R_4, R_5) whereas tags in the group g_3 are read by readers in the set (R_1, R_2, R_3, R_4, R_6, R_7).

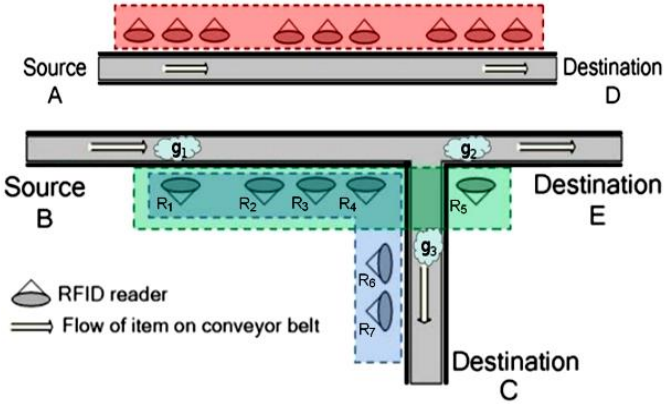

Figure. 11. Grouping of readers according to the data flow

Once tag groups are read by the corresponding readers, the diagnosis process can start. At this point, our algorithm RFID_Diag_Algo applies one of the aforementioned monitoring approaches. This could be ATTV, RETR or the PT approach. The exact applied monitoring approach is not of importance here, as only the obtained monitoring results matter. Table II shows different inventories collected for a set of readers (R_1, R_2, ..., R_7) and 3 groups of tags g_1, g_2 and g_3. The value "1" indicates that the reader has ensured a correct inventory of all tags while the value "0" indicates a failure of that inventory. When a reader is not concerned by a group of tags, the corresponding value is the "-" symbol.

The analysis of these results is done by applying the principle of majority voting in the following two cases, using a performance parameter that denotes the accuracy of each reader results and whose calculation is presented in [17][18]:

1. If the majority of readers meet the considered performance parameter, then the rest of the readers (i.e., the minority) are considered faulty; i.e., the minority that shows poor performance is considered faulty.
2. If the majority of readers do not meet the performance parameter set for the given group of tags, so this group of tags, as well as the other readers, are considered faulty.

TABLE II. COMPARISON OF TAG INVENTORY RESULTS

		Tag groups		
		g_1	g_2	g_3
RFID readers	R_1	1	0	1
	R_2	0	1	1
	R_3	1	0	1
	R_4	1	0	1
	R_5	1	0	-
	R_6	-	-	1
	R_7	-	-	0
	$R(g_i)$	$\{R_2\}$	$\{R_2, g_2\}$	$\{R_7\}$

By applying this analysis, we obtain, for each group of tags g_i, a set of likely faulty readers and/or tags, as shown on the row labelled $R(g_i)$ on Table II.

This phase of probabilistic diagnosis ends up with the determination of a confidence parameter corresponding to the quality of the diagnosis and hence, the trust level that the user can have. At this stage, it is necessary to ensure the following two points:

- A valid reader must be identified as valid (this case is called "correct negative" and noted CN).
- A faulty reader must be identified as failing. To simplify probabilistic calculations, we consider the complementary case, that is to say the case where a faulty reader is considered correct (this case is called a "false negative" and noted FN)

We define the identifiability as being the probability of correctly identifying the state of diagnosed readers. This measure indicates the ability of the diagnosis process to "distinguish" the faulty readers from those who are not. This probability is the diagnosis accuracy and its calculation is detailed in [15][16]. To give a little insight within this process, let us consider again the example of Table II. We may simply state that reader R_2 is determined twice as being faulty whereas reader R_7 is determined only once as being faulty. So, in the process of minimizing fault positives, the objective of the diagnosis calculation is to determine more precisely the probability of each reader of being actually faulty. This may lead to consider that R_7 is actually fault-free.

