

**DYNAMIC THRESHOLD BASED ROUTING FOR
SENSOR NETWORKS**

**M.Sc. Thesis by
Osman KORKUTAN**

Department : Computer Engineering

Programme: Computer Engineering

FEBRUARY 2006

**DYNAMIC THRESHOLD BASED ROUTING FOR
SENSOR NETWORKS**

**M.Sc. Thesis by
Osman KORKUTAN
504031523**

Date of submission : 19 December 2005

Date of defence examination: 03 February 2006

Supervisor (Chairman): Assist.Prof.Dr. Feza BUZLUCA

Members of the Examining Committee Prof.Dr. Emre Harmancı

Prof.Dr. Şebnem Baydere (Y.Ü.)

FEBRUARY 2006

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**DUYARGA AĞLARINDA DİNAMİK EŞİK TABANLI
YÖNLENDİRME**

**YÜKSEK LİSANS TEZİ
Osman KORKUTAN
504031523**

Tezin Enstitüye Verildiği Tarih : 19 Aralık 2005

Tezin Savunulduğu Tarih : 03 Şubat 2006

Tez Danışmanı : Yrd.Doç.Dr. Feza BUZLUCA

Diğer Jüri Üyeleri : Prof.Dr. Emre Harmancı

Prof.Dr. Şebnem Baydere (Y.Ü.)

ŞUBAT 2006

ACKNOWLEDGEMENTS

First of all, I am deeply appreciated to Ass.Prof. Feza BUZLUCA for his supervising and his kind tolerance to me. His support and encouragement made me write this thesis.

I dedicate this thesis to my father and mother who supported me in every phase of my educational life and all my teachers who thought me analysing, researching, and determined working. Without them everything would be extremely difficult for me.

Also, I would like to thank to the leader of my department in Siemens A.Ş., Mrs. Nesrin Keçik, and my colleagues in my department for supporting me in my educational life.

December 2005

Osman KORKUTAN

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
ABBREVIATIONS	vi
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF SYMBOLS	ix
ÖZET	x
SUMMARY	xi
1. INTRODUCTION	1
1.1. Introduction to Wireless Sensor Networks	1
1.2. Objective	2
2. RELATED WORK	3
2.1. A Classification	3
2.1.1. LEACH	3
2.1.2. TEEN	4
2.1.3. APTEEN	4
2.1.4. Drawbacks of TEEN and APTEEN	5
2.2. Sleeping/awaking Mechanisms	6
2.2.1. ELECTION	7
2.3. Localization	7
2.4. The Proposed Protocol	7
3. THE NEW APPROACH	8
3.1. The Model	8
3.2. The Effective Timeout	8
3.3. The Lower Bound	9
3.4. The Historian Matrix	9
3.5. Characteristic Steps of New Protocol	10
3.6. Threshold Generation	12
3.6.1. Soft Threshold Generation	12
3.6.2. Hard Threshold Generation	13
4. SIMULATIONS	15
4.1. Simulation Environment and Assumptions	15
4.2. First Group of Simulations	16
4.3. Second Group of Simulations	21
4.4. Third Group of Simulations	21
CONCLUSION	23
REFERENCES	25

APPENDIX A : SOFT THRESHOLD GENERATION ALGORITHM	27
APPENDIX B : HARD THRESHOLD GENERATION ALGORITHM	28
APPENDIX C : SIMULATED TRENDS AND BEHAVIOURS OF PROTOCOLS	29
BIOGRAPHY	37

ABBREVIATIONS

LEACH	: Low-energy Adaptive Clustering Hierarchy
TEEN	: Threshold-sensitive Energy Efficient Sensor Network Protocol
APTEEN	: Adaptive Periodic Threshold-sensitive Energy Efficient Sensor Network Protocol
TDMA	: Time Division Multiple Access
ELECTION	: Energy-efficient and Low Latency Scheduling Technique for Wireless Sensor Networks

LIST OF TABLES

<u>Number</u>		<u>Page</u>
Table 4.1.	Random number generators and their mean values.....	15
Table 4.2.	Simulation parameters.....	16
Table 4.3.	Simulation results for Test-1 when R_1 is the case	25
Table 4.4.	Simulation results for Test-2 when R_2 is the case	25
Table 4.5.	Simulation results for Test-3 when R_3 is the case	25
Table 4.6.	Simulation results for Test-4 when R_4 is the case	25

LIST OF FIGURES

	<u>Page Number</u>
Figure 4.1 : Total number of transmissions of all three protocols as the mean value of random number varied.....	27
Figure 4.2 : Total remaining energy of all three protocols as the mean value of random number varied.....	28
Figure 4.3 : Network lifetimes of all three protocols as the mean value of random number varied.....	29
Figure 4.4 : The change in the network lifetime of new protocol with four different number generators as subgroup count in the cluster varied	30

LIST OF SYMBOLS

C_{SG}	: Subgroup Count
T_E	: Effective Timeout
B_L	: Lower Bound
M_H	: Historian Matrix
S_H	: Historian Size
V_S	: Sensed Value
V_{LT}	: Last Transmitted Value
C_C	: Number of Cycles Since the Last Transmission
T_S	: Transmission Time
T_{TE}	: Total Time Elapsed
T_A	: Average Periods of Time
C_M	: Absolute value of the minimum change in the sensed values
T_S	: Soft Threshold
B_{HT}	: Hard Threshold Value Bound
V_M	: Maximum value of the transmitted values
T_H	: Hard Threshold
R_1, R_2, R_3, R_4	: Random Number Generators
V_M	: Mean Value of the Uniform Distribution
V_i	: Sensed Value in i th Cycle

DUYARGA AĞLARINDA DİNAMİK EŞİK BELİRLEMEYLE YÖNLENDİRME

ÖZET

Duyarga ağları yönlendirme protokolleri, bu ağların limitli enerji seviyeleri, kısıtlı bant genişliği, kısıtlı sistem kaynakları, güvenilir olmayan iletişim ortamı ve hareketlilik gibi çözülmesi gereken sorunları olmasından dolayı ilk kullanılmalarından itibaren hala güncelliğini koruyan geniş bir araştırma alanıdır. Bu protokoller genelde özel olarak belirli ihtiyaçları karşılamak ve belirli uygulamalarda kullanılmak üzere geliştirilmektedirler. En önemli iki duyarga ağı protokol grubu veriye dayalı protokoller ve olay bazlı protokollerdir.

Olay bazlı yönlendirme yapan ağ protokolleri, çevredeki değişimlere daha kısa sürede cevap vermelerinden ve periyodik veri iletim yüklerini taşımamalarından dolayı veriye dayalı yönlendirme yapan ağ protokollerine göre gerçek zamanlı uygulamalara daha uygundurlar. Bu tür protokollerde ortamdan hissedilen verinin iletilme kararının verilmesinde eşik kullanımı yaygın olarak kullanılan bir yöntemdir. Yöntemin verimliliği ise uygulama için doğru eşik değerinin seçilmesine çok bağlıdır. Uygulama alanındaki değişimler olağansa, doğru eşik değerlerini seçmek çok zor değildir ancak ilgilenilen niteliğin değişimi tutarsız ise, eşik değerlerinin sabit olması ihtiyaçları karşılamaktan uzaklaşır.

Bu ihtiyaçlardan hareketle, ortamdan hissedilen niteliğin değişim hızına göre grup bazlı eşik değerleri üreterek değişimlere ayak uydurabilen, duyargaları altgruplara ayırıp, bu alt gruplar üzerinde uyuma çizelgeleri oluştururak enerji tüketimini artırıp ağ ömrünü uzatan yeni bir olay bazlı ağ yönlendirme protokolü geliştirdik. Yaptığımız testlerde yeni geliştirilen protokol ilgilenilen niteliğin değişimini izleme, yeni koşullara uygun eşik değerleri oluşturma, ağ çalışmasını öngörülemez değişimlere göre uygunlaştırma ve düğümler arası haberleşmeyi azaltarak duyarga enerjilerini koruma özellikleri ile varolan protokollerden çok daha iyi bir performans sergilemiştir.

