Comparison of Transmission Control Protocols Based on EPC C1G2 Standard

Fariha Baloch and Ravi Pendse

Department of Electrical Engineering and Computer Science, Wichita State University, Wichita, KS, USA
Email: fxbaloch@wichita.edu

Abstract: a transmission control scheme plays an important role in determining a protocol's performance. Protocol performance can be measured based on a variety of factors. In this paper we compared four protocols that are compatible with EPC C1G2 standard using four units of performance, namely System Efficiency, Time System Efficiency, Total Number of Slots and Total Time.The results show not only the strengths and weaknesses of the protocols measured, but also highlight the importance of evaluating all relevant performance factors when comparing protocols as opposed to weighing performance using one or two units.

Keywords: EPC C1G2; Transmission Control Scheme; Performance.

I.   Introduction

  Radio Frequency Identification (RFID) technology is gaining popularity among businesses and in the scientific community. While it promises businesses with efficient asset tracking and cost reducing options, it presents many unique challenges. One of the challenges in RFID networks, as with any other wireless protocols, is packet collisions in a multi-node environment. In RFID's multi-tag environment, if several tags choose the same slot to communicate with the reader, collisions can occur. Another type of slot that negatively affects the performance of a protocol is the empty slot. While these issues are inherent to any wireless protocol, there are transmission control schemes that try to mitigate the effect of the wasted slots on the performance of the protocol.

Several medium access protocols have been proposed for RFID [2, 3, 4, 5, 6, and 7]. Each proposed protocol compares its performance with either ideal results of the FSA protocol, SchouteProtocol [8], Vogt Protocol [9] or with the Q-algorithm of EPC C1G2. In this paper we compared protocols presented in [2], [3] and [4] with the Q-Algorithm as presented in [1]. The protocols presented in [2, 3 and 4] are compatible with the standard EPC C1G2 therefore these will be evaluated based on the C1G2's defined parameters. The main unit used by protocol creators in the publications [2, 3 & 4]istotal number of slots required to scan a population of tags, in combination with units such as total collided slots in [4]and redundant time in [2]. These units or combination of does not show weaknesses of protocols when units such as total time, for example, is taken into account. In order to completely evaluate each protocol's performance the units of measurements used for this study are System Efficiency (SE), Time System Efficiency (TSE) [6], Total Number of Slots and Total Time.The combination of four units give a better understanding of each protocol compared of one dimensional view.

In the next section the method for accessing tags used in EPC C1G2 explained. This method will be the same for all the transmission control schemes used in this study. For example, the time consumed by a successful (or collided or empty) slot will remain the same for the Q-algorithm of EPC C1G2 [1], Fast-Q

algorithm of [4], Dual Dias Q-Algorithm (DBQ) [2] and for RFID Explicit Tag Estimation Scheme (RETES) [3]. The frame size will be calculated using the power of 2 technique and the Query (Query Rep or Query Adjust) packet will lead every slot as described in [1]. Section III contains the details of the transmission control schemes used in this paper. Section IV will explain the units of measurements used to compare the protocols. Section V contains the simulation results and analysis and Section VI ends the paper with conclusions.