2) Description of the second phase: Diagnosis of a faulty reader. After faulty readers have been identified, an additional study allowed us to pinpoint the causes of the observed failures. To this end, we analyzed the communication between the middleware and the RFID readers based on the LLRP protocol. Although LLRP is a complete and complex communication protocol that allows notifying the communication errors between the middleware and the readers, it can neither detect reader failures that are due to some misconfigurations, nor determine the causes of an observed failure. Therefore, it is not suitable as is for use in applications where dependability demands are critical, especially since the tag-reader interface is very sensitive to external disturbances and thus features a very random behavior. Furthermore, the functioning of the LLRP protocol is flexible and provides a wide autonomy to the application to specify the inventory operations and access to tags. This can lead to configuration errors resulting in a faulty behavior of the readers (for example, the reader cannot identify all tags in its reading range, the reader does not find the correct information on the tags, etc.).

Our work on the LLRP protocol mainly allows overcoming these limitations. The study of the LLRP protocol led to its modeling as a finite state machine. Let G denotes the finite state machine of the LLRP protocol. $G=(S, I, O, \delta, \lambda)$; where I, O and S are respectively a finite set of input symbols, a finite set of output symbols and a finite set of states.

- $\lambda: S\times I \rightarrow S$ is the state transition function.
- $\delta: S\times I \rightarrow O$ is the output function.

When an RFID reader or the middleware is in a state s $\in$ S and receives input i $\in$ I it produces a specified output $o=\lambda(s, i)\in O$ and transits to a state $s = \delta(s, i)\in S$. Details of this FSM are provided in [17]. For design or configuration mistakes, the faulty behavior is associated with an inconsistent state of the reader or the middleware. Indeed, the entity (reader or middleware) that is in an inconsistent state does not correctly interpret the received data and then adopts an inappropriate behavior. To tackle this type of mistakes, we applied to the finite state machine of the LLRP protocol standard techniques of model-based testing. More precisely, we used the distinguishing sequences approach (Distinguishing sequences) [18]. The application of this technique allows to simply retrieving the state in which the system was at the time of the failure occurrence [18]. However, this technique has some limitations as we cannot determine a distinctive sequence to all the states represented within the FSM.

We also analyzed failures that are due to the runtime environment, such as: slow execution, no data capture, etc. Such faults cannot be related directly to a system state, since they are mainly due to the execution environment. Thus, it is not possible to simply apply the above approach. We therefore proposed an extension of the state machine to include the causes of this category of failures in the diagnosis process in the form of an extended LLRP model [19][20]. The extensions made to the LLRP protocol to determine the exact or likely causes of observed failures are, to the best of our knowledge, new features to RFID middlewares.

C. *Tag Robustness Enhancement*

Thanks to RFIM, the most sensitive parts of the tag digital baseband architecture have been identified through fault injection campaigns [21]. The fault injection campaigns consist in measuring for a given time period the number of times the tag is detected by a reader while faults are injected in a part under analysis of the tag digital baseband. This experiment has also been done when several tags are in front of the reader in order to evaluate the faulty tag effects on other tags. The experiments have been carried out on all the functional registers (i.e., the registers storing parameter values dedicated to the communication between tag and reader) of the digital baseband in order to identify the most sensitive ones. Experimental results [22] show that only a few registers dramatically decrease the system performance (i.e., the tag read rate). Figure 12 hereafter gives the influence of the fault injection on the number of times the tag has been successfully identified. Light gray gives the value in case no faults are injected; dark gray gives the resulting number in case faults have been injected within the parameters given in horizontal axis.

At a first glance, we can see that all parameters are not equally sensitive. While some faulty parameter registers reduce the number of times the tag has been identified from 4500 to less than 500, other ones have a very limited influence on the tag response. This can be explained by the role played by the parameter during an inventory round, and the refreshment rate of the value during the same round.

We have proposed in a first approach to use hardware redundancy to decrease the fault effects. A Triple Modular Redundancy (TMR) has been applied on the most sensitive registers identified thanks to the previous fault injection campaigns. Since the tag digital baseband architecture is powered wirelessly and has a limited resource, the TMR was chosen to protect the most sensitive registers only. Moreover such registers are very small, which makes the cost acceptable. As shown in Figure 13, the TMR technique consists on the triplication of the target component to be protected. The three resulting outputs from triplication are connected to a voter block that compares the three received data and elects the data with the majority.

