2018 2nd International Conference on Modeling, Simulation and Optimization Technologies and Applications (MSOTA 2018)
ISBN: 978-1-60595-594-0

A General Collision Tree Protocol for RFID Tag Identification to Handle the Capture Effect in RFID System

Xiao-lin JIA[1,2,*], Miodrag BOLOC[2], Yu-hao FENG[3] and Ya-jun GU[1]

[1]School of Computer Science and Technology, Southwest University of Science and Technology, China

[2]School of Electrical Engineering and Computer Science, University of Ottawa, Canada

[3]School of Social & Behavior Science, University of Arizona, USA

*Corresponding author

Keywords: Uncertainty factor, Capture effect, Capture probability, Hidden tag problem.

Abstract. This paper presents a simple and efficient anti-collision protocol, named General Collision Tree Protocol (GCT), that was designed to cope with the capture effect in RFID system. The goal is to ensure that all tags are completely identified including the hidden tags and missing read tags which exist in conventional anti-collision protocols because of capture effect and other uncertainty factory in RFID system. Analysis and simulation results indicate that the performance of GCT outperforms that of the other state-of-the-art general RFID tags identification anti-collision protocols. The identification efficiency of it is always above 50% no matter what the capture probability of the RFID system is.

Introduction

Radio frequency identification system (RFID) is one of the most important and fundamental technologies of Internet of Things (IOT) because it allows for identifying, locating, tracing, and monitoring things or objects [1]. In RFID system, tag collision occurs when multiple tags respond to the reader's query simultaneously, and the reader cannot identify any tag [2,3]. Therefore, effective anti-collision protocol is very important for RFID tags identification. There are two categories of RFID tag identification protocols: probabilistic protocols and deterministic protocols [2,3]. Probabilistic protocols are ALOHA-based methods such as Slotted ALOHA Protocol, Frame Slotted ALOHA Protocol, and Dynamic Frame Slotted ALOHA Protocols, etc. Deterministic protocols are tree-based methods such as Query Tree Protocol (QT) [4,5], Binary Tree Protocol (BT) [5], Collision Tree Protocol (CT) [6,7], etc.

All these conventional protocols generally assume that the RFID reader and tags work in an ideal application environment. But in practical RFID applications, many uncertainty factors such as capture effect, detection error, channel quality, and noise impact the correctness and completeness of RFID tags identification [8-10]. Especially, the capture effect makes that RFID tags cannot be identified by the reader. In RFID system, when multiple tags respond to reader's query, some of them cannot be detected by the reader because their backscattered signals are weaker than those of the other tags and the weaker signals are captured by the stronger ones. This phenomenon is the capture effect in RFID system, and the weaker tags are called hidden tags. The conventional anti-collision protocols cannot identify the hidden tags and ensure that all tags are identified completely. This paper focuses on how to handle the capture effect and identifies the hidden tags during RFID tags identification.

Several variations of the classical anti-collision protocols have been proposed to address the capture effect during RFID tags identification [8-10]. For example, Generalized Query Tree Protocols (GQT1, GQT2) [8] are based on QT, General Binary Tree Protocol (GBT) [9] is base on BT, and Tweaked Query Tree Protocol (TQT) and Tweaked Binary Tree Protocol (TBT) [10] are based on GQT2 and GBT respectively. These protocols solved the capture effect, but their identification efficiency is still below 36.8%.

In [6] and [7], an efficient and stable anti-collision protocol for RFID tag identification is introduced, called Collision Tree Protocol (CT), which improves the tag identification efficiency up to 50% by eliminating the idle cycle or idle slot (in which no tag responds to the reader's query) during tags identification. At the same time, CT provides a good reference and foundation for the research of new efficient anti-collision protocols [11,12].

This paper presents a General Collision Tree Protocol (GCT) based on CT to cope with the capture effect and hidden tag problem in RFID system and ensure that all tags could be identified correctly and completely. Theoretical analysis and numerical simulation results indicate that GCT could identify all tags in RFID system under uncertain environments and the identification efficiency of it is always above 50%. And the identification performance of GCT is better than other general protocols such as GQT, GBT, TQT, and TBT.

The rest of this paper is organized as follows. Section II presents the General Collision Tree Protocol (GCT). Section III analyzes the identification performance of GCT. Section IV illustrates the simulation results of GCT. Section V is the conclusion of the paper.

