

An Efficient Approach to Detecting Missing Tags in RFID Data Stream

Nur'Aifaa Zainudin¹, Hairulnizam Mahdin², Deden Witarsyah³, Mokhairi Makhtar⁴,
Mohd Izuan Hafez Ninggal⁵, Zirawani Baharum⁶

^{1,2,6}Faculty of Computer Science and Information Technology, Universiti Tun Hussien Onn Malaysia

³Department of Information Systems, School of Industrial Engineering, Telkom University, Bandung, Indonesia

⁴Faculty of informatics and computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia

⁵Faculty of Computer Science and Information Technology, Universiti Putra Malaysia

Article Info

Article history:

Received May 20, 2018

Revised Jun 21, 2018

Accepted Jun 25, 2018

Keywords:

False Negative

Missing Tags

RFID

ABSTRACT

RFID technology is a Radio frequency identification system that provides a reader reading the data item from its tag. Nowadays, RFID system has rapidly become more common in our life because of its autonomous advantages compared to the traditional barcode. It can detect hundreds of tagged items automatically at a time. However, in RFID, missing tag detection can occur due to signal collisions and interferences. It will cause the system to report incorrect tag's count due to an incorrect number of tags being detected. The consequences of this problem can be enormous to business, as it will cause incorrect business decisions to be made. Thus, a Missing Tag Detection Algorithm (MTDA) is proposed to solve the missing tag detection problem. There are many other existing approaches has been proposed including Window Sub-range Transition Detection (WSTD), Efficient Missing-Tag Detection Protocol (EMD) and Multi-hashing based Missing Tag Identification (MMTI) protocol. The result from experiments shows that our proposed approach performs better than the other in terms of execution time and reliability.

Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Hairulnizam Mahdin,

Faculty of Computer Science and Information Technology,

Universiti Tun Hussien Onn Malaysia.

Email: hairuln@uthm.edu.my

1. INTRODUCTION

RFID technology uses radio-frequency waves to automatically identify people or objects. Based on this market and trajectory, it is expected that RFID will become the most common pervasive devices in our life [1]. RFID has been used in a wide variety of applications such as apparel [2], supply chain management [3], transporting system [4] and food industry [5]. However, there are still challenges to be solved in an RFID system, and one of them is the missing tag detection problem. Missing tag detection can happen due to signal collision [6], electrical interference [7] and unknown resources [8]. When an RFID reader missed detecting a tag, it will affect the data integrity of the system. The system will send an incorrect data to the backend processing and it will feed user with misleading reports. For example, in a supermarket, the reader is installed on a rack to read all the tagged items [9], [10]. When a reader misses reading the items, it will feed the system with the incorrect total which is less than the actual. The less number of items will trigger the system to instruct respective personnel to do item replenishment on the rack. The wrong instruction has been issued here because of the incorrect readings that have been made by readers. In another example, in some factory, they implement RFID to automate their system. When RFID miss read the item moving on the conveyor, the system will trigger incorrect action to the item at the next stop in the production. In case of car assembly plant [11], incorrect paint colour may be applied to the wrong parts because of the

misreading problem. In the next section, this paper will discuss in the background of the problem, in section III the current approaches to solving the problem is presented, while section IV will discuss the simulation results and section V is the conclusion.

2. RELATED WORKS

There are many approaches has been proposed to solve this problem such as Window Sub-range Transition Detection (WSTD) [12], Efficient Missing-Tag Detection Protocol (EMD) [13]and Multi-hashing based Missing Tag Identification (MMTI) protocol [14]. However, simply detecting the missing-tag event is not enough. In addition, we need to obtain the detailed information about the missing items so as to assess the seriousness of the loss and take different countermeasures.

2.1. Background

Based on a survey in 2014, the total market value for RFID is \$9.2 billion and is expected to reach \$30.24 billion in 2024 [15]. The main advantage of RFID technology is automated identification. However, there are still problems regarding RFID reading including noise, duplicate and missed reading. In this paper, we focused on solving the missing data problem. Missed reading will cause object being missed from being detected. Missing tag detection can happen due to the sensitivity of RFID tag and reader performance to the operating environment [16].