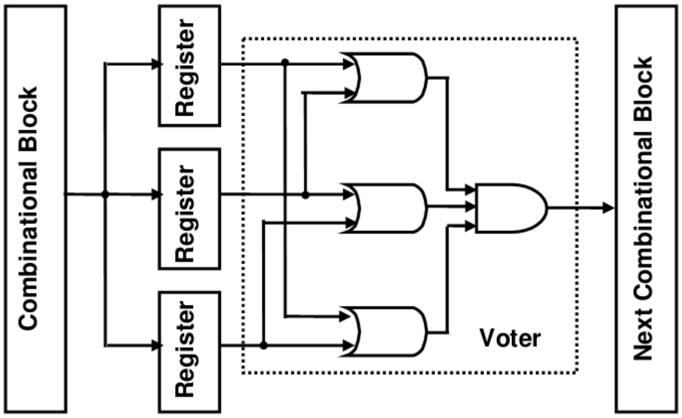

Figure 13. Triple Modular Redundancy Protection

.

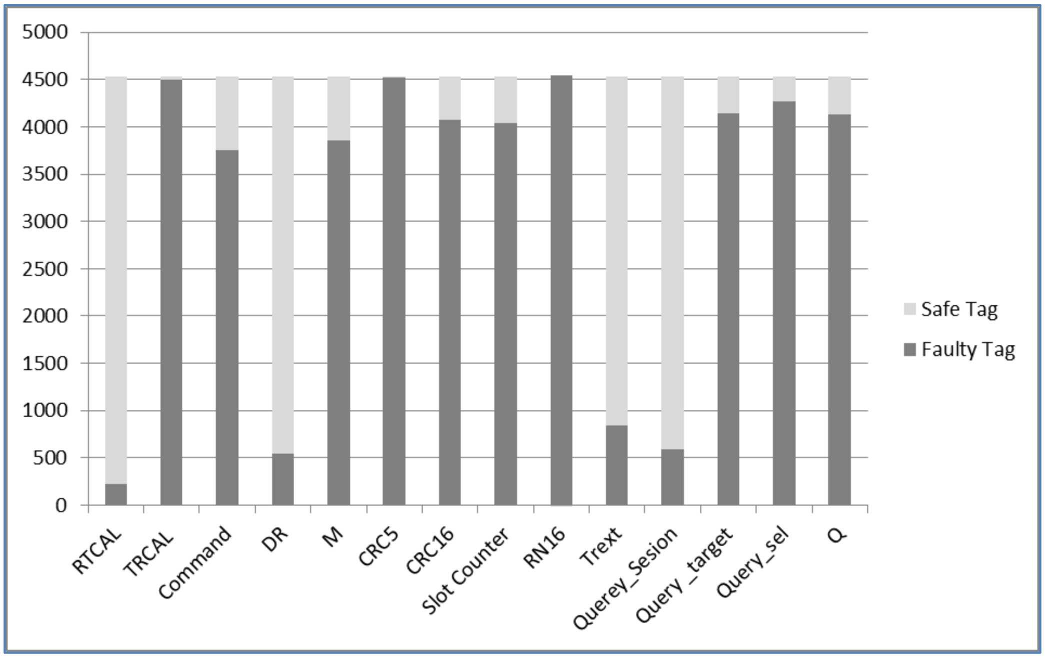

Figure 12. Successful Tag Identifications number (y axis) for the most sensitive protected (Safe) or unprotected (Faulty) registers (x axis)

If one of the three components fails or suffers a direct SEU then the fault is masked. In Figure 13, the target component is a specific register storing a sensitive parameter. This register is protected only against direct SEU impacting it. All the faults impacting the combinational block before this register will be propagated.

TMR technique implies an area increase of the redundant part of more than 200% due to the component triplication. It also needs a voter that is implemented just with some OR and AND gates for each bit of the triplicated component. We have experimentally noted that the use of this TMR improves the read rate in the presence of faults into sensitive registers. The proposed protection only adds 30 flip-flops to the whole circuit. Although expensive, TMR is in this case an acceptable method since thanks to the fault injection campaigns the most sensitive elements have been identified in a real RFID context, limiting the TMR use to only a few bits. The TMR can thus be tuned in order to replicate only flip-flops, which have been identified as the ones having the higher influence on the tag read rate in case of errors.