DYNAMIC THRESHOLD BASED ROUTING FOR SENSOR NETWORKS

SUMMARY

Developing wireless sensor network protocols has been a great challenge since first sensor networks were built due to the limitations and problems such as limited energy, limited bandwidth, limited system resources, unreliable transmission media and mobility. These protocols are usually developed in order to satisfy different needs and to be used in different applications. Two major groups of them are data-centric and event-based wireless sensor network protocols.

Unlike the data-centric sensor network protocols, event-based protocols are more appropriate for time-critical applications because of having short respond time for the changes of interested parameters and eliminating periodic data acquisition. Thresholds are commonly used to decide to transmit the sensed data on sensor nodes in event-based sensor network protocols. The efficiency of this technique is based on choosing the right thresholds for different applications. If ordinary changes of interests are observed in network environment, it is not so difficult to retrieve efficient thresholds; nevertheless, if the amount of change in the interests is unusual, fixed thresholds would not satisfy the needs.

We propose an event-based sensor network protocol that generates cluster-based thresholds in order to adapt the sensor system to the changes in the environment dynamically and manage sleeping schedules on sensor nodes in subgroups which are formed within clusters in order to provide energy efficiency and increase network lifetimes. Simulation results show that our protocol outperforms existing protocols by observing actual behavior of interested attributes, generating appropriate thresholds, adapting the network to unpredictable changes and increasing network lifetimes by decreasing transmission count between sensor nodes.

1. INTRODUCTION

1.1. Introduction to Wireless Sensor Networks

By the help of improvements in sensor technology, sensor networks have begun to be widely used due to having many advantages. They have a wide range of use on many different applications. On the other hand, sensor networks still have several limitations and problems that result in great research challenges such as energy efficiency and fault tolerance.

Energy efficiency is provided by using different techniques in recent protocols such as sleep scheduling, selective data transmission, data aggregation. In the approach of using thresholds which is implemented to satisfy the needs of time-critical applications, energy efficiency is mostly provided by transmitting sensed data less frequently.

According to a classification methodology [1], which is based on the functionality and application type of the network, sensor networks are grouped in two main families: *Proactive Networks* and *Reactive Networks*. *Proactive Networks* are those networks in which the sensor nodes sense the environment periodically and transmit the data as proposed in LEACH [2]. Nodes in sensor networks of the second group, *Reactive Networks*, sense the environment periodically but transmit the data if there is an interesting change in the value of relevant interest. The networks of this group react immediately to these changes; consequently, these networks are proper for time-critical and real time applications.

Reactive Networks work with the approach of threshold based data transmission. The main concept of threshold based data transmission is to transmit sensed data if the data exceeds predefined and announced static thresholds. Using static thresholds would be efficient if the changes in the interests are usual and the threshold values are well predicted. Nevertheless, when the thresholds are not appropriate for the network, sensor nodes can run out of energy in a short time because of frequent

transmissions or users can not get enough information about the current status of the network because of rare transmissions.

In [3], design and implementation concepts of a reactive network for environmental monitoring are given and the effectiveness of the network for data gathering is measured.

1.2. Objective

In this work, we present a new technique to generate thresholds in order to dynamically adapt the sensor network to the changes in the environment. By using historical data, new thresholds are generated and these values are announced along the network so the network automatically adapts itself to changes. Moreover, by forming subgroups of sensor nodes within clusters and managing sleeping schedules by keeping awake one of the nodes in each subgroups and sleeping the others, transmission count between nodes are decreased. As a result, the network lifetime is increased because only the awake nodes in clusters transmit data and transmissions take place in reasonable intervals; consequently, energy is used optimally. Simulation results show that our algorithm generates near optimal thresholds and helps the nodes in the sensor network to keep up with the changes in the behavior of sensing attributes in the media. Furthermore, network lifetimes increased considerably by eliminating transmissions between sensor nodes that carry redundant data.

2. RELATED WORK

2.1. A Classification

When sensor networks are classified by the means of data gathering, they can be grouped into two: *Periodic Data Gathering* and *Event Based Data Gathering*. In the first approach, sensor nodes sense the media and transmit the data periodically even though having no significant data. In the second approach, sensor nodes listen for a specific change in the environment which could be called an event.

2.1.1. LEACH

A hierarchical clustering based and energy-efficient protocol, LEACH [2], is a good example of *Periodic Data Gathering Protocols*. In this protocol, some nodes are selected as cluster head nodes on which data aggregation takes place. The cluster heads aggregate the data that is transmitted by the nodes in the related cluster. By aggregating data, the amount of information transmitted to the base station is reduced. LEACH also rotates the cluster head role in order to distribute the load among the whole network.

In LEACH, data collection is centralized and is performed periodically; consequently, this protocol is appropriate when there is a need for constant monitoring by the sensor network. A user may not need all the data immediately so periodic data transmissions are unnecessary which may consume much more energy of the sensor nodes. After a given interval of time, the role of the cluster head is given to another node so that uniform energy dissipation in the sensor network is obtained.

The operation of LEACH is separated into two phases, the setup phase and the steady state phase. In the setup phase, the clusters are organized and cluster heads are selected. In the steady state phase, the actual data transfer to the base station takes place. The duration of the steady state phase is longer than the duration of the setup phase in order to minimize overhead.

LEACH increases the network lifetime but there are a number of issues about the assumptions used in this protocol. LEACH assumes that all nodes can transmit with enough power to reach the base station if needed. Therefore, it is not applicable to networks deployed in large regions. It also assumes that nodes always have data to send and nodes located close to each other have similar data. Moreover, dynamic clustering brings extra overhead which causes losses in the saved energy.

2.1.2. TEEN

Unlike the other hierarchical routing protocols which are introduced in [4], [5], [6], [7] and [8], TEEN [1] is a protocol which is proposed for time-critical applications and an example of *Event Based Data Gathering Protocols*. In TEEN, two kinds of thresholds are used: hard threshold and soft threshold. Sensor nodes leave their sensors on and sense the medium continuously. They transmit the sensed attribute if only it is greater than the hard threshold or it differs from the sensed value which is stored before by an amount greater or equal to the soft threshold value.

The hard threshold tries to reduce the number of transmissions by allowing the nodes to transmit only when the sensed attribute is greater than an upper bound.

The soft threshold also reduces the number of transmissions which may occur when there is little or no change in the sensed attribute. A smaller value of the soft threshold gives a more accurate picture of the network; nevertheless, it will increase the energy consumption. The user can control the trade-off between energy efficiency and data accuracy.

The experiments have shown that TEEN outperforms LEACH in terms of energy consumption and network lifetime.

2.1.3. APTEEN

APTEEN [9] is a hybrid protocol that uses both periodic and event based data gathering. In addition to TEEN, a set of parameters are announced by cluster heads. These are *Attributes* (user interests - physical parameters), *Schedule* (TDMA schedule) and *Count Time*. The TDMA schedule is used to avoid collisions and each node in the cluster is assigned a transmission slot. Thresholds are used in the same way with TEEN; moreover, when a sensor node does not sense a value exceeding

thresholds for a period equal to *Count Time*, it transmits the sensed value. This results in the user to have a picture of complete network.

The main feature of APTEEN is combining both proactive and reactive policies. It offers flexibility by allowing the user to set the *Count Time* and the threshold values for the energy consumption can be controlled by changing the count time as well as the threshold values.

An analytical model for information retrieval using APTEEN is introduced in [10] and this work is said to be the first step of analytically determining the delay characteristics of a wireless sensor network.

The experiments have shown that APTEEN's performance is somewhere between LEACH and TEEN in terms of energy consumption and network lifetime because of the periodic reports and additional complexity to manage TDMA schedule.

2.1.4. Drawbacks of TEEN and APTEEN

The main drawbacks of TEEN and APTEEN are failing of communication when the thresholds are not received and the efficiency lost when the thresholds are not well defined or changes of the relevant interests are not consistent. The thresholds are fixed and announced among all nodes in the network but in large sensor networks, thresholds might not be valid for the whole network and the change in sensed values might not behave the same in different clusters. Moreover, the probability of transmitting multiple copies of same events is very high in both protocols because adjacent nodes sense same regions and this causes unnecessary energy consumption.