Figure 1 illustrates the missing tag scenario in RFID system. The tags can be missed to be read at certain checkpoint causing the incorrect count of items being reported to the system. The missed read could indicate either the items are really missing because of theft, misplaced or system error. In the next section we will look into the existing approaches.

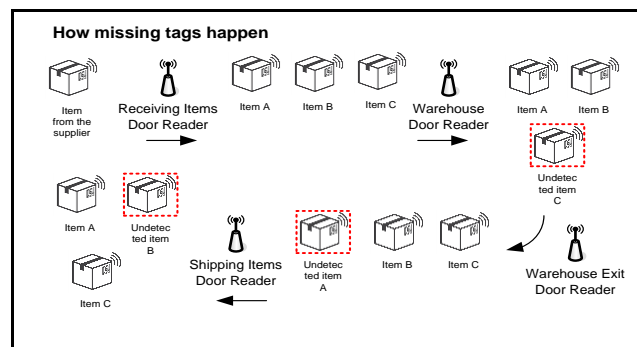


Figure 1. RFID missing tags scenario

2.2. Window Sub-range Transition Detection (WSTD)

In WSTD [12], the goal is to give each tag opportunity to be read within the window by reducing or eliminate dropped readings. WSTD is used as a data cleaning mechanism for low-level RFID data processing tasks within middleware system. WSTD requires two opposing application requirements which are to ensure completeness for the set of tag readings due to tag reader system unreliability. Large window sizes are good for ensuring completeness by smoothing out the missed readings. However, it's not efficient in detecting tag transitions.

The second requirement is capturing tag dynamic based on tag movement in and out of the reader's detection region. Small window sizes are able to detect tag transitions, but they are not capable of compensating the missed readings. Small window leads to false negative errors which the tag is mistakenly assumed to be absent while it actually present.

For missing tags, the variation within the window also become the cause and not only due to the transition. To reduce the number of false positives and false negative due to the transition, the WSTD is reducing the window size. WSTD also slide its window per single read cycle and produces output by reading the corresponding to the midpoint of the window after the entire window has been read. If the tag is moving out and was not detected in the second half of the window size, the tag is assumed exists in the detection range. However, the weakness is that the premature exit transition detection will lead to a false negative reading due to a small window size.

2.3. Efficient Missing Tag Detection Protocol (EMD)

Proposed by [13], this protocol objective is to reduce the execution time of the missing tag detection protocol. For EMD, the solution is focusing on the similarity of two small subsets of tags in a large RFID system. The similarity also has a large probability of sharing a common tag. An example, let M be the set of m missing tags, K be a subset of k tags that the reader randomly selects from the inventory list N of n tags currently in the system. The reader performs the operation to verify the presence of these k tags. The IDs of these tags it transmits one after another. The reader waits for a short period and listens for a response after transmitting and ID. After the tag is received its ID, it will acknowledge its presence by sending a response. If the reader does not receive any response back from ID, it reports the missing tag event. If k is reasonably large, K and M will have a good chance to share at least one common tag. The reader will find that the presence of at least one tag in K cannot be positively confirmed. Hence, the missing tag event is detected.

The EMD is that to address the limitations of the intermediate protocol. The RFID reader initiates the protocol execution by broadcasting a polling request. The request, each tag decides with a sampling probability p whether to participate in the polling. It will randomly select a slot in the subsequence frame to respond when it decided to participate. If it not to participate, it will enter to sleep mode and turn out at the next scheduled time for the protocol execution. All the decisions are made pseudo-randomly and predictable by the reader.

However, the limitations of this proposed first are not time-efficient. It is because it takes an amount of time to verify the presence of each selected tag although a subset of tag IDs is selected. Second, each selected tag makes just one short transmission, but it has to receive a large number of bits which makes the aggregate receiving energy significantly.

2.4. Multi-Hashing Based Missing Tag Identification (MMTI)

To increase the proportion of the expected singleton slots is by improving the utilization of time frame. Multi-hashing based Missing Tag Identification (MMTI) proposed by [14] is to reduce the proportion of expected empty slot and expected collision slot. The proposed challenge is how to guarantee that the achieved singleton slots will not be selected in the next hashing process.

Specifically, the slot occupation states of the frame are predictable to the reader. It could construct which '1' indicate the singleton slots that cannot be selected in the next hashing process and '0' indicate the collision or empty slot that can be selected in the next hashing process. The reader will guide it through the next hashing process.