General Collision Tree Protocol

General Collision Tree Protocol (GCT) uses an iterator flag, say iFlag, to repeat the identification process in order to identify the hidden and unidentified tags. The tags identification procedure of GCT is described in Fig.1 with the pseudo code of the reader and the tags.

```
Reader-GCT
iFlag=1, PUSH(Φ), tagID=Φ
while (Stack<>NULL and iFlag>0)
   prefix=POP()
   if (prefix==Φ and iFlag>0)
      PUSH(Φ)   // add a confirmation cycle
   end if
   send Query(prefix)
   waiting the response of tags
   if (no response)    //  all tags are identified
      iFlag=iFlag-1    //  to end the identification
   else
      get receivedID
      if (collided in receivedID)   // collided cycle
         c=the index of the first collided bit in receivedID
         prefix=prefix+receivedID[0, ⋯, c-1]
         PUSH(prefix+"1")   //  put new prefix into Stack
         PUSH(prefix+"0")   //  put new prefix into Stack
      else
         tagID=prefix+receivedID  //  readable cycle
         send Sleep(tagID)  // sleep the identified tag
      end if
  end if
end while
```

(a) for reader

```
Tags-GCT
while (unidentified)
    waiting Query(prefix)
    // extract the prefix parameter
    get prefix
    k=length(prefix)
    if (tagID[0, ⋯, k-1]==prefix)
       // answer the reader's query
       transmits tagID[k, ⋯, n-1]
    end if
end while
```

(b) for tags

Figure 1. Pseudo code of the work process of GCT.

The work process of the reader of GCT is shown in Fig.1(a). Initially, iFlag=1 (the initial value of iFlag could be adjusted to another integer depending on the RFID system), tagID=Φ, and Φ is pushed into the stack. Here, Φ is an empty binary string, and stack is a buffer pool to store prefixes. The reader pops a prefix from the stack as the parameter of query command. If the prefix is Φ and iFlag>0, the reader pushes another empty string Φ into the stack to add a confirmation cycle to ensure all tags are identified successfully. If any tags (hidden, missing read, or unidentified) existed in the query cycle or confirmation cycle, the reader repeats the identification process and identifies them until all tags are identified.

The reader sends Query(prefix) command to query the tags and waits the tags' response. If no tag responded to the reader's query, which indicated that all tags are identified, the reader decreases iFlag by 1 to end the identification process. If collision occurred in the receivedID, the reader generates two new prefixes and pushes them into the stack, which will be used in the following query as new parameter of query command. If no collision occurred in the receivedID, the reader identifies a new tag whose tagID is prefix+receivedID. Once a tag is identified, the reader sends Sleep(tagID) command to inform the tag not to answer the reader's following query command.

The work process of the tags of GCT is shown in Fig.1(b). The unidentified tag waits for the reader's Query(prefix) command and extracts the prefix from it. If the tag's ID matched with the prefix, the tag transmits its ID except for the part of the ID that is the same as the prefix, i.e., tagID[k, ..., n-1], here k is the length of prefix, and n is the length of tag ID. Accordingly, GCT also belongs to memoryless protocol because the responses of the tags only depend on the current query of the reader and not on the history of the reader's query and the tags' response. Therefore, GCT can be used in passive RFID tags identification system, in which the tags have no built-in power source and memory to provide energy and store history data.

Table 1 gives an example of GCT used to identify six tags with IDs: 0001, 0010, 0101, 1001, 1010, and 1100, in which Φ is a empty string, x represents a collided bit. In the first cycle, the six tags respond to the reader's query, but tags 0001, 0010 and 1010 are hidden because their signals are captured by that of the other tags. The collided string "xxxx" received by the reader is the fusion signal of tags IDs 0101, 1001 and 1100. Tag 0001 is hidden in cycle 2 again and is identified in cycle 7. Tag 0010 hidden in the first cycle is identified in cycle 8. Tag 1010 is captured in cycle 6 and identified in cycle 9, because its signal is weaker than that of the tags 0001 and 0010. The last cycle is an idle cycle, no tag answers the reader's query, which indicates that all tags are identified successfully.

On the other hand, all prefixes popped from the stack in cycles 1, 6, 9 and 10 are empty string Φ. But the empty string in cycle 1 is used to initialize and start the identification process, and those in cycles 6, 9 and 10 are used to confirm that there are hidden and unidentified tags or not. If the reader received response in confirmation cycle, the reader identifies the tag(s) as shown in cycle 6 and 9. If no response is received, the reader finishes the tag identification by decreasing iFlag by 1, as shown in cycle 10 in Table 1 and lines 02, 09 to 10 in Fig.1(a).