II. EPC C1G2

In the EPC C1G2 standard [1], a simple medium access protocol called the "Q-algorithm" is described to identify tags in a multi-tag environment. The objective of this algorithm is to change the protocol parameters based on the conditions of the medium. The standard is based on the legacy slotted-ALOHA protocol where tags send their data in a specific slot. According to [1], a reader starts an inventory round with a Query packet that contains a parameter "Q". "Q" is an integer and its value is between 0 and 15. Each tag has the capability to extract this parameter and load a value between 0 and (2Q -1) into its 15-bit slot counter. The Query packet not only sends out the initial information needed to start the interrogation round, but also energizes the passive tags.The reader then starts a continuous wave that the tag with 0000h in its slot counter uses to backscatter a 16-bit randomly generated number called (RN16). If RN16 is received by the reader successfully, the reader acknowledges the tag by sending an ACK packet. Upon reception of the ACK, the tag backscatters its unique Tag-ID (EPC) to the reader. The reader then either sends a Query Repeat (QueryRep) or Query Adjust (QueryAdj) to all the tags depending upon the new "Q" value computed by the "Q-algorithm. The Query command is 22-bits long and is sent at the start of the query round. QueryRep is 4-bits long and is sent to the tags if the frame size needs to remain same for the current slot. QueryAdj is 9 bits long and is transmitted to the tags when the frame size needs to be changed. EPC C1G2 also defines guard times which are used to avoid unnecessary interference on the channel.Figure 1shows the time of a successful, empty and collided slot using EPC C1G2 defined parameters.
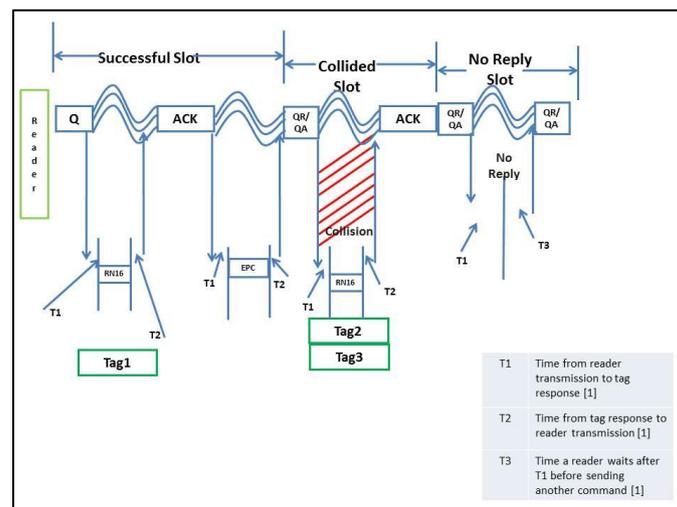

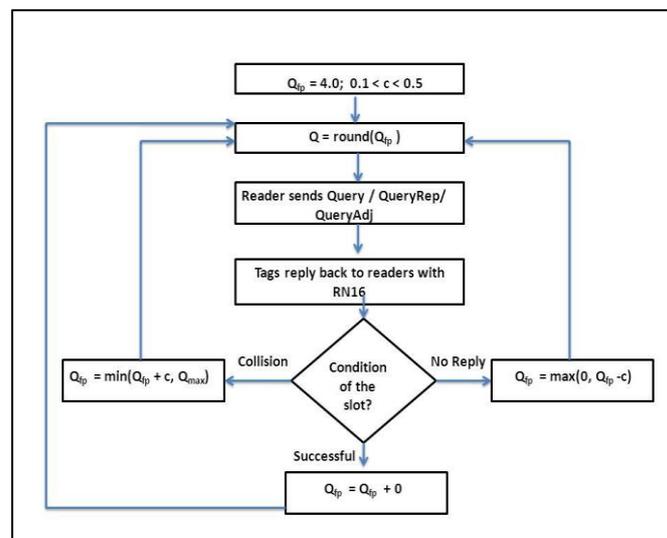**Figure 1.** Example of successful, collided and empty slot using EPC C1G2 protocol

Figure 2. Q-algorithm

III. OVERVIEW OF PROTOCOLS

The transmission control scheme dictates which query command needs to be broadcasted to the tags during a scan round. The research presented in [1, 4, 5 and 10] is based on the concept of changing the frame size after each slot, similar to C1G2. In [2] and [3], a slightly different version of C1G2 is presented. Instead of changing the frame size on a per slot basis, the decision is made on frame-by-frame basis. Based on these papers it can be inferred that the frame size can either be changed after every slot or after every frame. Therefore in this paper the comparison study will be performed based on both of these versions of EPC C1G2. From following four protocols, Q and Fast-Q algorithm use slot by slot assessment method to change query command during scan round whereas RETES and DBQ use frame by frame method to do the same.

