VOL. 11, NO. 24, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2016 Asian Research Publishing Network (ARPN). All rights reserved.

www.arpnjournals.com

AN OVERVIEW ON VARIOUS RFID DATA FILTERING TECHNIQUES BASED ON BLOOM FILTER APPROACH

Siti Salwani Yaacob and Hairulnizam Mahdin
Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Parit Raja, Johor, Malaysia
E-Mail: hairuln@uthm.edu.my

ABSTRACT
The unreliability data reading such as noise, missed reading and duplicate reading that were produced by RFID reader has become the primary factor limiting the widespread adoption of RFID technology. It is compulsory to filter the raw data to maintain its reliability of data reading because a slight decreasing of effective read rate will reduce the accuracy and reliability of RFID. One of the approaches that used to filter data in RFID is Bloom filter. Bloom filter is a probabilistic data structure that checks whether the data in the filter or not. It is a typical approach that filters incorrect raw data, such as noise and redundancy in the distributed system. This paper provides overview of various techniques that used Bloom filter approach to filter RFID data reading to maintain its reliability and efficiency.

Keywords: RFID, data filtering, noise reading, missed reading, duplicate reading, bloom filter.

INTRODUCTION
Radio Frequency Identification (RFID) is a technology that uses radio waves to transfer detecting data between readers and tagged objects from a distance without line of sight. The RFID system essentially consists of RFID readers the tagged objects and antenna. Through radio frequency waves, readers can read tags in a short range of time. The greatest advantage of RFID deployment is, it allows wireless interaction between tagged objects and readers to automatically identify large groups of items simultaneously. Despite of the various advantages of RFID system, incorrect data generated by the reader become challenging issues in the RFID technology [1-2]. RFID data contain redundant, missed and unreliable data because of the various parallel transponders present in the vicinity of the reader. It is compulsory to identify the presence of tag correctly because the RFID environment is not clean. There are several types of reading that the reader encounters; true positive readings, true negative readings, false negative readings, false positive readings and duplicate readings [3].

BASIC OF DATA FILTERING
The process of data filtering takes place at RFID middleware. There are two types of filtering: (i) low level data filtering [4] and (ii) semantic data filtering [5]. At low-level data filtering, it cleans raw RFID data streams. The semantic data filtering filters data based on demands from system such as list of manufacturers that did the shipment in July. In this paper, we will discuss on low-level data filtering, which focus on removing noise and duplicate readings. Figure 1 depicts the basic filtering concept of noise reading and duplicates readings. The problem of noise data filtering in (A) is when noise TXX has been removed, the duplicate readings of T1 and T2 are still exist in the data stream. The same scenario occurred in (B), where the noise reading still exists as the duplicate readings T1 and T2 has been removed. As RFID data arrives rapidly and in high capacity, RFID readers play a critical role in order to filter out incorrect RFID data readings. It has been the major hindrance for RFID to be implemented in large-scale that it could generate over 30% of incorrect data reading [6].

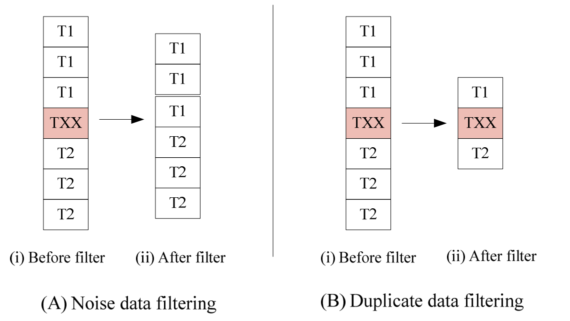

Figure-1. Noise data filtering and duplicate data filtering.

THE BLOOM FILTERS
A Bloom filter is a space-efficient probabilistic data structure that tells either the data is in the set or not [7]. Figure-2 shows a basic structural design of Bloom filter. It represents data in its bit array of size m using k number of hash functions. Whenever the data has been hashed, all bits in array that are initially set to 0 will be substitute to 1.

