Evaluating and Optimizing Power Consumption of Anti-Collision Protocols for Applications in RFID Systems†

Feng Zhou[1], Chunhong Chen[2], Dawei Jin[1], Chenling Huang[1], Hao Min[1]
[1] ASIC & Systems State's Key Lab, Fudan University, Shanghai, China
[2] Department of Electrical and Computer Engineering, University of Windsor, Ontario, Canada N9B 3P4
fengzhou@fudan.edu.cn
cchen@uwindsor.ca

ABSTRACT
For low-cost RFID systems, the design of passive tags is a key issue in anti-collision protocols where lower power consumption allows a longer working distance between tags and the reader. In this paper, we look at anti-collision protocols in tags' processing for their power optimization. We propose a new criterion, which takes into account both energy consumption and time complexity, to evaluate anti-collision protocols. An improved protocol is also presented for power savings.

Categories and Subject Descriptors
B.7.1 **[Integrated Circuits]**: Types and Design Styles – *algorithms implemented in hardware.*

General Terms
Algorithms, Design, Performance.

Keywords
Radio-frequency identification, low power, anti-collision protocols.

1. INTRODUCTION
Radio-Frequency Identification (RFID) systems consist of radio frequency (RF) *tags* and networked electromagnetic *readers* [1, 3, 4]. The reader tries to obtain within its read range the unique ID number of each tag. The need for implementing low-cost automatic identification systems requires passive electronic tags that get their energy from the electromagnetic waves emitted from the reader. For frequencies above 900MHz, the working distance of 1 meter or above makes tags fall into the far field region of the reader's antenna. The energy received by the tag is generally less than 100 μW. Such an energy supply requires the function of the tag to be as simple as possible so that its power consumption can be reduced. The data transmission from tag to reader is done by scattering the wave energy back to the reader. Depending upon the data being sent back to the reader ('0' or '1'), the tag chooses to scatter or not to scatter the wave energy, or alternatively modulates the carrier with two different frequencies while scattering. A *collision* problem may arise when more than one tags enter the working zone of a reader. In response to the reader's inquiry, each tag sends a series of data to the reader. The reader will receive a mixture of scattered waves and cannot tell individual IDs of the tags. Therefore, an *anti-collision* mechanism is needed in such an RFID system and has to be implemented in the digital integrated circuits within the tags.

While prior research focussed on the anti-collision problem in wireless communication systems, passive RFID systems deserve special attention. The limited power supply for tags makes it infeasible to establish the communication among the tags to coordinate their scattering behaviors. Thus, the anti-collision scheme has to be in the mode of "*reader-inquiring*" and "*tag-answering*". The process of inquiring and answering is repeated for a number of cycles until the job of accessing all the tags is done. A passive tag is power-limited, rather than energy-limited. One can assume that the reader always keeps the energy supply during the inquiring period so that the problem of "battery life" does not exist. The key issue is the tag's power consumption, which restricts the maximum working distance from the tags to the reader [1]. Since the anti-collision circuit constitutes the main part of the base-band processing unit on a tag, low power implementation of anti-collision protocols promises to be critical in the design.

In a passive RFID system, no anti-collision protocols can be evaluated accurately in terms of power consumption without a detailed implementation. This is because the restriction on power consumption has come to such an extent that both the protocol and circuit have to be optimized aggressively. In this work, we present a criterion for power evaluation of anti-collision protocols and explore their power optimization at the protocol- and circuit-level. An improved protocol is also proposed for power savings. The remainder of the paper is organized as follows. In Section 2, we describe a cost function that takes into account both power consumption and time complexity in order to evaluate the anti-collision schemes. In Section 3, two practical schemes are discussed and an improved scheme is presented. The three schemes are compared in Section 4, followed by our conclusions given in Section 5.

† This work was supported in part by National Natural Science Foundation of China (# 90307008).

ISLPED'04, August 9–11, 2004, Newport Beach, California, USA.

