# Bin-slotted Hybrid Search Algorithm for Multiple RFID Arbitration

**Ji-Yoon Kim** **Bong-Soo Kang** **Jeong-Woo Jwa** **Doo-Yeong Yang**

Department of Telecommunication Engineering
Cheju National University
Jeju, Korea

***Abstract*** *- In this paper, bin-slotted hybrid search algorithm (BHS) combined slotted ALOHA procedure with bin-tree procedure is proposed and analyzed. Also, the performance of proposed anti-collision algorithm is evaluated as comparing the BHS algorithm with a standard bin-slotted algorithm (BSA) through the simulation. The performance of the proposed BHS algorithm is improved by dynamically identifying the collided-bit position and the collided bins stored in the stack of the reader. As the results, the reduction ratio of the number of request command that the reader sends to tags in its interrogation zone and the total recognition is about 59% for 1000 tags as compared with those of BSA algorithm. Therefore, the tag identification performance is fairly improved by using the proposed BHS algorithm in resolving a collision problem.*

***Keywords:*** *RFID, bin-slotted hybrid search algorithm, bin-slotted algorithm, anti-collision algorithm*

## 1 Introduction

Radio frequency identification (RFID) is considered as a core technology of ubiquitous sensor network (USN). Recently, international organization for standardization (ISO) and EPC-global organization have announced their documents for RFID Air-interface which is a standard communication protocol between a reader and passive tags at the 860~960MHz (ISO 18000-6, EPC Class1 protocol ). In USN, An RFID system supports numerous useful applications such as supply chain management, inventory tracking and access control [1][2]. The RFID system can simultaneously recognize several tags within the reader's interrogation zone by using anti-collision algorithm. Also, it can identify the unique tag ID, or other information included in microchip of the tag. A typical RFID system consists of a reader, tags and a middleware. To identify the tag ID, the reader communicates with tags through a radio frequency (RF) communication link. An individual tag is first identified from a group of tags in the reader's interrogation zone by using anti-collision algorithm, and then the communication occurs between the identified tags and the reader. A tag consists of an antenna and a microchip stored the object ID. A middleware processes, controls, and manages the object data collected from the RFID reader. If there are several tags in the reader's interrogation zone, a collision occurs between data received from the tags. The collision can be resolved by using anti-collision algorithms[3][4]. The capability of tag recognition and selection depends on the performance of anti-collision algorithm. The anti-collision algorithms are generally divided into binary search procedure (BS) and ALOHA procedure [5]-[7]. In the BS procedure, the tag is selected from a group by intentionally causing data collision in the serial numbers transmitted to the reader following a request command. If this procedure is to succeed, the reader is capable of determining the precise bit position where the collision detected by using a suitable data processing scheme. In the ALOHA procedure to resolve a collision, tags only transmit the defined data packets on the synchronous time slots to the reader. The synchronization of all tags needed for this procedure is controlled and checked by the reader.

In this paper, bin-slotted hybrid search (BHS) algorithm combined the slotted ALOHA procedure with bin-tree procedure is presented and analyzed. Also, the performance of the proposed BHS algorithm and of the bin-slotted algorithm (BSA) which is a standard EPC class1 protocol are compared and investigated. The remainder of the paper is organized as follows. In section II, it explains RFID system based on the bin-slotted algorithm (BSA), which is presented on the EPC Class1 protocol. In section III, the bin-slotted hybrid search algorithm (BHS) is proposed. In section IV, the performance of those algorithms is analyzed, and the conclusion is given in section V.