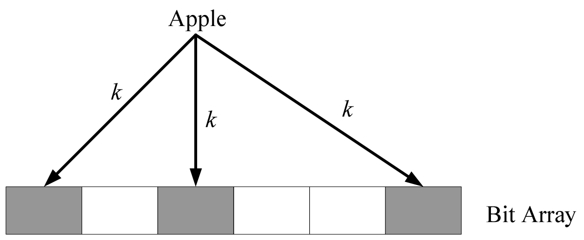

Figure-2. Basic structure of bloom filter.

The basic Bloom filter supports two operations: test and add. The test is used to check whether a given element is in the set or not. If it returns false, then the element is definitely not in the set. If it returns true, the

VOL. 11, NO. 24, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences
©2006-2016 Asian Research Publishing Network (ARPN). All rights reserved.

element is probably in the set. Add, simply adds an element to the set. To add an element to the Bloom filter, we feed it to the *k* different hash functions and set the bits in the resulting result. To test if an element is stored in the filter, again we feed it to the same k hash function. If one or more of these bits is not set, then the queried element is definitely not present in the filter.

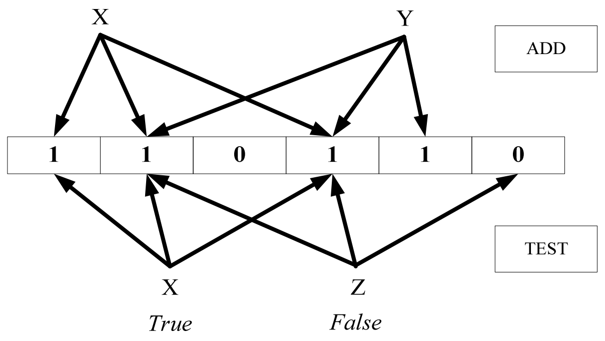

Figure-3. The operation of bloom filter.

Figure-3 visualizes how a Bloom filter operates. The Bloom filter simply adds data, such data *x* and data *y* in the Bloom filter. If any of the bits are 0, for example data *z*, then the element definitely does not exist in the filter. If all of the bits are 1, there is probably that the element exists in the filter. Generally, Bloom filter has been used to filter duplicate data. Removal and deletion are not allowed in normal Bloom filter. This is for the reason that a single counter in Bloom-filter can be hashed number of times with different data. Turning counter to 0 will disturb other data that are not involved in the deletion. Previous research shows that the Bloom filter has been extended to allow deletion in order to filter duplicate readings and noise readings in RFID data streams. Deng *et al.* [8] has used Bloom filter approach to eliminate duplicate data in the data stream applications. The bits in the regular Bloom filter are changed into cells consisting one or more bits. In order to eliminate old data, each cell is set to the maximum value and decrement the values of randomly selected cells whenever data arrives. However, this approach has still produced false positive errors and false negative errors.

Xiaowei *et al* [9] has proposed local duplicates filtering and global duplicates filtering in order to filter local duplicates within each stream to overcome their space and time cost feasibility for high-speed data streams. There are remote sites and coordinator site. Each remote site, the duplicate data readings are filtered by using Bloom filter technique and then send clean data to the coordinator site. Each local stream has both local duplicates and global duplicates. Each local filtering can only filter all local duplicates, but not global duplicates. In order to filter both local and global duplicates, each remote site must maintain the history of stream because each remote site can only communicate with the coordinator site. The remote site must share global history via the coordinator site. To share the global history, the coordinator site must update its filter with the new reading and sent it to the other remote site. If the same reading being produced again at any of the remote sites, it will be ignored because it has been inserted in the filter. As RFID generates a flood of data, this approach is not suitable because it need to update the coordinator site each time new reading is entered. This process can cause bottlenecks as data at the remote site need to be filtered before send clean data to the coordinate remote for querying process.