All of the values of the sensed attribute are reported by sensor nodes in both protocol if the values are greater than the hard threshold; nevertheless, if the values exceed the hard threshold for a long time, sensor nodes will consume much more energy and the network traffic will be overloaded.

In TEEN, if sensor nodes do not sense values exceeding thresholds, they will not transmit any data. This will result in users not to have any information about the network and the status of sensor nodes. This problem is solved in APTEEN by using *Count Time*. If sensor nodes do not sense a value exceeding thresholds for a period equal to the *Count Time*, they will transmit the last values they sensed which will give general information to the user about the network and status of the related

attribute. Unfortunately, adjacent nodes will probably report the same value and multiple copies of same data will be delivered through the network.

2.2. Sleeping/awaking Mechanisms

A widely used technique in sensor networks is to use sleeping/awaking mechanisms in order to save energy. This technique is based on the fact that sensor nodes consume much less energy during the sleep mode than the idle mode. Three major approaches are made by using this technique: Timer-Based, On-Demand and Hybrid [11]. In Timer-Based approach, nodes go into sleep mode after setting their timers to wakeup at a static and pre-determined time. They wakeup as soon as the sleep interval are elapsed. A well-defined sleep schedule algorithm must be handled in this approach to prevent data loss due to long intervals.

Sensors are slept forever until they are awoken by other nodes in On-Demand approach. In this approach, usually a second, low-power radio is added to sensor nodes to awake neighbors on demand. Low energy consumption is provided by simpler hardware with a lower bit-rate or less decoding capability and periodic listening using a radio with identical physical layer as data radio. “Wakeup” messages can be directed in order to awake several nodes or a broadcast to awake all nodes in the network. The steps which are performed by sleeping nodes are given below.

1. Sense “Wakeup” channel periodically.
2. If “Wakeup” channel is sensed as busy:
 - A. Turn on data radio.
 - B. Receive packet on data channel.
 - C. If “Wakeup” message is for another node, return to sleep.

Awaking all nodes in a network, which is called a “Full Wakeup”, results in high energy usage; consequently, directed “Wakeup” messages are usually preferred. Hybrid approach is a combination of the other two approaches.

2.2.1. ELECTION

ELECTION [12] is an Event Based Data Gathering Protocol that also uses a Hybrid-like sleep scheduling technique which adaptively schedules the sleep cycles of sensor nodes. Unlike TEEN and APTEEN, nodes do not take the samples of the environment periodically. They sense the environment once at each wake-up and transmit the sensed attribute if there is an abnormality. Sleep cycle scheduling is very important in this protocol because inappropriate sleep cycles would result in long response time to the changes in the media.

2.3. Localization

Localization, which is defined as determining the physical positions of nodes in a network, has been a great challenge in wireless networks due to many reasons such as limited resources, mobility. Solutions that are announced in [13], [14], [15] and [16] propose techniques to handle the localization problem. Recently, many wireless network protocol designs are built with the assumption of geographic location information of sensor nodes in the network and depend on these localization techniques. One of the best technique, which was a simple and useful connectivity metric technique, explored in [13] for localization in outdoor environments which uses the inherent radio frequency communication of sensor nodes without using GPS.

2.4. The Proposed Protocol

We propose a periodic and cluster based threshold generation technique by using historical values that are transmitted to the cluster head in time. Our technique will eliminate the failures when the thresholds are not received or fixed all over the network by generating cluster specific thresholds and distributing them within clusters periodically. We will use an On-Demand-like sleeping/awaking mechanism that proposes to keep only one node awake at a time in a subgroup which consists of neighbor nodes within a cluster. In order to form subgroups within clusters, positional information of nodes in the network is needed and this information will be provided by using the technique announced in [13].

3. THE NEW APPROACH

3.1. The Model

At the beginning of our work, we build a model sensor network to analyze the performance of our algorithm and made some assumptions.

- Our model is based on the hierarchical clustering scheme announced in LEACH [2]. It is assumed that all cluster based operations such as cluster forming, data aggregation and inter-cluster routing are done by this protocol. Cluster heads are elected among the sensor nodes within clusters periodically by LEACH in order to distribute the energy load caused by this role.
- It is also assumed that all nodes in the model network have the same initial energy level and are homogenous.
- Sensor nodes are usually scattered to application areas redundantly because when sensor nodes die, it may not be possible to replace them with new ones. Furthermore, clusters which are formed by LEACH consist of nodes which are geographically nearer so nodes in a cluster are usually close to each other. Because of being adjacent, neighbor nodes in a cluster are assumed to sense similar data.
- Localization is assumed to be provided by using “GPS-less Low Cost Outdoor Localization For Very Small Devices” announced in [13].

3.2. The Effective Timeout

Count Time, which was introduced in APTEEN, provides several functionalities. One of them is that *Count Time* is used like a timeout mechanism because as soon as this time period is elapsed without sensing any interesting change in the media, last sensed values are sent by sensor nodes immediately; consequently, information about living nodes in the network is provided. We used a constant called *Effective Timeout* (T_E) that has a *Count Time* like functionality. Just like in APTEEN, if a sensor node

does not transmit data during a time period equal to *Effective Timeout*, it transmits the last sensed value. On the other hand, *Effective Timeout* has an additional usage. It represents an effective period that is used as an upper bound during threshold generation phase, which is explained in 3.6. This constant can be set primarily by users when the network installation takes place. All sensor nodes in the network have to know the value of this constant at the beginning.

3.3. The Lower Bound

Another constant which is attached to the algorithm which is called *Lower Bound* (B_L). All of the sensed values which are greater than the hard threshold are have to be transmitted in both TEEN and APTEEN. If the related attribute gets values greater than the hard threshold for a long period of time in a region in the application area, not only the energy consumption of sensors nodes in that region will increase in order to report the same event, but also network traffic will be overloaded. On the other hand, after informing the cluster head about an event, sensor nodes do not have the right to transmit a new packet until a number of cycles equal to *Lower Bound* is elapsed, where a cycle is equal to the time duration between two consecutive senses performed by a sensor node. The functionality preserves the network to be blocked and the sensor nodes from transmitting same information again and again. This constant can also be set by users during the installation phase.

3.4. The Historian Matrix

Sensor nodes in a network sense the application area periodically and check whether the sensed value has to be reported as an event or not. The sensed value is transmitted by the sendor node if one of the conditions that are given below is provided, where C_C is the number of cycles since the last transmission, V_S is the last sensed value, V_{LT} is the last transmitted value, T_S is the soft threshold, T_H is the hard threshold, B_L is the *Lower Bound* and T_E is the *Effective Timeout*.

- $((V_S - V_{LT} \geq T_S) \vee (V_S \geq T_H)) \wedge (C_C \geq B_L)$
- $C_C \geq T_E$

Matrixes are maintained on cluster heads, which are called *Historian Matrixes* (M_H), with a row count of *Historian Size* (S_H). *Historian Matrixes* are used to store old values that are transmitted to the cluster head. Each row consists of two fields: *Sensed Value* (V_s) and *Transmission Time* (T_s). As the cluster head receives information from sensor nodes, it stores the sensed values with their receive time.

$$M_H = \begin{bmatrix} V_{S1} & T_{S1} \\ V_{S2} & T_{S2} \\ \dots & \dots \\ V_{SHH} & T_{SHH} \end{bmatrix} \quad (3.1)$$

Cluster heads maintain separate *Historian Matrixes* for each *Subgroup*, which consists of a group of sensor nodes in the network and is introduced in 3.5. An index, which points to the last stored row, is kept for each *Historian Matrix* and the next received value from a node of the related *Subgroup* is stored to the next row with its transmission time.

3.5. Characteristic Steps of New Protocol

After cluster formation and cluster head selection phases, some characteristic steps of the algorithm are performed. At the beginning of these steps, every cluster in the network is divided into subregions and subgroups of sensor nodes that are formed by cluster heads within clusters to reduce transmission counts and energy consumption. Number of subgroups is a parameter which can be selected by user (C_{SG}). In order to form subgroups, cluster heads use the location information of nodes that is determined by using the technique in [13] and group nodes which are geographically near.

