Monitoring en ligne des systèmes RFID en vue du diagnostic de fautes

Rafik Kheddam, Oum-El-Kheir Aktouf, Ioannis Parissis

MajecSTIC 2012
Lille, France, du 29 au 31 novembre 2012

Monitoring en ligne des systèmes RFID en vue du diagnostic de fautes

Rafik Kheddam, Oum-El-Kheir Aktouf et Ioannis Parissis

LCIS - Grenoble Institute of Technology, 50, rue B. de Laffemas, 26000 Valence - France

Contact : `firstname.lastname@lcis.grenoble-inp.fr`

Résumé
La RFID[1] (Radio Frequency IDentification) voit son champ d'application s'étendre vers les domaines critiques tels que les domaines du transport ou de la santé, alors qu'auparavant, elle était essentiellement utilisée dans la gestion de la chaîne logistique. Pour faire face à ce nouveau besoin en sûreté de fonctionnement, bien que plusieurs projets aient été déjà menés, plus de travaux de recherche sont nécessaires pour répondre à la demande croissante en tolérance aux fautes de cette technologie. Dans le cadre de cette thématique, nous proposons une approche de diagnostic en ligne[2] des systèmes RFID (lecteurs et tags[3]). Cette approche consiste principalement en un algorithme probabiliste intégré au middleware RFID. Il porte le nom de « *RFID diagAlgo* ». Il se base sur une analyse statistique des résultats des lecteurs pour identifier de possibles défaillances des lecteurs ou des tags RFID.

Abstract
Few years ago, RFID systems were only deployed in single site scenarios, usually with few readers. Nowadays, this kind of architecture is not suitable for the new business needs, such as data sharing with partners' companies. To meet these new requirements, a novel type of software, called RFID middleware has been designed for managing RFID devices and for processing the huge volume of generated raw data. Besides, current RFID systems are increasingly used in critical domains such as medical field or real-time processing domains. So, significant efforts are made to enhance the reliability of these systems. Despite that, more research is needed to meet the increasing dependability requirements. In this article, we propose a probabilistic diagnosis algorithm (called "RFID diagAlgo") that identifies faulty components of an RFID system on the basis of a statistical analysis of their read results.

Mots-clés : sûreté de fonctionnement, fiabilité, diagnostic en ligne, test, algorithme probabiliste.

Keywords: dependability, reliability, online diagnosis, test, probabilistic algorithm.

1. Introduction

Depuis quelques années, nous assistons à une explosion de l'utilisation des systèmes RFID dans les domaines critiques tels que le domaine médical, le ferroviaire ou l'aérien où la notion de temps réel est vitale. Ainsi le besoin en sûreté de fonctionnement se fait sentir de plus en plus. FIG. 1 montre un exemple d'utilisation de la technologie RFID pour la gestion et l'acheminement des bagages vers différentes destinations. Les lecteurs RFID jouent un double rôle dans cet exemple. Ils permettent d'une part le bon acheminement des bagages, et d'une autre part, la localisation de ces der-

[1] La technologie RFID est développée et promue par l'organisation à but non lucratif EPCglobal Inc. (www.epcglobalinc.org).
[2] Ces travaux font partie du projet SAFERFID supporté par l'Agence Nationale de la Recherche ANR (www.agence-nationale-recherche.fr) et dont le but est la proposition d'approches de tolérances aux fautes pour les systèmes RFID.
[3] Un tag RFID est une étiquette apposée sur un objet pour son identification (technologie similaire au code-barres).

niers. Il est clair que si le système RFID mis en place est défaillant (*i.e.*, lecteur en panne, erreurs de lecture, *etc.*), le système de manutention des bagages ne pourra pas remplir ses fonctions. Ainsi, il serait impossible de localiser un bagage qui serait tombé du tapis ou bien d'acheminer un bagage vers la bonne destination. Ceci introduit la nécessité d'améliorer la fiabilité des systèmes RFID.

