An Improved SMURF Scheme for Cleaning RFID Data

He Xu[1,2*], Jie Ding[1,2], Peng Li[1,2], Daniele Sgandurra[3], Ruchuan Wang[1,2]

1. School of Computer Science,
Nanjing University of Posts and Telecommunications,
Nanjing 210023, China;
2. Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks,
Nanjing 210003, China
3. School of Mathematics and Information Security,
Royal Holloway, University of London,
Surrey TW20 0EX, UK
Email: xuhe@njupt.edu.cn; djlibainan@126.com; lipeng@njupt.edu.cn;
Daniele.Sgandurra@rhul.ac.uk; wangrc@njupt.edu.cn

Abstract: With the increasing usage of Internet of Things' devices, our daily life is facing Big Data. RFID technology enables the reading over a long distance, provides high storage capacity and is widely used in the Internet of Things environmental supply chain management for object tracking and tracing. With the expansion of the RFID technology application areas, the demand for reliability of business data is increasingly important. In order to fulfil the needs of upper applications, data cleaning is essential and directly affects the correctness and completeness of the business data, so it needs to filter and handle RFID data. The traditional statistical smoothing for unreliable RFID data (SMURF) algorithm dynamically adjusts the size of window according to tags' average reading rate of sliding window during the process of data cleaning. To some extent, SMURF overcomes the disadvantages of fixed sliding window size; however, SMURF algorithm is only aimed at constant speed data flow in ideal situations. In this paper, we overcome the shortage of SMURF algorithm, and an improved SMURF scheme in two aspects is proposed. The first one is based on dynamic tags, and the second one is the RFID data cleaning framework, which considers the influence of data redundancy. The experiments verify that the improved scheme is reasonable in dynamic settings of sliding window, and the accuracy of cleaning effect is improved as well.

Keyword: RFID; data cleaning; Internet of Things; sliding window

Biographical notes: He Xu received his PhD degree in Information Network from the Nanjing University of Posts and Telecommunications (NUPT) in 2012. Currently, he is an Associate Professor at School of Computer Science. His research interests include information security and Internet of Things (IoT).

Jie Ding is a graduate in Software Engineering at the Nanjing University of Posts and Telecommunications. Her research interests include RFID based IoT.

Peng Li received his PhD degree from Nanjing University of Posts and Telecommunications (NUPT) in 2013. Currently, he is an Associate Professor at NUPT. His main research interests include computer communication networks and information security.

Daniele Sgandurra is currently Lecturer at the Information Security Group of Royal Holloway, University of London. His research focuses on cybersecurity of systems, in particular operating systems/virtualisation, mobile and IoT.

Ruchuan Wang is a Professor of School of Computer Science at NUPT. His research interests are in the area of information security, including computer network, RFID and wireless sensor networks.

Copyright © 200x Inderscience Enterprises Ltd.

1 Introduction

1.1 Background

Internet of Things (IoT) is most commonly described as an ecosystem of technologies monitoring the status of physical objects and capturing meaningful data [14-16]. Data creation occurs at a fast rate, referred to as Big Data [23], and has emerged as a widely recognized trend [24]. Wireless Sensor Networks (WSN) and Radio Frequency Identification (RFID) are the two of the most prominent IoT technologies [17]. RFID is a type of automatic recognition of communication technology [1]. With the detection, identification and monitoring of the electromagnetic signals, the related objects' data can be obtained, and their transmission is performed without physical contact between the recognition system and the target, which can be recognized. RFID technology has the features of allowing the reading over a long distance, providing high storage capacity, and these are some of the reasons explaining why it is so widely used in the Internet of Things environment for object tracking and tracing, and it also has high-profile application prospect [2-6]. For example, by integrating the RFID with wireless sensors, it is possible to enable the communication between RFID tags and other equipment. RFID technology not only can help a company to improve the efficiency of information management, but also can facilitate the interconnection among enterprises, such as sales and manufacturing enterprises, as to enable a more accurate control and feedback, and provide an optimized supply chain system.

With the expansion of the RFID technology application areas, the demand for reliability of business data is increasingly important in IoT environment [18-19]. With the increasing usage of RFID devices, our daily life is facing Big Data [20-22]. To fulfil the needs of upper-level applications, data cleaning is essential since it directly affects the correctness and completeness of the business data; therefore, we need to filter and handle RFID data properly [7].

RFID practical application in physical environments is complex; in fact, RFID system can only typically read around 60% to 70% of the tags in the process of receiving data flow. This means that the data that cannot be read can be as high as 40%. Furthermore, a large percentage of data reading is leaked, which further increases the difficulty of tracing objects. In order to improve the reliability, accuracy and integrity of RFID data, the RFID system needs a self-contained module able to clean huge volume of error data of tags. Normally, the error data types in RFID system are false negative, false positive and data redundancy [8].

