Efficient RFID Devices

Yair Wiseman
Computer Science Department
Holon Institue of Technology
Holon, Israel
yairw@hit.ac.il

***Abstract*— RFID equipment usually makes use of EPC which has many unexploited codewords. Actually, the unexploited codewords are the vast majority of the codewords. When there is a code with so many unexploited codewords, many compression algorithms can significantly reduce the average codeword size. In this paper we explain how to select the most effective compression algorithm for RFID equipment.**

I. Introduction

Electronic Product Code (EPC) is an agreed tag defines a unique code for every item anywhere in the world [1]. EPC format is well-defined and well-known [2].In point of fact, EPC standard's most important purpose is facilitating RFID tags to have an effective code for items that make use of RFID equipment. RFID and EPC suggest a technique for distinctiveness of every single item even where there are numerous of items.

More than decade ago the RFID tags have been rolled out on many products at Wal-Mart's more than 3,750 U.S. stores [3]. Some branches of the US government like Federal Aviation Administration (FAA), The Department of Defense and The Department of Homeland Security also initiated the employ of EPC tags more than a decade ago for several functions like warehouse supply chain, tracking armaments, smart borders etc. [4]. The selling price of RFID equipment has been decreased because of the more widespread usage by many organizations [5].

There are some other purposes for the EPC standard besides the RFID mechanism [6]. EPC information can be completely made up based on reading of optical data producers like various linear bar code versions; however EPC points towards defining an exclusive label to any single object in the world, whereas old-style barcodes normally only specify a manufacturer and a class of products.

Methods aiming at reducing the data of RFID equipment have been researched during the years so as to save power [7]. The power of low-cost passive RFID tag is very restricted because passive RFID tags do not have any internal power source. Passive RFID tags produce the electric power using the interrogating radio waves sent by the code transmitter. As a result the memory size for the entire EPC consumes most of its memory. Furthermore, the communication bandwidth should also take into account the limited potential power resources and send fewer bits. Consequently, saving power of RFID tags by reducing the EPC size is essential [8].

II. Compression of EPC codes

Alteration of EPC information has been already done for cryptography [9]. We also suggest altering the information of EPC for RFID tags, but instead of cryptography, our aim is compressing the information. Both cryptography and compression tends to omit redundant data, so the tasks are similar.

There are two main categories of compression: one for applications where the correlation between the bits is strong [10] and the other category where the correlation between the bits is weak [11,12,13]. In our compression scheme, we will employ one of the entropy encoders which are the reasonable choice for applications where the correlation between the bits is almost does not exist.

The most popular entropy encoders are Huffman Coding [14] and Arithmetic Coding [15]. The compression ratio of the Arithmetic Coding is always not worse than the compression ratio of Huffman Coding, therefore it is very common to choose the Arithmetic Coding; however, Huffman Coding can recover after an error in transmission, while Arithmetic Coding cannot. So, in an untrustworthy environment where errors are unexceptional, Huffman Coding can be preferred over the Arithmetic Coding [16].

It has been shown in [17] that an optimal compression algorithm gives every item with a probability P, a codeword in length of $-\log_2 P$ bits. Huffman Coding almost at all times produces non-optimal codewords because its algorithm rounds the codeword sizes in bits to an integer. Only if the round does not change all the looked-for codeword size i.e. all the looked-for codeword sizes are anyway integer, the produced code of Huffman will be optimal. If the distribution of an item set is Dyadic [18], all the looked-for codeword sizes will be anyway integer, but this is a very rare case; therefore, Huffman Coding is almost always less efficient than Arithmetic Coding. Unlike Huffman Coding, Arithmetic Coding does not require codewords in an integer size; however, the execution time of Arithmetic Coding is always much longer than Huffman Coding.

Even though Arithmetic Coding is quite slow, nowadays there are strong enough processors that can handle even heavy processes and even in Real Time systems, Arithmetic Coding is an option [19]. Furthermore, there have been many studies on how to alleviate the long execution time problem of the Arithmetic Coding [20].