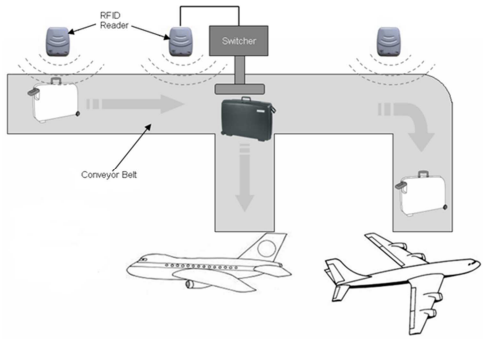

FIG. 1 - Système de manutention de bagages

Le reste du papier est organisé comme suit. La section 2 présente brièvement les systèmes RFID, suivie d'un état de l'art sur les techniques de monitoring utilisées dans ces systèmes en section 3. Enfin, les sections 4 et 5 présentent respectivement les détails de l'algorithme de diagnostic probabiliste et l'évaluation du modèle probabiliste utilisé par ce dernier.

2. Système RFID

La technologie RFID permet d'identifier plusieurs objets en même temps « sans contact ni vision direct » [1] [2]. Elle représente une technologie d'identification semblable à celle du code-barres. Elle est composée de lecteurs RFID, de tags RFID et d'un middleware.

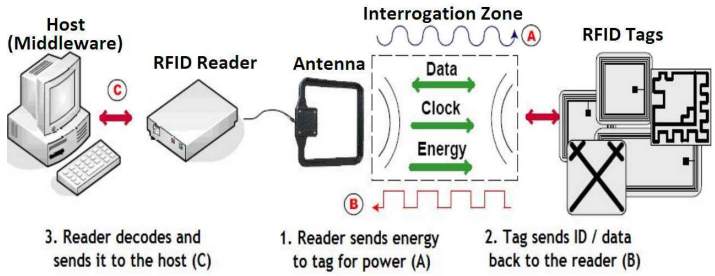

FIG. 2 - Architecture d'un système RFID [10]

- *Tag RFID* : étiquette disposant d'une mémoire pour le stockage de diverses informations, comme l'identifiant du produit étiqueté.
- *Lecteur RFID* : appareil dont le rôle est l'inventaire et l'accès aux mémoires des tags.
- *Middleware* : programme dont le rôle est la gestion des lecteurs et des tags.

3. Techniques de monitoring

Plusieurs techniques ont fait leur apparition pour répondre aux exigences des systèmes RFID actuels. Ces techniques sont classées en deux catégories [3] :

3.1. Monitoring de l'état des lecteurs

Dans ce type de monitoring, le middleware vérifie que le lecteur RFID est alimenté, connecté au réseau, que ses antennes sont opérationnelles, *etc.* Il peut être réalisé par des requêtes SNMP (Simple Network Management Protocol) (*e.g.*, commande *ping*). Ce type de monitoring est considéré comme *monitoring intrusif* car le middleware a besoin d'injecter des données supplémentaires dans le système surveillé pour collecter des informations qui ne sont pas disponibles lors du fonctionnement normal de ce système.

3.2. Monitoring des performances des lecteurs

Dans ce type de monitoring, le middleware exploite les données disponibles dans le système (*monitoring non-intrusif*) afin de calculer certains paramètres de performance comme le taux de lecture, la précision de lecture, la fréquence des erreurs d'un lecteur RFID, *etc.* Les techniques de cette catégorie, à l'inverse de la première, prennent en considération le caractère aléatoire des lecteurs[4]. Ainsi, elles permettent de détecter certaines défaillances suivant les variations du paramètre de performance choisi.

Néanmoins, toutes ces techniques semblent insuffisantes pour détecter toutes les défaillances des systèmes RFID, car elles ne considèrent qu'un seul lecteur à la fois. Cela conduit souvent à un diagnostic pauvre. En effet, lorsque certains tags ne sont pas lus correctement (ce qui se traduit par *un taux de lecture faible*), il est impossible de vérifier si le problème vient des tags ou du lecteur. Nous proposerons alors, une nouvelle approche sous forme d'un algorithme probabiliste qui sera intégrée au middleware afin d'améliorer la fiabilité de tout le système RFID.