Moreover, a sleeping/awaking mechanism is used in the protocol to reduce the energy consumption of the sensor nodes. Sensor nodes in a cluster are awoken only by the cluster head according to their remaining energy levels. Nodes, which are awake, sense the media during a period called *Sensor Cycle*, which represents total count of cycles that a node remains awake before fall into sleep automatically.

After forming the subgroups in a cluster, the cluster head asks the sensor nodes in its cluster to announce their remaining energy levels. According to this information, cluster heads appoints a node which has the maximum remaining energy in each subgroup as watchdog and sleeps the others. This results in only one node in a subgroup remains awake and cluster head receive data from only four nodes at a time. During the *Sensor Cycle*, awake nodes of the subgroups sense the media and transmit data to the cluster head before they fall into sleep. At the end of each *Sensor Cycle*, sensor nodes wakeup automatically, the cluster head asks the sensor nodes in its cluster to announce their remaining energy levels and assigns a node as the watchdog in each subgroup.

With this assumption of being adjacent and sensing similar data, information received from only awake sensor nodes in a cluster is stored in a *Historian Matrix* at a time in order to reduce the amount of data that will be used by threshold generation algorithm. After selecting watchdogs in each subgroups, cluster head starts to store the information sensed and transmitted by these nodes. Awake nodes are appointed as ‘archive sources’. Number of *Historian Matrixes* are equal to C_{SG} which results in cluster head to store each subgroup’s information separately.

All of the steps performed by new protocol are summarized and listed below:

1. Clusters are formed (by LEACH).
2. Nodes announce their positions.
3. Subgroups are formed by cluster heads.
4. Until the next cluster formation phase, following steps are performed in a cycle:
 - A. Nodes announce their remaining energy levels.
 - B. The node which has the greatest remaining energy is appointed as Watchdog in each subgroup, other fall into sleep.
 - C. Awake nodes sense the application area and transmit data if one of the conditions which were given in 3.4 is provided during a *Sensor Cycle*.
 - D. All nodes wakeup after the *Sensor Cycle* is elapsed. Thresholds are regenerated and announced to the nodes by the cluster head.

3.6. Threshold Generation

One of the most important contribution of this study is the generation of thresholds. The soft and the hard thresholds are generated independently.

3.6.1. Soft Threshold Generation

After each *Sensor Cycle* is completed, soft threshold generation phase starts. As it was told before, cluster head maintains *Historian Matrixes* with a count which equals to the subgroup count. Soft threshold generation algorithm is applied to these matrixes independently and each execution generates a threshold. At the end, the smallest soft threshold is selected as the main soft threshold.

At the begining of the algorithm, total time elapsed (T_{TE}) since the beginning of the data collection is needed. To find out this value, the difference of *Transmission Time* values of consecutive rows in *Historian Matrix* is calculated and summed. Related formula is below.

$$T_{TE} = \sum_{i=1}^{S_H} M_H[i+1].T_s - M_H[i].T_s \quad (3.2)$$

In the second step, the average period of time elapsed (T_A) until the last threshold generation phase is found by the formula given below.

$$T_A = \frac{T_{TE}}{S_H - 1} \quad (3.3)$$

After that, the absolute value of the minimum change in the sensed values (C_M) between consecutive rows in *Historian Matrix* is calculated by the formula given below.

$$C_M = \left| \{i \mid 1 \leq i \leq S_H, \text{MIN}(M_H[i+1].V_s - M_H[i].V_s)\} \right| \quad (3.4)$$

Finally, the soft threshold (T_s) is calculated by the following formula using T_E , T_A and C_M :

$$T_s = \frac{C_M * T_{TE}}{T_A} \quad (3.5)$$

Here, it can be easily seen that as the average period of time elapsed, T_A , increases - which means the sensed values are sent less frequently-, the soft threshold decreases resulting in the sensor nodes to become more sensitive to the changes in the media.

After the algorithm is applied to each *Historian Matrix* and soft thresholds of each subgroup is calculated, the smallest one is selected as main soft threshold and the algorithm is finalized.

The main goal of this algorithm is to select an appropriate soft threshold so that number of transmissions become independent from the size of the change in the related attribute and sensor nodes perform nearly same number of transmissions in each case that the related attribute changes unpredictably. The algorithm handles a tradeoff between energy consumption and data delivery which results in trends of sensed attributes with reasonable and acceptable intervals.

3.6.2. Hard Threshold Generation

As it is introduced in TEEN [1], the sensed attribute by a sensor node is transmitted if the value is greater than the hard threshold. On the other hand, hard threshold is a user specific parameter. It fully depends on the application which the sensor network is used for. Hard thresholds represent the critical values of interested attributes and all sensed values exceeding hard thresholds would be announced to cluster heads. Generating hard threshold periodically causes this ‘alarm-level’ to be changed.

For this reason, our algorithm has two options: static hard threshold and dynamic hard threshold. In the first case, hard threshold is used just like in TEEN. It is announced periodically with soft threshold but never changes. In the second case, the hard threshold value is generated and announced periodically by cluster heads as it is done in generation of the soft threshold.

When dynamic hard threshold option is the case, the algorithm needs an additional constant to generate the hard threshold, *Hard Threshold Value Bound* (B_{HT}). This constant is a lower bound for hard threshold which is represented in percentage (%). Its value can be set primarily by users when the network installation takes place as it is done for the constant, *Effective Timeout*.

In the first step of hard threshold generation phase, the maximum value of the transmitted values (V_M) among those which are stored in each *Historian Matrix* is calculated independently as it is shown below.

$$V_M = \text{MAX} \{i \mid 1 \leq i \leq S_H, M_H[i].V_S\} \quad (3.6)$$

Next, the hard threshold generation step is performed, in which *Hard Threshold Value Bound* is used, for each subgroup as shown below.

$$T_H = \frac{V_M * B_{HT}}{100} \quad (3.7)$$

As it is in soft threshold generation phase, hard threshold is selected among the ones that are belong to subgroups but differently, the greatest one is selected as main hard threshold.

On the other hand, a *Historian Matrix* is maintained for one attribute. The number of matrixes maintained by cluster heads is equal to the attribute count which the user is interested in.

4. SIMULATIONS

4.1. Simulation Environment and Assumptions

We have done high level simulations to evaluate the performance of our algorithm. The main goal of the simulations is to compare the performances of pure TEEN, APTEEN and the proposed protocol for different behaviors of sensing attribute.

We have chosen cluster-based simulations because as it is told above, we have assumed that all cluster-based operations are performed by LEACH so we have decided to focus on simulating the real-time behavior.

The energy cost of LEACH has been discarded in each simulation because all related protocols uses clustering so considering energy load of LEACH is not needed for this simulation of comparison.

In our simulations, we have used static hard threshold option of the new protocol. As it's told before, hard thresholds are fully user and application dependent; consequently, our tests are focused on soft threshold generation.

We have used several data sources to simulate the sensed attributes by sensor nodes in the cluster and performances have been evaluated in each case for each protocol. The data sources are random number generators that produce values which are in a uniform distribution with different mean values to satisfy all cases. These are given in Table 4.1.

Table 4.1. Random number generators and their mean values

Random Number Generator	R_1	R_2	R_3	R_4
Mean Value	1	5	25	100

Generation of sensed values is depended on the last sensed data on sensor node and these values are directly used to generate the next sensed value. Related equation is given below where the sensed value in the cycle i is represented with V_i , the sensed value in the cycle $i+1$ with V_{i+1} , mean value of the uniform distribution with V_M and

random number which is generated each cycle with R_j where j is one of the random number generator options ($j = 1, 2, 3, 4$) described below.

$$V_{i+1} = V_i + R_i - V_M \quad (4.1)$$

4.2. First Group of Simulations

In the first group of simulations, performances of three protocols are compared by choosing same initial energy levels on sensor nodes and after a number of cycles, total remaining energy levels of sensor nodes are observed.