2. COST FUNCTION FOR ANTI-COLLISION SCHEMES

In passive RFID systems, low power implementation allows a longer working distance between the reader and the tag. The tag works by extracting the electromagnetic power from the reader, which is given by [1]:

$$S = \frac{\lambda^2 \cdot P \cdot G_s \cdot G_r}{(4\pi R)^2} \quad (1)$$

where λ is the wavelength of the emitted electro-magnetic wave, P the power injected into the reader's antenna, R the distance between the reader and the tag, G_s the gain of the reader's antenna, and G_r the gain of the tag's antenna. It is the maximum *instant power consumption* of the tag that determines the tag's longest working distance from the reader. Thus, for passive tags, the objective is to minimize the maximum instant power of the circuit. In a practical tag, the requirement on the maximum instant power can be relaxed. This is shown in Figure 1 where the capacitor C in the rectification circuit provides energy compensation when the received power is less than the instant power consumption of the inner circuits. When the compensation occurs, the capacitor discharges with its voltage decreased. Let T be the system's clock period. For any duration of $(k_1 - k_2)*T$, where k_1 and k_2 denote the numbers of clock cycles experienced by the time the duration starts and ends, respectively, we have

$$\frac{1}{2}CV_{dd}^2 - \frac{1}{2}C(V_{dd} - \Delta V)^2 = E_{k1,k2} - P_{in}(k_1 - k_2)T \quad (2)$$

where ΔV is the voltage change on the capacitor, P_{in} the input power, and $E_{k1,k2}$ the total energy consumed by the circuit over $[k_1, k_2]$.

Assuming $\Delta V << V_{dd}$, (2) can be approximated as:

$$CV_{dd}\Delta V = \frac{1}{2}V_{dd}^2 C_{load}\sum_{i=t1}^{t2} A_i - P_{in}(k_2 - k_1)T \quad (3)$$

where A_i is the circuit's total number of *0-to-1* transitions occurring at clock cycle i, and C_{load} is the average load capacitance of the circuit. If we use V_{drop} to denote the maximum voltage drop that is acceptable under a given performance specification, we require:

$$CV_{dd}V_{drop} \geq \frac{1}{2}V_{dd}^2 C_{load}\sum_{i=t1}^{t2} A_i - P_{in}(k_2 - k_1)T \quad (4)$$

or

$$P_{in} \geq \frac{V_{dd}^2 \cdot C_{load}}{(k_2 - k_1)T}\left(\sum_{i=t1}^{t2} A_i - \frac{C}{C_{load}} \cdot \frac{V_{drop}}{V_{dd}}\right) \quad (5)$$

We define:

$$P_{in,\min} = \max\{P(k_2, k_1)\} \quad (6)$$

where k_1 and k_2 ($k_1 < k_2$) may refer to any duration within which the anti-collision scheme is taking place, and $P(k_2, k_1)$ is given by the right side of (5):

$$P(k_2, k_1) = \frac{V_{dd}^2 \cdot C_{load}}{(k_2 - k_1)T}\left(\sum_{i=t1}^{t2} A_i - \frac{C}{C_{load}} \cdot \frac{V_{drop}}{V_{dd}}\right) \quad (7)$$

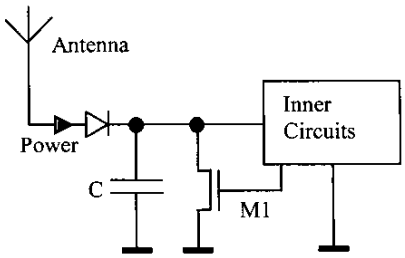

Figure 1: Rectification circuit in a practical tag.

Before going further, we explain the function of *M1* in Figure 1 as follows. When the consumed power is less than the input power, *M1* bypasses the surplus current, preventing the voltage on the capacitor from increasing too much. In other words, if the voltage across the capacitor reaches V_{dd}, the surplus power (the difference between the input power and the consumed power) will be bypassed or "*abandoned*", rather than accumulated on the capacitor.

To introduce our cost function, we first describe the following two theorems (the proofs are omitted due to space limitation).

Theorem 1: If the $P_{in,min}$ is found within the duration $[k_1, k_2]$, then the total energy needed by the circuit for any duration $[k_0, k_1]$ (where $0 \leq k_0 < k_1$) is less than $(k_1 - k_0)T \cdot P_{in,min}$.

Theorem 2: If the $P_{in,min}$ is found within the duration $[k_1, k_2]$, then the energy needed by the circuit for any duration $[k_1, k_3]$ (where $k_1 < k_3 < k_2$) is more than $(k_3 - k_1)T \cdot P_{in,min}$.