Arithmetic Coding can allocate fractions of bits as codewords [21]; not like Huffman coding that can use a string of bits for each item in the original file. In Arithmetic Coding one bit can be "owned" by several items. For that reason, any codeword can be denoted by a non-integer numbers of bits (e.g. by 4.7 bits). The Arithmetic Coding algorithm is shown in this pseudocode:

Let A be a set of items.

Each item i in A has a probability Pi within [0,1], such that:

$$\sum_{i \in A} P_i = 1$$

Each item is denoted by the interval:

$$[\sum_{j < i} P_j , \sum_{j \le i} P_j)$$

Repeat until EOF:

The current interval is divided into sub-intervals according to the items‘ probabilities.

Replace the current interval by the sub-interval of the items that were read.

Write into the compressed file the shortest binary fraction available in the current interval.

III. Adapting of Arithmetic Coding for EPC Compression

The conventional version of EPC is 96 bits long and includes four fields:

1. Header – 8 bits that classifies the length, type, structure, version, and generation of the EPC.
2. EPC Manager Number – 28 bits that classifies the manufacturer, the business or the company of a line of items for selling (28 bits).
3. Object Class – 24 bits that classifies a category of items.
4. Serial Number – 36 bits that classifies a specific item.

For example the EPC tag for a specific Herbal Essences Shampoo of Clairol is

Header is 0011 0000 for Serialized Global Trade Item Number (SGTIN-96)

EPC Manager Number is 0000 0000 0000 0000 1011 0100 0111 for Clairol Company.

Object Class is 0000 0000 0000 0010 0110 1111 for Herbal Essences Shampoo.

Serial Number is 0000 0000 0000 0010 0111 1000 1011 0110 0000 for a specific bottle of Herbal Essences Shampoo.

Accordingly, the complete EPC tag for this particular Herbal Essences Shampoo of Clairol is:

0011 0000 0000 0000 0000 0000 1011 0100 0111 0000 0000 0000 0010 0110 1111 0000 0000 0000 0010 0111 1000 1011 0110 0000

Provided that EPC allocates 28 bits for the manufacturer/business/company which gives 2^{28} possibilities for manufacturer/business/company, many of the potential combinations obviously will not be in use. The same wastefulness allocation of codes also takes place in the Object Class field. EPC allocates 2^{24} would-be objects, but on the whole, manufacturers/businesses/companies do not have so many items for consumption. Similarly, EPC allocates for the Serial Number 2^{36} possible particular objects and this allocation is also almost always very wastefulness; however, it should be noticed that the actual codes of this field tend to be scattered throughout the possible codes, because most of the manufacturers/businesses/companies begin a new series of codes in a new year, a new line or other similar reasons.

The codes of EPC tags have some characteristics because of these many unused potential codes:

- Most of the potential codes are not in use.
- The actual used codes tend to cluster together within several groups of codes.

Several compression algorithms are able to efficiently deal with data with such qualities. Two of them are JPEG [22] and MP3 [23]. We have taken their concept and have adjusted it to our circumstances. MP3 and JPEG divide their data into pairs of:

1. A recurring sequence.
2. The rest of the data.

After that, they assign each of the pair a unique codeword (When Huffman Coding is used) or a unique interval (if Arithmetic Coding is used).

We adapted this technique and adjusted it to EPC tags compression. We explored the probabilities of the possible pairs and built tables of intervals for the Arithmetic Coding algorithm that fit the EPC data.

IV. Results

The dataset that we have used to evaluate our compression scheme were collected form the selling information of a retail corporation during 2015.

We have built an independent table of intervals for every segment of the RFID tags. The use of the current statistics obviously creates tables of intervals that are just right for this current statistics. Usually the statistics changes and this change can make our tables of interval outdated and as a result the compression will be less efficient; however, usually the statistics does not significantly change, so the compression will not be ineffective. Besides, a change in the tables of intervals can be made from time to time.