The MMTI is based on the slotted aloha communication mechanism. The communication between the reader and tags is in a time-slotted away. The readers synchronize the slots by broadcasting the *end_slot* command. Each tag has a *slot_clock* which is initialized with a random slot number. A tag counts its *slot_clock* one each time when the reader indicates that the current slot has ended. A tag responds when its *slot_clock* reaches zero. If each tag response is at least 10 bits, the reader can categorize the type of slot. First, the empty slot in which no tag responds and second is the collision slot in which more than one tag responds. Otherwise, if the tag response is less than 10 bit, the readers can categorize two kind types of the slot which is *idle* slot were is no tag response and the *busy* slot is at least one tag responds.

However, due to the channel error which is path loss, its response is not sensed by the reader, and then it is wrongly considered as a missing tag. On the other hand, if an expected singleton slot corresponds to an actually missing tag, due to the channel errors, this missing tag is wrongly verified as a present one, namely the false negative.

2.5. Intersection Algorithm

Intersection algorithm [17] is for comparing the RFID data between the two readers. The situation that often happens is low-power hardware where false negative usually happen when the multiple tags are didn't detected and will cause the reader to identifying the current tags that frequently dropped reading by the readers. The detection is through time slot when the reader signal will organize the clocks of tags were 1 as single slot, 1 or more than 1 is collision slot and 0 as an empty slot. Intersection algorithm will compare the EPC data between the two readers without including the empty slot for detecting the false negative reads.

The intersection plays a role where the data will store by setting up the array. Figure 2 illustrates the intersection is when there will be two readers as an input which is Reader1 and Reader2. Begin with reader read every data of Reader1 and Reader2. If every data from Reader1 is equal with every data from Reader2 then the data will be placed into the Set A as a new output. Set A is a new set of data that merge the same data between data of Reader1 and Reader2.

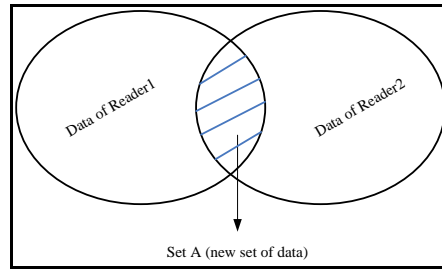


Figure 2. Intersection data between the two readers

Nevertheless, the research is focusing on detecting false negative reads by the reader. This algorithm is only comparing the data between the two readers to find out if same data exist in both readers and will allocate into a new set. After all, the algorithm is selected for to comparing the data to find out the different data that exist among the readers.

2.6. R-PRN Algorithm

R-PRN algorithm [18] is used removing the false negative reads in the RFID data reader. The R-PRN algorithm starts the process set the pre-condition and Timestamp for the algorithm identify the status of RFID data. The algorithm can be identified when the slot is '0' for empty slot and '1' for the available slot. Set t as time for every data where $(t - 1 = 1; t + 1 = 1; t = 0)$ including the precondition = 1 or Timestamp = 1, the algorithm will detect the data as false negative read. Otherwise if $(t - 1 = 0; t + 1 = 0; t = 1)$ including precondition = 0 or Timestamp = 1 the algorithm will detect false positive read.

Figure 3 shows how the algorithm detects the false negative and false positive read. However, this algorithm is used for removing the false negative reads. The algorithm wants to clean the data in the RFID data to deal with anomalies false negative of high accuracy and less complexity. In this research, the algorithm is selected to merge with intersection algorithm. The algorithm is used for detecting the false negative reads from the new set of data that has been allocated by the intersection algorithm.

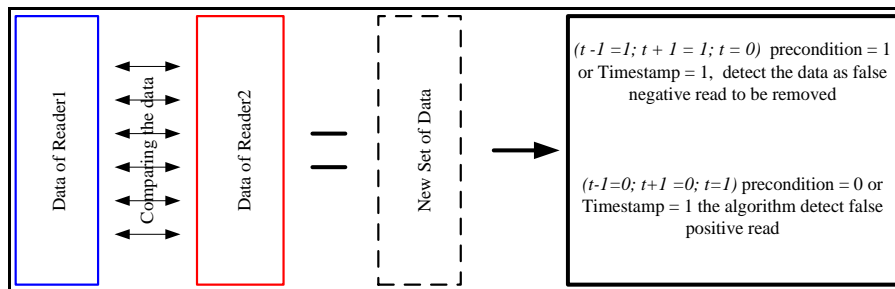


Figure 3. R-PRN algorithm detecting the false negative and false positive read

3. +PROPOSED ALGORITHM

The proposed algorithm is included intersection algorithm [17] with hashing for comparing the RFID data tags meanwhile R-PRN algorithm [18] will use for detecting false negative read. The situation that often happens that frequently dropped reading in low-power hardware where false negative usually happen when the multiple tags cannot detect.