The most commonly used method to solve the problem of tag misreading is based on statistical, smooth, adaptive sliding window adjustment cleaning methods – in particular the statistical smoothing for unreliable RFID data (SMURF) algorithm [11]. The traditional SMURF algorithm is used to dynamically adjust the size of smoothing-window according to tags' average reading rate of sliding windows during the process of data cleaning. This approach, to some extent, overcomes the disadvantages of fixed sliding window size. However, SMURF algorithm is aimed at constant speed data flow in ideal situations. That is to say, if the speed of data flow is non-uniform, the algorithm shows difficulty in accurately detecting tags, and cannot identify the dynamic process when tags enter and leave the reader's range. Furthermore, the algorithm also produces a lot of repetitive data when trying to fill the data that has misread, which cannot effectively guarantee the quality of data cleaning. Due to the shortcomings of SMURF algorithm, we propose an improved version.

1.2 Related Works

In the original data cleaning system, the mechanism using a slide time window technology to smooth filtering data-flow is put forward by Y.Bai [9] from UCLA University and is used for washing original tag data flow of RFID. In

Copyright © 200x Inderscience Enterprises Ltd.

most of the cleaning models of RFID data, the basic idea of data cleaning is equivalent to "smoothing filter based on time". The concept of "smooth window" refers to a window that has a certain length and slides over time. In the whole life cycle of the system implementation, a fixed value represents the size of the time window that will continue to move forward as time goes on. This method is simple and fast, but it does not adapt to the requirements of dynamic environments. The window size can be very hard to property set: if it is too small, it can cause misreads (false negative), and if it is too large it can lead to repeated reads (false positive), thus resulting in the problem of data delay. The ideal size of sliding window must be set to ensure tags can be read completely, and tags' data can be captured dynamically. Generally, although this method has shortcomings, it still has strong practicability because of its simple implementation.

Contraposing the characteristics of RFID data flow, Jeffrey et al. [10] propose a data cleaning model based on the structure of pipeline's data cleaning model (ESP) by introducing the definition of space and time granularity, which is also known as extensible data flow cleaning model. ESP uses different cleaning steps which form a pipe to process the raw data. ESP can be used by all kinds of practical applications through a simple configuration. This algorithm is suitable for data leaking and repeated read. ESP can clean a variety of RFID data effectively, such as from a single reader to multiple readers, from a single space to multiple space, and from a single type to multiple types. It has extensive data cleaning scope. Moreover, this model is very flexible, and can select processing phase according to different requirements. However, setting the ideal time and space granularity parameters in ESP can be very difficult. Only by choosing the optimal particle size it can ensure the quality of cleaning and minimize the average error rate of cleaning. Therefore, a solution to deal with this issue is to adaptively adjust the time and space granularity.

Jeffery et al. [11] have introduced the data cleaning strategy based on time correlation, which is based on probability model. Through dynamic changing the window's size, it is mainly used to solve the problem of leaking of read data. They give the definition of adaptive smooth filtering of RFID data cleaning method. This method puts RFID data flow as a random event in probability statistics, by using the probability theory to fill the misread data. This strategy has the advantages of adaptively updating window's size according to the size of tag reading rate, and reducing the false negative and false positive rate that are caused by unreasonable selection of window size. However, for dynamic tag data, for example when moving tags leave the reader's reading range quickly, it can make the reading rate suddenly decrease. Instead, the method of SMURF uses large windows smooth that can lead to more false positive. The shortcomings of sliding window are that it cannot avoid false negatives and false positives, because SMURF is also based on sliding window.

H.Gonzalez et al. [12] put forward the RFID data cleaning algorithm based on Dynamic Bayesian Network (DBN), through accessing data cleaning results' accuracy and balance among the cost which is needed to pay to realize the cleaning cost optimization. The fewest resources are used to clean the most original tag data under the premise of ensuring the accuracy of the cleaning's results. DBN uses an index called implicit model, the actual value of which is the noise value, to determine the real location, and predicts the tag data by observing historical data, and finally calculates a probability value as a standard to measure tags whether exist or not. Compared with sliding windows, ESP and SMURF, the advantage of DBN is that it does not need to record the latest tag data, because the relationship of observed value and predicted value are combined and the weight of new data is given higher. On the other hand, its shortcomings are that it cannot guarantee the dynamic update because predicted values and the observed values are gained from historical data, and the results of cleaning dynamic tags are not very ideal.