Currently, only several tens of thousands of manufacturers/businesses/companies make use of RFID tags with EPC. EPC standard stipulates that the length of EPC Manager Number segment is 28 bits, but for the existing tens of thousands of manufacturers/businesses/companies only 15 bits are actually required.

The dataset that we have used reflects only the data of one retail corporation, so most of the tens of thousands of manufacturers/businesses/companies do not appear in our dataset. Actually, we have in our dataset only 978 manufacturers/businesses/companies. We have built the tables of intervals according to this dataset and we have considered the manufacturers/businesses/companies that do not exist in the dataset as very infrequent. Definitely, if we used a broader dataset, the compression and the tables of the intervals could be very different.

Usually, most manufacturers/businesses/companies sell only few items to retail corporations. In Figure 1 we detail the number of manufacturers/businesses/companies that have sold N items in 2015 to the retail corporation we have checked, where N was range from 1 to 23, because there have been no manufacturer/business/company that sold more than 23 items to the retail corporation.

Fig. 1. Number of items per manufacturer/business/company

EPC standard stipulates that the Object Class field will be in length of 24 bits. This length allows for each of the manufacturers/businesses/companies more than 16 million classes of items; however, this is much more than what is needed for them. This excess length can be resolved the Arithmetic Coding.

The Serial Number field contains a specific number for each individual item and its length also always excesses the filed real requirements. According to the EPC standard, the field length is 36 bit. This length means that essentially there can be more than 68 billion items in each class i.e. more than 68 billion of the very same item.

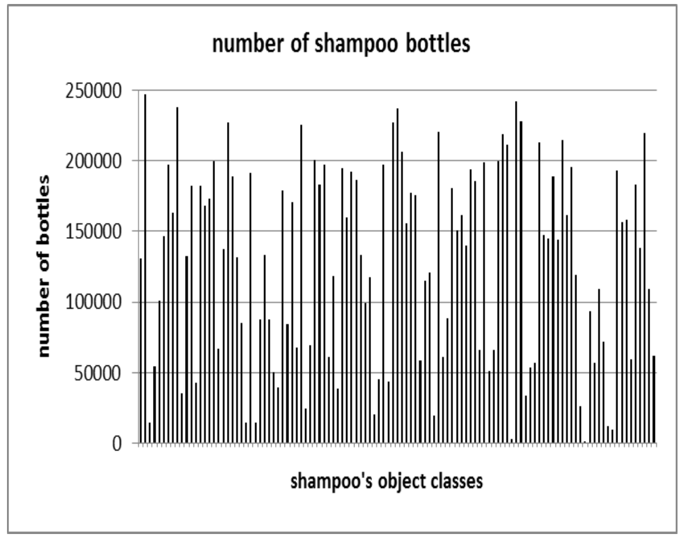

Fig. 2. Number of Clairol shampoo bottles sold in 2015.

We counted the number of items of several types of Clairol shampoo bottles. There were 127 different Clairol shampoo bottles in the retail corperation.

Figure 2 details the number of Clairol shampoo bottles for each Object Class sold in 2015. Each one of the bars in the chart stands for one kind of Clairol shampoo. The titles of the Clairol shampoos are not written within the diagram since there is no room for 127 titles in the diagram and they are not necessitated for understanding how we have compressed the data.

More than 68 billion items is far too much as can be seen in Figure 2. Even the most sold shampoo demanded just 18 bits to denote how many times it has been sold.

Additionally, the Arithmetic Coding is an entropy encoder; therefore, if most of the data for compression is identical, the entropy will be at a low level and the Arithmetic Coding will work at its best.

Over all, as was mentioned EPC tag is in length of 96 bits and it has been averagely compressed to 32.94 bits which are 34.3% of the original length.

This result was obtained as follow:

The header of the EPC has been omitted because just SGTIN-96 has been employed in our system.

The EPC Manager Number field length has been averagely reduced from 28 bits to 9.41 bits.