The intersection with hashing plays a role where the data will store by setting up the array. The description for the intersection with hashing is when setting a as first array and b as the second array. Meanwhile for the length of the first array will be set as n and second array the length will set as m . If $(n = m)$ the result will be equal where the data from n are same with data in m .

However, the hashing will help the intersection algorithm be more easier when the $i = 1$ will set into n , the hash will contain $a(i)$ then $hash[a(i)] = hash[a(i)] + 1$. Otherwise, $a(i)$ will be mapped to 1 to the hash. For second array, $j = 1$ will set into m , with the hash contain $b(j)$ and $hash[b(j)] > 1$. Then $b(j)$ will append to intersection where $hash[b(j)] = hash[b(j)] - 1$. Otherwise, if $hash[b(j)] AND hash[b(j)] = 0$ append into intersection or remove $b(j)$ from the hash.

After data has been hashing, the R-PRN algorithm will start the process to identify which one of the data is the false negative. The algorithm can be identified when the slot is '0' for empty slot and '1' for the available slot. Set t as time for every data in hash where ($t - 1 = 1$; $t + 1 = 1$; $t = 0$) including the precondition = 1 or Timestamp = 1, the algorithm will detect the data as false negative read. Otherwise if ($t - 1 = 0$; $t + 1 = 0$; $t = 1$) including precondition = 0 or Timestamp = 1 the algorithm will detect false positive read. Figure 5 shows the intersection algorithm with hashing merge with the R-PRN algorithm.

The algorithm in Figure 4 is showing the Missing Tag Detection Algorithm to detect the false negative read in the intersection sets. From line 1 until 5 is the intersection to place the data into the array. IF the first array n is bigger than the second array which is m , the data will be placed into the intersection new set. Otherwise, the first array will be hash into the hash table. In line 6, it will process the data into the hash table when i will be set as 1. Line 7, the data will be hash as $a(i)$ THEN the $a(i)$ has been inserted into the empty array and map i to 1 as show in line 10. In line 11, the data will be hash the second array that set as j . IF the data from the m is contained $b(j)$ AND its bigger than 1 , the data will be sent to the intersection new set as shown in line 13 and 14. Otherwise, when the $b(j)$ is 0, the data will be placed to the intersection set for the second array as shown in line 15 and 16. From the line 1 until 17, is the process of data to be hash to the hash table and send it to the intersection set to be compared.

```

INPUT: Reader 1, Reader2, Reader3, Reader4
STATUS: 0 (not available)
1(available)
FUNCTION Intersection (n, m)
  n - Length of array set 1
  m - Length of array set 2
BEGIN
1: intersection - resulting array
2: IF (n > m)
3: intersection = Intersection( m, n);
4: ELSE
5: hash – hash table
6: ENDIF
7: FOR i = 1 to n
8: IF hash contains a(i) THEN
9: hash[a(i)] = hash[a(i)] + 1
10: ELSE
11: ADD mapping (i) -> 1 to hash
12: ENDIF
13: ENDFOR
14: FOR j = 1 to m
15: IF hash contains b(j) AND hash[b(j)] > 1 THEN
16: b(j) to intersection
17: hash[b(j)] = hash[b(j)] + 1
18: ELSEIF hash contains b(j) AND hash[b(j)] = 0 THEN
19: b(j) to intersection
20: ENDIF
21: ENDFOR
22: FOR (every tag in reader x(x = n, m) at time t) DO
23: IF (t - 1 = 1; t + 1 = 1; t = 0) AND (Precondition = 1 OR Timestamp = 1) DO
24: RETURN "False Negative to be detected";
25: ELSEIF (t - 1 = 0; t + 1 = 0; t = 1) AND (Precondition = 0 OR Timestamp = 0)
26: RETURN "Non False Negative data";
27: ENDIF
28: ENDFOR
29: RETURN intersection;
30: END
END