## 2 Communication Interface for Passive RFID System

To initiate a passive back-scattered RFID system, a reader transmits a modulated signal with periods of continuous RF carrier, which is received by the tag antenna. The RF energy developed on antenna terminals during the RF carrier period is converted to DC. This energy powers up the tag chip, which sends back the information by varying its front end complex RF input impedance. The impedance typically toggles between two different states - conjugate match, or complete mismatch - effectively modulating the back-scattered signal. In UHF EPC Class1 protocol, a communication between a reader and tags is performed in a packet based manner, where a single packet contains a complete command from a reader and a complete response from a tag. The communication method used between them is a half-duplex scheme, meaning that a reader talks and tags listen, and the intended interrogation zone of a reader is nominally 2~10m area. A reader transmits a complete command which is modulated by RF signal to a tag. The tag receives both information and operating energy from RF signal of a reader. And then the tag transmits its stored data to the reader by back-scattered modulation signal. Tag ID of EPC Class1 is formatted by [HEADER] [DOMAIN MANAGER] [OBJECT CLASS] [SERIAL NUMBER], and its numeric is composed of 64 bits, or 96 bits [8] [9].

For a reader command, a complete command from a reader consists of eight fields and five parity bits over those fields. The commands enable selection based on the tag's CRC, EPC and password. The command format is as follows: [PREAMBLE][CLKSYNC][SOF][CMD][P1][PTR][P2][LEN][P3][VALUE][P4][P5][EOF]. [PREAMBLE] is the continuous wave (CW) signal that supply energy to tags, and its duration is 64μs. [CLKSYNC] consists of a series bit string of 20 binary zeros for on-tag clock synchronization. [CMD] specifies the command being sent to the tag and the number of bit is eight. [PTR] is the starting point for tags to attempt a match with data specified in the [VALUE] field and the number of [PTR] is represented by using eight bits. [LEN] is length of data being sent [VALUE] field, and the number of [LEN] bit is represented by using eight bits. [VALUE] is data that the tag will attempt to mach against its identifier, and length of [VALUE] is variable. [SOF] is the start of frame indicator and [EOF] is the end of frame indicator.

For tag response, a Class1 compliant tag will change its internal state or perform backscatter modulation in response to the commands. If a tag in interrogation zone is activated by a continuous wave signal (CW) of suitable power from a reader, the tag gets into 'Awake' state that can be changed into all possible state. The tag receives the required commands from a reader, gets into 'Reply' state, and transmits its response signal to a reader. Whenever a reader selects tags, the reader transmits 'Quiet' command to the identified tag because the signal from the completely identified tag act as interference to the others. And the tag received the command gets into 'Asleep' state for the while to do not respond. Reader commands mainly used in tag recognition process are PingID and ScrollID. PingID command requires signaling of the tag, and ScrollID command matches and identifies the tag. The tag gets into 'Reply' state whenever it receives PingID and ScrollID command.

Figure 1 shows tag response for PingID command and the bin-modulation procedure that represents the process. There are five distinct phases in reader-to-tag communication. The reader-to-tag communication begins with transaction gap in the first phase, followed by 64μs CW period in the second phase. The first and the second phase comprise the [PREAMBLE] of the reader command. The third phase is data modulation window that transmits command packet of the reader. A short tag setup period in the fourth phase provides time for the tags to interpret and execute the command just issued by the reader. Thereby, the reader enters a low modulation phase during the tags response to the just communicated command. At the beginning of data modulation, the reader provides a master clock signal to tags. This is [CLKSYNC] period of the command. The time between the negative-going edges of the low intervals determines the reader-to-tag data rate. In order for the tag to be able to detect the next transaction gap, the reader must start the next transaction within the coast interval (Tag Setup + Tag Response Window). The coast interval has been regulated by maximum 20ms in the EPC Class1 protocol. Tags transmit its Tag ID at the fifth phase in response to PingID and ScrollID command. In tag recognition process, a reader progresses the bin-modulation work of 3-bit to tags in its interrogation zone after transmitting PingID. The tags matching [VALUE] beginning at the location [PTR] respond by sending 8-bit of the tag identifier beginning with the bit at the location [PTR]+[LEN]. The 8-bit reply is transmitted during one of 8 bins delineated by bin-modulation from the reader. If several tags respond in one of bins, the reader repeatedly transmits PingID adding the bin data on [VALUE] to the tag. And then if there is no collision between data transmitted during each bin, the reader transmits ScrollID adding the bin data on [VALUE] to the tag. The tag matched by ScrollID responds its entire ID to the reader. By checking the CRC of Tag ID, the reader completely identifies the tag.