The Arithmetic Coding average code length is the entropy of the compressed items. This average code length can be computed by this expression:

$$\sum_{i=1}^{n} P_i Log(P_i)$$

Where n is the number of the compressed items and P_i are the items' probability. Substituting the data of figure 1 in this expression will yield 6.34 or in other words, Object Class field has been compressed from 24 bits into 6.34 bits

Similarly, the Serial Number filed can be calculated according to the same expression and data shown in figure 2. The result of this calculation is 17.19 bits which means the 36 bits of the Serial Number filed have been compressed into just 17.19 bits.

Figure 3 specifies the compression ratio of the EPC tag fields.

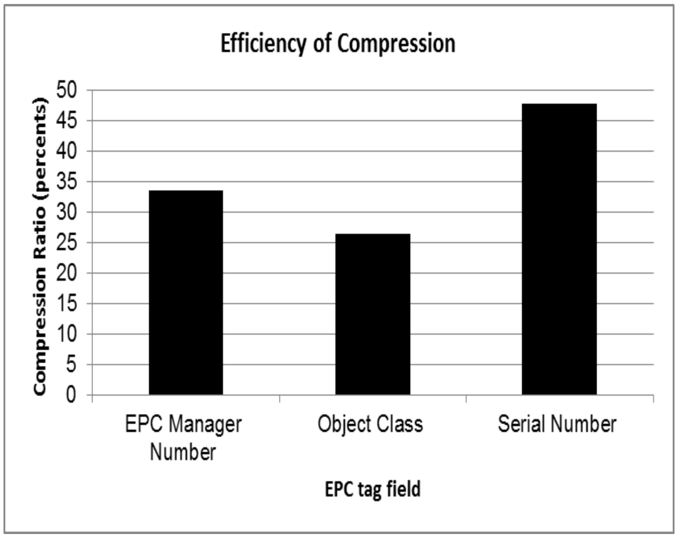

Fig. 3. Compression ratio percent for each of EPC tag fields

V. Conclusions and Future work

The results of this paper are encouraging. EPC average tag length has been compressed into 34.3% of the original length. Such results can facilitate the use of passive RFID tags in many more utilities.

In the common literature, two main methodologies have been proposed with the aim of encountering the restricted communication bandwidth and the restricted electric power of passive RFID devices [24]:

- Decreasing the average EPC tag length into fewer bits.
- Constructing low power energy consuming passive tags.

This paper implements the first methodology. The suggested compression scheme efficiently decreased EPC tag lengths. This decrease can improve the propagation of the RFID devices in the market because it is easier and less expensive to produce RFID devices that consume less electric power.

In the future, we consider generating more comprehensive dataset. Additionally, we consider reworking our implementation so as to enable adapting the Arithmetic Coding tables of interval according to up-to-date adjusted lists of manufacturers/businesses/companies and up-to-date adjusted lists of merchandises.

References

[1] K. Myny, S. Steudel, S. Smout, P. Vicca, F. Furthner, B. van der Putten, A. K. Tripathi, G. H. Gelinck, J. Genoe, W. Dehaene, and P. Heremans, "Organic RFID transponder chip with data rate compatible with electronic product coding." Organic Electronics Vol. 11, no. 7, pp. 1176-1179, 2010.

[2] H. Hada, and J. Mitsugi, "EPC based internet of things architecture", In Proceedins of IEEE International Conference on RFID-Technologies and Applications (RFID-TA), pp. 527-532, 2011.

[3] M. Bustillo, "Wal-Mart radio tags to track clothing", Wall Street Journal, July, 2010.

[4] US. Government Accountability Office, "Information Security: Radio Frequency Identification Technology in the Federal Government", Report to Congressional Requesters, GAO-05-551, 2005.

[5] A. Sarac, N. Absi and S. Dauzère-Pérès, "A literature review on the impact of RFID technologies on supply chain management", International Journal of Production Economics, Vol. 128, no. 1, pp. 77-95, 2010.