We have also proposed and validated a complementary approach allowing fault detection and diagnosis. This approach consists in adding hardware checkers into the tag circuit. Some of these checkers are provided by the synthesizable assertions available in the Open Verification Library (OVL) and others are designed to monitor tag finite state machine transitions. The faults detected by the checkers are counted and saved within the tag memory. Then, a user can read this information through the RFID reader and thereafter acquire diagnosis information. This approach has been implemented and evaluated on RFIM. Details on these robust architectures are given in [22].

IV. Conclusion and Future Work

The SafeRFID project addresses the dependability issues in RFID systems. The proposed framework considers both hardware and software components as well as analog and digital aspects of RFID systems. Three main layers have been identified: the hardware layer with tags and readers, the communication layer and the software layer including the RFID middleware. The main results of this work are: (1) the development of a fault simulator (SERFID) and of an FPGA based emulator (RFIM) that allows fault injection and test method evaluations, (2) the design and implementation of a robust LLRP-compliant RFID middleware prototype that provides fault detection and diagnosis new services, and (3) the development of a tag robust architecture with self-diagnosis capability. The main perspective of this work is to consider fault attacks and security issues related to RFID Systems. This issue is a major concern in the context of Internet of Thing deployment. Then we will use the two developed platforms SERFID and RFIM to validate new secure tags and system architectures. These tags and systems will embed security functions and authentication protocols.

Acknowledgment

This work has been supported by the French National Research Agency project “SafeRFID” [ANR 2010 JCJC 0305 01]

References

[1] V. Beroulle, O. Aktouf, and D. Hély, “System-Level Simulation for the Dependability Improvement of UHF RFID Systems,” ICWMC 2016, The Twelfth International Conference on Wireless and Mobile Communications, November 13 - 17, 2016 - Barcelona, Spain.

[2] R. Shorey, A. L. Ananda, M. C. Chan, C.-C. Chu, and W. T. Ooi, Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions, Chapter “WinRFID – A middleware for the enablement of Radio Frequency Identification (RFID) based Applications,” B. S. Prabhu, X. Su, H. Ramamurthy, C.-C. Chu, and R. Gadh, John Wiley and Sons Inc., 2006.

[3] Sun Microsystems, Inc., "Sun Java™ System RFID Software 3.0 Administration Guide," February 2006.

[4] A. Sengupta and S. Z. Schiller, “FlexRFID: A design, development and deployment framework for RFID-based business applications,” in Information Systems Frontiers, Vol. 12, n° 5, pp. 551-562, November 2010.

[5] N. Ahmed, “Reliable Framework for Unreliable RFID Devices,” in 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mannheim, 2010.

[6] F. Thornton, “How to Cheat at Deploying and Securing RFID,” Syngress Publishing ©2007, ISBN 1597492302 9781597492300

[7] C. Angerer and R. Langwieser, “Flexible Evaluation of RFID System Parameters using Rapid Prototyping,” in IEEE International Conference on Digital Object Identifier: 10.1109/RFID.2009.4911188 Publication Year: 2009 , pp. 42 – 47.

[8] C. Floerkemeier and S. Sarma, “RFIDSim—A Physical and Logical Layer Simulation Engine for Passive RFID,“ in Automation Science and Engineering, IEEE Transactions on Volume: 6 , Issue: 1 Digital Object Identifier: 10.1109/TASE.2008.2007929 Publication Year: 2009 , pp. 33 – 43.

[9] C. E. Palazzi, A. Ceriali, and M. Dal Monte, “RFID Emulation in Rifidi Environment,” in Proc. of the International Symposium on Ubiquitous Computing (UCS'09), Beijing, China, August 2009.

[10] EPCglobal, EPC Radio Frequency Identity Protocols Classe-1 Generation-2 UHF RFID, Protocol for Communications at 860 MHz 960 MHz, version 1.2.0, 2008.