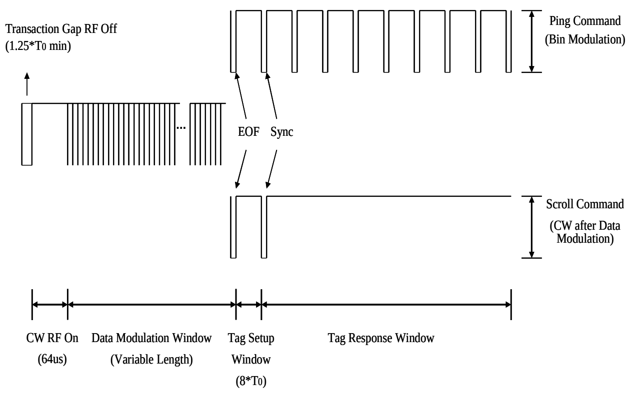


Fig. 1. Reader-to-tag modulation

# 3 The Bin-slotted Hybrid Search Algorithm

The bin-slotted hybrid search algorithm (BHS) is a hybrid algorithm combined a slotted ALOHA procedure with bin-tree procedure. In the binary tree algorithm, a reader queries to all tags for the next bits of their serial IDs and typically explores by bit-by-bit verifying scheme. The bin-slotted algorithm (BSA) explores by bin-by-bin detection scheme in bin tracking and sends request commands repeatedly to recognize all tags without storing the collided bin response data. However, the bin-slotted hybrid search algorithm proposed in this paper explores the 3-bit up to the maximum 8-bit in each bin using the collided bin response data stored in the stack of the reader and a hierarchical bin-by-bin detection scheme. On detecting the collided bins, the reader splits the respond set and queries until there is only one tag response by using the BHS algorithm. The bin exploration form is illustrated in Fig. 2.

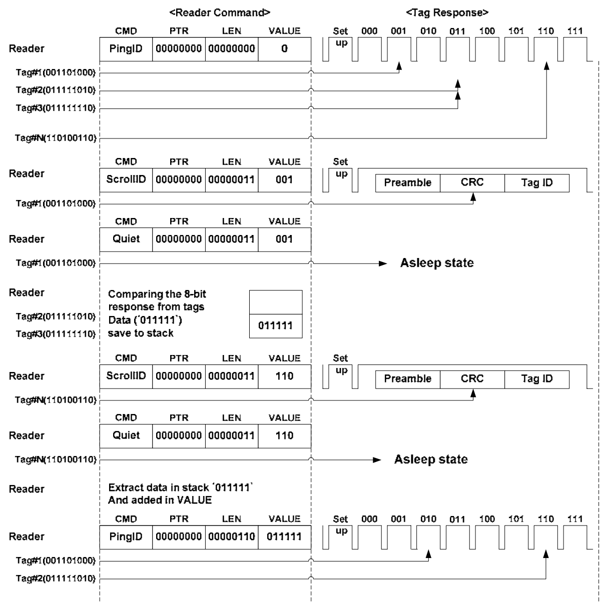


Fig. 2. Bin-slotted hybrid search algorithm.

In the BSA procedure, tag recognition process is begun by that the reader transmits PingID command including 'Value = 0, Length = 0' to tags in the reader's interrogation zone. The reader executes bin-modulation work, and the tag coinciding with the bin data replies in each bin. The reader examines bins, and if the collision does not occur, the reader sends ScrollID command adding the bin data on [VALUE] to tags. The tag consistent with the value of ScrollID sends its entire ID to the reader as the response for ScrollID command. The reader performs CRC check to detect whether there is error for the received tag ID or not. However, if the collision occurs in bins, the reader restarts processing procedure by PingID command adding the bin data on [VALUE]. The examination of bins begins from the first bin (the value of '000') again.