Our simulation environment consists of one cluster with 20 sensor nodes. All nodes have an initial energy size of 100000 units. All simulation parameters are given in Table 4.2.

Table 4.2. Simulation parameters

Parameter	Value
Sensor Count Within a Cluster	20
Subgroup Count (C_{SG})	4
Cost of Sensing	1 Unit
Cost of Sleeping	0.03 Unit
Cost of Transmission Within a Cluster	10 Units
Initial Energy Value of Each Node	100000 Units
Sense Period Of a Sensor	1 Second
Simulation Duration	100000 Seconds
Hard threshold (For all protocols)	1000 Units
Soft threshold (For TEEN and APTEEN)	10 Units
Effective Timeout (T_E)	100 Seconds
Lower Bound (B_L)	10 Seconds
Count Time (For APTEEN)	100 Seconds
Last sensed value at the beginning in each case	500 Units

In our first groups of simulations, four separate group of tests are performed for each protocol. In each group of test, a different random number generator option is used. Figures C.1, C.5, C.9, C.13 show the generated random numbers in each case with a seed of 500 Units which is assumed to be the value of the related attribute at the beginning. The trends given in the figures simulate the change in the related attribute in the media. Figures C.2, C.6, C.10, C.14 show the behavior of a selected node using TEEN while Figures C.3, C.7, C.11, C.15 show the behavior of a node using

APTEEN. Figures C.4, C.8, C.12 and C.16 show the behavior of the awake nodes of randomly selected subgroup in a cluster which uses the new protocol.

Tests have been performed with the simulated sensed values given above. Each sequence of values were applied to pure TEEN, APTEEN and the new protocol to which our algorithm is attached and we got results in each case. Tables 4.3, 4.4, 4.5 and 4.6 show the results of the simulations for each protocol in each case. Number of senses is 200000 for TEEN and APTEEN while it is 50000 for the new protocol because only the awake nodes continues sensing according to the algorithm.

Table 4.3. Simulation results for Test-1 when R_1 is the case

	TEEN	APTEEN	New Protocol
<i>Number of Senses</i>	200000	200000	50000
<i>Number of Transmissions</i>	603	2082	3078
<i>Total Remaining Energy</i>	1794573	1781262	1911898

Table 4.4. Simulation results for Test-2 when R_2 is the case

	TEEN	APTEEN	New Protocol
<i>Number of Senses</i>	200000	200000	50000
<i>Number of Transmissions</i>	21335	21354	3324
<i>Total Remaining Energy</i>	1607985	1607814	1909684

Table 4.5. Simulation results for Test-3 when R_3 is the case

	TEEN	APTEEN	New Protocol
<i>Number of Senses</i>	200000	200000	500000
<i>Number of Transmissions</i>	142592	142600	3482
<i>Total Remaining Energy</i>	516672	516600	1908262

Table 4.6. Simulation results for Test-4 when R_4 is the case

	TEEN	APTEEN	New Protocol
<i>Number of Senses</i>	200000	200000	50000
<i>Number of Transmissions</i>	187765	187767	3653
<i>Total Remaining Energy</i>	110115	110097	1906723

In the first group of tests in which the random number generator 1 (R_1) is used, a few transmissions were done by sensor nodes in TEEN because the changes in the user interested attribute were very small and the thresholds were rarely exceeded. This situation could avoid the user from having information about the status of the nodes in network. By the help of *Count Time*, sensed values were transmitted more frequently in APTEEN and the problem in TEEN did not exist any more. Moreover, simulation results show that the selected thresholds are appropriate for networks that

work in these circumstances. The trend of related attribute did not comprise sudden and great changes so TEEN and APTEEN worked properly. With our new protocol, sensor nodes made some more transmissions because as it's told above, aim of the threshold generation algorithm is to generate appropriate thresholds so that sensor nodes perform nearly same number of transmissions in each case that the related attribute changes unpredictably.

Because of the increase in the change of the sensing attribute, more transmissions were done by sensor nodes in TEEN and APTEEN in the second group of tests but for both protocols, thresholds seemed to be still applicable. As the simulation results of new protocol is observed, it can be seen that its performance was much more better than other two protocols. This shows that the algorithm was successful while handling changes in the related attribute and generating appropriate thresholds. It can be also seen that number of transmissions is not so different from the one in the first group of tests which shows the algorithm was also successful on forcing sensor nodes perform not so different number of transmissions in each case, independent from the change in the sensed attribute.

In the third group of tests, thresholds were exceeded very frequently and became a really big problem for TEEN and APTEEN. Nearly 70% of the sensed values are transmitted by sensor nodes which resulted in great energy loss. Our new protocol had again great performance on adapting changes in the related attribute and sensor nodes made not so different number of transmissions as it was in the first two group of simulations.

In the fourth and last group of tests, it can be easily observed that thresholds did not help filtering of sensed data TEEN and APTEEN. Nearly all sensed data were transmitted by the sensor nodes and final energy levels of sensor nodes were very low. As the simulation results of new protocol is observed in this group of tests, it can be seen that it greatly outperformed other two protocols. Sensor nodes conserved much energy and transmitted sensed values in reasonable and acceptable intervals.

On the other hand, APTEEN behaved like TEEN in the second, third and fourth group of tests because transmissions were performed very frequently and *Count Time* was never elapsed; consequently, number of transmissions was very near for TEEN and APTEEN in the last three groups of tests.

If Figures C.9 and C.13 are observed, it will be seen that many of the sensed values in the third and fourth group of tests exceeded the hard threshold that resulted in nodes which used TEEN and APTEEN to transmit the sensed value immediately. By the help of *Lower Bound*, transmission count of sensor nodes which used the new protocol did not increased so much and energy is preserved greatly.

These simulation results also show that the new protocol had near performance levels in each test. After a time period was skipped from the beginning, soft thresholds were started to be generated and the network was adapted to the changes in the media. The slight increase in the number of transmissions in the third and fourth group of tests was partially caused by the exceeds of hard threshold. The hard threshold was static and never changed; moreover, a serious percentage of the generated sensed values were greater than 1000, which was the static hard threshold. Although the algorithm greatly saved energy by the help of *Lower Bound*, exceeding the hard threshold many times resulted in more transmissions and energy usage. This problem could be solved by selecting the dynamic hard threshold option.

Figure 4.1 is the graphical representation of simulation results which are given in Tables 4.3, 4.4, 4.5 and 4.6 and shows the changes in the total number of transmission for each protocol as the mean value of random number varied.

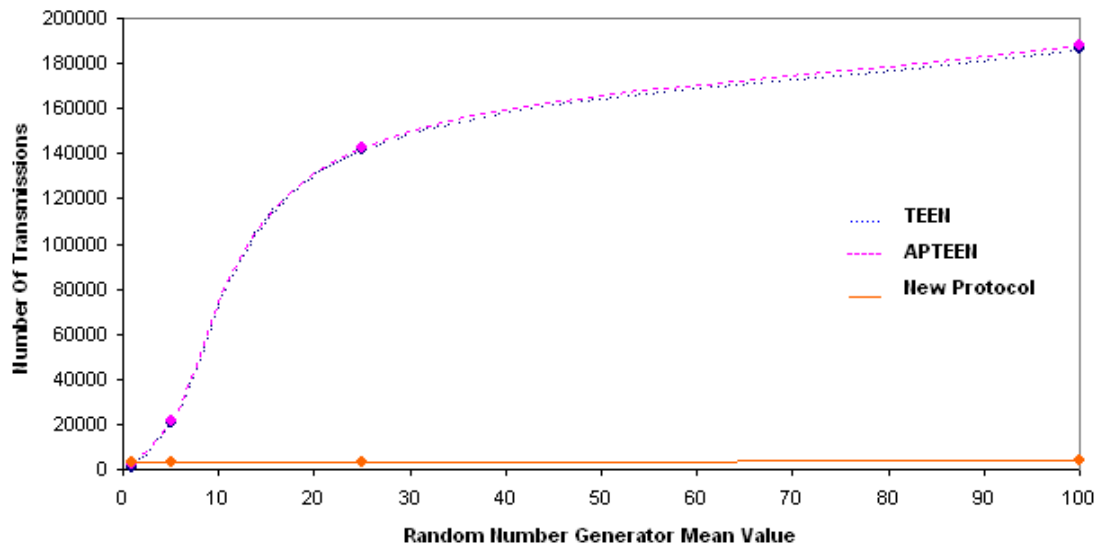


Figure 4.1. Total number of transmissions of all three protocols as the mean value of random number varied

Figure 4.17 clearly shows that number of transmissions hugely increased in TEEN and APTEEN as the related attribute changes greatly while with our new protocol, nodes perform nearly same number of transmissions in all cases.

