Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Improved Adaptive Cleaning Method for RFID Data Stream

Duo Zhang
Department of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, China
zd0451@gmail.com

*Chunkai Zhang
Department of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, China
ckzhang@hotmail.com

Keguang Ma
Department of Computer Science and Technology
Harbin Institute of Technology Shenzhen Graduate School
Shenzhen, China
zd0451@yahoo.com.cn

***Abstract*—Significant missed readings and tag transitions pose a challenge for real-time RFID middleware in data reliability. Fixed cleaning method proposed can only address the data completeness while neglect to capture the motion of tags. Shawn R. Jeffery introduced an adaptive cleaning method which can compensate missed readings and the dynamic tags, but we find its accuracy falls sometimes when tag moves out the detected region. In this paper, we present an improved method, which can provide a better accuracy in both smoothing missed readings and detecting tag transitions, and experiment result had shown that our method can improve data reliability clearly by about 15% to gain a better quality of data.**

***Index Terms*—RFID, Filter, Middleware.**

I. Introduction

RFID (Radio Frequency IDentification) technology has been recognized as one of the most promising identity technologies in the professions such as manufacture,logistics and retail[1,8]. They are getting cheap and functional enough to replace the existing bar-code tag and to become a new generation tag — Electronic Product Code(EPC). The basic operation of an RFID system contains tags and readers, the reader generates the signal which send out from the antenna and the radio wave signal may be received by an RFID tag, which in turn generates some of the energy it received from the reader[9], so reader acts both sender and receiver. The characteristics of RFID can be viewed as RF non-contact and fast base on this principle of RFID. As a result, these features lead to data unreliability directly. "Garbage in = Garbage out", poor quality of data can cause the bad result in application layer which is we do not expected, that is to say, inaccurate source data leads to inaccurate decision directly or indirectly. Especially in RFID middleware system, data stream flows fast, so it needs an efficient way to ensure data reliability.

The operation mode of RFID system can generate a mass of RFID data which comes with multiple errors, including the dropped readings, wrong readings and data redundancy etc. The inherent unreliability of RFID data streams becomes the key factor of limiting the widespread applications of RFID technology. There are three undesired scenarios[2]:

1) False positive readings: Due to the difference of readers' performance, some tags stay out of the detected range may receive reader's signal occasionally, which may be captured by the reader in turn. This case is also known as noisy readings. Some unknown factors can cause this noisy readings too, for example, tag decoder accidentally makes wrong message.

2) False negative readings: The tag get to appear in the reader's detected range but the reader failed to capture it. False negative readings also mean missed readings.Signal collision may cause this missed reading when serval signal overlaps together; weak signal will not received correctly by readers; and environmental disturbance, like metal objects, can also contribute to false negative readings

3) Duplicate readings: Base on UHF Gen2 protocol, a reader can scan 1500 tags per second[10]. As a result, massive data will be produced by readers and the huge data stream will present redundant information which is duplicated [2].

The false positive readings and the false negative readings can give rise to data unreliability. Yijian Bai had introduced a method to reduce noise and duplication in an effective and simple way, and it indicated that false positive readings happen occasionally in the data stream. So false positive readings can contribute little to data unreliability. According to survey [3, 4], there are almost 30% readings lost that reduce data reliability a lot when acquiring tags from readers. So the fundamental concern may focus on how to smooth false negative readings.

RFID data stream is huge, discrete and coming up as time goes by. In order to smooth the lost data in time, old data discarded and processing only carries on with the recent elements[11]. Base on this technical characteristic, sliding window technique is taken into consideration to smooth RFID raw data. Traditional RFID middleware systems use a smoothing filter— a sliding window of fixed size to compensate the missed readings. However, the fixed window size was

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

initialized by system at the beginning, the large window can compensate missed readings well but can't detected tag's movement, for example, a tag move out of the reader's detect region(transition behavior); the small one can capture the tag transition easily, but is poor to build the data completeness.