```

Figure 4. Missing Tag Detection Algorithm (MTDA)

Starting from line 19 until 25, the process will detect the false negative read from the intersection set. In line 19, every data from the first and second array will be gathering. The data will be scan by following the time which set as t . The conditions is set as $t = 0$, $t - 1 = 1$ and $t + 1 = 1$ as show in line 20. The conditions are set to figure out which data is a false negative read. When the data is fulfilling the requirement, the data will be listed as a false negative read from the intersection set as shown in line 21. However, if the data is having a criteria $t - 1 = 0$, $t + 1 = 0$ and $t = 1$ as shown in line 22, the data will be listed

as a non-false negative as shown in line 23. The data is distinguished between the set in the first and second array of intersection set. In Table 1 show the comparison between the approaches.

Table 1. The Difference Between the Approaches

Year	Approach	Advantage	Disadvantage
2007	Intersection Algorithm	<ul style="list-style-type: none"> - The data has been separated by the intersection - The data can be compared to the readers 	<ul style="list-style-type: none"> - Collision slot still available - Cannot detect which tags are missing
2011	EMD	<ul style="list-style-type: none"> - The protocol allows energy-time trade-off that achieve the minimum execution time - Smaller energy cost - Improve transition detection mechanism - Use binomial sampling concepts to calculate the appropriate window size 	<ul style="list-style-type: none"> - Just for one short transmission but will receive a large of bits numbers
2012	WSTD	<ul style="list-style-type: none"> - Use comparison for two windows sub-range - Uses smaller window size - Takes shorter processing time. 	<ul style="list-style-type: none"> - Produces slightly more negative errors - Competitive performance in compensating for missed readings.
2013	R-PRN Algorithm	<ul style="list-style-type: none"> - Detecting other than false negative using pre-condition or timestamp - To reduce the expected empty slots and the expected collision slots. 	<ul style="list-style-type: none"> - Removing false negative reads - Collision slot is still exist
2014	MMTI	<ul style="list-style-type: none"> - Not suffer the limitation of storage. - Better utilize the time slots thereby relieving the deficiency of the IIP scheme. 	<ul style="list-style-type: none"> - The channel errors often degrade the performance MMTI or give rise to the false of the identification results. - Consume more time in the scenarios with channel errors

According to [19], the problem of missing tags with a large number of tags is yet under-investigated by the research community. However, there are many existing works that concentrate on the missing tag detection problem [15], [20]. As we discussed earlier, by ignoring the impact of false negatives on the performance of RFID systems may lead to an overestimation of the benefits achievable with RFID [21].

4. SIMULATION RESULT

The purpose of this simulation is to measure the performance of the searching scheme by using Matlab R2017b. The performance of the searching scheme will be measured in term of time execution and accuracy on detecting false negative rate. The result of this simulation was compared with Window Sub-range Transition Detection (WSTD), Efficient Missing-Tag Detection Protocol (EMD) and Multi-hashing based Missing Tag Identification (MMTI) protocol. A slight modification is in intersection algorithm and R-PRN algorithm for the Missing Tag Detection Algorithm (MTDA) to solve the problem. Data samples are prepared and tested in each algorithm. The result that gained from simulation will be analyzed in term of time execution, the rate of false negative reading to determine the performance of each algorithm. The result from simulation has been showing into tables and charts. There are two criteria measuring for this simulation:

- a. Time execution analysis
- b. The rate of false negative detection

As a result, this will show the significant of the detecting false negative in RFID system performance for each algorithm. The result that been shown is the details analysis of the simulation result for each algorithm. By comparing the result between the proposed algorithm with other methods is shown that the proposed algorithm get the advantage of others algorithm.

4.1 Time Execution Analysis

Time execution for detecting false negative is measure through elapsed time for every second false negative has been detected for each algorithm by using Matlab R2017b as a platform. The analysis conducted by comparing elapsed time for each percentage. The analysis has been divided between the first RFID reader's data set as *Reader1*, second RFID reader's data set as *Reader2* and third RFID readers' dataset as *Reader3* that has been compared with the starting point reader. The result shows that the elapsed time between they reduce slightly.