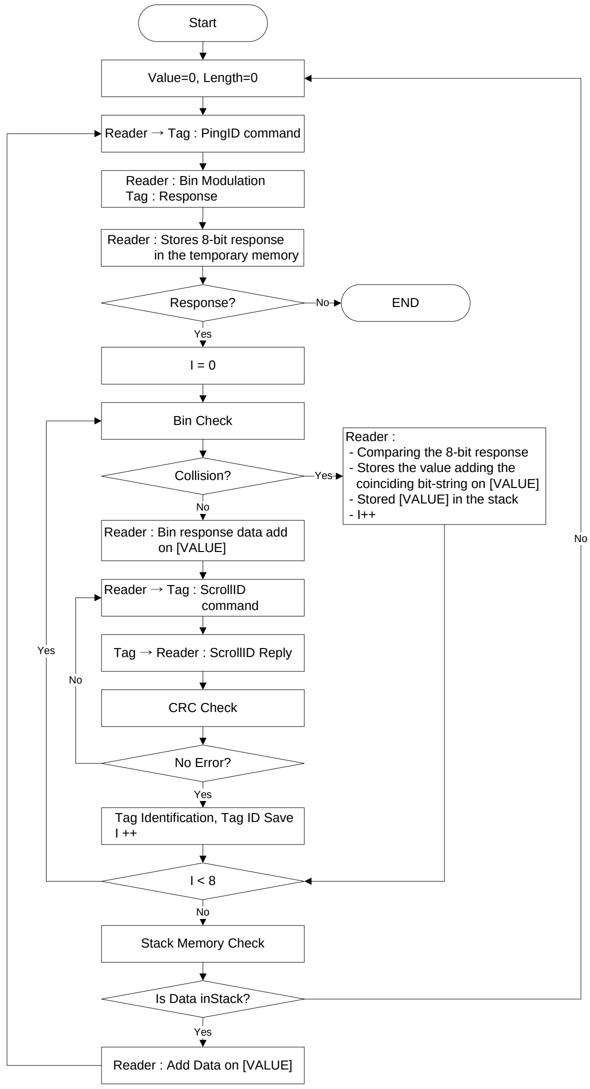


Fig. 3. Flow chart of the proposed BHS algorithm.

In this paper, by introducing a concept of stack and dynamic bin search in the bin-slotted algorithm (BSA), we reduce the number of reader command, and the recognition time of tags. First, tag recognition process is initiated by the PingID command transmitted from reader to tags in its interrogation zone. The reader performs bin-modulation after transmitting PingID command, and the tags respond the 8-bit of tag identifier in their bins coinciding with the next 3-bit at the location [PTR]+[LEN] during the bin-modulation. And then the reader examines the bins. If the collided bins do not exist, the reader adds the bin data on [VALUE], sends ScrollID command to tags and recognizes the tag ID. Otherwise, if the collided bins exist, the bin response data are stored in the stack of the reader. In order to identify the collided tags, the reader dynamically examines the collided-bit position of stored data in the stack. And then the reader transmits PingID command adding the data until the just previous position of the first collision-bit on [VALUE] to the tags. The collision resolution process is repeated until there is no collided data in the stack. The BHS algorithm procedure is briefly represented as Fig.3.

For instance, as shown in Fig. 2, Tag#1 responds ‘00110100’ in the second bin (001), Tag#2 and Tag#3 respond ‘01111101’, ‘01111111’ in the forth bin (011), Tag#N responds ‘11010011’ in the seventh bin (110) when [PTR]=0. Then the reader stores all of the response data of 8-bit in the temporary memory of the reader, and examines the bins according to ascending order of bin. In case that Tag#1 responds in the second bin without the collision, the reader transmits ScrollID adding the first bin data ‘001’ on [VALUE]. The Tag#1 matched by [VALUE]=’001’ sends its entire ID to the reader. By checking the CRC with the received ID, the reader completely identifies the ID ‘001101000’ of Tag#1 if there is no error. The procedure of Tag#N also is the same with the procedure of Tag#1. However, in case that Tag#2 and Tag#3 respond in the fourth bin with the collision, after comparing the 8-bit response data ‘01111101’ of Tag#2 with the 8-bit response data ‘0111111’ of Tag#3 by bit-string comparison, the reader stores the value adding the coincided bit-string ‘011111’ on [VALUE] in the stack. And then the reader extracts the value in the stack, retransmits PingID command including the [VALUE]=‘011111’ to the tags. Tag#2 and Tag#3 receiving the PingID respond the next 8-bit of their IDs in the third bin (010) and the seventh bin (110), respectively. The Tag#2 matched by [VALUE]=‘011111010’ sends its entire ID to the reader. By checking the CRC with the received ID, the reader completely identifies the ID ‘011111010’ of Tag#2 if there is no error for the ID. The procedure of Tag#3 also is the same with the procedure of Tag#2.