4. Algorithme de diagnostic probabiliste « RFID diagAlgo »

RFID diagAlgo se décompose en trois étapes. La première[5] vise à regrouper les lecteurs suivant les groupes de tags qu'ils sont amenés à lire. Le fait d'organiser les lecteurs en groupes facilite dans la deuxième étape, la comparaison des différents résultats de ces derniers afin d'identifier les tags ou les lecteurs qui sont « potentiellement » défaillants. Dans la troisième étape, *RFID diagAlgo* associera une probabilité de défaillance aux éléments aberrants suivant un modèle probabiliste adapté depuis les travaux de D. Fussell et S. Rangarajan sur le diagnostic des systèmes multiprocesseurs [4] [5]. *RFID diagAlgo* a une complexité de $O(n \times (t+2))$ [6] dans la version présentée dans l'annexe. Nous avons délibérément choisi cette représentation (sans optimisation) pour plus de clarté. A titre indicatif, le temps d'exécution de *RFID diagAlgo* en faisant abstraction du délai[7] nécessaire pour l'obtention de tous les résultats des lecteurs ne dépasse pas 10 *millisecondes* (néanmoins, cela varie selon la configuration matérielle du système hôte).

4.1. Partitionnement des lecteurs en groupes

Si nous reprenons l'exemple du système de manutention de bagages dans un aéroport (FIG. 1), nous pouvons partitionner les lecteurs qui se trouvent le long du tapis roulant suivant le chemin que les différents bagages prennent pour atteindre leurs destinations. FIG. 3 montre un partitionnement possible des lecteurs. Ainsi, tous les lecteurs qui sont situés sur le chemin {*A*,*D*} seront regroupés ensemble (*zone rouge sur la figure*). Ce partitionnement est naturel, d'autant plus que chaque bagage dispose d'une destination et donc d'un chemin bien spécifique à suivre. De ce fait, *RFID diagAlgo* connaît tous les lecteurs qui doivent identifier ce bagage et ainsi, est évité les cas où l'algorithme peut prendre le silence d'un lecteur à propos d'un ou plusieurs bagages pour une défaillance. Ce partitionnement sera aussi utilisé pour comparer les résultats des lecteurs à l'intérieur d'un même ensemble. Mais lors du processus de comparaison des résultats, différents ensembles de lecteurs peuvent y participer (*i.e.*, un lecteur peut appartenir à des groupes différents suivant les bagages (les tags) qu'il analyse).

4.2. Comparaison des résultats des lecteurs

Dans cette étape, la comparaison ne se fait pas directement sur les résultats de lecture de chaque lecteur mais plutôt sur un des paramètres de performance des lecteurs (par abus de langage nous utiliserons l'expression « résultats des lecteurs » pour indiquer si les tags ou les groupes de tags analysés respectent bien le paramètre de performance choisi). Ce paramètre peut être :

- *Read Error to Total Read Rate* [3] : représente le ratio entre le nombre d'erreurs de lecture et le nombre total de tentatives de lecture de tous les tags.

---

[4] La précision de lecture pour les lecteurs RFID est à environ 70% [11]. Ils sont aussi très sensibles à leur environnement notamment à la présence d'obstacles comme le métal, l'eau ou les ondes électromagnétiques des autres appareils.

[5] Cette approche s'inspire des travaux d'Ahmed Nova qui a utilisé le type des données qui circulent dans le système pour former des chemins virtuels en vue de faciliter le traitement des données [9].

[6] n et t sont respectivement le nombre total des lecteurs en cours d'analyse et le nombre total des groupes de tags.

[7] Ce délai dépend de la distance qui sépare chaque lecteur des autres et de la vitesse de déplacement des tags.