When deriving equation (2), we assume that the voltage on the capacitor begins to drop from V_{dd} at clock k_1. Theorem 1 ensures that if the input power equals to $P_{in,min}$, then k_1 actually lies at the beginning of the voltage-dropping process. Put another way, there is no voltage drop at clock k_1. On the other hand, Theorem 2 implies that no input power would be "abandoned" within $[k_1, k_2]$, since the lost charges on the capacitor will never be completely restored within $[k_1, k_2]$ (the needed power is always larger than the input power).

Based on the definition of $P_{in,min}$ and the two theorems, one can say that $P_{in,min}$ is the maximum gap between the power required by the circuit and the power that can be compensated by the charges on the capacitor. Therefore, $P_{in,min}$ is the lower bound for the input power of tags, and is hence used as our criterion for evaluating anti-collision schemes. Theorem 2 rules out a possibility that some abandoned input power might be mistakenly included in calculation of $P_{in,min}$.

In general, the time complexity of an anti-collision scheme is not a big issue unless a large number of tags are to be read within limited accessible time. To enable a fair comparison, however, we are using a same period of time, T_{INQ}, for all schemes, and define $T_{INQ} = T \cdot Cyc$, where Cyc is the total number of clock cycles needed to complete a scheme. Thus, (7) can be rewritten as:

$$P(k_2,k_1)=\frac{Cyc\cdot V_{dd}^2\cdot C_{load}}{(k_2-k_1)T_{INQ}}\left(\sum_{i=k_1}^{k_2}A_i-\frac{C}{C_{load}}\cdot\frac{V_{drop}}{V_{dd}}\right) \quad (8)$$
$$\propto\frac{Cyc}{(k_2-k_1)}\left(\sum_{i=k_1}^{k_2}A_i-\frac{C}{C_{load}}\cdot\frac{V_{drop}}{V_{dd}}\right)$$

The above equation is used as a cost function in evaluating power consumption of anti-collision schemes to be discussed in the following sections.

3. EVALUATION OF PRACTICAL ANTI-COLLISION SCHEMES

In this section, we evaluate the power consumption with our cost function for two existing anti-collision schemes and one proposed scheme. Throughout the discussions, we use n to denote the length of IDs configured inside the tags and k the number of tags simultaneously appearing within a working zone of the reader.

3.1 Binary-Tree Scheme

Binary-tree scheme [3] is such a protocol that requires tags to remember the previous inquiring results, thus reducing the average inquiring time. In this scheme, a tag has to completely finish an inquiring processing before responding to the next reader. If more than one reader work near the tag, the task of coordinating the readers becomes complicated. Also, memory protocols are sensitive to the power supply.

In this protocol, a tag is "*killed*" once it has been completely identified. There is a pointer within each tag. Each time a tag is reset, the pointer points to the highest bit of the tag's ID and, with the ongoing of inquiring, it moves toward a lower bit. During the inquiring, the reader sends one inquiring bit at one time. The tags whose pointed bit is identical to the inquiring bit send back their next bits to the reader, while the rest convert to a state of "*quiet*" and will not answer the remaining inquiries in this round of inquiring until one tag has been killed and all the remaining tags have been reset. If the reader detects a non-collision answer, it is used as a next inquiring bit. Otherwise, the reader uses '0' as the next inquiring bit. Thus, for every cycle of inquiry, one and only one tag will answer all the n times of inquiring, with its pointer eventually getting to the lowest bit. Then, the identified tag is killed, the tags with a state of "quiet" are reset, and a new round of inquiring begins from the highest bit. After k times of inquiring, the IDs in the k tags will be identified. An example of inquiring process on 3 tags, whose IDs are 001, 011 and 100, is shown in Table 1 where the asterisk (*) denotes a collision detected by the reader. This process is equivalent to searching on a binary tree k times, from the root to the k leaves, as shown in Figure 2.

To identify one tag, the numbers of clock cycles required for both inquiring and answering are $n-1$ each. The $2k(n-1)$ clock cycles are needed for k tags. In addition, it takes 3 clock cycles for the reader to know there are no more tags whose IDs start with '0' alive (refer to the 5th bit sent by the reader in Table 1). In other words, the number of clock cycles required is:

$$Cyc_{binary}=2k(n-1)+3 \quad (9)$$

Table 1: The inquiry process of Binary-tree protocol

Reader sends	Tags answer	Identified tag (remaining tags reset)
0	*	
0	1	001
0	1	
1	1	011
0		
1	0	
0	0	100

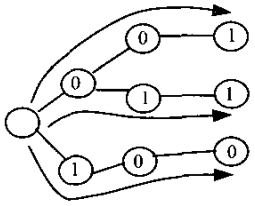

Figure 2: Search on a binary tree.