# 4 Simulation Results

In this section, the simulation of BSA and proposed BHS algorithm are carried out. And the performance comparison between them is investigated. The first is BSA algorithm based on the EPC Class1 protocol. The communication between the reader and tags and the recognition procedure are begun from the starting point of the procedure without depending on whether the collision on the bin response data is occurred during bin modulation or not. The second is BHS algorithm that tag identification performance of it is improved by checking the stored response data of collision-bins in the stack. Performance comparison for two methods is analyzed by the simulation program coded by C language.

In order to calculate the number of reader command identifying tags, the number of bit string of tag ID sets by 64 bits, or 96bits. The tag set is from 10 to 1000 tags.

The simulation results for the anti-collision algorithms are shown in Table I and Table II. The calculated parameters are an average number of PingID commands (No. of PingID), an average number of total reader command (No of T.C), an average length of [VALUE] field per a PingID command (VLP), an average length of [VALUE] field per a ScrollID command (VLS) being sent from the reader to tags, total tag recognition time (T.R.T), and total recognition time decrements of the BHS against the BSA algorithm. Table I shows the results for 64-bit tag ID. Table II shows 96-bit case. In detecting the collided bin response data, the BHS-static algorithm explores by the fixed 3-bit in the 8 bins. The BHS-dynamic algorithm explores from the 3-bit up to the maximum 8-bit in the same bins.

TABLE I
DEFAULT VALUES USED IN THE COMPUTER SIMULATIONS

| No.of Tags | Algorithm | No.of PingID | No.of T.C | VLP[bit] | VLS[bit] | T.R.T[ms] | Reduction of T.R.T[%] |
|---|---|---|---|---|---|---|---|
| 10 | BSA | 17.0 | 37.0 | 1.41 | 5.10 | 62.89 | 0 |
| | BHS_static | 5.0 | 25.0 | 3.00 | 6.54 | 39.85 | 36.6 |
| | BHS_dynamic | 4.8 | 24.8 | 3.50 | 6.90 | 39.47 | 37.3 |
| 30 | BSA | 68.2 | 128.2 | 2.34 | 6.82 | 224.88 | 0 |
| | BHS_static | 14.6 | 74.6 | 5.05 | 8.08 | 120.45 | 46.4 |
| | BHS_dynamic | 12.8 | 72.8 | 5.48 | 8.37 | 116.95 | 48.0 |
| 50 | BSA | 123.0 | 223.0 | 2.52 | 7.38 | 395.02 | 0 |
| | BHS_static | 22.8 | 122.8 | 6.42 | 9.02 | 198.95 | 49.6 |
| | BHS_dynamic | 21.2 | 121.2 | 6.08 | 9.06 | 195.81 | 50.4 |
| 100 | BSA | 282.8 | 482.8 | 3.07 | 8.48 | 868.28 | 0 |
| | BHS_static | 47.8 | 247.8 | 7.16 | 10.00 | 405.55 | 53.3 |
| | BHS_dynamic | 42.4 | 242.4 | 6.94 | 9.06 | 195.81 | 54.5 |
| 200 | BSA | 634.6 | 1034.6 | 3.58 | 9.52 | 1889.94 | 0 |
| | BHS_static | 91.0 | 491.00 | 7.87 | 10.74 | 810.14 | 57.1 |
| | BHS_dynamic | 87.6 | 487.6 | 8.14 | 11.18 | 803.39 | 57.5 |
| 500 | BSA | 1807.6 | 2807.6 | 4.21 | 10.85 | 5219.74 | 0 |
| | BHS_static | 248.2 | 1248.2 | 9.43 | 12.35 | 2091.38 | 60.0 |
| | BHS_dynamic | 222.0 | 1222.0 | 9.54 | 12.58 | 2038.82 | 60.9 |
| 1000 | BSA | 3950.2 | 5950.2 | 4.68 | 11.85 | 11189.56 | 0 |
| | BHS_static | 485.4 | 2485.4 | 10.32 | 13.33 | 4193.77 | 62.5 |
| | BHS_dynamic | 437.2 | 2437.2 | 10.45 | 13.52 | 4096.45 | 63.4 |