Table 1. Example of GCT identifying tags: 0001, 0010, 0101, 1001, 1010, and 1100.

Cycle	Prefix	ReceivedID	Channel State	Response Tags	Hidden Tags	Prefix Stack
1	Φ	xxxx	collided	0001,0010,0101, 1001,1010,1100	0001, 0010, 1010	0,1,Φ
2	0	001	tagID:0101	0001,0010,0101	0001,0010	1,Φ
3	1	x0x	collided	1001,1010,1100	1010	10,11,Φ
4	10	01	tagID:1001	1001,1010	1010	11,Φ
5	11	00	tagID:1100	1100		Φ
6	Φ	00xx	collided	0001,0010,1010	1010	000,001,Φ
7	000	1	tagID:0001	0001		001,Φ
8	001	0	tagID:0010	0010		Φ
9	Φ	1010	tagID:1010	1010		Φ
10	Φ		idle	no response		

Performance Analysis

This section discusses the time complexity and identification efficiency of GCT based on the performance of CT and the correlations between GCT and CT.

The time complexity of anti-collision protocol is the number of query-response cycle required to identify all RFID tags. From literatures [6] and [7], the time complexity of CT is

$$T_{CT}(n) = 2n - 1, \quad (1)$$

here, n is the number of tags.

As described in section II, GCT increases one idle query cycle (no tag response) to make sure that all tags are identified, e.g. the last cycle in Table 1. The other part of query and response process of GCT is the same with that of CT. Therefore, if no tag is hidden or missing read during tags identification, the time complexity of GCT is

$$T_{GCT\text{-}1}(n) = T_{CT}(n) + 1 = 2n - 1 + 1 . \quad (2)$$

In practical RFID system, if some tags are hidden or missing read in some query cycles, GCT identifies them after the other unhidden tags are identified. Actually, GCT divides the tags into two groups according to their state hidden or not, and identifies them respectively. In each group, the identification process of GCT is the same with that of CT. Assuming the capture probability, which is the ratio between the number of captured tags or hidden tags and the total number of tags in the tags group, of the RFID system is α, the number of tags in the two groups are $n_1 = n - n\alpha$ and $n_2 = n\alpha$. Therefore, the time complexity of GCT in this case is

$$\begin{aligned} T_{GCT-2}(n) &= (2n_1 \text{ -}1) + (2n_2 \text{ -}1) + 1 \\ &= (2n - 2n\alpha \text{ -}1) + (2n\alpha \text{ -}1) + 1 \cdot \\ &= 2n - 2 + 1 \end{aligned} \quad (3)$$

If the capture effect occurred again in the second group with probability α, the tags in the second group are divided into two groups also. The number of tags in the second group changes to $n_2 = n\alpha - n\alpha^2$, and the number of tags in the third group is $n_3 = n\alpha^2$. In this case, the time complexity if GCT is

$$\begin{aligned} T_{GCT-3}(n) &= (2n_1 \text{ -}1) + (2n_2 \text{ -}1) + (2n_3 \text{ -}1) + 1 \\ &= 2n - 3 + 1 \end{aligned} . \quad (4)$$

Iterating this process with same capture probability α, if the tags are divided into k groups, the number of tags in the k groups are $\mathrm{n}_j = n\mathrm{a}^{j-1} - na^j$, j is an integer from 1 to k -1, and $n_k = n\alpha^{k\text{-}1}$. The time complexity of GCT with same capture probability (SCP) is

$$\begin{aligned} T_{GCT\text{-k-SCP}}(n) &= \sum_{\mathrm{i}=1}^{k} (2n_i - 1) + 1 \\ &= \sum_{i=1}^{k-1} (2na^{i-1} - 2na^i - 1) + (2na^{k-1} - 1) + 1 \cdot \\ &= 2n - \mathrm{k} + 1 \end{aligned} \quad (5)$$

More generally, assuming α_1, α_2, α_3, ..., α_k are the capture probability of groups from 1 to k, here $\alpha_k = 0$ because no tag is hidden or missing read in the last group k, the number of tags in these groups are $\mathrm{n}_j = n\prod_{i=1}^{j-1} a_i - n\prod_{i=1}^{j} a_i$, j is an integer from 2 to k. Note that $n_1 = n - n\alpha_1$. The time complexity of GCT with different capture probability (DCP) is

$$\begin{aligned} T_{GCT\text{-k-DCP}}(n) &= \sum_{\mathrm{i}=1}^{k} (2n_i - 1) + 1 \\ &= (2n - 2na_1 - 1) + \sum_{\mathrm{j}=2}^{k} (2n\prod_{i=1}^{j-1} a_i - 2n\prod_{i=1}^{j} a_i - 1) + 1 \cdot \\ &= 2n - \mathrm{k} + 1 \end{aligned} \quad (6)$$

Therefore, $T_{GCT} = T_{GCT-k-\mathrm{DCP}} = T_{GCT\text{-k-SCP}}$.