Figure 3 shows the schematic implementing the protocol (the tag end). The inner circuits of the macro modules in Figure 3 are shown in Figures 4 and 5. In the register of Figure 5, only one bit can be '1' at any time, making just one of the bits in ID appear at the port "DATA" (see Figure 3). This bit is to be compared with the received inquiring bit. Figure 6 shows the state diagram of the tag's state machine. If a tag is not at the "quiet" or "killed" state, it shifts between the "receiving" and "sending" states. The longest string of continuous "receiving" and "sending" covers $2(n-1)$ clock cycles between k_1 and k_2 before the tag is read out.

While different tags consume different amount of energy during the whole inquiring process (the sooner a tag is identified and killed, the less energy it would consume), they have the same cost since the operations for them to be identified remain the same. To calculate the cost of the protocol, we first look at the transitions at the node "DATA" of module "ID-BIT" in Figure 3. On average, the probability of the occurrence of l consecutive '1's in an ID is $1/2^l$, and the average length of consecutive '1's is given by $\sum_{l=1}^{n} l\cdot\frac{1}{2^l}=2-\frac{n+2}{2^n}$, which is close to 2 for large n (for a practical system, n could be 32, 96, or even bigger). Thus, the average number of transitions at node "ID_OUT" (the right-side port of module "ID-BIT") is $n/2$. The operation of "killed" causes 10 transitions. In the inquiring process, there are $(n-1)$ times of receiving and $(n-1)$ times of sending. Each operation of receiving and sending introduces 7.5 and 12.5 transitions, respectively. Therefore, we have (refer to equations (8) and (9)):

$$\left.\begin{aligned} &k_2-k_1=[2(n-1)+1]=2n-1,\\ &\textstyle\sum A_i=7.5(n-1)+12.5(n-1)+10,\\ &and\\ &P_{binary}=[2k(n-1)+3]\left(10-\frac{1}{2n-1}\cdot\frac{C}{C_{load}}\cdot\frac{V_{drop}}{V_{dd}}\right)\end{aligned}\right\} \quad (10)$$

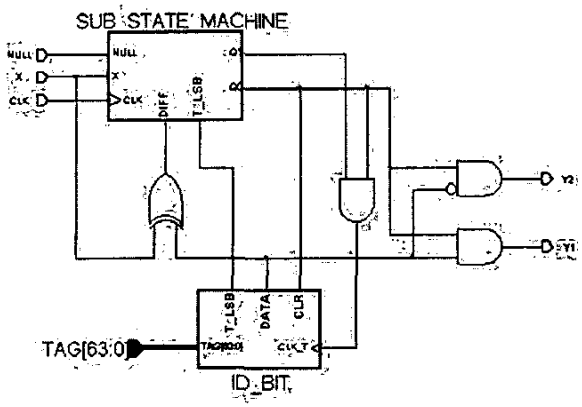

Figure 3: The schematic of binary-tree protocol.

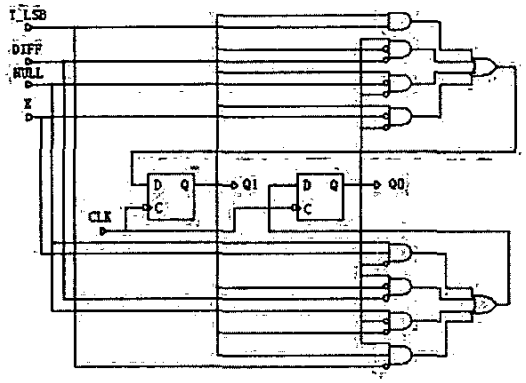

Figure 4: The schematic of the state machine in Figure 3.

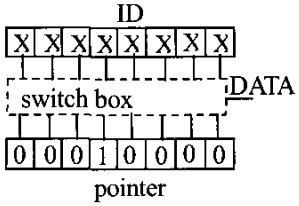

Figure 5: The register for ID_BIT module in Figure 3.