Mahdin *et al.* focuses their works to filter duplicate reading and redundancy in RFID data stream [10-11]. These approaches consider the duplicates and redundancy reading at the reader level. However, this approach has low false positive rates, which illustrates the improved correctness of the filtering process and have better performance. Lee *at al.* [12] has extends original Bloom filter to support sliding window and proposed time Bloom filter in order to detect duplicate data reading on RFID data streams. As the process of filtering duplicates took place at the server side, a lot of bandwidth wasted during transfer the duplicates. Hence, three algorithms; Bloom filter, time Bloom filter and time interval Bloom filter were proposed to eliminate each duplicate data arrive. The time interval Bloom filter was used in fault detection and elimination and this algorithm need more space than the time Bloom filter. In this research, the time interval Bloom filter needs more space, compare with time Bloom filter. Time Bloom filter depends on time information to check whether the data is duplicated or not. Even though it does not create false negative errors, the major problem with this technique is bottleneck will occur as the data has to pass through this module in the server side.

Jiang *et al* [13] proposed two-layer filtering approach to eliminate the local duplicates at the reader level, then detecting the global duplicates in the server side. At the reader level, local duplicates will be filtered before the data enters at the server site. Next, all global duplicates will be removed at the server site. However, this approach produces a small error rate and caused latency at the reader level as it need to clean the duplicate reading before it have to pass through to global duplicate filtering.

Yongsheng *et al.* [14] proposed a data filtering approach to detect and eliminates duplicates data reading in RFID. They proposed a technique that used counting Bloom filters with the dynamic chain lists to detect and removed duplicates reading at the data level. The dynamic chain is proposed as a linked list. They used an array of time information to update the expired historical data. By this way, they can detect the duplicates in RFID. The number in counter position will be incremented and decremented when input is added or deleted in the filter. The bit in bit vector is set whenever counter changes from zero to one. While bit in the bit-vector is clear whenever the counter changes from one to zero. However, this approach is still not suitable for RFID application because the reader will generate too many readings. Whenever a duplicate detected, they have to find a match in the double linked list. Hence, there will be a latency to find a match during searching duplicate data before they can eliminate the duplicate data.

Mezzi *et al* [15], has proposed an algorithm that detects and removes duplicates reading in data streams in order to store into data warehouse to enhance the decision making for supply chain. They proposed a technique to detect and eliminate the duplicate by looking at the difference of time of the flow detection and the time of the previous stream detection. However, the technique is not suitable to process massive data of RFID. The data has to process with the difference of time to detect and eliminate the duplicates of data. This will slow down the process of filtering and it will disturb the efficiency of decision making for supply chain.

From this overview, an efficient data filtering technique should perform fast screening to detect and remove unwanted RFID data such as data redundancy and noise readings. The multiple layering process such as [9] [12-13] often requires large bandwidth and latency. Bandwidths in RFID data stream can be defined as how much the reader can transfer data from the source to the server. While latency can be defined as how fast data can be transferred from the reader to the target destination. Logically, RFID system needs to handle massive data that requires large bandwidth to make sure the data flow in multiple layer data filtering run effectively for as much time as possible. The larger the bandwidth, the more data can be transferred from the client to the server. This is to support on-demand business requirement that needs to process data at the right time and in right quantity. However, to maintain such network with large bandwidth requires a lot of energy and costly. If the system unable to support real time massive data filtering process for as much as possible, this can cause bottlenecks. The most important things in the RFID filtering process, the result must be free from false positive errors and false negative errors. In supply chain business, small error rates are costly and such items can turn into trashy. The accuracy of the results using Bloom filter depends on the size of the filter, the number of hash functions used in the filter and the number of elements in the set. The more elements are added to a Bloom filter, the higher the probability result of false positives [16]. This makes Bloom filter useful for many tasks, especially to filter the flood of RFID data.