A. Q-Algorithm

The Q-Algorithm is the transmission control protocol used in the EPC C1G2 standard [1]. The flow diagram of the Q-algorithm is shown in Figure 2. The diagram shows a variable Q_{fp} which denotes the floating-point representation of Q. The value of Q is determined based on the integer nearest to Q_{fp}. The decision process used by this algorithm is based upon the three conditions a slot can have at a given time. The three conditions are "successful slot", "no reply (or empty slot)" and "collided slot". If a reader successfully receives information from a tag, the value of Q remains unchanged. If reader receives multiple RN16s from various tags, a collision will occur and the reader will increment Q_{fp} by a value of parameter "c". "c" is a real number whose values are between 0.1 and 0.5. The standard dictates that Q_{fp} will not exceed 15. The Q is determined as Q=round (Q_{fp}). If a reader receives no reply in a slot, it will decrease Q_{fp} by a value "c" and round it to the nearest integer to assign the value to Q. If the value of Q becomes negative, Q_{fp} is set to zero. After every slot is queried, the value of Q is updated by the reader based on the outcome of the Q-algorithm. The EPC C1G2 standard has suggested to start Q at Q0 = 4, and then either decrease or increase Q as the process of reading tags continues. By decreasing Q, the number of slots is being reduced and by increasing Q, number of slots is being increased. The Q-algorithm does not require

any complicated computations and is fairly simple algorithm. However, since it is very simple, it is not a perfect prediction of slots in a frame.

B. Fast-Q Algorithm

The Fast-Q algorithm introduced in [4] points out that the Q algorithm of EPC C1G2 does not change the frame size fast enough as it only relies on one variable "c" to increase or decrease the number of slots in next frame. Instead of only one value of "c" for increasing or decreasing the frame size, the Fast-Q algorithm uses one value of "c" when collisions occur and another value when an empty slot is detected. The authors claim that there are two factors that influence the total number of slots produced by a protocol, one is the duration of the collided and empty slots and other is their probabilities. Therefore to find the optimal value of Ccoll and Cidle both of these factors are considered. In [11] the ratio between the probabilities of collided and empty slots was derived and it was determined to be "e – 2". The authors of [4] take that value and multiply it by the ratio of the duration of collided and empty slots to find the optimal Ccoll and Cidle. Combining these two components give the complete ratio between collided and empty slot. Equation (1) shows this value.

$$\frac{Ccoll}{Cidle} = \frac{Tcoll}{Tidle} \text{ x } \frac{Pcoll}{Pidle} = 1.98 \text{ x } (e-2) \qquad (1)$$

$$=1.4122$$

In [4], the simulation results for the Fast-Q algorithm show that by adjusting Q with this method a significant decrease in the number of slots is observed when compared to Q-algorithm. In fact the results came closer to ideal DFSA (Dynamic Frame Slotted ALOHA) results. It can be observed that this protocol is dependent upon the time a collision or empty slot takes which for different data rates will be different. Because of this dependency the ratio $\frac{Tcoll}{Tidle}$ is will different for different system parameters. In [4] the value 1.98 was calculated when TRrate was set to 62.5Kbps and RTrate was set to 64Kbps. In our paper TRrate is set to 15.3Kbps and RTrate is set to 64Kbps which makes the ratio 4.92. Ccoll is set to 0.283 and Cidle is set to 0.08. The results from our simulation prove that by considering just one set of systems parameter the same results cannot be achieved.

C. DBQ Algorithm

In [2] a frame size estimator is proposed which is an advance version of the Q-algorithm. This protocol predicts the next optimal frame size based on the network's current state instead of simply adjusting it by a constant value "c". If Nc is the total number of collided slots in the current frame, Ne is the total number of empty slots in the current frame and Qi is the Q of the current frame "i",than Qi+1 is predicted by DBQ as follows

$$Q_{i+1} = round(Q_i + k_1Nc - k_2Ne) \qquad (2)$$

The values k1 and k2 are bias values which were calculated using the Least Square Method on a mathematical simulation of the Dynamic Frame ALOHA protocol. The protocol is thus named Dual-Bias