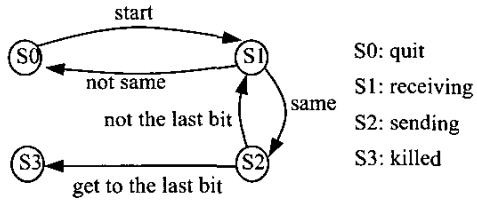

Figure 6: The state diagram for binary-tree protocol.

3.2 Query-Tree Scheme

Query-tree scheme [4] is a *memoryless* protocol in which the tags do not have to remember their inquiring history. Instead of sending only one bit for each inquiring, the reader sends a *prefix* which may have the length of 1 to $n - 1$ bits. The tags will send back the remaining bits of their IDs if the prefix matches the first bits (counting from the highest bit) of their IDs. From the tags' response, the reader can tell the bit at which the collision is occurring. After that, the reader overrides the non-collision bits and extends the prefix directly to the collision bit. Receiving data of no collision implies that an ID has been read. After recording the ID, the reader revises the prefix by either changing the last bit or changing the next-to-last bit with the last bit "abandoned", and continues the inquiring process. An example is shown in Table 2 where there are 4 tags with the IDs of 0001, 0011, 1000 and 1100. Figure 7 shows the state diagram for the query-tree protocol (the schematic implementations can be done in a similar way as above and are hence omitted).

Table 2: The inquiring process of query-tree protocol

Reader sends	Tags answer	Identified tag
start	****	
0	0*1	
000	1	0001
001	1	0011
1	*00	
10	00	1000
11	00	1100

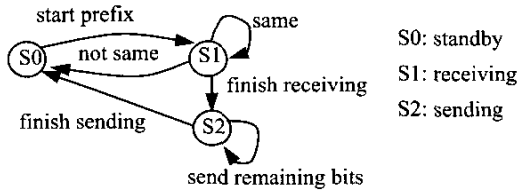

Figure 7: The state diagram for query-tree protocol.

Assuming that the IDs are evenly distributed, the above inquiring process can be represented by a binary tree with depth $m = \lceil \log_2 k \rceil$, where k is the number of tags. Once the length of the prefix increases to m, no collisions could occur. This suggests that the remaining parts of the ID be sent back without any further inquiry. This is equivalent to a *depth-first traversal* over a full binary tree where a node represents a prefix.

In order to inform the tags of the start and end of the received data, the data are enclosed with a pair of "*NULL*". An extra operation, "*wait*", is needed for the reader to deal with the sent-back data. The nodes on the query tree can be divided into two classes: *Type-I* which corresponds to those lying on the levels less than m, and *Type-II* which corresponds to those on level m of the tree. During the depth-first traversal, each arrival at a node of Type-I requires the tag to experience $\lceil n$ (either receiving or sending data) +2(NULL) + 1] clock cycles. The arrival at a node of Type-II implies that a tag is identified, and two extra clock

cycles are needed (in addition to $n + 2 + 1$ clock cycles as required for Type-I node) for the reader to record the read-out ID. Thus, the total number of clock cycles required is given by (note $m = \lceil \log_2 k \rceil$):

$$Cyc_{query} = (n+3)(2^0 + 2^1 + \cdots + 2^{m-1}) + (n+5)\cdot 2^m$$
$$= k(2n+8) - (n-3) \quad (11)$$

We see that with this protocol, different tags have different costs. In particular, the first read-out tag generates the maximum cost. To read the first tag, the protocol has to experience m Type-I nodes and 1 Type-II node, and the number of clock cycles required is: $k_{1st} = (2 + n + 1)m + (2 + n + 1 + 2)$. A total of $(2 + m - 1)$ more clock cycles is needed to identify the rest of tags. In other words, $k_2 - k_1 = k_{1st} + 2 + m - 1 = (n + 4)(m + 1) + 2$. In our implementations, each operation of "receiving" and "sending" generates 7 and 8 transitions, respectively. The two "NULL" operations introduce 14 and 9 transitions. The operation of "wait" causes 7 transitions. No transitions occur during the two clock cycles for recording the identified tag. Therefore, we have

$$\left.\begin{aligned} \sum A_i &= \sum_{i=0}^{m}[14+9+7i+8(n-i)+7]+[14+9+7(m-1)] \\ &= 8(n+4)(m+1)+5m+14-\tfrac{1}{2}(m^2+m), \\ \text{and} \\ P_{query} &= [k(2n+8)-(n-3)]\cdot[8-\frac{m^2-9m+4}{2(n+4)(m+1)+4} \\ &\quad -\frac{1}{(n+4)(m+1)+2}\cdot\frac{C}{C_{load}}\cdot\frac{V_{drop}}{V_{dd}}] \end{aligned}\right\} \quad (12)$$