The adaptive cleaning method — SMURF (Statistical sMoothing for Unreliable RFid data) [6] proposed to trade-off data completeness and tag transition. SMURF technique creatively employs a statistical sample method to calculate the most proper size of window when sliding. The average detection rate within a window is taken into account to build a Bernoulli distribution model to estimate the right window size, then according to the bias between observed and expected sample to determine the transition behavior. But its result is not much satisfied when transition had processed, this attribute to some flawed transition detection principle and output scheme.

Our paper focuses on the filter that smooth the false negative readings. We presented an improved algorithm (ISMURF)base on SMURF, the statistical method of calculate adaptive window size is kept, and we adjust the scheme of transition detection. Then we build a similar experimental scenario as built in SMURF to give a comparison between the methods.

The remainder of this paper is structured as follows. Section 2 describes the background of smooth filter. Section 3 describes proposed algorithm and provided theoretical base. Section 5 shows some numbers from experiments comparing the performance of other smooth filter. And Section 6 concludes.

II. Smoothing Filter

A. Fixed Cleaning Method

A tag won't flash in the reader's detected region; on the other hand, it will appear in the detected region for a while short or long, motion or motionless. According to this behavior, fixed cleaning method employs a slide window technique to smooth missed data. The slide window is a concept of time scale which is made up of one or several continuous read cycles [5], we call them units. The rule of the algorithm is: if a certain tag is observed within a window's unit, we output tag at the middle of the window, then with the window sliding forward and data will be continuous as time goes on. Thus, filling the tag to the units can make complement to the data stream. The simplicity of this algorithm gives beneficial promotion to the quality of RFID middleware system, but its drawback is obvious too. The window size is unchangeable and it needs to be initialized before the program. If window size chooses too small, the missed data will not be compensated enough when the signal is poor. On the contrary, the addition will be redundant and inaccurate when the reader detection effect(we call it read-rate in the the following text) is good if the large window is selected. That is to say the tag has left the reader's detected region in the real world and the large window can't discover it but the small one can. Fixed cleaning method can reduce false negative readings for the motionless tag and the circumstances that the read-rate is stable, but is not adequate for the changeable scenarios.

B. Adaptive Cleaning Method

In general, RSSI (Received Signal Strength Indication) level determines reader's read-rate for tags[7]; it attenuates from reader's antenna to margin of detected region by some way and it is difficult to maintain a stable read-rate for a reader. So fixed window can't satisfied data accuracy. The adaptive cleaning method (SMURF) solved this drawback of fixed slide window. In order to estimate the fittest size of window at current time, statistical methods are adopted. SMURF views a RFID tag which observed in a window as a non-probability sampling. Suppose a certain tag t, in this sampling model, the population is X which denotes the set of tag t observed in the window W in the idealized situation, and use a unequal-probability to sample a subset of X, let S denote, which is actually observed in W. The objective of building sampling model is to estimate the unknown population X to some extent. It refers a concept of interrogation cycle in [6]; reader scans all tags for a time by RF signals, this operation is called an interrogation cycle. Fixed amount of interrogation cycles make into a read cycle. Base on this interrogation, read-rate $\mathrm{P_t^i}$ - the probability of reading a certain tag t can be learnt in certain unit i of window, where

$$P_t^i = \frac{number\ of\ observed\ tag\ in\ the\ unit\ i}{number\ of\ interrogation\ cycles\ in\ a\ unit}$$

We can notice that each $\mathrm{P_t^i}$ of tag t within a window calculated is similar. It can be understood that the read-rate is stable within a window to some degree. Further more,We can use a variable P_t^{avg} to denote each P_t^i within a window, where

$$P_t^{avg} = \frac{\sum P_t^i}{number\ of\ observed\ tag\ in\ the\ unit\ i}$$

So SMURF lets the number of observed tagt $|S|$ follow Bernoulli distribution $|S| \sim B\left(|W|, P_t^{avg}\right)$, where P_t^{avg} is the mean of each P_t^i, and the expectation and variance is $E\left[|S|\right] = |W|\, P_t^{avg}$ and $Var\left[|S|\right] = |W| P_t^{avg}\left(1 - P_t^{avg}\right)$ respectively. Tag t is not observed at all in window W is not expected, so let$(1 - P_t^{avg})^{|W|} \leq \delta$ where δ is a small constant of completeness confidence. Finally the proper window size gained:

$$W = \lceil (1/P_t^{avg}) \ln (1/\delta) \rceil$$

SMURF uses bias between sample observation and statistical expectation to determine transition behavior:

$$||S| - |W| P_t^{avg}| \geq 2\sqrt{|W| P_t^{avg} (1 - P_t^{avg})}$$

The algorithm of adaptive cleaning method is presented as below, and we can see SMURF is superior to the fixed cleaning method for its adaptive window, but data accuracy is not satisfied after the transition detection.