Q-Algorithm (DBQ-Algorithm). The authors ran simulations under the assumption that the reader has knowledge about the number of tags in its vicinity. They called this a perfect estimator since in real world the reader does not know the exact number of tags in a network. The purpose of these simulations was to find value of Q2 when Q1is known.The expected number of collided (Nc) and empty slots (Ne) were also discoveredusing binomial equations where the frame size and respective number of tags' were found from simulations. The authors then calculated several values of Q2 by keeping Q1, L and N constant for "m" inventory rounds and applied Least Square (LS) method on these values with the values obtained from perfect estimator.   The values of k1 and k2 were determined by using equation (3)

$$\begin{pmatrix} k1 \\ -k2 \end{pmatrix} = (H^T H)^{-1} H^T (\mathbf{Q_2} - \mathbf{Q_1}) \tag{3}$$

  H is the averaged collision pattern matrix, Q1 is the vector of the values Q1 and Q2 is the vector of Q2 found by keeping Q1the same for "m" rounds.The authors created a table of the biases (k1, k2) for different values of Q. The reader uses this table to determine best size for the next frame (Qi+1) given Qi, Nc and Ne using equation (2). The simulation results show that the number of slots required to read tags decrease substantially using DBQ algorithm compared to Q-algorithm (when Q-algorithm is considered to change frame based on frame by frame method as oppose to slot to slot method). The authors do not provide any other method to evaluate their protocol other than redundant time and total number of slots required to scan tags. Since in RFID each slot has different lengths, by decreasing total number of slots does not guarantee that total time to read the tags will shrink down as well.

D.   RETES Protocol

  In [3], a new anti-collision protocol named RETES (RFID Explicit Tag Estimation Scheme) is introduced which estimates the accurate number of tags around a reader and is compatible with EPC C1G2. In this paper the authors compared their protocol with "Schoute backlog estimation technique" [8] and with "Lower Bound (LB) backlog estimation technique" [9]. Schoute and LB techniques have proved to be the best in tag estimation methods and authors of [3], with experimentation, show that RETES can provide better results with certain parameter values. In Schoute's method, the frame size is estimated based on equation (4) and in LB frame size is estimated with equation (5). It can be noticed that in these schemes only collision slots are considered to estimate the next frame size. In RETES, the authors suggest using equation (6) to estimate the backlog. This equation considers two variables V1 and V2, where the value of V1 is between 2.0 and 2.5 and the value of V2 is between 0.1 and 0.5. In equation (6) the next frame size is estimated by considering the collision happened in current frame and by estimating the number of empty slots in next frame. The authors do not give any explanation of the values of the variable V2 except that since empty slots do not engage any tag, they assume the variable's value fall between 0.1 and 0.5. Experiments were performed to evaluate the number of slots needed for different initial values of Q and varying numbers of tags. The results show that if the initial Q is high (8 in these experiments), RETES requiredfewer slots to read 200 and 250 tags. For these results the values of V1 were between 2.0 and 2.2 whereas the values of V2 were between 0.1 and 0.2. However, as the number of tags increased to where 2^Q was less than the number of tags, the performance degraded. With these results, the authors concluded that the backlog is affected not only by the current collided slots but also the current empty slots. Also since the performance of RETES is affected by the initial value of Q and the values of V1 and V2, the authors propose to adaptively adjust these values based on the number of tags to achieve best results. The idea of using an initial Q appropriate for the number of tags to be read is not practical in RFID systems since this

information is usually not available. If this assumption is not taken into consideration RETES will not perform better than Schoute or LB protocols.

$$\text{Backlog or FrameSize} = 2.39 \times C \tag{4}$$

$$\text{Backlog or FrameSize} = 2 \times C \tag{5}$$

$$\text{Backlog or FrameSize} = \text{Round}(V_1 \times C + V_2 \times E) \tag{6}$$

IV.   UNITS OF MEASUREMENT

In this section we define the units we have used to compare the four transmission control schemes. The two popular units among protocol creators are System Efficiency and the Total Number of Slots used to read a set of tags. We utilize both of these units as well as the Total Time needed to read the tags and unit created in [6] called Time System Efficiency.