[6] J. Mitsugi, Y. Sato, T. Yokoishi, T. Tashiro, T. Eda, K. Sato and Y. Kishi, "Regional purchase orders dissemination and shipments aggregation of agricultural products with interworking EPC network and EDI system" In Proceedings of IEEE International Conference on RFID-Technologies and Applications (RFID-TA) , pp. 1-6, Johor Bahru, Malaysia, 2013.

[7] Y. Wiseman, "Compression scheme for RFID equipment", Proceedings of IEEE International Conference on Electro/Information Technology (EIT 2016), Grand Forks, North Dakota, USA, 2016.

[8] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li and D. Chen, "Efficient tag identification in mobile RFID systems", In Proceedings of IEEE International Conference on Computer Communications, pp. 1-9, San Diego, California, 2010.

[9] H. J. Chae, M. Salajegheh, D. J. Yeager, J. R. Smith, and K. Fu, "Maximalist cryptography and computation on the WISP UHF RFID tag", In Wirelessly Powered Sensor Networks and Computational RFID, pp. 175-187, Springer New York, 2013.

[10] Y. Wiseman, "The Relative Efficiency of LZW and LZSS", Data Science Journal, Vol. 6, pp. 1-6, 2007.

[11] Y. Wiseman, "Enhancement of JPEG Compression for GPS Images", International Journal of Multimedia and Ubiquitous Engineering, Vol. 10, No. 7, pp. 255-264, 2015.

[12] Y. Wiseman and E. Fredj, "Contour Extraction of Compressed JPEG Images", Journal of Graphic Tools, Taylor & Francis publications, Vol. 6, No. 3, pp. 37-43, 2001.

[13] Y. Wiseman, "Alleviation of JPEG Inaccuracy Appearance", International Journal of Multimedia and Ubiquitous Engineering, Vol. 11, No. 3, pp. 133-142, 2016.

[14] D. A. Huffman, "A method for the Construction of Minimum Redundancy Codes", In Proceedings of the Institute of Radio Engineers, Vol. 40, pp. 1098-1101, 1952.

[15] Y. Wiseman, "A Pipeline Chip for Quasi Arithmetic Coding", IEICE Journal - Trans. Fundamentals, Tokyo, Japan, Vol. E84-A No.4, pp. 1034-1041, 2001.

[16] S. T. Klein and Y. Wiseman, "Parallel Huffman Decoding with Applications to JPEG Files", The Computer Journal, Oxford University Press, Swindon, UK, Vol. 46(5), pp. 487-497, 2003.

[17] C. E. Shannon, "A Mathematical Theory of Communication, Bell System", Tech. Journal, vol. 27, pp. 398-403, 1948.

[18] G. Bocherer and R. Mathar, "Matching Dyadic Distributions to Channels" In proceedins of IEEE Data Compression Conference, Snowbird, Utah, pp. 23-32, 2011.

[19] G. Fuchs, V. Subbaraman and M. Multrus, "Efficient context adaptive entropy coding for real-time applications" In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 493-496, 2011..

[20] M. Tao, M. Hempel, D. Peng, and H. Sharif. "A survey of energy-efficient compression and communication techniques for multimedia in resource constrained systems", IEEE Communications Surveys & Tutorials, Vol. 15, No. 3, pp. 963-972, 2013.

[21] Y. Wiseman, K. Schwan and P. Widener, "Efficient End to End Data Exchange Using Configurable Compression", Operating Systems Review, Vol. 39, No. 3, pp. 4-23, 2005.

[22] Y. Wiseman, "The still image lossy compression standard – JPEG", Encyclopedia of Information and Science Technology, Third Edition, Vol. 1, Chapter 28, pp. 295-305, 2014.

[23] J. Sterne, "MP3: The meaning of a format", Duke University Press, 2012.

[24] C. R. Valenta and G. D. Durgin, "Harvesting wireless power: Survey of energy-harvester conversion efficiency in far-field, wireless power transfer systems", IEEE Microwave Magazine Vol. 15, no. 4, 108-120, 2014.