III. Improved Algorithm

We are going to improve adaptive cleaning method in three aspects: condition of transition detection, scheme to adjust window when transition, and output scheme.

Algorithm 1: Adaptive cleaning method (SMURF)

Initialization: *set completeness confidence* δ *;*
set $|W| = 1$ *of tag t*;
loop (when coming the next read cycle)
  calculate the new window size $|W_n|$;
  if $|W_n| > |W|$ **then**
    $|W| \leftarrow \max\{\min\{|W| + 2, |W_n|\}, 1\}$
  else if $||S| - |W|P_t^{avg}| \geq 2\sqrt{|Var[|S|]}$ **then**
    $|W| \leftarrow \max\{|W|/2, 1\}$;
  end if
  slide window W;
end loop

A. Condition of transition detection

Only if the new window is calculated larger than current window will the transition detection be enabled. The growth of window indicates read-rate declines. On the contrary, a decrease indicates the read-rate goes up. However, transition may happen occasionally when read-rate grows. So no matter how the new calculated window alters, larger or smaller, we detect the transition, this can ensure some transitions won't miss.

We find out the transition was wrongly detected frequently in SMURF and cause false negative readings. It is indicated that the read-rate is low and the number of observed tag doesn't reach expect number in the window when$|S| < |W|P_t^{avg}$, then the window grows to compensate data; but when $|S| > |W|P_t^{avg}$ the number of observed data is more than the number expected, it indicates that the read-rate is getting better, and it seems that the transition is not likely to happen. So we change the conditional judgment of transition to:

$$|W|P_t^{avg} - |S| \geq 2\sqrt{|W|P_t^{avg}(1 - P_t^{avg})}$$

B. Scheme to adjust window when transition

To compensate data better, SMURF grows the window size gradually when read-rate performs poor; to discover the tag's dynamic state easily, SMURF cuts the current window in half when satisfied the condition of transition. This AIMD (Additive Increase Multiplicative Decrease) scheme represents a linear growth of the congestion window, combined to an exponential reduction when congestion takes place in TCP Congestion Avoidance[12]. In our opinion, multiplicative decrease is unreasonable to control window size when transition detected. The probability of transition can't be high when the read-rate is good, so half decrease may be too much if detection result is not true; and transition is more likely to happen when the read-rate changes to be very low, so it needs to reduce more than half a window to prevent it, and how much to reduce is depend on the read-rate a lot. Based on the read-rate, we adjust the window size when by $\lambda|W|P_t^{avg}$ after the transition detected, where λ is a constant. We find out that the effect is good when λ equals to 1.25 in our experimental environment.

C. Output scheme

Tags outputted from window are continuous in the fixed cleaning method, but discontinuous in varying window. We fill the "gap", which not mentioned in SMURF, between two output points of window as shown in Figure 1.

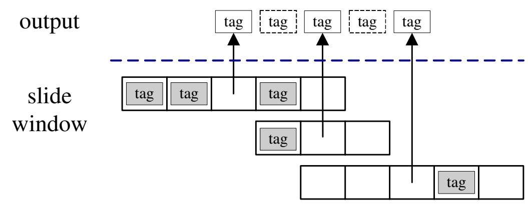

Fig. 1. Output the tag and fill the "gap".