RFID-Particle Filter Cleaning (RPFC)[13] is a type of tracing cleaning algorithm based on particle filtering to

Copyright © 200x Inderscience Enterprises Ltd.

perform Bayesian filter using abstract particles to simulate movements of tags under real situations and get the mobile tags' location. The original data is converted to smooth uncertainty data flow in regard to the state of the tag location. Non-deterministic cleaning method, which puts forward in RPFC, starts from two parts: semantic cleaning and misread cleaning. Semantic cleaning is based on uncertainly RFID data semantic, and uses weighted particle swarms to track, simulate and show changes of moving tags' location information. The effective probability value is used to express uncertainty of RFID data semantics. Misread consists of reader's false negative and false positive. RPFC uses RFID tag data' characteristic of uncertainly semantics to fill the false negative data from adjacent readers, which exists as an adjacent relation of single tag reader's mobile path. Though certain readers read data for a long time but it has not a high success rate, and the misread of tag data can be filled by other readers, and it will not lead to false positive. This algorithm is very effective to solve the problem of single reader's false negative while it limits error to a small range.

1.3 The Contributions

Filtering RFID data will make the RFID systems not to cause much redundant data for IoT systems, which may also help to improve the RFID systems to fast process data in big data environment. In this paper, we overcome the shortage of SMURF algorithm, and an improved SMURF scheme is proposed in the following two aspects. One is based on dynamic tags, and the other is the cleaning framework which considers the influence of data redundancy. The experiments verify the improved scheme is reasonable in dynamic settings of sliding window, and the accuracy of cleaning effect is improved as well. In addition, the redundant data is processed by the proposed cleaning framework to greatly reduce the amount of redundant RFID data.

1.4 Structure of the Paper

The rest of the paper is organized as follows. In Section 2 we describe the SMURF algorithm, while Section 3 presents improvements for SMURF algorithm. Experiments are given in Section 4. Finally, Section 5 concludes the paper.

2. SMURF algorithm

Jeffery et al. from the University of California at Berkeley proposed an adaptive cleaning algorithm based on time window size, which is based on probability model. Through changing the window size dynamically, it mainly solves the problem of false negatives. The definition for adaptive smooth filtering of RFID data cleaning method is "SMURF" (statistical smoothing for unreliable RFID data).

SMURF algorithm adaptively changes the window size, and the RFID data stream is regarded as a reader to identify and read data from RFID tags, after then the data will be selected in random sampling. The adjust and filter strategy of SMURF algorithm uses the theory of probability through sampling, which is based on the nonequivalent probability when sampling and modeling that is originated from the real world's tags. This approach makes the SMURF algorithm to adopt statistical methods to measure the unreliability of readers and dynamic tags, which always changes the size of sliding window to generate accurate data for the specific application. SMURF algorithm is based on an indicator to determine the size of the sliding window, which is the *tag reading rate*. The algorithm adopts a big window to improve the passive reading when tag reading rate is low. On the contrary, it adopts small window to reduce active reading in RFID data obtaining process.

The main innovation of this method is to put RFID data flow as a random experiment in probability statistics, using the binomial distribution probability model to fill the data which have been misread. That is, it regards the process of reading a tag at one time as a random experiment, when tags have been read, which means the random event happened. The frequency of the tag that can be read is the event occurrence probability which considers each reading cycle epoch's detection value in smooth window as repetitive random experiment's results and constructs the binomial distribution model.

The specific comparisons of RFID data and random event model are shown in Table 1.

Table 1. Comparisons of RFID data and random event model

RFID data flow	Random event model
Detection period	One random trial
Tags being read	Random event occurrence
Frequency of tags reading	Probability of event occurrence
Value in the smoothing window	Results of repeated random events

The steps of SMURF algorithm are listed as follows: set the size of the sliding window as w_i reading epoch, and set the whole data flow epoch as t, then the size of sliding window is $W_i = (t - w_i, t)$. Assume the probability of tag i appears in reader's range during the entire sliding window time and the probability of each reading epoch in time slice tags which have been read in the window as p_i, every reading epoch are regarded as once Bernoulli experiment which has the probability p_i. The sliding window time appears in the reader range and each reading epoch within the window time slice for tag is set in probability, which regards each reading cycle as a probability for Bernoulli experiment, and the amount of tags which is being read in the sliding window distribution obey the random variable of binomial distribution.

Set the times of tag reading as $S_i \subseteq W_i$, then the average reading rate of the tag is:

$$p_i^{avg} = \sum_{t \subset S_i} \frac{p_i}{|S_i|} \tag{1}$$

In this way, $|S_i|$ is regarded as free variables of Bernoulli model $B(w_i, |S_i|)$, and then we get:

$$E[|S_i|] = w_i p_i^{avg} \tag{2}$$

The variance is:

$$Var[|S_i|] = w_i p_i^{avg} (1 - p_i^{avg}) \tag{3}$$

Using the probability model based on Bernoulli trials to look tag i, if the probability of tag that has been under average read in w_i reading cycle time slice is p_i^{avg}, so the probability of tag which has not been read in the whole process of sampling is $(1 - p_i^{avg})^{w_i}$.