CONCLUSIONS

The Bloom filter provides interesting features because it offers space and process efficient method in data filtering. However, most of the previous research focused on a single type filtering method. For example, most of the filtering approach only focusses to remove duplicates in the RFID data stream. As the filter removed duplicate reading, noise reading still exists in the data stream. Both types of readings need to be removes to ensure reliability of information produced from the data streams. This problem clearly shows that further improvement and innovations on effectively filtering noise and duplicates RFID data are necessary. In future research, we planned to overcome this problem, to filter noise and duplicate data by using Bloom filter technique to exploit the efficiency of Bloom filter in filtering massive data successfully.

ACKNOWLEDGEMENT

This work is sponsored by Ministry of Education through ERGS grant vote E054 and Gates IT Solution Sdn Bhd.

REFERENCES

[1] K. Venkata and P Nikitin. 2006. Method and System for Reading Objects Having Radio Frequency Identification (RFID) Tags inside Enclosures. U.S. Patent Application 11/404,510.

[2] D. Roozbeh, M. E. Orlowska, and X Li. 2007. RFID Data Management: Challenges and Opportunities. IEEE International conference on RFID. pp. 175-182.

[3] Prabhu B. S., Xiaoyong Su, Harish Ramamurthy, Chi-Cheng Chu, and Rajit Gadh. 2006. WinRFID: A Middleware for the Enablement of Radio Frequency Identification (RFID) Based Applications, Mobile, Wireless, and Sensor Networks. Wiley-Interscience. pp. 313-336.

[4] W. Lisa, R. J. Barker, M. A. Kim, and K. A. Ross. 2013. Navigating Big Data with High-Throughput, Energy-Efficient Data Partitioning. ACM SIGARCH Computer Architecture News, 41(3): 249-260.

[5] W. Yanbo, Q. Z. Sheng and S. Zeadally. 2013. RFID: Opportunities and Challenges, In: Next-Generation Wireless Technologies. Springer London. pp. 105-129.

[6] Aggarwal Charu C. and J. Han. 2013. A Survey of RFID Data Processing, In: Managing and Mining Sensor Data. Springer US. pp. 349-382.

[7] T. Sapna, A. Q. Ansari, and M. A. Khan. 2010. Dynamic Threshold Based Sliding-Window Filtering Technique for RFID Data. In: Advance Computing Conference (IACC), IEEE 2nd International, IEEE. pp. 115-120.

[8] Deng F. and Rafiei, D. 2006. Approximately Detecting Duplicates for Streaming Data Using Stable Bloom Filters. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data ACM. pp. 25-36.

[9] W. Xiaowei, Q. Zhang, and Y. Jia. 2008. Efficiently Filtering Duplicates over Distributed Data Streams.

In: Computer Science and Software Engineering, 2008 International Conference. pp. 631-634.

[10] M. Hairulnizam and J. Abawajy. 2010. An Approach to Filtering Duplicate RFID Data Streams, In U-and E-Service, Science and Technology. Springer Berlin Heidelberg. pp. 125-133.

[11] M. Hairulnizam and J. Abawajy. 2010. An Approach for Removing Redundant Data from RFID Data Streams. Sensors 11(10): 9863-9877.

[12] L. Chun-Hee and C.-W. Chung. 2011. An Approximate Duplicate Elimination in RFID Data Streams, Data and Knowledge Engineering 10(12): 1070-1087.

[13] J. Wen, Y. Wang and G. Zhang. 2012. A Two-Layer Duplicate Filtering Approach for RFID Data Streams. In Internet of Things, Springer Berlin Heidelberg. pp. 226-233.

[14] Yongsheng, H. A. O. and G. E. Zhijun. 2013. Redundancy Removal Approach for Integrated RFID Readers with Counting Bloom Filter. Journal of Computational Information Systems 9(5): 1917-1924.

[15] M. Nahla and J. Akaichi. 2013. Supply chain Duplicate Transportation RFID Data Stream Filtering. International Journal of Application or Innovation in Engineering and Management 2(5): 484-493.

[16] T. Sasu, C. E. Rothenberg and E. Lagerspetz. 2012. Theory and Practice of Bloom Filters for Distributed Systems, Communications Surveys and Tutorials. IEEE. 14(1): 131-155.