Similarly, Figure 4.2 is the graphical representation of simulation results which are given in Tables 4.3, 4.4, 4.5 and 4.6 and shows total remaining energy of nodes in the network for each protocol as the mean value of random number varied.

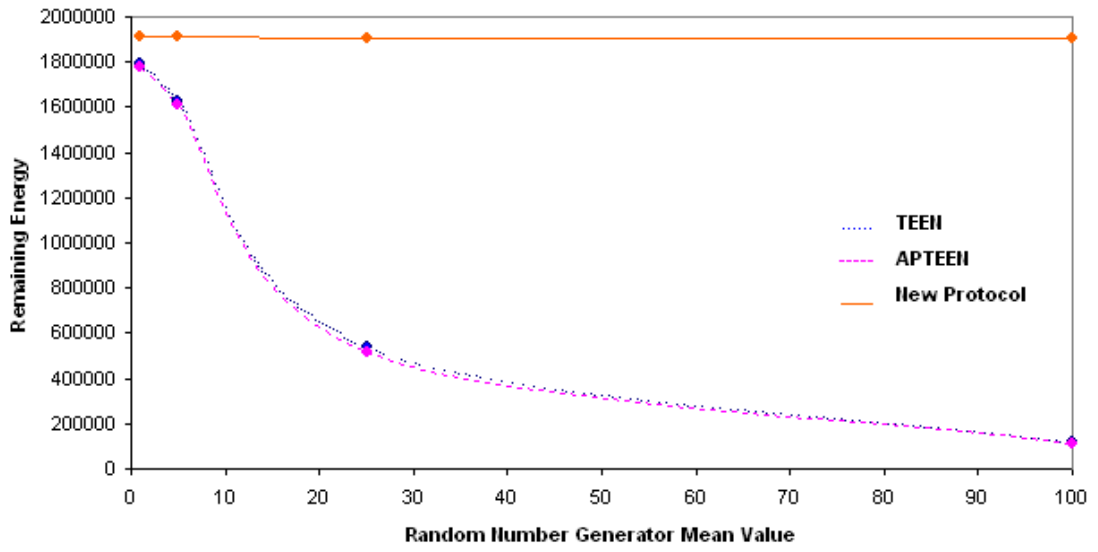


Figure 4.2. Total remaining energy of all three protocols as the mean value of random number varied

Figure 4.2 also shows that total remaining energy hugely decreased in TEEN and APTEEN as the related attribute changes greatly while with our new protocol, energy consumption is nearly the same in all cases. Moreover, total remaining energy is also greater than other protocols when the mean value of the random number generator was 1 while total number of transmissions for that value is greater in our protocol. The reason is that sensor nodes to which our protocol was applied performed less senses. Sensing results in less energy consumption than transmission but still causes energy loss. In our case, nodes which used TEEN and APTEEN performed 200,000 senses while the ones with our protocol performed only 50,000 senses. For this reason, when the random number generator mean value was 1, sensor nodes to which our protocol was applied made more transmissions but still had greater remaining energy than the others.

4.3. Second Group of Simulations

In the next group of simulations, we used the same simulation parameters but did not stop the simulation after a period of time. We ran the simulations until 90% of total energy of sensor nodes in the network were consumed. This helped us to observe and compare network lifetimes of the protocols in different circumstances. Figure 4.3 shows the results of this group of simulations.

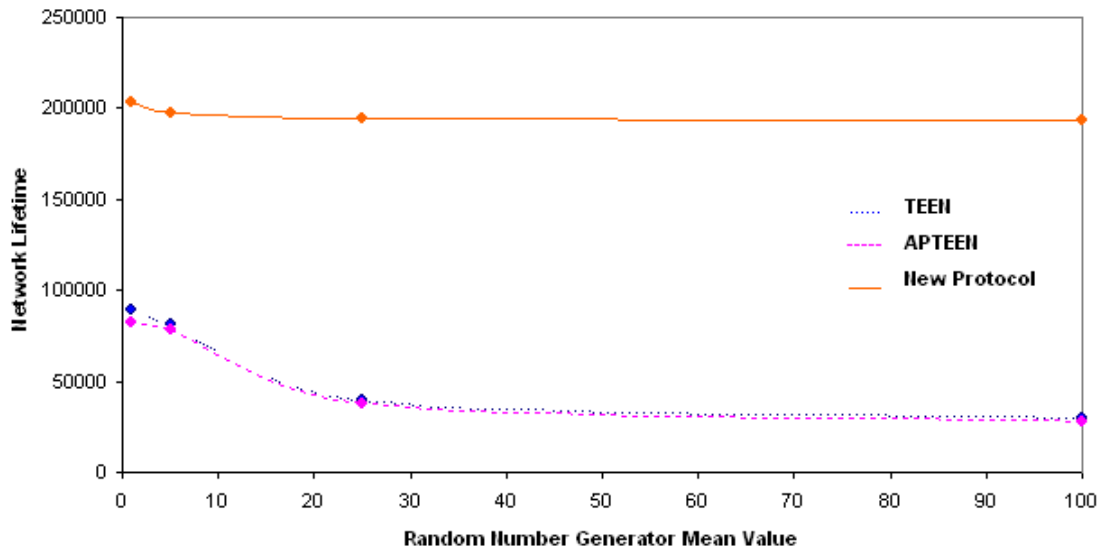


Figure 4.3. Network lifetimes of all three protocols as the mean value of random number varied

It can be easily seen that our protocol outperformed TEEN and APTEEN by the means of network lifetime. These results are expected and reasonable because energy consumption is minimized by new protocol while the other two protocols do not provide energy saving.

4.4. Third Group of Simulations

Lastly, we perform a separate simulation for our new protocol in order to observe how the number of subgroups effects the performance. Simulation parameters given in Table 4.2 were again applied, the simulation was run until 90% of total energy of sensor nodes in the network were consumed and also subgroup count was varied. Figure 4.4 shows lifetimes of the network to which our new protocol applied as subgroup counts varied. Four trends were generated for different random number

generator mean values which were used to generate the trend to simulate the change in the related attribute.

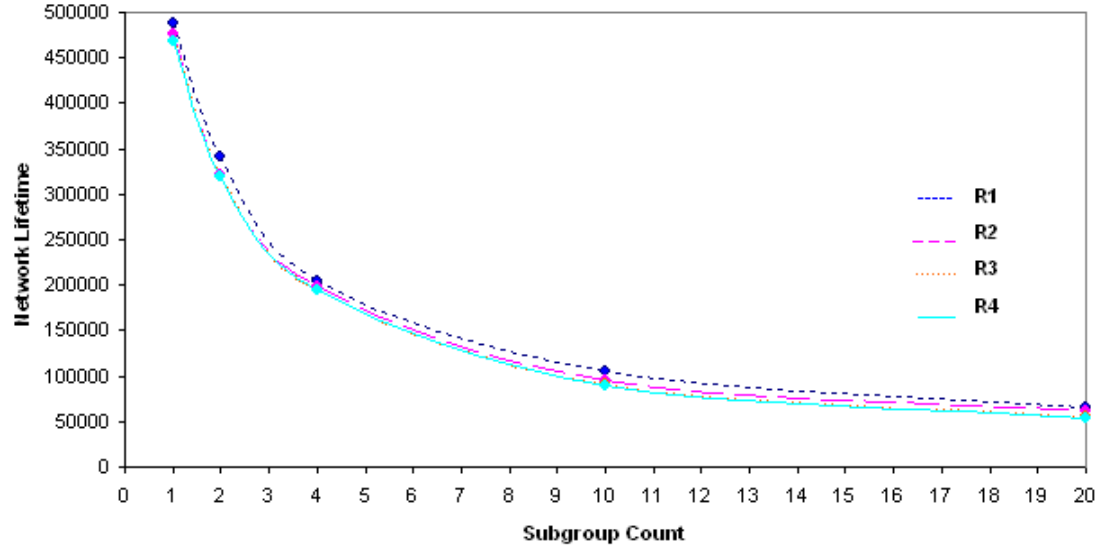


Figure 4.4. The change in the network lifetime of new protocol with four different random number generators as subgroup count in the cluster varied