When the confidence is δ, we get:

$$(1 - p_i^{avg})^{w_i} \le \delta \tag{4}$$

Taking the logarithm on both sides at the same time in the formula (4), we obtain:

$$w_i \ln(1 - p_i^{avg}) \le \ln \delta \tag{5}$$

And through the equation $-x \ge \ln(1-x), x \in (0,1)$, we get:

$$-w_i(-p_i^{avg}) \ge -w_i \ln(1 - p_i^{avg}) \ge -\ln \delta \tag{6}$$

In order to ensure the data integrity, the size of the sliding window w_i needs to meet:

Copyright © 200x Inderscience Enterprises Ltd.

$$w_i \geq \frac{\ln(1/\delta)}{p_i^{avg}} \qquad (7)$$

In the Formula (7), the confidence δ is the data constant that is given according to experience, that is, no matter how the actual situation changes, the confidence is always the same.

If the current sliding window cannot meet the demand of integrity, then according to the formula (8) we need to adjust the sliding window size as follows:

$$w_i = \max\{\min\{w_i + 2, w_i^*\},1\} \qquad (8)$$

Where w_i^* is set to satisfy the integrity of the minimum of sliding window size, that is, it should meet the minimum w_i in the formula (7).

In order to keep the dynamic tag, the size of sliding window needs to meet:

$$\left\| S_i \right| - w_i p_i^{avg} \right| > 2\sqrt{w_i p_i^{avg}(1 - p_i^{avg})} \qquad (9)$$

If the state is changed, then we need to adjust the results of sliding window size according to the following formula (10):

$$w_i = \max\{\min\{w_i / 2, w_i^*\},1\} \qquad (10)$$

If the state does not change and it meets the requirements of integrity, the sliding window remains the same.

Traditional SMURF algorithm puts the size of the sliding window w_i initialization as 1 at first in the process of implementation, and then it dynamically adjusts the window size according to the current specific reading recognition condition. Dynamic discrimination is to test tags, whether the state has changed or not, when entering and leaving the reading range of reader. When the tag's state is changed, the algorithm will adaptively adjust the window size according to the formula (7); when the window has no reading event, SMURF algorithm sets the slide window size as the initial window size to 1, and then SMURF algorithm continues to slide window in epoch. If the current window size meets the requirements of integrity, namely the sliding window size w_i meets the formula (7), then the SMURF algorithm will continue to test tags according to the formula (9) whether it has changed dynamically or not; if tag's state has changed, SMURF will reduce the size of current window to $\frac{w_i}{2}$ according to the formula (10), which is the half size of the original window; if there is no tag's state change after detection, the SMURF algorithm outputs the sliding point which is the current centered size of sliding window, then slides the length of an epoch and proceeds next process. If the current window size cannot meet the demand of integrity, the SMURF algorithm will add 2 as a step to enlarge current window according to formula (8), then it can proceed the next slide after outputting the middle point of the current window.

3. The improvements of SMURF algorithm

In order to overcome the shortage of SMURF algorithm, we need to improve it in the following two directions. One is our proposed cleaning method named "Dynamic SMURF" which is based on dynamic tags; the second one is the cleaning framework that considers the influence of data redundancy.

3.1 Dynamic tags scheme

For dynamic tags, the smaller the value δ is, the lower the accuracy is. Because the larger the window is set and the reading rate of dynamic tag is very low, large windows are needed to ensure the data integrity. While the value δ

becomes small, the sliding window needs to be set up bigger, which finally causes more misread.

Through the above analysis, the movement velocity of tag is introduced into the adjustment of confidence level to improve the accuracy of the confidence value δ. Considering that the adjustment of the confidence value δ uses the movement velocity of tag, our proposed "Dynamic SMURF" is based on dynamic tags.