A. System Efficiency (SE)

The System Efficiency (SE) is defined as the ratio between total number of successful slots and the total number of slots required to read a set of tags. The higher the number of total slots used by a protocol, the lower the efficiency. This unit has been used in several wired and wireless FSA based protocols for decades, however as pointed out by the authors of [6] this unit does not give the whole picture of the efficiency of a protocol for an RFID system due to the fact that the slots are of different lengths.

$$SE = \frac{Total\ Number\ of\ Successful\ Slots}{Total\ Number\ of\ Slots} \tag{7}$$

B. Total Number Slots

This is self-descriptive unit defined as the total number of slots needed for a protocol to read a set of tags. This number includes all the successful slots, collided slots and empty slots. Equation (8) shows the total slots needed to read tags in "m" rounds.

$$TotalSlot = \sum_{i=1}^{m} Nsm + \sum_{i=1}^{m} Ncm + \sum_{i=1}^{m} Nem \tag{8}$$

C. Total Time

This is the total end to end time taken by a reader to read a set of tags. This includes the time spent onthe first Query command and the time spent on each slot in the frame.The time spent on individual successful, empty and collided slot is given by equations (9), (10) and (11) respectively.

$$T_S = 2 * (T_1 + T_2) + T_{Query} + T_{RN16} + T_{ACK} + T_{EPC} \qquad (9)$$

$$T_e = T_1 + T_2 + T_{Query} \qquad (10)$$

$$T_c = T_1 + T_2 + T_{Query} + T_{RN16} \qquad (11)$$

D. Time System Efficiency (TSE)

The unit TSE was introduced in [6]. TSE is the ratio between the time taken to read all tags if there were no collisions and the actual time spent on reading same set of tags. If TSE is 0.6, it means that the 60% of the time was spent on identifying tags whereas 40% of the time was spent on collisions and empty slots. Equation (12) represents TSE. The definition of TSE is similar to SE but with slot time consideration. It is important to take time into account with EPC C1G2 protocol as the lengths of the slots accessing tags are different compared to other protocols where length of each slot is constant. The authors of [6] point out that many protocol designers try to achieve maximum SE by reducing total number of slots to read the tags without considering the amount of different types of slots. If the aim of a protocol is to maximize TSE it would be acceptable to increase smaller empty slots if there is a large reduction in collision slots.

$$TSE = \frac{\sum Ts}{\sum Ts + \sum Tc + \sum Te} \qquad (12)$$

V. SIMULATION RESULTS

We created a software simulation of the four protocols using the C language. The data rates, size of the query command, RN16 and EPC length remained constant for all protocols. For the Q and Fast-Q protocols, the approach of slot-by-slot frame change is implemented whereas for DBQ and RETES protocols the decision of changing the frame size was performed at the end of the frame. The simulations were executed for 10-1000 tags assumed to be within a reader's range. The effect of noise and other environmental factors were not considered during the simulation. The results shown below were obtained by averaging 100 runs. These resultswere compared for a small number of tags (10-100) and a large number of tags (200-1000)separately based on the four metrics SE, TSE, Total Slots and Total Time.
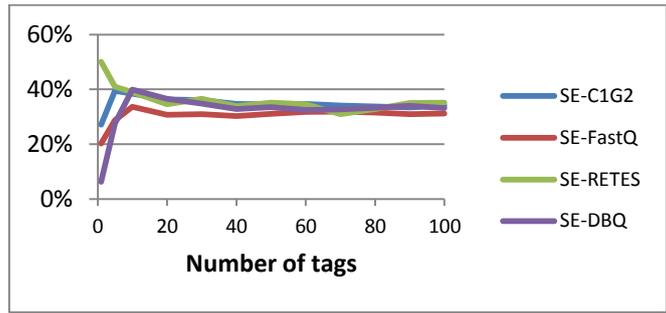
A. Results for Small RFID Networks