The elapsed time between the comparison algorithm has been tested and be recorded in Figure 5. The time execution for differentiate between the reader are following the ratio 10%, 20%, 30%, 40% and

50%. Differentiate for the Reader1, is illustrate that the proposed algorithm has the best time running for detected false negative read. The table shows that the proposed algorithm is running better than other existing approaches.

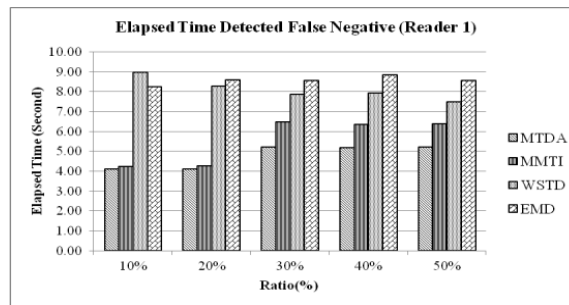


Figure 5. Time execution of detected false negative for reader1 ratio

Besides that, the difference between Reader2 data with the fourth set of data is showing that the time execution for the MTDA continues the momentum to be the best time for detected false negative. However, MMTI almost reaches the running time with the proposed algorithm. Different with WSTD and EMD that still require more time than MMTI and the proposed algorithm. The result is illustrated in Figure 6.

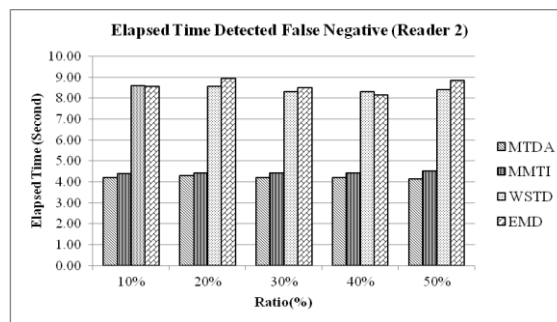


Figure 6. Time execution of detected false negative for reader2 ratio

Time execution comparison between Reader3 data with the fourth set of data is shown in Figure 7. The table shows that once again that the MTDA has the best run time for detected false negative read but MMTI algorithm shows that the algorithm can reach a fastest running time for detecting the false negative. Meanwhile, WSTD and EMD are still taking a long time for detected false negative.

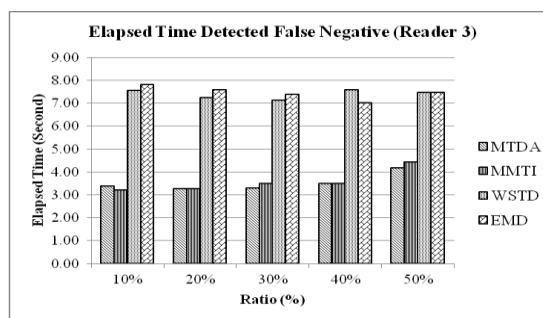


Figure 7. Time execution of detected false negative for reader3 ratio

4.2. Rate of False Negative Detection

The comparison for false negative detection rate has been recorded for each *Reader1*, *Reader2* and *Reader3*. The result also shows the missing tags from ratio 10%, 20%, 30%, 40% and 50%. Figure 8 show the results of successful detected false negative in the *Reader1* data sets. The table shows that MTDA achieves 100% of false negative read follow by MMTI in 10% ratio. However, MMTI algorithm has nearly achieved in ratio 30%. However, WSTD also can achieve detected false negative in ratio 10% and 20% and EMD in ratio 20% and 30%.

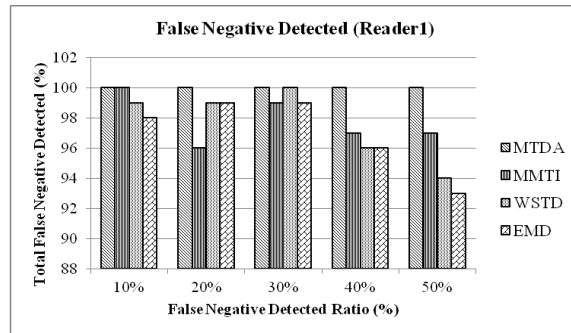


Figure 8. Total of detected false negative for *reader1* ratio

Figure 9 illustrate the result comparison between the *Reader2* data with the fourth set of data in detected false negative. For the *Reader2* percentage is show that in all ratio is detected achieve 100%. However, comparing with MMTI algorithm show that the detecting false negative is achieves 100% in ratio 10%, 20% and 50%. Meanwhile, the WSTD nearly reach 100% in ratio 20% and EMD almost achieve perfect percentage where the higher detect is 99% in ratio 20%.