By analyzing above-mentioned factors comprehensively, our improved algorithm (ISMURF) is presented as below:

Algorithm 2: Improved method based on SMURF (ISMURF)

1 **Initialization:** *set completeness confidence* δ *;*
2 *set the proper constant* λ *;*
3 *set* $|W| = 1$ *of tag t*;
4 **loop** (when coming the next read cycle)
5   calculate the new window size $|W_n|$;
6   **if** $|W|P_t^{avg} - |S| \geq 2\sqrt{|Var[|S|]}$ **then**
7     $|W| \leftarrow \max\{\lambda |W| P_t^{avg}, 1\}$
8   **else if** $|W_n| > |W|$ **then**
9     $|W| \leftarrow \min\{|W| + 2, |W_n|\}$;
10   **end if**
11   slide window W;
12   fill the gap between the output points;
13 **end loop**

Algorithm parameters should be initialized at the very beginning, the proper constant λ varies in different experimental situations, we adjusted it after several experiments and choose the most suitable one. Our algorithm firstly judges the condition of transition at line 6, and then to determine if it is necessary to enlarge the window size. algorithm fills the gaps after sliding window every time.

IV. Experiment Result and Analysis

In order to give a comparison, we build two experimental models which are similar to the SMURF.

A. Experiment model

1) Reader detection model: The model is established based on reader's characteristics of signal attenuation. We specify that the detection range is a circular area starts from reader's antenna to the detected margin. The read-rate is stable with some figure in reader's major detection area which is a circular sub region of detection range starts from reader's antenna and ends before the detected margin. We also specify a minor detection area which starts from the margin of major detection area to the detected margin; the signal intensity follows a linear attenuation base on the distance to the antenna in this area. We initialize the radius of detection range with 15 feet and the major read-rate with 0.8 then varied the major detected range to observe the experimental results.

2) Data generation model: Data generation model We generate tags and their behaviors like velocity and direction of movement to built realistic scenario in our experiment; it can be fixed or changed randomly in each read cycle by the generator. Distance to the antenna, read-rate and velocity will totally determine whether the tag will be detected by reader. It will generate 25 tags in our experiment, and run the algorithms track each of them.

B. Experiment 1.

For experiment one, we generate 5000 read cycles and each tag has an initial status with a random velocity, motion or motionless, the movement will change randomly per 100 read cycles. The range of movement will be in 20 feet around the antenna and the velocity will be controlled in 3 feet per read cycle. We change the major detection area percent to observe the algorithms' performance.

The comparison of several algorithms performance showed as we see in Fig.2, raw denotes raw data gathered from reader; static denotes fixed cleaning method, the number followed specifies the window size. We also lists the detailed experimental result in the Table.1, p# denotes the major detection area percent. The average errors are calculate as below:

$$Average\ Errors = \frac{\sum number\ of\ errors\ in\ read\ cycle}{number\ of\ total\ read\ cycles}$$

SMURF performs even worse than static5 sometimes when major detection area percent is at 40% and 50%. We can see our improved algorithm performs best and has much lower

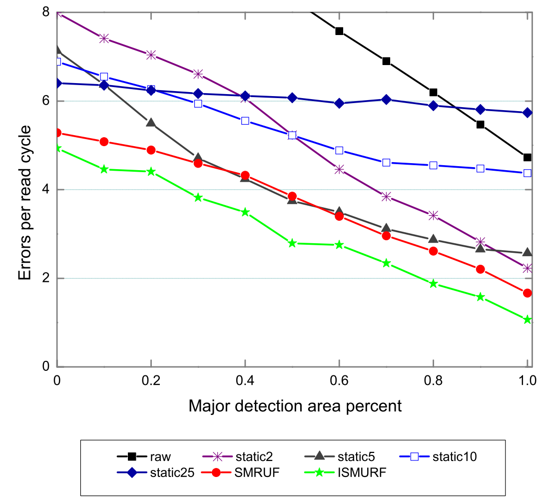

Fig. 2. Average errors per read cycle as major detection area percent changes.