TABLEII
DEFAULT VALUES USED IN THE COMPUTER SIMULATIONS

| No.of Tags | Algorithm | No.of PingID | No.of T.C | VLP[bit] | VLs[bit] | T.R.T[ms] | Reduction of T.R.T[%] |
|---|---|---|---|---|---|---|---|
| 10 | BSA | 18.2 | 38.2 | 1.78 | 5.46 | 68.90 | 0 |
| | BHS_static | 5.2 | 25.2 | 3.00 | 6.30 | 44.59 | 35.3 |
| | BHS_dynamic | 4.8 | 24.8 | 3.63 | 6.64 | 43.96 | 36.2 |
| 30 | BSA | 68.0 | 128.0 | 2.27 | 6.80 | 233.72 | 0 |
| | BHS_static | 15.8 | 75.8 | 5.16 | 8.34 | 136.37 | 41.7 |
| | BHS_dynamic | 13.2 | 73.2 | 5.30 | 8.40 | 131.40 | 43.8 |
| 50 | BSA | 124.6 | 224.6 | 2.63 | 7.48 | 412.57 | 0 |
| | BHS_static | 25.4 | 125.4 | 5.86 | 9.05 | 226.74 | 45.0 |
| | BHS_dynamic | 21.0 | 121.0 | 5.79 | 8.93 | 217.95 | 47.2 |
| 100 | BSA | 285.4 | 485.4 | 3.11 | 8.56 | 899.23 | 0 |
| | BHS_static | 50.8 | 250.8 | 7.07 | 10.10 | 457.35 | 49.1 |
| | BHS_dynamic | 42.4 | 242.4 | 6.94 | 10.03 | 440.51 | 51.0 |
| 200 | BSA | 634.6 | 1034.6 | 3.59 | 9.52 | 1930.43 | 0 |
| | BHS_static | 96.4 | 496.4 | 7.32 | 10.96 | 909.69 | 52.9 |
| | BHS_dynamic | 87.6 | 487.6 | 8.14 | 11.18 | 894.59 | 53.7 |
| 500 | BSA | 1810.6 | 2810.0 | 4.22 | 10.86 | 5291.99 | 0 |
| | BHS_static | 247.6 | 1247.6 | 9.32 | 12.32 | 2313.69 | 56.3 |
| | BHS_dynamic | 222.0 | 1222.0 | 9.54 | 12.58 | 2309.57 | 56.4 |
| 1000 | BSA | 3975.0 | 5975.0 | 4.75 | 11.93 | 11327.45 | 0 |
| | BHS_static | 481.6 | 2481.6 | 10.32 | 13.31 | 4635.22 | 59.1 |
| | BHS_dynamic | 435.2 | 2435.2 | 10.45 | 13.51 | 4548.13 | 59.8 |

As shown in Table I, in the case of BHS-static algorithm, the reduction ratio of the number of PingID command for the tag recognition is 83.1% for 100 tags, 87.7% for 1000 tags against that of BSA algorithm. And in the case of BHS-dynamic algorithm, the ratio is 85% for 100 tags, 88.9% for 1000 tags.

From Table II, in the case of BHS-static algorithm, the reduction ratio of the number of PingID command for the tag recognition is 49.1% for 100 tags, 87.9% for 1000 tags against that of BSA algorithm. And in the case of BHS-dynamic algorithm, the ratio is 82.2% for 100 tags, 87.9% for 1000 tags.