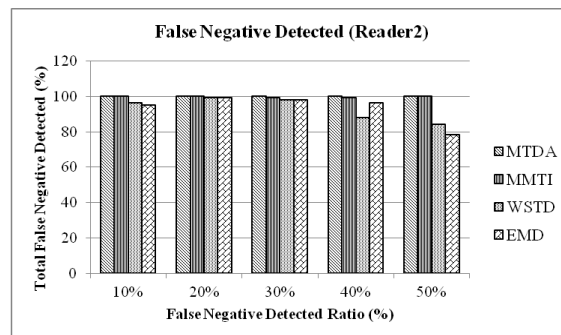


Figure 9. Total of detected false negative for *reader2* ratio

The time execution of MTDA is faster compare to MMTI, WSTD and EMD. However, there are certain parts that MMTI shows that the algorithm also can perform well if there are some improvements. Even thought MMTI can perform faster but the detection of false negative read is less than MTDA. This is because MTDA incoming data has been hash into the intersection sets before the data can be detected as false negative read different with MMTI where the algorithm is just hashing in the table without comparing it first for detecting the false negative reads.

The comparison between the *Reader3* with the fourth set of data to detect the false negative has been illustrated in Figure 10. For the *Reader3* results show that the MTDA has achieved 100% successful of detected false negative read. However, the MMTI in the ratio 10% the result shows that the successfulness for detect false negative is 100%. However, for the *Reader3* surprisingly show that EMD has achieved successfulness detect false negative in ratio 20% and 30% that is the same result with MMTI algorithm with 99%. Meanwhile, WSTD can't achieve the successfulness when the detection is higher only achieve 98%.

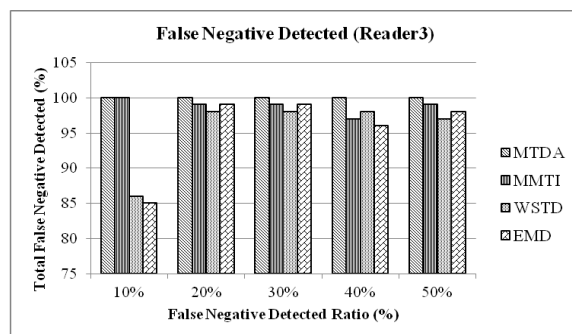


Figure 10. Total of detected false negative for *reader3* ratio

Besides that, WSTD and EMD also can achieve successfulness for detecting false negative read. However, there are too slow to compare with MTDA and MMTI. For WSTD has used sliding window approach to detect the false negative reads. This algorithm took a long time to detect the false negative reads. For EMD algorithm to detect the missing tags, the algorithm used the time frame that contains a slot. If the frame is large so the time to take for detected false negative is more. Also, that could be some of the false negative read are left behind and cannot to detect by the algorithms.

5. CONCLUSIONS

As RFID has become cheaper and more popular, we are bound to see more uses of the technology in many more fields. However, the missed tags reading problems need to be resolved to ensure user get the full use of RFID. This problem is affecting the efficiency and the accuracy of the data supplied by RFID. Even with a low number of incorrect data, RFID system can be considered useless. This paper presents some of the approaches that have been used to solve the problem of missing tags. However, there are still gaps to be solved in order to increase the effectiveness and the efficiency of the approaches. The future work that can be considered to solve the problem is by using MTDA that used intersection algorithm, which is to compare the tag existence between the readers and combine it with the R-PRN algorithm which is to detect the false negative read that is among the cause of missed readings.