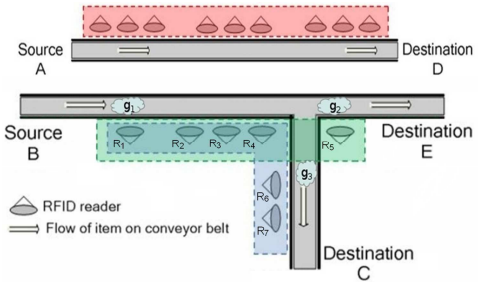

FIG. 3 - Partitionnement des lecteurs en groupes

TAB. 1 - Comparaison des résultats de lecture

M	R_1	R_2	R_3	R_4	R_5	R_6	R_7	**S(g_t)**
g_1	1	0	1	1	1	-	-	$\{R_2\}$
g_2	0	1	0	0	0	-	-	$\{R_2, g_2\}$
g_3	1	1	1	1	-	1	0	$\{R_7\}$
SF								$\{R_2, R_7\}$

- *Read Error Rate*[8] [6] : représente le ratio entre le nombre d'erreurs de lecture et le nombre total de tentatives de lecture de chaque tag.
- *Approche par Profil* [7] : concerne l'analyse d'un ensemble de tags en même temps (*e.g.*, une palette de produits). Chaque ensemble ou groupe de tags se voit assimiler un profil de lecture par chaque lecteur. Ce profil est représenté par une courbe qui correspond à l'ensemble des taux de lecture de chaque tag de ce groupe. Dans la suite de cet article, nous utiliserons cette approche pour la surveillance des lecteurs et des groupes de tags.

Si la valeur de ce paramètre dépasse une limite prédéterminée ou bien représente une valeur aberrante par rapport à celle des autres lecteurs, le lecteur est déclaré *potentiellement défaillant*.

Exemple : reprenons l'exemple de FIG. 3, où nous avons 7 lecteurs $R = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7\}$ et 3 groupes de tags $G = \{g_1, g_2, g_3\}$. TAB. 1 montre le résultat de la comparaison des profils de lecture de tous les groupes de tags.

- Nous appellerons M la matrice des résultats de lecture, représentée par TAB. 1 sans la dernière ligne et la dernière colonne.
- $M(R_n, g_t) = 1$ (*resp.*, $M(R_n, g_t) = 0$) indique que le groupe de tags g_t respecte bien (*resp.*, ne respecte pas) son profil lorsqu'il a été analysé par le lecteur R_n.
- g_1 et g_2 ne sont pas analysés par R_6 et R_7 car ils empruntent un chemin autre que celui où ces lecteurs se trouvent (idem pour g_3 par rapport à R_5). Ce qui montre l'utilité de la première étape de notre algorithme qui évite de prendre le silence de certains lecteurs sur certains groupes de tags comme une défaillance.
- $S(g_t)$ est une fonction qui répertorie les lecteurs et groupes de tags dont les résultats sont aberrants (par rapport à celui de la majorité) lors de l'analyse du groupe g_t. ; *e.g.*, la majorité des lecteurs disent que g_1 respecte bien son profil d'origine sauf le lecteur R_2, donc R_2 est considéré défaillant. $S(g_t)$ permet de nous donner les groupes de tags défaillants, mais aussi le nombre de fois où un lecteur est déclaré défaillant (qui peut être une défaillance temporaire ou intermittente) ; *e.g.*, R_2 est déclaré défaillant sur 2 groupes de tags et R_7 sur un seul groupe. Ce nombre est utilisé dans *SF* pour identifier les lecteurs qui sont réellement défaillants.
- *SF* permet de répertorier tous les lecteurs qui sont considérés comme défaillants après l'analyse de tous les groupes de tags. Sur notre exemple, un lecteur est déclaré défaillant s'il est déclaré défaillant au moins sur un groupe de tags. Nous pouvons varier le nombre de fois où un lecteur doit être déclaré défaillant dans $S(g_t)$ pour qu'il soit réellement considéré défaillant. Cela permet notamment de réduire les fausses alertes.

4.3. Calcul de la précision du diagnostic

Cette approche a pour rôle de valider les décisions prises dans l'étape précédente sur l'état des lecteurs et des tags en associant à ces derniers une probabilité de défaillance qui dépend de la configuration courante du système. Pour cela, nous définissons deux paramètres sur lesquels reposera notre modèle probabiliste. La précision du diagnostic lors de la phase de comparaison des résultats revient à respecter les deux cas suivants.