3.3 An Improved Query-Tree Scheme

In this subsection, we propose and analyze an improved version of the above-mentioned query-tree scheme. It differs from the query-tree scheme in that when the tags are sending back their remaining parts of IDs, the reader, once detecting a collision bit, gives the tags a signal to stop sending. This will lead to the reduced number of "sending" operations in the worst case. Table 3 illustrates the inquiring process of the protocol on the same example from Table 2. Again, the schematic implementation can be obtained accordingly. Also, the state diagram is similar to Figure 7 except that the edge labelled "finish sending" in Figure 7 should be replaced with the one labelled with "stop sending".

In this protocol, the reader needs one clock cycle to detect the collision and another clock cycle to send the "*stop*" signal. This means that the tags will receive the signal following two bits after the collision bit. With the assumption of evenly distributed IDs, the tags actually send only 3 bits back unless a node of Type-II is reached. This is because a collision occurs as frequently as the tags send back their first bit. Therefore, the number of clock cycles needed by a Type-I node is: i (receiving) + 2 (NULL) + 3 (sending back) = i + 5, where i is the node's level number in the tree. The nodes of Type-II take the same number of clock cycles as those in the query- tree scheme. As a result, the total number of clock cycles turns out to be:

Table 3: The process of improved query-tree protocol

Reader sends	Tags answer	Identified tag
start	*	
0	0*	
000	1	0001
001	1	0011
1	*	
10	00	1000
11	00	1100

$$Cyc_{improved} = \sum_{i=0}^{m-1}(i+5)\cdot 2^i + (n+5)\cdot 2^m$$
$$= k(n+m+8) - 3 \quad (13)$$

where $m = \lceil \log_2 k \rceil$. By similar analysis as in previous section, one can obtain: $k_2 - k_1 = n + \frac{1}{2}(m^2 + 11m + 12)$. In the case of our implementation, the two "NULL" operations generate 19 and 15 transitions, while the "receiving" and "sending" operations introduce 7 and 8 transitions, respectively. No transitions are needed for the reader to record the identified tags. This leads to:

$$\left.\begin{aligned} \sum A_i &= \sum_{i=0}^{m-1}(19+15+7i+3\times 8) \\ &\quad +[15+19+7m+8(n-m)] \\ &\quad +[15+19+7(m-1)] \\ &= 8n+\tfrac{1}{2}(7m^2+121m+122), \\ \text{and} \\ P_{improved} &= [k(n+m+8)-3]\cdot[8-\frac{m^2-33m-26}{2n+m^2+11m+12} \\ &\quad -\frac{2}{2n+m^2+11m+12}\cdot\frac{C}{C_{load}}\cdot\frac{V_{drop}}{V_{dd}}] \end{aligned}\right\} \quad (14)$$

4. COMPARISON OF THE ANTI-COLLISION SCHEMES

In this section, we compare the cost functions for all three anti-collision schemes discussed above, as given equations (10), (12) and (14). They depend on n, m (or k) and many other parameters among which is a constant ratio of $CV_{drop}/C_{load}V_{dd}$. We use R to denote this ratio in the following discussions.

4.1 Comparison between Binary-Tree and Query-Tree Protocols

The comparison between the binary-tree protocol and the query-tree protocol is shown in Figure 8 where R is the vertical axis and m is the horizontal axis. The curves in this figure show the situations when P_{binary} equals P_{query} for different values of n. Above the curves is the zone where the binary-tree protocol is better than the query-tree protocol. Below the curves is the zone where the binary-tree protocol is worse than the query-tree protocol. It can be seen from these curves that:

- As n increases, so does the probability of the query-tree protocol outperforming the binary-tree protocol;
- With a smaller value of R, the query-tree protocol tends to be better than the binary-tree protocol.