Firstly, we set the speed of a single tag as V, the reader's time slot as T, which is the time interval for the process of communication between tag and reader, and the reader's identification range's radius as R. When the tag passes through the identification range of a reader, the reader can read the tag for several times, so the biggest times of tag that has been read are:

$$f = \frac{2R}{(V \times T)} \qquad (11)$$

In order to dynamically adjust the confidence δ, according to the meaning of confidence's parameter, the calculation formula of confidence δ is defined as:

$$\delta = \frac{p_i^{avg}}{f} \qquad (12)$$

Parameter δ is proportional to the reading rate, and is inversely proportional to the time of reading which is already known. Then the probability of tag that has not been read is:

$$(1 - p_i^{avg})^{w_i} < \delta = (p_i^{avg} / f) \qquad (13)$$

By taking the logarithm on both sides at the same time in formula (13), we get:

$$w_i \times \ln(1 - p_i^{avg}) < \ln(p_i^{avg} / f) \qquad (14)$$

Because $(1 - p_i^{avg}) < 1$, $\ln(1 - p_i^{avg}) < 0$, then we can get:

$$\ln(1 - p_i^{avg}) < -p_i^{avg} \qquad (15)$$

Thus:

$$w_i > \frac{\ln\left(f / p_i^{avg}\right)}{p_i^{avg}} \qquad (16)$$

By formula (16), we can solve the problems of oversize dynamic tags' sliding window.

3.2 Data redundancy cleaning

SMURF algorithm does not take redundant data into account, which exists in the RFID data flow. Because it does not process the redundant data instead of directly stored in the database, it will cause serious waste of storage and computing resources. In this section, we propose a cleaning framework for processing the redundant data on the basis of improved SMURF algorithm.

Firstly, the original data flow can be smooth-filtered and cleaned using the "dynamic SMURF". Then the EPC filter module is used to filter out the coding errors data and sets the tag ID that is produced by hardware. This process does not need to conform to the encoding rules. These new data sets are examined and judged whether they exist or not in the buffer. If it does not exist, it will be stored in the buffer, otherwise it will be discarded. A fixed time period is set for buffer which is a sliding window size. The data which is in the buffer will be output if it is over time.

The algorithm framework is shown in Figure 1:

Copyright © 200x Inderscience Enterprises Ltd.

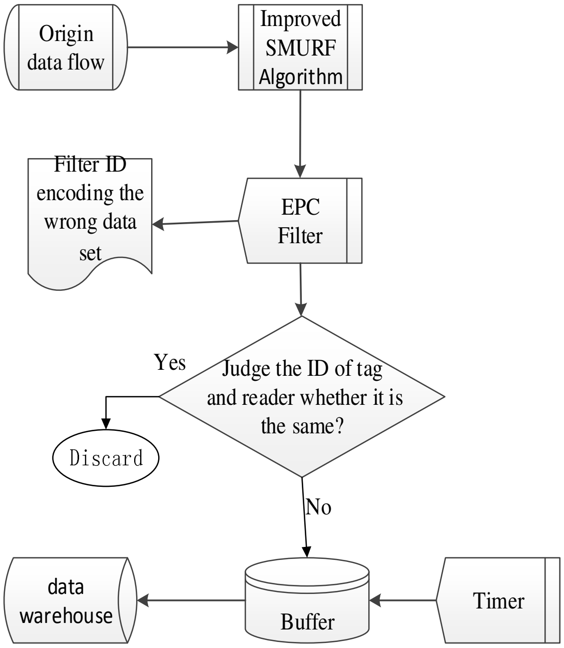
Origin data flow
Improved SMURF Algorithm
Filter ID encoding the wrong data set
EPC Filter
Yes
Judge the ID of tag and reader whether it is the same?
Discard
No
data warehouse
Buffer
Timer

Fig.1 Framework of RFID redundant data cleaning

4. The simulation experiment

This section describes the experiments we performed to verify the performance of our proposed algorithm, described method in Section 3, and analyzes the experimental data. The experiments are based on MATLAB R2016a, and are aimed at verifying the validity of proposed algorithm from the proper size settings of the sliding window and the accuracy of cleaning effect respectively. Finally, redundant data are used to analyze the effectiveness of the cleaning framework.

4.1 Experimental environment

The testbed used in the experiments is composed of the following equipment.

(1) Hardware environment:

CPU: Intel core i7-3537; Memory 4G; Disk capacity: 150G.

(2) Software environment:

1) Operation system: Microsoft Windows 7.

2) Simulation platform: MATLAB R2016a.

4.2 Experimental data sets

Experiments are set according to the real RFID application scenarios using a random-distribution RFID system. As shown in Figure 2, there is a number of readers and tags in the system, where we have method used the symbol * to signify the tags and the symbol ○ to represent the reader's reading area. When the tag in the range of readers, that is, * is located inside the ○, this means that the reader can read the tags. The simulation environment is created by randomly generating RFID readers and tags, which can represent major RFID application scenarios.