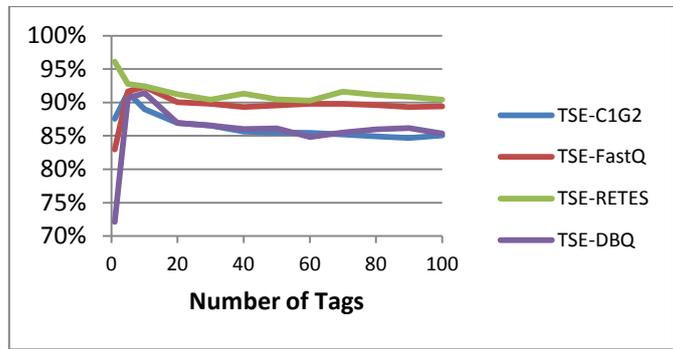**Figure 3.** System Efficiency comparison for small networks

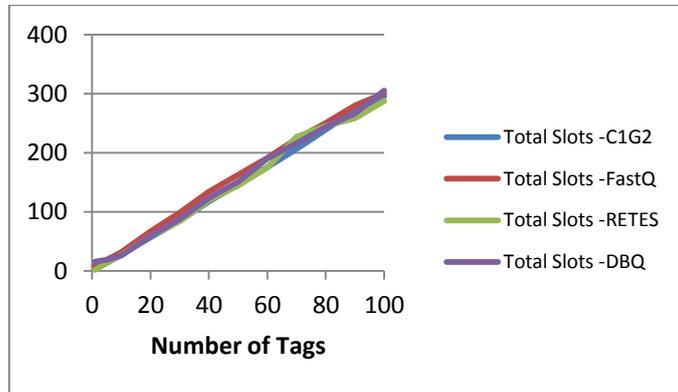**Figure 4.** Time System Efficiency comparison for small networks

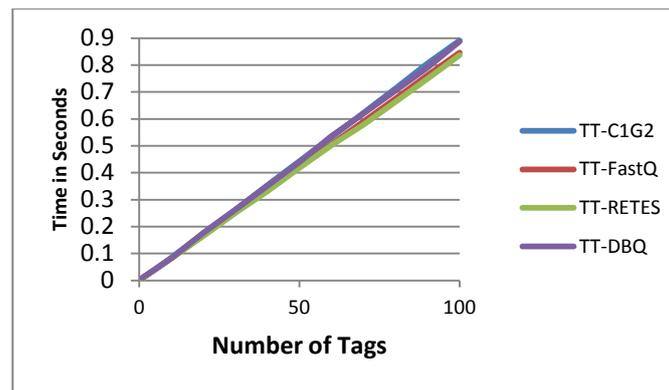**Figure 5.** Total Number of Slots comparison for small networks

Figure 6. Total Time comparison for small networks

From Figure 3 we can observe that SE for RETES is 50% when the number of tags in the network is between 1 and5. For between 6 and100 tags, SE does not changemuch and remains between 31-35%. DBQ and C1G2 protocols havealmost the same SE. Both start at lower SE when the number of tags is between 1 and 5 and reach a maximum value of 40% when the number of tags is close to 10. After 10 tags, the SE for DBQ and C1G2 falls between 30-35%. FastQachievesa maximum SE of 35% when the number of tags is 10. The SE of Fast-Q then decreases to 30% when the number of tags increases in a network.

Figure 4 shows that RETES acheives 97% TSE,which is best among the the four protocols, when the number of tags in the network is between 1 and 5. The TSE for RETES then drops to 90% and remians almost constant as the number of tags increase. The TSE for the DBQ protocol is 72% which is the lowest TSE among the four protocols when the number of tags is between 1and 5. The DBQ protocol does gain TSE as the number of tags increase, and it reaches a maximum value of 92% when the number of tags is 10. The TSE of DBQ decreases to 85% after the network size grows to 25 tags, and it remains at that value as number of tags increase beyond 25. It can be observed that the EPC C1G2 protocol follows the same TSE valuesas the DBQ protocol. Fast-Q starts the TSE at 84% for network sizes of 1 to 5 tags, reaches a maximum value of 92% when the number of tags is 10, and becomes a constant (90%) as number of tagsincrease.