- « Un lecteur valide doit être identifié comme valide (*Correct Positive (CP)*) ».
- « Un lecteur défaillant doit être identifié comme défaillant ». Pour des raisons de simplicité, nous nous intéressons au cas inverse de ce dernier ; *i.e.*, le cas où un lecteur défaillant est considéré comme valide (*False Positive* [9]*(FP)*).

[8] Ce paramètre est utile pour la surveillance individuelle des tags RFID.
[9] False Positive représente les fausses alertes qu'il faut réduire.

La capacité de bien identifier l'état des lecteurs (**Identifiabilité**[10] (*I*) ou la précision du diagnostic) est donnée par la formule probabiliste suivante :

$$I(n,t,p,r) = (1-p) \times CP(n,t,p,r) + p \times (1 - FP(n,t,p,r))$$

Où :

- n est le nombre total des lecteurs.
- t est le nombre de groupes de tags.
- p est la probabilité initiale (temporaire) de défaillance d'un lecteur[11]. Dans l'algorithme présenté en annexe cette probabilité est propre à chaque lecteur et est égale au nombre de fois où le lecteur est déclaré défaillant dans le processus de comparaison des résultats sur le nombre total des groupes de tags analysés.
- r (*Rational Behavior)* est la probabilité que la défaillance d'un lecteur se manifeste (*i.e.*, les résultats du lecteur en question sont incorrects). Un programme erroné ne devient défaillant que lorsque le chemin d'exécution qui contient l'erreur est emprunté. r représente l'estimation de ce phénomène. Une grande valeur de r indique que la plupart, voire toutes les erreurs du programme ont des effets visibles sur le système.

1. *Correct Positive (CP)* : est représenté par la probabilité *CP(n, t, p, r)*. Il désigne le cas où un lecteur bon est considéré comme bon dans la phase de comparaison des résultats. Pour cela nous distinguons les deux cas suivants :
   - La majorité des lecteurs (autres que celui en cours d'analyse) sont bons (*i.e.*, leur nombre est entre $n/2$ et $n-1$ lecteurs) et donc, forcément leurs résultats concordent avec ceux du lecteur en cours d'analyse.
   - La majorité des lecteurs sont défaillants, mais sur chaque groupe de tags, il y a un certain nombre d'entre eux qui ont des résultats corrects (*i.e.*, ils ont un comportement non défaillant) de façon à ce que le nombre de lecteurs (défaillants ou pas) en accord avec le lecteur en cours d'analyse représente la majorité.

$$CP(n,t,p,r) = \begin{cases} \bullet \sum_{i=\frac{n}{2}}^{n-1} \left[C(i,n-1) \times (1-p)^i \times p^{n-1-i} + C(i,n-1) \times p^i \times (1-p)^{n-1-i} \right. \\ \left. \times \left(\sum_{j=i+1-\frac{n}{2}}^{i} \left(C(j,i) \times (1-r)^j \times r^{i-j} \right) \right)^t \right] \quad (\textbf{Si } n \text{ est pair}) \\ \bullet \sum_{i=\frac{n-1}{2}}^{n-1} \left(C(i,n-1) \times (1-p)^i \times p^{n-1-i} \right) + \sum_{j=\frac{n-1}{2}+1}^{n-1} \left[C(i,n-1) \times p^j \times (1-p)^{n-1-j} \right. \\ \left. \times \left(\sum_{k=j-\frac{n-1}{2}}^{j} \left(C(k,j) \times (1-r)^k \times r^{j-k} \right) \right)^t \right] \quad (\textbf{Si } n \text{ est impair}) \end{cases}$$

   C(x,y) désigne le nombre de combinaisons de x éléments parmi y éléments.

2. *False Positive (FP)* : est représenté par la probabilité *FP(n, t, p, r)*. Il désigne le cas où un lecteur défaillant est considéré comme bon. Cela se traduit par deux cas possibles :
   - Le lecteur défaillant traite correctement tous les groupes de tags (*i.e.*, la défaillance ne se manifeste pas).