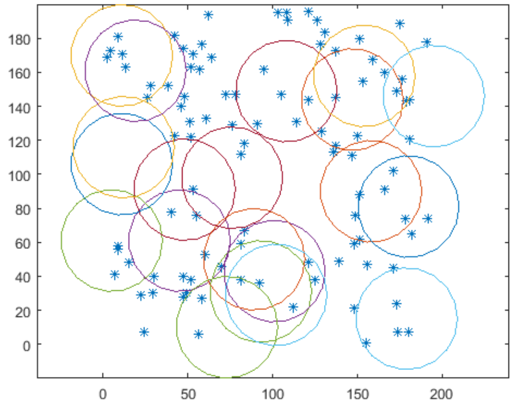

Fig.2 RFID network simulation topology

4.3 Experiment analysis

In this section, we compare the performance of our improved SMURF algorithm. The simulation data generation model is used to generate RFID data as shown in Figure 3, where tags information is randomly generated by the MATLAB software.

First, the simulated reader's reading process is described by data generator and tag movement behavior. Assume that the radio frequency range between all readers and tags is constant, specific detection effect under actual application scenario is simulated. As shown in Figure 3, the horizontal axis shows the distance between a tag and a reader, and the longitudinal axis shows the probability of tags which are identified by reader, which is also named the reading rate. The detection area is defined as the distance from the reader to the edge of reader's coverage, and each reader which belongs to the radio frequency range is divided into two different areas:

(1) The primary reading area: in this detection area, the tag has a very high detection rate (equal or greater than 95%).

(2) Secondary detection area: this area extends from the edge of the main detection area to the largest radio frequency range of reader. The detected rate of tag data is linearly attenuated in this area. In actual scenarios, it may be fluctuated, but this can be negligible in the experiment. Finally, it is decay to 0 in the edge of radio frequency identification range.

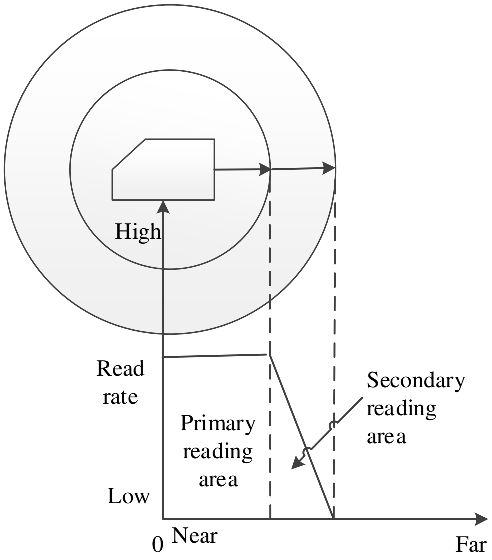

Fig. 3 Simulation data generation model

4.4 The comparison of algorithm validity

Copyright © 200x Inderscience Enterprises Ltd.

Experiment 1：Comparison of sliding window size.

Mobile tag will reduce the reading rate and the window of SMURF algorithm will be oversize. However, our improved SMURF algorithm (Dyanmic SMURF) sets up reasonable window size under the condition of dynamic tags.

In this experiment, the proportion of the reader's main detection area is changed within the whole reader recognition range. When the value is tested from 0 to 1, using different variation of main detection area proportion, we observe and compare the size of sliding window. The main area detection ratio refers to the percentage of the main area occupied all testing area. In the experiments, the δ of SMURF algorithm is 0.1 and 0.01 respectively. The results are shown in Figure 4.

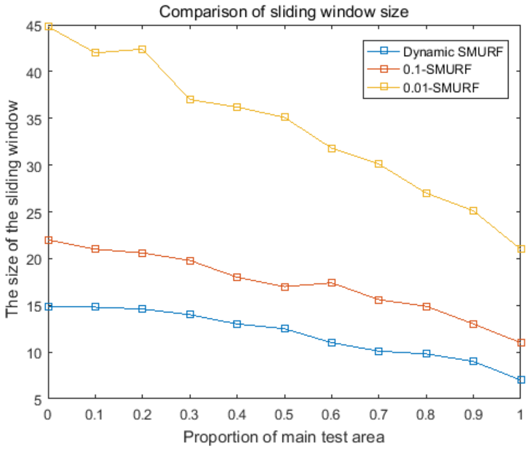

Fig. 4 Comparison of sliding window size

From Figure 4, we can clearly see that the confidence value δ reduces one magnitude, and the size of the window suddenly expands nearly twice. In the case of large windows, the confidence value δ directly affects the accuracy of the results, and improved SMURF algorithm sets up a suitable sliding window size which can ensure the integrity of dynamic tag data.

Experiment 2：Comparison of accuracy rate.

One performance metrics for evaluating RFID data cleaning algorithm is the accuracy rate of cleaning result data. The accuracy of the algorithm is defined as follows:

suppose the number of real tag is N , and the tags' number of each cleaning algorithm that has been cleaned is M , then the accuracy of the algorithm is defined as:

$$P = \frac{M}{N} \qquad (17)$$

In the experiments, the set of tags' numbers are set as the following:10, 20, 30, 50, 100, 200, 500.