According to the above discussions, the time complexity of GCT decreases with the increasing of the number of groups. If $k = 1$, i.e., all tags form one group, the worst time complexity of GCT is $T_{GCT-Worst} = 2n$, because no tag is hidden or missing read during the identification. If $k = n$, i.e., the n tags are divided into n groups, the best time complexity of GCT is $T_{GCT-Best} = n + 1$, because the reader identified the tags one by one, and no collision occurred during the tags identification.

The identification efficiency of anti-collision protocol is the ratio between the number of tags and the number of query-response cycles required to identify these tags (the time complexity). According to the time complexity discussed above, the identification efficiency of GCT is

$$E_{GCT}(n) = n / T_{GCT}(n) = n/(2n-k+1). \quad (7)$$

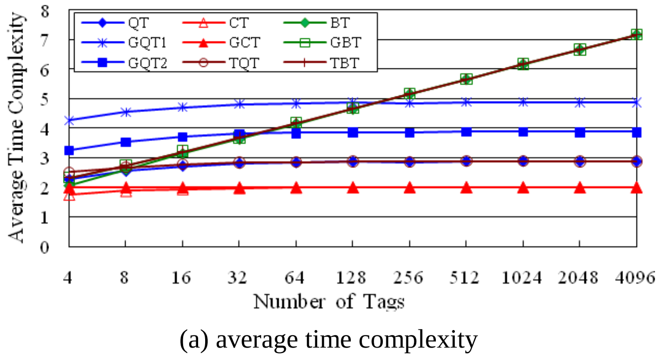
From Eq.2, without capture effect, i.e., $\alpha = 0$, the identification efficiency of GCT is always 50%. From Eq.7, for both n and k are integer and $n \geq k > 0$, $E_{GCT} \geq 50\%$. And when $k = 1$, the worst identification efficiency of GCT is $E_{GCT-Worst} = 50\%$. The best identification efficiency of GCT is $E_{GCT-Best} = n/(n+1)$ and $\lim_{n\to\infty} E_{GCT-Best} = 100\%$, when $k = n$. Therefore, the identification efficiency of GCT is equal to or above 50%.

Simulation Results

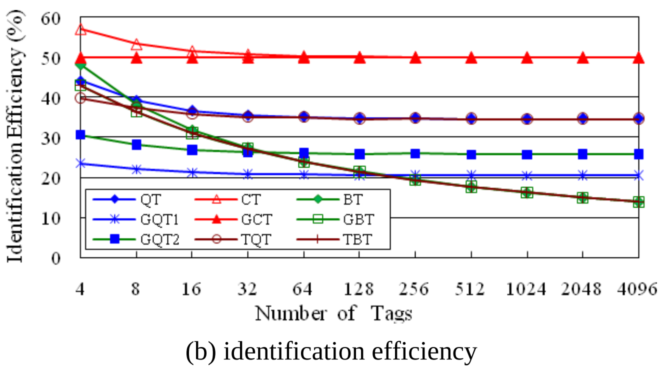
To verify the advantages of GCT, this section compares the performance of GCT with that of CT, QT, BT, GQT1, GQT2, GBT, TQT and TBT for different values of capture probability α. All RFID tags in the simulations have unique 96-bit IDs. The number of tags increases from 4 to 4096 and all tag IDs are distributed uniformly. The simulation results are illustrated in Fig.2 and Fig.3.

Fig.2 depicts the simulation results of GCT and the comparison protocols when no capture effect occurred during RFID tag identification, i.e., $\alpha = 0$. Fig.2(a) is the average time complexity (the average number of query-response cycle required for one-tag identification) of GCT and the other protocols. GQT1 and GQT2 increases average 2 or 1 cycles respectively for one tag identification compared to that of their original protocol QT. The average time complexity of the other general protocols (GCT, GBT, TBT and TQT) is similar with that of their original protocols (CT, BT and QT). GCT and CT require fewer cycles for one tag identification than the other protocols. The average time complexity of GCT is 2 cycles, while those of the other protocols are above 2.88 cycles.