[10] *Identifiabilité* est la capacité que notre l'algorithme *RFID diagAlgo* arrive à « bien identifier » les lecteurs défaillants et les lecteurs non défaillants.

[11] Pour ne pas alourdir les différentes formules probabilistes, nous considérons que les lecteurs ont la même probabilité de défaillance.

- Le lecteur défaillant traite quelques ou tous les groupes de tags incorrectement, mais pour chaque groupe de tags traité, il y a au moins la moitié des lecteurs qui le traite incorrectement (et qui sont donc défaillants) de façon à représenter la majorité à chaque fois.

$$FP(n,t,p,r) = CP(n,t,p,r) \times (1-r)^t + \sum_{i=1}^{t} \left[\left(\sum_{j=\lfloor \frac{n}{2} \rfloor}^{n-1} \left(C(j,n-1) \times (p \times r)^j \times \left((1-p) + p \times (1-r) \right)^{n-1-j} \right) \right)^i \times C(i,t) \times r^i \times (1-r)^{t-i} \times CP(n,t-1,p,r) \right]$$

5. Evaluation du modèle probabiliste

Pour étudier le comportement de l'*Identifiabilité* suivant les différents paramètres ; (*i.e.*, l'influence de chaque paramètre sur l'*Identifiabilité*), nous avons fait varier les paramètres r et p de 0 à 1, le nombre de lecteurs n de 2 à 20 et le nombre de tags t de 1 à 5.
Les figures 4, 5 et 6 montrent quelques résultats de cette simulation.

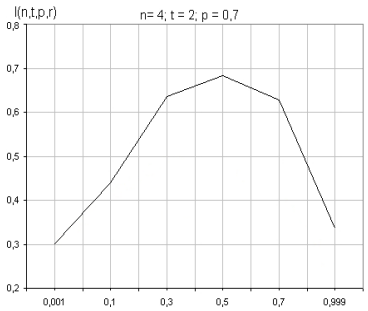

FIG. 4 - Variation de l'*Identifiabilité* suivant le paramètre *r*

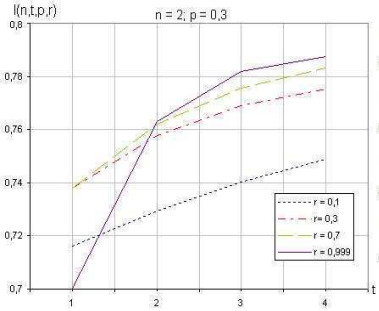

FIG. 5 - Variation de l'*Identifiabilité* suivant le nombre de groupes *t*

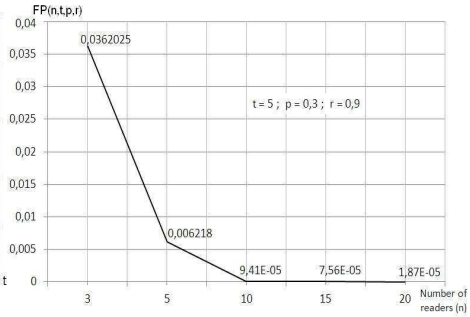

FIG. 6 - Effet du nombre de lecteurs sur *FP*

Explication
Avec $p = 0.7$ (FIG. 4), la plupart des lecteurs sont défaillants.

- Quand r est petit, la plupart des lecteurs défaillants ne manifestent pas leur défaillance, et donc ils ne sont pas détectés vu qu'ils ont les mêmes résultats que les lecteurs corrects (*i.e.*, la majorité des lecteurs sont défaillants et ne manifestent pas leurs défaillances).
- Avec l'augmentation de r jusqu'à atteindre environ 0.5 ; (*i.e.*, le nombre de lecteurs qui manifestent leurs défaillances augmente), l'*Identifiabilité* (*I*) augmente, vu que tous les lecteurs qui manifestent leurs défaillances représentent la minorité et sont donc détectés.
- Une fois que r dépasse 0.5, son effet sur I s'inverse ; *i.e.*, la plupart des lecteurs défaillants manifestent leurs défaillances et donc, ils représentent la majorité et par conséquence ils sont considérés comme non défaillants (*i.e.*, l'*Identifiabilité* diminue).