The number of total reader command transmitted from the reader to tags is sum of the number of PingID command, the number of ScrollID command and the number of Quiet command. In this simulation, the number of ScrollID command and the number of Quiet command are the same with the number of tag since it is assumed that tags transmit their IDs to the reader without error.

$$C_{Total} = N_P + N_S + N_Q = N_P + 2 \times N_{Tags} \quad (1)$$

where NP , NS and NQ are the required number of PingID command, ScrollID command and Quiet command, respectively. An average value of the number of total reader command for each anti-collision algorithm is shown in Fig. 4. In the proposed BHS algorithm, the number of total reader command for the tag recognition is decreased to 48~50% for 100 tags, 58 ~60% for 1000 tags against that of BSA algorithm. Furthermore, the number of total reader command is reduced remarkably in the proposed algorithm according as the number of tag increases.

In order to calculate the total tag recognition time, the pulse duration time(T0) sets to 14.25μs that is regulated on EPC protocol. Initiating signal from the reader to the tag consists of five phases as shown in Fig. 1. The first phase is the transaction gap (1.25T0) and the second phase is CW RF signal of 64μs. The third phase is the data modulation window, the fourth phase is the tag setup window (8T0) and the last is the tag response part (64T0 + 2.5T0). In formatting the reader command, data modulation window is variable because the length of [VALUE] is variable. To estimate tag recognition time accurately, the length (VLP) of [VALUE] field for each PingID and the length (VLS) of [VALUE] field for each ScrollID are calculated and shown in above Tables. Both VLP and VLS have the maximum length of 14-bit for 1000 tags recognition. The excepted part of [VALUE] in data modulation window transmitting reader command consists of 51-bit. In the proposed BHS algorithm, the data modulation window is allocated with 51-bit + VLP for PingID command and with 51-bit + VLS for ScrollID command or Quiet command.

The number of PingID command transmitted to the tags is equal to the number of bin-modulation. And the number of ScrollID and 'Quiet' command are equal to the number of identified tag. If the time required for processing one PingID command, one ScrollID command and one Quiet command between the reader and the tag are TP1, TS1 and TQ1, respectively. So, the total time for the tag recognition can be written as follows.

$$\begin{aligned} T_{P1} &= 1.25T_0 + 64\mu s + 51T_0 + [VLP]T_0 + 8T_0 + 64T_0 + 2.5T_0 \\ T_{S1} &= 1.25T_0 + 64\mu s + 51T_0 + [VLS]T_0 + 8T_0 + [tagbit]T_0 + 2.5T_0 \\ T_{Q1} &= 1.25T_0 + 64\mu s + 51T_0 + [VLS]T_0 + 8T_0 + 2.5T_0 \\ T_{Total} &= N_P T_{P1} + N_S T_{S1} + N_Q T_{Q1} \end{aligned} \quad (2)$$

The calibration time of RFID system and the CRC check time are excluded from Total because they have a little effect on the total recognition time.

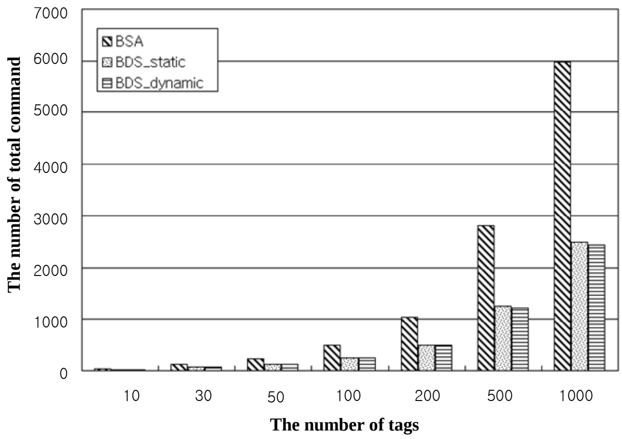

Fig. 4. The number of total command for tags (96-bit)