As the subgroup count in the cluster increased, network lifetime decreased because of the increase in awake nodes count and total number of transmissions. In our previous simulations, we used a subgroup count of four and this provides an optimal solution. When the nodes in a sensor network are near to each other, this results in nodes to sense redundant data and transmit same events. Nevertheless, if the event to be collected is so critical for an application, the probability of missing the event by sensor nodes have to be eliminated. Consequently, if the nodes in the network are not far from each other or the related attribute to be sensed is not so critical in an application area, the subgroup count can be decreased. On the other hand, if the nodes in the network are far from each other or the related attribute to sensed is critical, the subgroup count can be increased.

CONCLUSION

In this study, we have introduced a new protocol for event-based sensor network protocols which proposes dynamic threshold generation within clusters. This technique avoids the user to set inappropriate thresholds which are used to filter the transmitted data. If the thresholds do not reach the nodes in the network, the nodes never communicate in some of the recent protocols. Moreover, thresholds might not be valid for the whole network and the change in sensed values might not behave the same in different clusters in large sensor networks. Our protocol solves these problems by generating thresholds within clusters and announcing them by cluster heads periodically. The algorithm is suitable and performance effective in networks where the amount of change in user interested attributes is unusual and thresholds can not be easily estimated. The proposed technique can increase network lifetimes by adapting the network to the changing circumstances and ensuring that sensor nodes use their energy optimally.

By forming subgroups and using a sleeping/awaking mechanism, the protocol also eliminates sensing and transmitting of redundant data which are sensed by neighbor nodes; consequently, energy consumption is minimalized while network lifetime is maximized.

Protocol provides users to manage event reliability and network lifetime by setting subgroup count in a cluster. If the nodes in the application area are near to each other or the related attribute to be sensed is not so critical in an application area, the subgroup count can be decreased which will result in an increase in the network lifetime. If the nodes in the network are far from each other or the related attribute to sensed is critical, the subgroup count can be increased which will result in a decrease in the network lifetime. The preference is fully dependent to the application type and sensor density on plant.

The major drawback of the protocol is extra overhead in order to perform the threshold generation phases, additional messaging between sensor nodes such as remaining energy level announcements and wakeup messages and additional

complexity to maintain sleep/awaking mechanism; nevertheless, the energy saved by the protocol is much greater than the energy loss that is caused by the additional mechanisms required by the new protocol.

REFERENCES

- [1] **Manjeshwar, A. and Agrawal, D.P.**, 2001. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks, *International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, California, USA, August 7-9.
- [2] **Heinzelman, W., Chandrakasan A. and Balakrishnan, H.**, 2000. Energy-Efficient Communication Protocol for Wireless Microsensor Networks, *Proceedings of the 33rd Hawaii International Conference on System Sciences*, Hawaii, USA, January 9.
- [3] **Cardell-Oliver, R., Smettem, K., Kranz, M. And Mayer K.**, 2004. Field Testing a Wireless Sensor Network for Reactive Environmental Monitoring, *Technical Report UWA-CSSE-04-003*, Crawley, Australia.
- [4] **Lindsey, S. and Raghavendra, C.**, 2002. PEGASIS: Power-Efficient Gathering in Sensor Information Systems, *IEEE Aerospace Conference Proceedings*, **3**, 1125-1130.
- [5] **Rodoplu, V. and Meng, T. H.**, 1999. Minimum Energy Mobile Wireless Networks, *IEEE Journal Selected Areas in Communications*, **8**, 1333-1334.
- [6] **Li, L. and Halpern, J. Y.**, 2001. Minimum-Energy Mobile Wireless Networks Revisited, *IEEE International Conference on Communications*, **1**, 278-283.
- [7] **Fang, Q., Zhao, F. and Guibas, L.**, 2003. Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation, *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing*, pp.165-176, ACM Press, Maryland, USA.

- [8] **Al-Karaki, J. N., Ul-Mustafa, R. and Kamal, A. E.,** 2004. Data Aggregation in Wireless Sensor Networks - Exact and Approximate Algorithms, *Proceedings of IEEE Workshop on High Performance Switching and Routing*, 2004, Phoenix, Arizona, USA, April 18-21.
- [9] **Manjeshwar, A. and Agrawal, D.P.,** 2002. APTEEN: A Hybrid Protocol for Routing and Comprehensive Information Retrieval in Wireless Sensor Networks, *International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, Fort Lauderdale, Florida, USA, April 15-19.
- [10] **Manjeshwar, A., Zeng, Q. and Agrawal, D. P.,** 2002. An Analytical Model for Information Retrieval in Wireless Sensor Networks Using Enhanced APTEEN Protocol, *IEEE Transactions on Parallel and Distributed Systems*, **12**, 1290-1302.
- [11] **Miller, M. J. and Vaidya, H.V.,** 2004. Minimizing Energy Consumption in Sensor Networks Using a Wakeup Radio, *IEEE WCNC*, Atlanta, USA, March 25.
- [12] **Begum, S., Wang S., Krishnamachari B. and Helmy, A.,** 2004. ELECTION: Energy-efficient and Low-latency Scheduling Technique for Wireless Sensor Networks, *IEEE LCN*, Tampa, Florida, USA, November 16-18.
- [13] **Bulusu, N., Heidemann J. and Estrin, D.,** 2000. GPS-less Low Cost Outdoor Localization For Very Small Devices, *IEEE Personal Communications Magazine*, **5**, 28-34.
- [14] **Huang, C., Blum M. B., Stankovic J. A. and Abdelzaher, T.,** 2003. Range-Free Localization Schemes for Large Scale Sensor Networks, *Proceedings of the 9th annual international conference on Mobile computing and networking*, **9**, 81-95.
- [15] **Niculescu, D. and Nath, B.,** 2001. Ad Hoc Positioning System (APS), *GLOBECOM '01. IEEE*, **5**, 2926-2931.
- [16] **Brooks, A., Williams S. and Makarenko, A.,** 2004. Automatic Online Localization of Nodes in an Active Sensor Network, *IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, USA, March 12.

APPENDIX A: SOFT THRESHOLD GENERATION ALGORITHM

```
integer  TE;
float    TA, CM, TS;

TS = MAX_FLOAT;

FOR k = 1 TO CSG
BEGIN
    TE = 0;

    FOR i = 1 TO SH
    BEGIN
        TE = MH[k][i+1]->TS - MH[k][i]->TS;
    END

    TA = TE / (SH - 1);

    CM = MAX_FLOAT;

    FOR i = 1 TO SH
    BEGIN
        IF CM > ABS(MH[k][i+1]->VS - MH[k][i]->VS)
        BEGIN
            CM = ABS(MH[k][i+1]->VS - MH[k][i]->VS);
        END
    END

    IF TS > ((CM * TE) / TA)
    BEGIN
        TS = (CM * TE) / TA;
    END
END
```

APPENDIX B: HARD THRESHOLD GENERATION ALGORITHM

```
float    VM, TH;

VM = MAX_FLOAT;

FOR k = 1 TO CSG
BEGIN
  FOR i = 1 TO SH
  BEGIN
    IF VM < MH[k][i]->VS
    BEGIN
      VM = MH[k][i]->VS;
    END
  END

  IF TH < ((VM * BHT) / 100)
  BEGIN
    TH = (VM * BHT) / 100;
  END
END
```