FIG. 5 montre l'effet du nombre de groupes de tags (t) sur I. L'augmentation de t implique l'augmentation de I quelle que soit la valeur de r. Aussi, I augmente avec r sans effet inverse lorsque $r > 0.5$ comme nous l'avons déjà vu dans FIG. 4. Cela est dû à la probabilité de défaillance des lecteurs qui est faible ($p = 0.3$) ; *i.e.*, les lecteurs défaillants n'ont aucune chance de représenter la majorité pour qu'ils soient déclarés non défaillants.

L'objectif de *RFID diagAlgo* est de réduire les faux positifs (paramètre *FP*). FIG. 6 met l'accent sur ce phénomène qui est rendu possible grâce notamment à la 2ème étape de notre algorithme (*i.e.*, un lecteur n'est considéré correct par rapport à un groupe de tags que s'il dispose d'un même résultat que la *majorité*), mais aussi, au nombre de lecteurs qui augmente au sein du même groupe. Nous voyons sur cette figure que lorsque le nombre de lecteurs dépasse 10, *FP* est quasiment nul. Cela nous ouvre une autre utilité qui nous permet de fixer le nombre nécessaire de lecteurs et / ou de groupes de tags pour avoir un meilleur diagnostic.

6. Conclusion et perspectives

Nous avons présenté dans cet article un algorithme de diagnostic probabiliste basé sur une analyse statistique pour la surveillance des systèmes RFID. Il se compose en trois étapes. La première est le partitionnement des lecteurs en groupes afin de faciliter et d'optimiser l'analyse. La deuxième étape est la comparaison de tous les résultats de tous les lecteurs pour identifier les lecteurs et les tags défaillants. Ces éléments défaillants se voient associer des probabilités de défaillances pour appuyer la décision de l'algorithme *RFID diagAlgo* dans la troisième étape.

L'algorithme en lui-même est très rapide, mais dans le mode réel, la phase de récupération des résultats des lecteurs prend un temps considérable. De ce fait, il est nécessaire de proposer plusieurs implémentations en décomposant les groupes de lecteurs en sous-groupes, en définissant le nombre de groupes de tags à analyser à chaque étape et en adaptant la méthode de comparaison des résultats de façon à minimiser le temps d'attente des résultats, tout en gardant une précision de diagnostic acceptable. Il reste aussi à estimer le paramètre *Rational Behavior* en utilisant notamment les métriques de complexité d'Halstead[12] [8] pour estimer le nombre d'erreurs dans le programme testé et répartir ces erreurs sur l'ensemble des chemins d'exécution du graphe de flot de contrôle du programme pour déduire la probabilité qu'une erreur se manifeste. Enfin, nous simulerons cet algorithme dans des conditions d'exécution réelles avec injection de fautes (ondes radio parasites, présence d'obstacles, *etc.*) pour simuler des comportements défaillants du système.

Bibliographie

[1] P. Krishna and D. Husak, "RFID INFRASTRUCTURE," *IEEE Applications and Practice,* pp. 4-10, 2007.

[2] H. Taimur and C. Samir, "A Taxonomy for RFID," *Proceedings of the 39th Annual Hawaii International Conference on System Sciences,* vol. 08, no. 06, pp. 1-10, 2006.

[3] P. Sanghera, F. Thornton, B. Haines, F. Kung Man Fung, J. Kleinschmidt, A. M. Das, H. Bhargava and A. Campbell, How to Cheat at Deploying and Securing RFID, Syngress Publishing, Inc. Elsevier, Inc., 2007.

[4] S. Rangarajan and D. Fussell, "A Probabilistic Method for Fault Diagnosis of Multiprocessor Systems," *Eighteenth International Symposium on Fault-Tolerant Computing,* pp. 278-283, 1988.

[5] D. Fussell and S. Rangarajan, "Probabilistic Diagnosis of Multiprocessor Systems with Arbitrary Connectivity," *Nineteenth International Symposium on Fault-Tolerant Computing,* pp. 560-565, 1989.