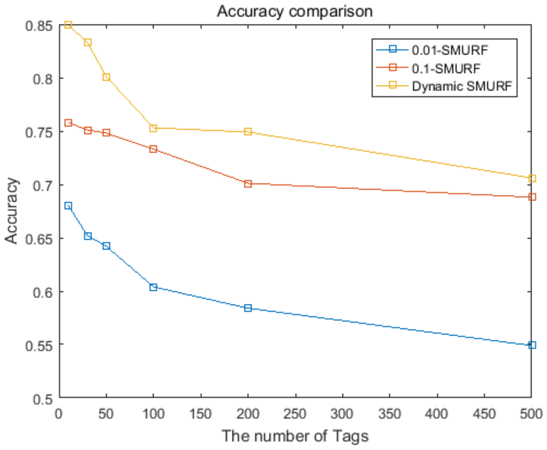

Fig.5 Accuracy results

From Figure 5, we can see that the value δ of SMURF algorithm is set as 0.1, and that the accuracy is much higher than that of 0.01. The figure also shows that selecting a reasonable value of confidence δ is difficult, which is also important to the cleaning results. The Dynamic SMURF algorithm we presented has the highest accuracy rate in the experiments. Because it can dynamically adjust and set the size of the sliding window, which reduces the false negative and false positive, it effectively improves the accuracy of the cleaning results.

Experiment 3：Validity of cleaning framework.

With the increasing number of tags in RFID reading area, the RFID data flow show redundant phenomenon, which can lead to waste storage space and data transmission delay. The tags data's redundancy amount is compared with the original RFID data flow in the tags' numbers of 100, 200, 300, 400, 500 and 1000 respectively. The results of comparison are shown in Figure 6.

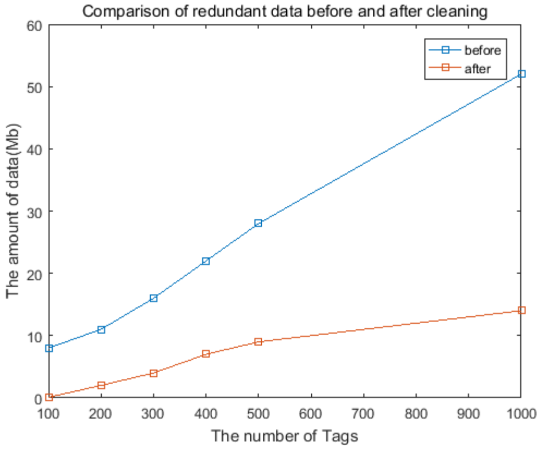

Fig.6 Comparison of redundancy data

From figure 6 we can see some redundant data which existed in the buffer that has been filtered after being disposed by our proposed RFID redundant data cleaning framework. The quantity of data is obviously decreased after cleaning, and RFID redundant data is greatly reduced.

5 Conclusions

For large-scale application of RFID technology, one of the critical issues is the low efficiency of communication overhead due to redundant data which increases energy consumption and causes time delay in RFID systems. To address this issue, an improved RFID data cleaning algorithm has been proposed in this paper; its feasibility and effectiveness have been verified via simulation experiments and compared with SMURF algorithm. We have described how to overcome the shortage of SMURF algorithm, and an improved SMURF scheme in two aspects is described. The first improvement is based on dynamic tags, and the second one is based on a cleaning framework that considers the influence of reading data's redundancy. The experiments have shown that the improved scheme is reasonable in dynamic settings of sliding window, and the accuracy of cleaning effect is improved as well. As a future extension to this work, we will consider multi tags data cleaning algorithms.

Acknowledgment

The authors thank all the reviewers and editors for their valuable comments and works. This paper is supported by the National Natural Science Foundation of P. R. China (No.61373017, No.61572260, No.61572261, No.61672296, No.61602261), the Natural Science Foundation of Jiangsu Province (No.BK20140886, No.BK20140888), Scientific & Technological Support Project of Jiangsu Province (No. BE2015702, BE2016185, No. BE2016777), Natural Science Key Fund for Colleges and Universities in Jiangsu Province (No.12KJA520002), China Postdoctoral Science Foundation (No.2014M551636, No.2014M561696), Jiangsu Planned Projects for Postdoctoral Research Funds (No.1302090B, No.1401005B), Jiangsu Postgraduate Scientific Research and Innovation Projects (SJLX16_0326), Project of Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks (WSNLBZY201509), and NUPTSF (Grant No. NY214060, No. NY214061).