4.2 Comparison between Binary-Tree Protocol and Improved Query-Tree Protocol

A family of similar curves is shown in Figure 9 to demonstrate the difference between the binary-tree protocol and improved query-tree protocol. While similar relationship can be obtained from the curves with respect to n and R, it is observed that the zone where the improved query-tree protocol outperforms the binary-tree protocol becomes larger compared to the conventional query-tree protocol.

4.3 Comparison between Query-Tree Protocol and Improved Query-Tree Protocol

If we plot the equi-cost curves for the query-tree protocol and its improved version, these curves lie in a quadrant with negative values of R, as shown in Figure 10. This verifies that the improved query-tree scheme is always better than the conventional query-tree scheme.

5. CONCLUSIONS

We have looked at power evaluation and optimization of anti-collision schemes in RFID systems by combining both protocol-level and circuit-level design considerations. We have introduced a cost criterion taking into account both energy and time complexity of the protocols, and presented an improved anti-collision scheme in comparison with the two existing schemes. It has also been shown that with proper protocol improvement and circuit design, memoryless protocols can have better power performance than memory ones.

6. REFERENCES

[1] Finkenzeller, K. and Waddington, R. *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. John Wiley & Sons, January 2000.

[2] Sama, A., Theeuwen, J. F. M. and Balakrishnan, M. Speeding up Power Estimation of Embedded Software. In *Proceedings of International Symposium on Low Power Electronics and Design*, Rapallo, Italy, July 2000, 191-196.

[3] MIT Auto-ID Center. *Draft protocol specification for a 900 MHz Class 0 Radio Frequency Identification Tag*. http://auto-id.mit.edu, February 2003.

[4] Law, C., Lee, K., and Siu K. Y. Efficient Memoryless Protocol for Tag Identification. In *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Boston, Massachusetts, August 2000, 75-84.

[5] Zhou, F., Jing, D., Huang, C., and Min H. Optimizing the Power Consumption of Passive Electronic Tags for Anti-collision Schemes. In *Proceedings of the 5th ASICON*, Beijing, China, October 2003, 1213-1217.

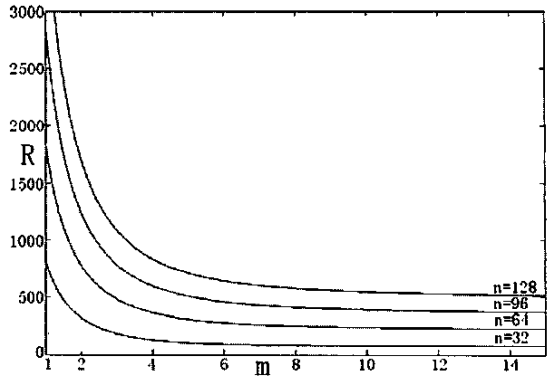

Figure 8: Equi-cost curves for binary-tree protocol and query-tree protocol.

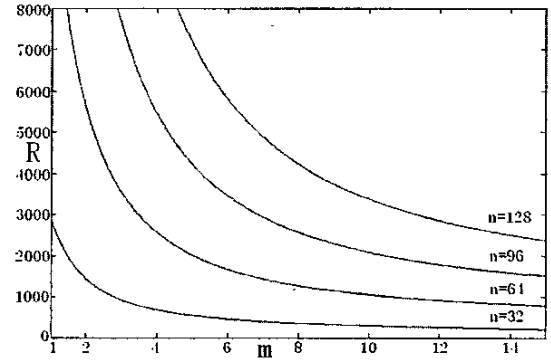

Figure 9: Equi-cost curves for binary-tree protocol and improved query-tree protocol.

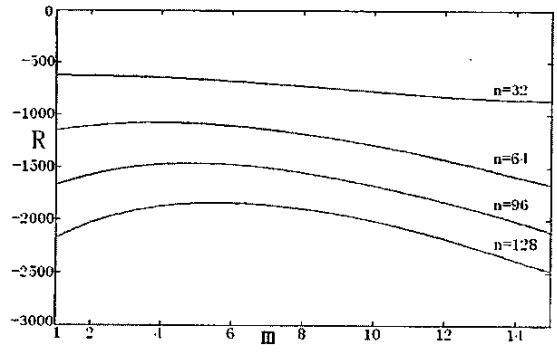

Figure 10: Equi-cost curves for query-tree protocol and its improved version.