[6] G. Fritz, V. Beroulle, O.-E.-K. Aktouf, M.-D. Nguyen and D. Hély, "RFID system on-line testing based on the evaluation of the tags Read-Error-Rate," *IEEE, Mixed-Signals, Sensors and Systems Test Workshop,* pp. 1-10, 2010.

[7] G. Fritz, B. Maaloul, V. Beroulle, O.-E.-K. Aktouf and D. Hély, "Read rate profile monitoring for defect detection in RFID Systems," *IEEE RFID-TA,* 2011.

[8] M. H. Halstead, Elements of software science, New York, USA: Elsevier Science Inc., 1977.

[9] N. Ahmed, R. Kumar, R. Steven French and U. Ramachandran, "RF²ID: A Reliable Middleware Framework for RFID Deployment," *21th International Parallel and Distributed Processing Symposium,* pp. 1-10, 2007.

[10] D. J. Glasser, K. W. Goodman and N. G. Einspruch, "Chips, tags and scanners: Ethical challenges for radio frequency identification," *Ethics and Information Technology,* vol. 9, no. 2, pp. 101-109, 2007.

[11] R. Derakhshan, M. E. Orlowska and X. Li, "RFID Data Management: Challenges and Opportunities," *IEEE International Conference on RFID,* pp. 175-182, 2007.

[12] Métriques introduites par Maurice Howard Halstead en 1977. Ce sont des métriques qui se basent sur le nombre d'opérateurs et d'opérandes dans un programme pour estimer le nombre d'erreurs qu'il est susceptible de contenir.

Algorithme : Diagnostic probabiliste des systèmes RFID

Inputs: Read results of each reader.
Outputs: Failure probabilities of each faulty reader and / or groups of tags.

```
    // R is a set of readers to be analyzed by the algorithm.
    // G is a set of groups of tags processed by R.
1   begin
2       n = |R|; t = |G|;
3       FT := ∅; // The set of Faulty Tags or groups of tags.
4       FR := ∅; // The set of Faulty Readers.
5       T[n] := {0,0, ... ,0}; // T[u]: number of groups of tags read by reader u.
6       r := 0.99; // Most failures of the faulty readers have visible effects on the system.
7       nbFailR[n] := {0,0, ... ,0}; // Failure counters for each reader, (array of n counters).
8       nbFailT[t] := {0,0, ... ,0}; // Failure counters for each group of tags, (array of t counters).
9       failureProba[n] = {0,0, ... 0}; // Failure probabilities of all readers.
10      x := 0; // number of groups of tags already processed.
11
12      for (g ∈ G){
13          x++;
14          for (u ∈ R_g) { // R_g is the set of readers that process the group g with R_g ⊂ R.
15              computeProfile(u,g); [13]
16              T[u] := T[u] + 1; // Counting the number of groups of tags processed by u.
17              if (D_u(g) = 0) {
18              // D_u(g)=0 means "g does not match its original profile according to the reader u".
19                  nbFailT[g] := nbFailT[g] + 1; // number of times g is declared faulty.
20              }
21          }
22          compute(S(g)) ; // S(g) contains the faulty components (faulty readers and/or tags).
23          for (u ∈ R) {
24              if (u ∈ S(g)) {// u is faulty on g.
25                  nbFailR[u] := nbFailR[u] + 1;
26                  failureProba[u] := nbFailR[u]/T[u] ;
27              }
28          }
29          if (g ∈ S(g)) { // g is a faulty group.
30              FT := FT ∪ {g};
31              print("g is faulty with a probability", nbFailT[g]/n × I(n, x, failureProba, r));
32          }
33      }
34      compute(SF); // SF contains the real faulty readers.
35      for (u ∈ SF) {
36          FR := FR ∪ {u};
37          print("u is faulty with a probability", I(n, t, failureProba, r));
38      }
39  end
```

[13] *computeProfile(u,g)* : est une fonction qui permet de calculer le profil du groupes de tags g suivant le lecteur u. Cette approche est présentée brièvement dans la section 4.2 et fait office d'un autre article [7].