REFERENCES

- [1] Kamaludin, Hazalila, Mahdin. H, Jemal H. Abawajy. *Filtering redundant data from RFID data streams*. Journal of Sensors 2016 (2016).
- [2] Liu, Xuan, Bin Xiao, Shigeng Zhang, Kai Bu, Alvin Chan. *STEP: A time-efficient tag searching protocol in large RFID systems*. *IEEE Transactions on Computers* 2015; 64(11): 3265-3277.
- [3] Yu, J., Chen, L., Zhang, R., Wang, K., Finding needles in a haystack: Missing tag detection in large RFID systems. *IEEE Transactions on Communications* 2017; 65(5), 2036-2047.
- [4] Deng. J. *Architecture Design of the Vehicle Tracking System based on RFID*. TELKOMNIKA Indonesian Journal of Electrical Engineering. 2013; 11(6): 2997 – 3004.
- [5] Evizal A.K, Shamsudin S.M, Supriyanto E., Sutopo W & Rosa S.L. Food Traceability in Supply Chain Based on EPCIS Standard and RFID Technology. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2015; 13(1):187-194.
- [6] Chen, W. T., Optimal frame length analysis and an efficient anti-collision algorithm with early adjustment of frame length for RFID systems. *IEEE Transactions on Vehicular Technology* 2016; 65(5): 3342-3348.
- [7] Zhao, J., Li, W., & Li, D. A., *Identifying the missing tags in categorized RFID systems*. International Journal of Distributed Sensor Networks 2014; 10(6), 582-951.
- [8] Ma, C., Lin, J., & Wang, Y. (2012, June). *Efficient missing tag detection in a large RFID system*. IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). 2012:185-192.
- [9] Xiong, Z., Song, Z.Y., Scalera, A., Sottile, F., Tomasi, R. and Spirito, M.A., *Enhancing WSN-based indoor positioning and tracking through RFID technology*. Fourth International EURASIP Workshop on RFID Technology, 2012:107-114.
- [10] Xie, L., Yin, Y., Vasilakos, A. V., & Lu, S. Managing RFID data: challenges, opportunities and solutions. *IEEE communications surveys & tutorials* 2014;16(3),1294-131.
- [11] Chen, M., Luo, W., Mo, Z., Chen, S., & Fang, Y., An efficient tag search protocol in large-scale RFID systems with noisy channel. *IEEE/ACM Transactions on Networking* 2016; 24(2), 703-716.

- [12] Massawe, L.V., Vermaak, H. and Kinyua, J.D., *An adaptive data cleaning scheme for reducing false negative reads in RFID data streams*. IEEE International Conference on RFID, 2012:157-164.
- [13] Luo, W., Chen, S., Li, T., & Chen, S., Efficient missing tag detection in RFID systems. Proceedings *IEEE INFOCOM, 2011*:356-360.
- [14] Liu, X., Li, K., Min, G., Shen, Y., Liu, A. X., & Qu, W., A multiple hashing approach to complete identification of missing RFID tags. *IEEE Transactions on Communications* 2014, 62(3), 1046-1057.
- [15] Wang, L., Gu, T., Tao, X., & Lu, J. Toward a wearable RFID system for real-time activity recognition using radio patterns. *IEEE Transactions on Mobile Computing*, 2017; 16(1), 228-242.
- [16] Zhu, X., Mukhopadhyay, S. K., & Kurata, H., *A review of RFID technology and its managerial applications in different industries*. Journal of Engineering and Technology Management 2012;29(1), 152-167.
- [17] Pupunwivat, P., & Stantic, B., *Location filtering and duplication elimination for RFID data streams*. International Journal of Principles and Applications of Information Science and Technology 2007, 1(1), 29-43.
- [18] Leema, A. A., & Hemalatha, M., *Proposed prediction algorithms based on hybrid approach to deal with anomalies of RFID data in healthcare*. Egyptian Informatics Journal 2013, 14(2), 135-145.
- [19] Zhu, Y., Jiang, W., Zhang, Q., & Guan, H., Energy-efficient identification in large-scale RFID systems with handheld reader. *IEEE Transactions on Parallel and Distributed Systems* 2014; 25(5), 1211-1222.
- [20] Shahzad, M., & Liu, A. X. *Expecting the unexpected: Fast and reliable detection of missing RFID tags in the wild*. IEEE Conference on Computer Communications (INFOCOM), 2015; 1939-1947.
- [21] Zheng, Y., & Li, M. Fast tag searching protocol for large-scale RFID systems. *IEEE/ACM Transactions on Networking (TON)* 2013; 21(3), 924-934.