Figure 5 shows the total number of slots required by a protocol to read up to 100 tags. All protocols show a linear increase in the number of slots as the number of tags increase. By observing Fast-Q's total slots in Figure 5, it can be concluded that this protocol requires slightly more slots to read tags compared to the other three protocols. When the network size is 40 tags, Fast-Q produces approximately 134 slots whereas the other three produce around 120 slots. In Figure 6, the total time to read up to 100 tags is shown. The graph shows a linear relation between the number of tags and the total time for all four protocols. The RETES and Fast-Q protocols read 100 tags in 0.85 seconds whereas C1G2 and DBQ protocols took 0.9 seconds to read the same amount of tags. For less than 50 tags all four protocols take the same amount of total time.

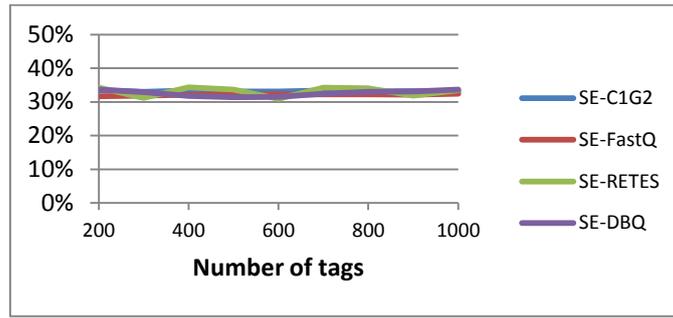*B.   Results for Large RFID Networks*

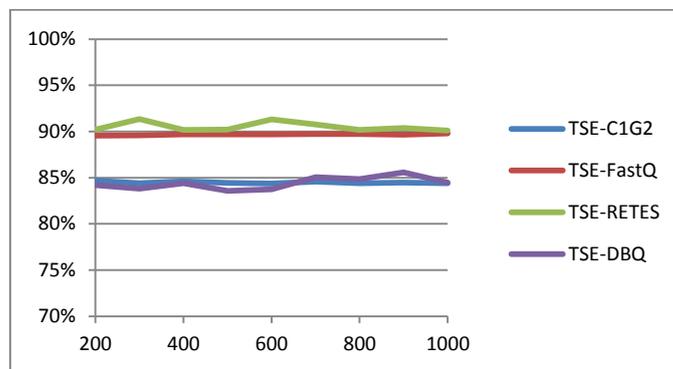**Figure 7.** System Efficiency comparison for large networks

Figure 8. Time System Efficiency comparison for large networks
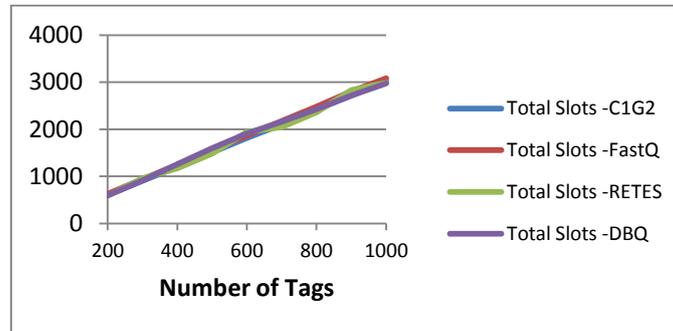

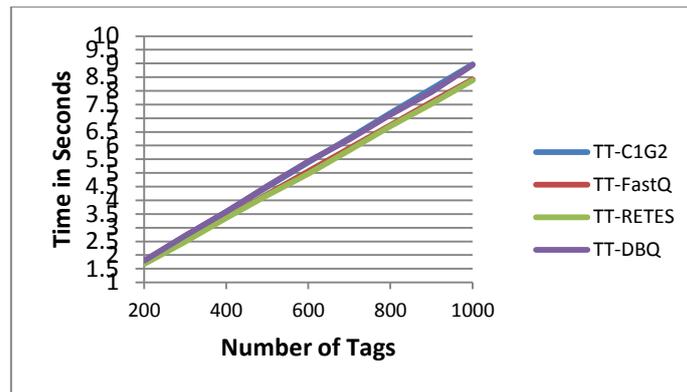**Figure 9.** Total Number of Slots comparison for large networks