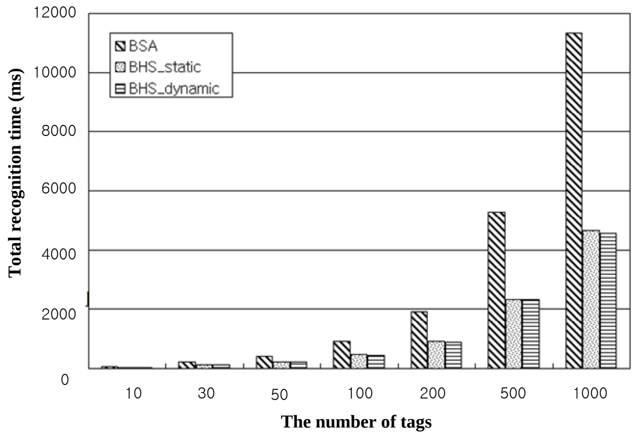

Fig. 5. Total recognition time for tags (96-bit)

Figure 5 shows the total recognition time for each algorithm, and shows the decrements of BHS against BSA. For 64-bit ID tag, in the case of BHS-static algorithm, the reduction ratio of the total recognition time is 53.3% for 100 tags, 62.5% for 1000 tags against that of BSA algorithm. And in the case of BHS-dynamic algorithm, the ratio is 54.5% for 100 tags, 63.4% for 1000 tags.

For 96-bit ID tag, in the case of BHS-static algorithm, the reduction ratio of the total recognition time is 49.1% for

100 tags, 59.1% for 1000 tags against that of BSA algorithm. And in the case of BHS-dynamic algorithm, the ratio is 51% for 100 tags, 59.8% for 1000 tags. As the results, the total recognition time for the proposed BHS algorithm is shorter than that of BSA algorithm according to increasing the number of tag. And the reduction of total recognition time for the proposed BHS algorithm is noticeable according to increasing the number of tag ID bit.

# 5 Conclusion

In this paper, as anti-collision algorithm for RFID system, the BHS algorithm is proposed and analyzed. Also the performance comparison between the proposed BHS algorithm and the BSA algorithm based on EPC Class1 protocol is carried out. The proposed algorithm processes ahead the bins that the collision does not occur, and then the collided bins stored in the stack are processed dynamically by searching the collided-bit position of collided bin data. By comparing the tag recognition time and the number of reader command, the performance improvement of the proposed algorithm is proven. For 96-bit ID tag, the reduction ratio of the number of reader command and tag recognition time of BHS algorithm for 1000 tags are 59.2% and 59.8%, respectively. Consequently, the tag identification performance of the proposed algorithm is much better than that of BSA algorithm according to increasing the number of tag.

Acknowledgment

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment, IITA-2005-C1090-0502-0009)

# 6 References

[1] K. Finkenzeller, RFID Handbook, Wiley & Sons, 2003.

[2] L. Bolotnyy, G. Robins, “Multi-tag radio frequency identification systems, Available : http://www.cs.virginia.edu/robins.

[3] [3] H. Vogt, "Multiple object identification with passive RFID tags," IEEE International Conference on Systems, Man and Cybernetics, Oct. 2002. vol. 3, pp. 6-9.

[4] F. Zhou, D. Jin, C. Huang and M. Hao, "Optimize the power consumption of passive electronic tags for anti-collision schemes," ASIC Proceedings of 5th International Conference, Oct. 2003, vol. 2, pp. 1213-1217.

[5] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," IEEE International Symposium on Information Theory, Aug. 1998, pp. 107.

[6] B. Zhen, M. Kobayashi, M. Shimizu, "Framed ALOHA for multiple RFID objects identification," The Institute of Electronics Information and Communication Engineers, 2005, pp. 991-999..

[7] H. S. Choi, J. R. Cha, H. H. Kim, "Fast wireless anti-collision algorithm in ubiquitous ID system," Vehicular Technology Conference, VTC2004-Fall, 2004 IEEE 60th vol. 6, Sept. 2004, pp. 4589 – 4592.

[8] Technical Report, “860-930MHz Class 1 radio frequency identification tag radio frequency & logical communication interface specification candidate recommendation, version1.0.1,” Auto-ID Center Massachusetts of Technology, Nov. 2002..

[9] EPC™ generation 1 tag data standards version 1.1 Rev.1.27, EPC global Inc. Standard Specification, May 2005.