Fig.2(b) illustrates the identification efficiency of GCT and the comparison protocols when $\alpha = 0$. The identification efficiency of GCT is always 50%, and is very close to that of CT. Similarly, the methods adopted in GBT, TQT and TBT not visibly impact the identification efficiency compared to their original protocols when no capture effect occurred during tags identification. But GQT1 and GQT2 reduce the identification efficiency to 20.5% and 25.8% because they increase the number of idle cycles when dealing with the hidden tag problem. This is because $2n$ and n idle cycles are increased in GQT1 and GQT2 respectively for solving the capture effect [8][10].



(a) average time complexity



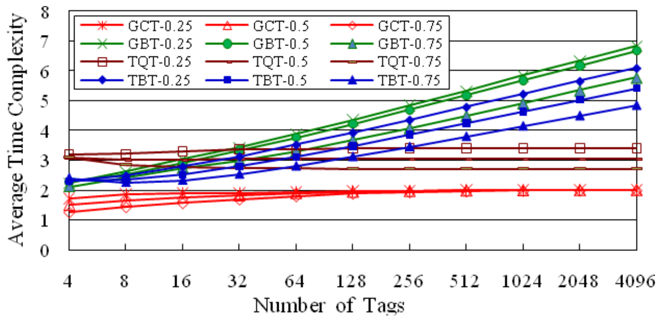
(b) identification efficiency

Figure 2. Identification performance of GCT when $\alpha = 0$.

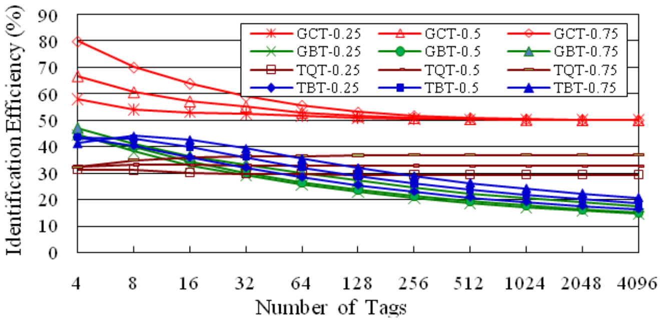
Fig.3 shows the simulation results of GCT, GBT, TQT and TBT when $\alpha = 0.25$, 0.5, and 0.75. The performance curves of GQT1 and GQT2 are omitted in Fig.3 because their performance is the same for all α and the results for $\alpha = 0$ are shown in Fig.2. As shown in Fig.3(a), the average time complexity of GCT is the lowest one among these protocols. With the increasing of α, the average time complexity of GBT, TQT and TBT decrease gradually, because the tags are divided into more groups that are smaller and the number of collided cycles is fewer in smaller groups than that in bigger ones.

The identification efficiency of GCT is always above 50% and outperforms that of the other protocols as indicated in Fig.3(b). The efficiency of TQT is better than that of its original protocol GQT2 (25.8%), and even above 36.7% when $\alpha = 0.75$. The efficiency of TBT decreases with increasing the number of tags, but it is still better than that of GBT. The efficiency of GBT, TQT and TBT are much lower than that of GCT because they product many idle cycles during tags identification.

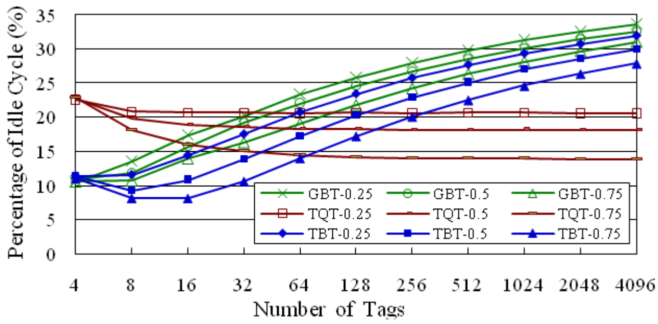
Fig.3(c) shows the percentage of idle cycle (the ratio of idle cycles to total cycles) of GBT, TQT and TBT. The idle cycle ratios of them decrease gradually when the capture probability increases from 0.25, 0.5 to 0.75. But with the number of tags increasing the idle cycle ratios of GBT and TBT also increase, while those of TQT trend to be 20.5%, 18.2%, and 13.9% respectively. It is worth to note that only one idle cycle exists in GCT, i.e., the last cycle, which is just used to confirm and indicate that there is no hidden and unidentified tag in the reader's range. This is also why GCT is not given in Fig.3(c).