[11] G. Fritz, V. Beroulle, O. Aktouf, and D. Hély, “SystemC Modeling of RFID Systems for Robustness Analysis,” in 19th International Conference on Software, Telecommunications and Computer Networks IEEE SoftCOM 2011Split - Hvar - Dubrovnik, September 15 – 17, 2011, IEEE Catalog Number: CFP1187A-CDR; ISBN 978-953-290-027-9.

[12] M. Hutter, J-M. Schmidt, and T. Plos. 2008. “RFID and Its Vulnerability to Faults,” in Proceeding of the 10th International Workshop on Cryptographic Hardware and Embedded Systems (CHES '08).

[13] G. Fritz, V. Beroulle, O. Aktouf, M. D. Nguyen, and D. Hély, “RFID System On-line Testing Based on the Evaluation of the Tags Read-Error-Rate,” in Journal of Electronic Testing:

Volume 27, Issue 3 (2011), pp. 267-276, (DOI: 10.1007/s10836-010-5191-6).

[14] G. Fritz, B. Maaloul, V. Beroulle, O-. Aktouf, and D. Hély, “Read Rate Profile Monitoring for Defect Detection in RFID Systems,” in IEEE International Conference on RFID-Technologies and Applications (RFID-TA 2011), pp. 89-94, Sitges, Barcelona, Spain, on September 15-16, 2011, IEEE Catalog Number: CFP11RFT-CDR ; ISBN: 978-1-4577-0026-2.

[15] G. Fritz, V. Beroulle, . Aktouf, and D. Hély, “Evaluation of a new RFID System Performance Monitoring Approach,” in Design, Automation & Test in Europe, (DATE 2012), Interactive Presentation, Dresden, Gremany, 12-16 March 2012.

[16] D. Fussell and S. Rangarajan, “Probabilistic Diagnosis of Multiprocessor Systems with Arbitrary Connectivity,” in IEEE 19th International Symposium on Fault-Tolerant Computing, FTCS-19. Digest of Papers., Chicago, IL, pp. 560-565, 1989.

[17] R. Kheddam, O. Aktouf, and I. Parissis, “Saferfid-MW: Safe and Fault-Tolerant RFID Middleware,” in Journal of Communications Software and Systems (jcomms), Special Issue on RFID Technologies and Internet of Things, Vol. 9, n° 1, March 2013, pp. 57-73.

[18] R. Kheddam, O. Aktouf and I. Parissis, “On-line Monitoring and Diagnosis of RFID Readers and Tags,” in 20th IEEE International Conference on Software, Telecommunications and Computer Networks (softcom 2012), Split, Croatia, 11-13 September 2012, pp. 1-9.

[19] R. Kheddam, O. Aktouf and I. Parissis, “An Extended LLRP Model for RFID System Test and Diagnosis,” in 8th Workshop on Advances in Model Based Testing, Montreal, Canada, 17-21 April 2012, pp. 529-538.

[20] R. Kheddam, O. Aktouf, I. Parissis and S. Boughazi, “ Monitoring of RFID Failures Resulting from LLRP Misconfigurations,” in 21st IEEE International Conference on Software, Telecommunications and Computer Networks (softcom 2013), Split, Croatia, Septembrer 2013, pp. 1-6.

[21] O. Abdelmalek, D. Hély, and V. Beroulle “Fault Tolerance Evaluation of RFID Tags,” in IEEE Latin America Test Workshop (LATW 2014), Fortaleza, Brésil, 13-16 March 2014.

[22] O. Abdelmalek, D. Hély, and V. Beroulle “Emulation of Faults Injection on UHF Transponders,” in 17th IEEE Symposium on Design and Diagnosis of Electronic Circuit and System (DDECS 2014), Warsaw, Poland, 23-25 April 2014.

[23] I. Mezzah, O. Kermita, H. Chemali, O. Abdelmalek, D. Hély, and V. Beroulle, “Assertion based on-line fault detection applied on UHF RFID tag,” in 8th IEEE International Design & Test Symposium 2013, Maroc (2013).