TABLE I
Errors per read cycle in different major detection area percent

$p\#$	raw	static2	static5	static10	static25	SMURF	ISMURF
0	10.614	7.9924	7.1342	6.886	6.4004	5.282	4.9342
0.1	10.058	7.412	6.3702	6.5512	6.3586	5.084	4.459
0.2	9.771	7.0396	5.494	6.265	6.2404	4.8956	4.4049
0.3	8.8218	6.6066	4.7092	5.9414	6.1676	4.5926	3.8192
0.4	8.8254	6.0648	4.2406	5.5536	6.1132	4.3242	3.4888
0.5	8.2996	5.2296	3.744	5.2246	6.0752	3.8506	2.7883
0.6	7.5798	4.4584	3.4932	4.8838	5.9466	3.4002	2.7511
0.7	6.9	3.8432	3.1148	4.6116	6.0354	2.957	2.339
0.8	6.1974	3.4164	2.8698	4.547	5.8962	2.6124	1.875
0.9	5.4662	2.8212	2.652	4.4764	5.807	2.2044	1.5778
1	4.7292	2.2256	2.5692	4.3706	5.7352	1.6676	1.0652

average errors than the SMURF and the other fixed cleaning schemes. In 25 tags, our algorithm produces about 5 errors even in the noisiest environment, and almost reduces to 1 error when major detection area percent is 100%. In general, the average errors reduces about by 15% than SMURF in the experiment because of the improvement in transition detection scheme. Fig.3 displays the comparison between the percentage of data reliability of SMURF and ISMURF, we can see the significant improvement in our work, and the accuracy reaches to 95.7% when the major detection area percent is 100%.

In fixed cleaning algorithm, large window (static25) shows good accuracy when the major detection area percent is low,

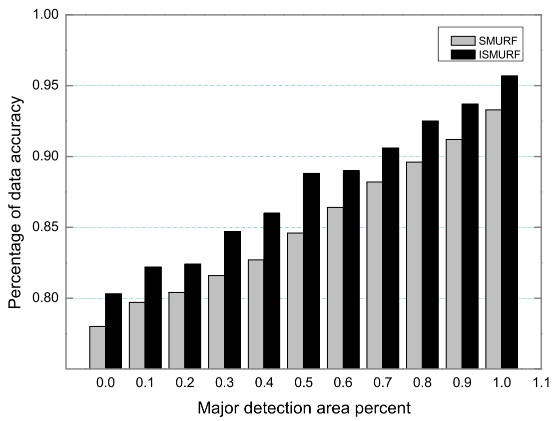

Fig. 3. Percent of data accuracy per read cycle as major detection area percent changes.

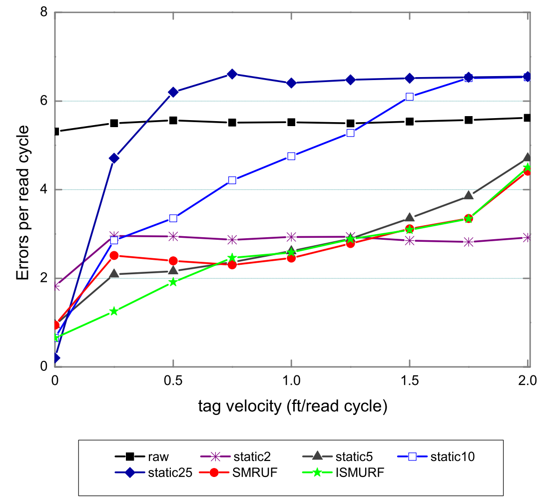

Fig. 4. Average errors per read cycle as tag velocity changes from 0 to 2 feet per read cycle.

and turn even worse than raw data when percent goes up, this due to data can be well compensated when the read-rate is very poor and wrongly filled when environmental condition is good (less noisy); small window (static2) has good accuracy only when the major detection area percent is high.

C. Experiment 2.

For experiment two, we also keep 5000 read cycles and the velocities are controlled in a fixed value and varied in 0 2 feet per read cycle to observe the average errors.

As presented in Fig.4, large window produces nearly no errors when tags are motionless, because there is no transition happen, but the false positive readings goes up sharply when tags turn dynamic. Small window shows a much stable average errors, static2 performs best(less than 3 errors per read cycle as showed in Table.2) when the velocity goes high, for its sensitive detection for transition.

Errors in SMURF presents higher than the improved scheme and staic5, when tags have a low speed, because the defect of transition detection contributes many false positive readings when observed reading is more than expected readings within a window. Improved algorithm performs best as the velocity is low (0.25 0.5 feet per read cycle); and the errors follows the SUMRF as the velocity goes up. When tags move fast (faster than 1.5 feet per read cycle), the transition is on too frequently, so it can be more effective to detect it than the small window.