(a) average time complexity



(b) identification efficiency



(c) percentage of idle cycle (%)

Figure 3. Identification performance of GCT when $\alpha = 0.25$, 0.5, 0.75.

Summary

This paper presented a novel RFID tag identification protocol called General Collision Tree Protocol (GCT) which can identify all RFID tags correctly and completely including hidden tags or missing read tags. GCT solved the capture effect and hidden tag problem by making modifications to the original protocol CT. This method used in GCT can also be applied to all other protocols based on CT.

Importantly, GCT inherited the advantage and characteristic of CT, such as lower system overhead, simple and easy implementation, and high performance, while increased the practicability and robustness by overcoming hidden tag problem caused by capture effect and other uncertainty factors. Therefore, GCT can be applied to many kinds of RFID and IOT systems, such as massive data collection, automatic production control, and large-scale warehouse management.

Acknowledgement

This research was financially supported by the National Natural Science Foundation of China (NNSF) under grant 61471306, 61601381, the Foundation of China Scholarship Council (CSC) under grant 201709390005, the National Key Research and Development Program of China under grant 2016QY04W0801, the Science and Technology Foundation of Sichuan Province under grant 2016ZC1720, 2017KZ0023, and the Scientific Research Fund of Sichuan Provincial Education Department under grant 18ZA0488.

References

[1] A. Magruk, The Most Important Aspects of Uncertainty in the Internet of Things Field - Context of Smart Buildings, Procedia Engineering, Elsevier, 2015, pp. 220-227.

[2] M. Bolic, D. Simplot-Ryl, I. Stojmenovic, RFID Systems: Research Trends and Challenges, Wiley, 2010.

[3] D.K. Klair, K.W. Chin, R. Raad, A Survey and Tutorial of RFID Anti-collision Protocols, IEEE Communications Surveys and Tutorials, vol. 12, no. 3, 2010, pp. 400-421.

[4] C.Law, K.Lee, K.Y.Siu, Efficient Memoryless Protocol for Tag Identification, in Proc. Fourth International workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIALM'00), New York, NY, USA, 2000, pp. 75–84.

[5] M. A. Bonuccelli, F. Lonetti, F. Martelli, Instant Collision Resolution for Tag Identification in RFID Networks, Ad Hoc Networks, Elsevier, vol. 5, no. 8, Nov. 2007, pp. 1220–1232.

[6] X.L. Jia, Q.Y. Feng, C.Z. Ma, An Efficient Anti-collision Protocol for RFID Tag Identification, IEEE Communications Letters, vol. 14, no. 11, Nov. 2010, pp. 1014-1016.

[7] X.L. Jia, Q.Y. Feng, L.Sh. Yu, Stability Analysis of an Efficient Anti-collision Protocol for RFID Tag Identification, IEEE Transactions on Communications, vol.60, no.8, Aug. 2012, pp. 2285-2294.

[8] V.K.Y. Wu, R.H. Campbell, Using Generalized Query Tree to Cope with the Capture Effect in RFID Singulation, IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, USA, Jan. 2009, pp. 1-5.

[9] Y.Ch. Lai, L.Y, Hsiao, General Binary Tree Protocol for Coping with the Capture Effect in RFID Tag Identification, IEEE Communications Letters, vol. 14, no. 3, Mar. 2010, pp. 208-210.

[10] C.T. Nguyen, A.T.H. Bui, V.D. Nguyen, A.T. Pham, Modified Tree-based Identification Protocols for Solving Hidden-tag Problem in RFID Systems over Fading Channels, IET Communications, vol. 11, no. 7, Jul. 2017, pp. 1132-1142.

[11] J. Su, Z.G. Sheng, G.J. Wen, V.C.M. Leung, A time efficient tag identification algorithm using dual prefix probe scheme (DPPS), IEEE Signal Processing Letters, vol. 23, no. 3, Mar. 2016, pp. 386-389.

[12] L.J. Zhang, W. Xiang, X.H. Tang, Q. Li, Q.F. Yan, A time- and energy-aware collision tree protocol for efficient large-scale RFID tag identification, IEEE Transactions on Industrial Informatics, Nov. 2017. [on line]