Figure 10. Total Time comparison for large networks

Figure 7 shows that for a large number of tags, the SE for all four protocols is about 34% as can be expected for any FSA based protocol. The TSE for Fast-Q and RETES from Figure 8 is about 90% and the TSE for the EPC C1G2 and DBQ protocolsis about 85%. Clearly TSE shows a different picture of a system's performance when compared to SE. The total slots required to read a large number of tags increases linearly as the number of tags increase. The maximum number of slots was recorded at 1000 tags and its value was 3000. It can be observed from Figure 9 that all four protocols end up using total slots that are three times the number of tags. For example to read 200 tags every protocol used 600 slots. The same is true for other numbers of tags. The total time required to read a large number tags increases linearly as the number of tags increase. This can be observed from Figure 10. RETES and Fast-Q protocols require the least amount of time to read 1000 tags whereas EPC C1G2 and DBQ require the most although the difference is very minute (about 1 second).

VI. CONCLUSION

From the four protocols we compared in this paper, RETES showed the highest system efficiency, highest time system efficiency and lowest reading time. The better performance of RETES over the other three protocols was due to the fact that it assumes that the size of the network is known before starting the query process. This assumption is not valid when it comes to RFID networks since depending upon the application the number of tags may not be known. From the other three protocols we conclude that Fast-Q provides the best results in three out of four units of measurement.This proves that when comparing performance it is important to consider multiple units. Also it can be concluded from this study that in RFID networks it pays off when the frame size is adopted quickly to the network's conditions. This is the reason why Fast-Q performed better than the DBQ protocol since in DBQ we wait until the end of the current frame to make a decision on the next frame size.

References

[1]   EPCglobal, "EPC Radio-Frequency Identity Protocols Class-1 Generation- 2 UHF RFID Protocol for Communications at 860 MHz -960MHz Version 1.0.9." EPCglobal, 2005

[2]   D. Lee, O. Bang, S. Im and H. Lee, "Efficient Dual Bias Q-Algorithm and Optimum Weights for EPC Class 1 Generation 2 Protocol", 14th European Wireless Conference, 2008.

[3]   P.Pupunwiwat and B.Stantic, "A RFID Explicit Tag Estimation Scheme for Dynamic Framed-Slot ALOHA Anti-Collision", 6th International Conference on Wireless Communications Networking and Mobile Computing, September 2010.

[4]   J.Teng, X.Xuan and Y.Bai, "A Fast Q Algorithm Based on EPC Generation-2 RFID Protocol", 6th International Conference on Wireless Communications Networking and Mobile Computing, September 2010.

[5]   C.Floerkemeier, "Bayesian Transmission Strategy for Framed ALOHA based RFID Protocols", IEEE International Conference on RFID, 2007

[6]   T.LaPorta, G.Maselli and C.Petrioli "Anti-collision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization", IEEE Transaction on Mobile Computing, February 2011.

[7]   M. Daneshmand, W. Chonggang, K. Sohraby, "A New Slot-Count Selection Algorithm for RFID Protocol", Communications and Networking in China, CHINACOM, 2007

[8]   F.Schoute, "Dynamic Frame Length ALOHA", IEEE Transactions on Communications, volume 3, 1983.

[9]   H. Vogt, "Efficient Object Identification with Passive RFID Tags", International Conference on Pervasive Computing, 2002.

[10]   D. Liu, Z. Wang, J. Tan, H. Min, J. Wang, " ALOHA Algorithm Considering the Slot Duration Difference in RFID system", IEEE International Conference on RFID, 2009

[11]   D. Lee, K. Kim and W.Lee, "Q+-Algorithm: An Enhanced RFID Tag Collision Arbitration Algorithm", 4th International Conference on Ubiquitous Intelligence and Computing, July 2007.