References

[1]. Zoltan K. Using RFID And GIS Technologies For Advanced Luggage Tracking[J]. SEA-Practical Application of Science, 2015,2(8): 229-234.

[2]. Sakurai S. Prediction of sales volume based on the RFID data collected from apparel shops[J]. International Journal of Space-Based and Situated Computing, 2011, 1(2-3): 174-182.

[3]. Srinivasan S, Ranganathan H. RFID sensor network-based automation system for monitoring and tracking of sandalwood trees[J]. International Journal of Computational Science and Engineering, 2013, 8(2): 154-161.

[4]. Kumar R, Chatterjee A, Verma S, et al. RFID-based navigation of autonomous patient vehicle for hospital management[J]. International Journal of Medical Engineering and Informatics 2, 2014, 6(4): 281-288.

[5]. Chen T, Wang Y C. A study on the construction of a ubiquitous customer relationship management system with RFID–a case study[J]. International Journal of Information and Communication Technology, 2012, 4(1): 13-26.

[6]. Wang L. RFID-based information technology and management in agriculture and food supply chains[J]. International Journal of Information Technology and Management, 2012, 11(3): 225-239.

[7]. Managing and mining sensor data[M]. Springer Science & Business Media, 2013.

[8]. Xie L, Yin Y, Vasilakos A V, et al. Managing RFID data: challenges, opportunities and solutions[J]. IEEE Communications Surveys & Tutorials, 2014, 16(3): 1294-1311.

[9]. Bai Y, Wang F, Liu P. Efficiently Filtering RFID Data Streams[C]. CleanDB, 2006:1-8.

[10]. Jeffery R, Alonso G, Franklin M J, et al. A pipelined framework for online cleaning of sensor data streams[J]. Computer Science, 2005:1-12.

[11]. Jeffery R, Garofalakis M, Franklin M J. Adaptive cleaning for RFID data streams[C]. Proceedings of the 32nd international conference on Very large data bases(VLDB Endowment), 2006: 163-174.

[12]. Gonzalez H, Han J, Shen X. Cost-conscious cleaning of massive RFID data sets[C]. IEEE 23rd International Conference on Data Engineering. IEEE, 2007: 1268-1272.

[13]. Want T, Zhang J. Research on RFID data cleaning method based on particle filter[J]. Electronic Technology & Software Engineering, 2014(1):214-215.

[14]. Pereira R, Pereira E G. Future internet: trends and challenges[J]. International Journal of Space-Based and Situated Computing, 2015, 5(3): 159-167.

[15]. Sula A, Spaho E, Matsuo K, et al. A new system for supporting children with autism spectrum disorder based on IoT and P2P technology[J]. International Journal of Space-Based and Situated Computing, 2014, 4(1): 55-64.

Copyright © 200x Inderscience Enterprises Ltd.

[16]. Wang X. The architecture design of the wearable health monitoring system based on internet of things technology[J]. International Journal of Grid and Utility Computing, 2015, 6(3-4): 207-212.

[17]. Yao X, Wang L. Design and implementation of IOT gateway based on embedded μTenux operating system[J]. International Journal of Grid and Utility Computing, 2017, 8(1): 22-28.

[18]. Atanasov I, Nikolov A, Pencheva E, et al. An Approach to Data Annotation for Internet of Things[J]. International Journal of Information Technology and Web Engineering (IJITWE), 2015, 10(4): 1-19.

[19]. Baarah A, Mouttham A, Peyton L. Architecture of an event processing application for monitoring cardiac patient wait times[J]. International Journal of Information Technology and Web Engineering (IJITWE), 2012, 7(1): 1-16.

[20]. Moore P, Thomas A, Tadros G, et al. Detection of the onset of agitation in patients with dementia: real-time monitoring and the application of big-data solutions[J]. International Journal of Space-Based and Situated Computing, 2013, 3(3): 136-154.

[21]. Hoarvar A R, Sami A. Extracting Usage Patterns from Power Usage Data of Homes' Appliances in Smart Home using Big Data Platform[J]. International Journal of Information Technology and Web Engineering (IJITWE), 2016, 11(2): 39-50.

[22]. Mary S A, Malarvizhi M. An Efficient and Accurate Discovery of Frequent Patterns Using Improved WARM to Handle Large Web Log Data[J]. International Journal of Information Technology and Web Engineering (IJITWE), 2014, 9(2): 47-64.

[23]. Hashem I A T, Yaqoob I, Anuar N B, et al. The rise of “big data” on cloud computing: Review and open research issues[J]. Information Systems, 2015, 47: 98-115.

[24]. Chen M, Mao S, Liu Y. Big data: A survey[J]. Mobile Networks and Applications, 2014, 19(2): 171-209.