TABLE II
ERRORS PER READ CYCLE IN DIFFERENT VELOCITY

v#	raw	static2	static5	static10	static25	SMURF	ISMURF
0	5.3126	1.824	0.9471	0.6508	0.2019	0.9418	0.6499
0.25	5.5003	2.9535	2.0876	2.853	4.7109	2.5129	1.2544
0.5	5.5615	2.9432	2.1591	3.3551	6.2018	2.3947	1.9107
0.75	5.5139	2.8702	2.3641	4.2112	6.608	2.3014	2.4591
1	5.5204	2.9297	2.6107	4.7524	6.4049	2.4574	2.5812
1.25	5.4943	2.9349	2.8904	5.2811	6.4786	2.7854	2.8812
1.5	5.5338	2.8508	3.3517	6.0982	6.5124	3.1175	3.0924
1.75	5.5715	2.8214	3.8504	6.5192	6.5376	3.35	3.341
2	5.6207	2.9171	4.7088	6.5349	6.5505	4.4172	4.5211

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper, we study the fixed cleaning method and adaptive cleaning method, and analyze the advantages and disadvantages of them. The experiments above show that the improved algorithm based on adaptive cleaning method is more accurate and efficient than the original one to improve data reliability. Our approach can perform a better accuracy in both missed readings and tag transitions, and it fits for the tags which speed is not very high.

B. Future Works

The algorithm we proposed is designed for single tag and single reader but not multiple ones. Although the reliability of data can be enhanced, the cost cannot be satisfied when we apply it to every tag and reader. In the future work,on one hand, we are going to study an efficient way to cleaning multiple tags with multiple readers' information. On the other hand, we plan to combine schemes which can clean both false positive readings and false negative readings.

REFERENCES

[1] Pearson, J., Securing the Pharmaceutical Supply Chain with RFID and Public-key Infrastructure (PKI) Technologies, *Texas Instruments White Paper, June 2005*.

[2] Bai, Y. and Wang, F. and Liu, P., "Efficiently filtering rfid data streams", *CleanDB Workshop*, 2005, pp 50-57.

[3] Jeffery, S.R. and Garofalakis, M. and Franklin, M.J., "A pipelined framework for online cleaning of sensor data streams", *Computer*,2005.

[4] Sullivan, L., "RFID Implementation Challenges Persist, All This Time Later", *Information Week*, 2005.

[5] Traub, K. and Bent, S. and Osinski, T. and Peretz, SN and Rehling, S. and Rosenthal, S. and Tracey, B., "The Application Level Events (ALE) Specification, Version 1.0", *EPCglobal Proposed Standard*, 2005.

[6] Jeffery, S.R. and Garofalakis, M. and Franklin, M.J., "Adaptive cleaning for RFID data streams", *Proceedings of the 32nd international conference on Very large data bases*, VLDB Endowment, 2006, pp 163-174.

[7] Kenney, T.J. and Niva, I. and Gray, S.D. and Dowling, J., "Device and method for detection and reduction of intermodulation distortion", *US Patent 6,009,129*,Dec 1999.

[8] Dyson, E. and Dean, E., "Rfid: Logistics meets identity", *Release 1.0: Esther Dysons Monthly Report*, 2003.

[9] Hodges, S. and Harrison, M., "Demystifying rfid: Principles & practicalities", *Cambridge University Engineering Department, Technical Report CUED/E-MANUF/TR-028*, 2003.

[10] Inc, E.P.C., "Class 1 generation 2 UHF air interface protocol standard version 1.0. 9", *EPCglobal Proposed Standard*, 2005.

[11] Braverman, V. and Ostrovsky, R., "Smooth Histograms for Sliding Windows", *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*,IEEE Computer Society Washington, DC, USA, 2007, pp 283-293.

[12] Yang, Y.R. and Lam, S.S., "General AIMD congestion control", *Proc. IEEE ICNP*, 2000, pp 187-198.