APPENDIX C: SIMULATED TRENDS AND BEHAVIOURS OF PROTOCOLS

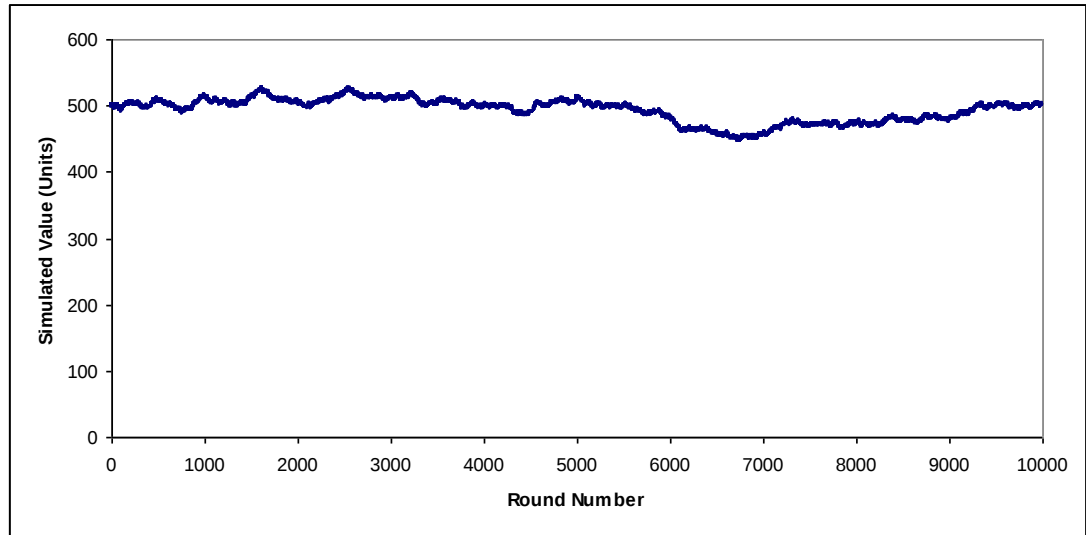


Figure C.1. Simulated values generated using R_1

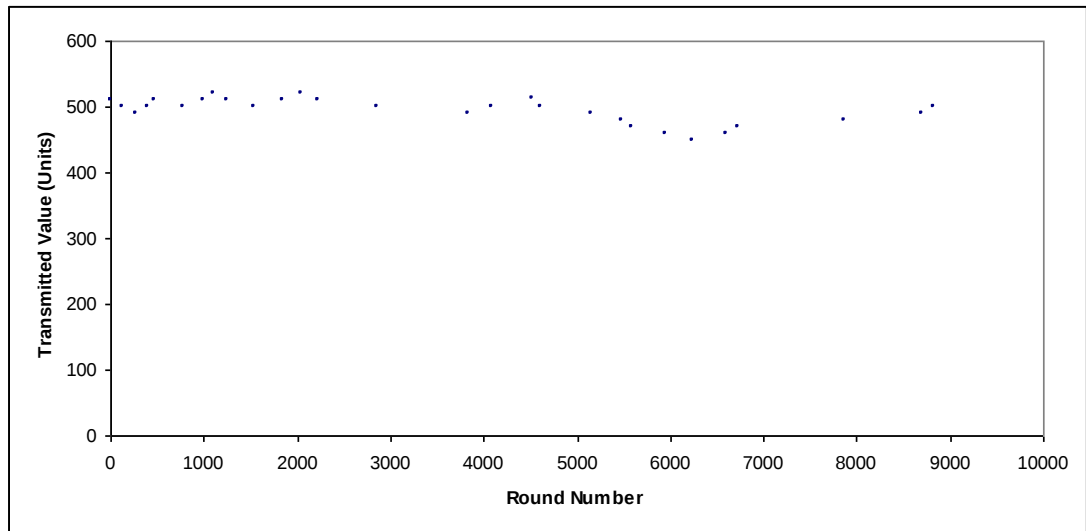


Figure C.2. Transmitted values among the ones in Figure C.1 by the sensor nodes using TEEN

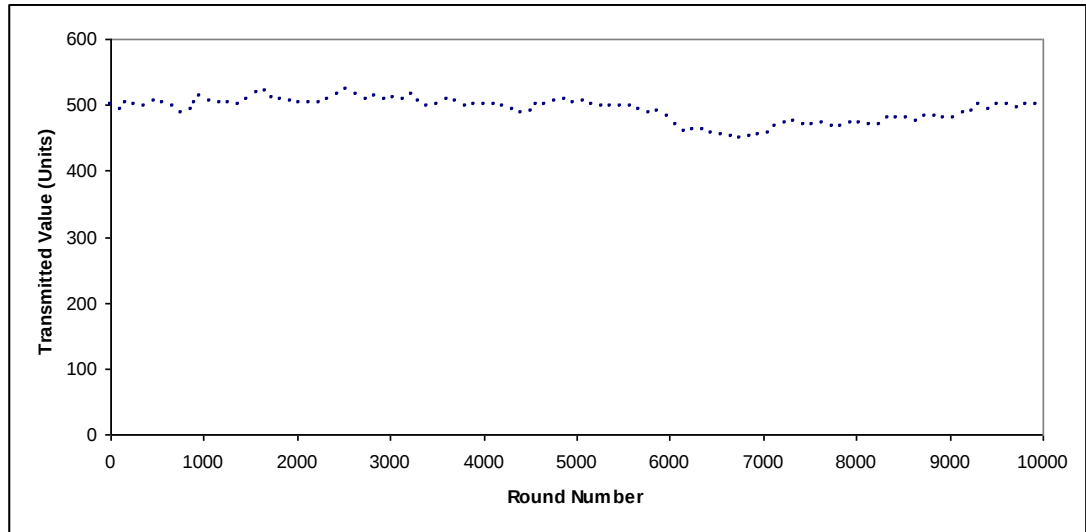


Figure C.3. Transmitted values among the ones in Figure C.1 by the sensor nodes using APTEEN

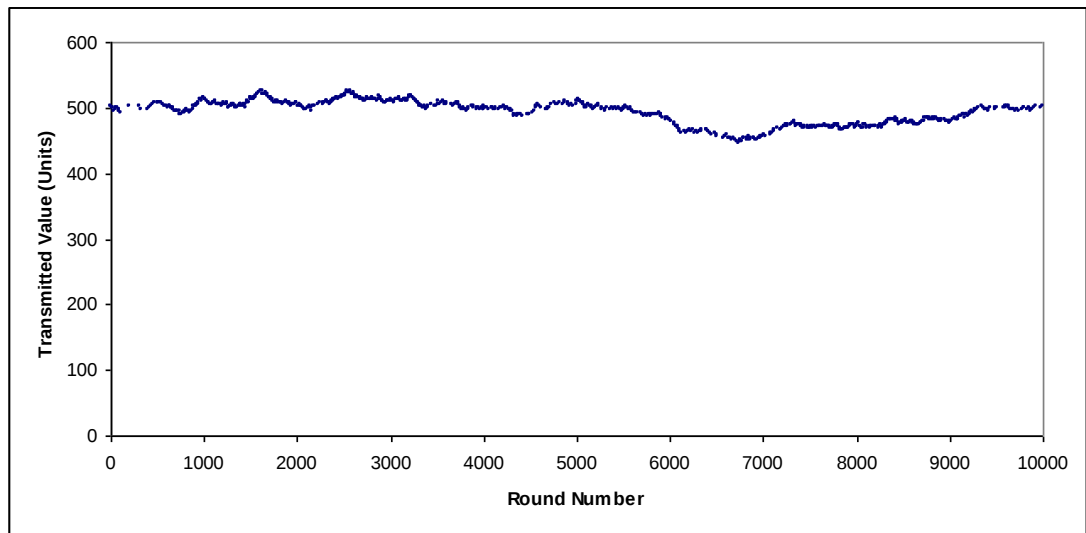


Figure C.4. Transmitted values among the ones in Figure C.1 by the a selected sensor node using the new protocol

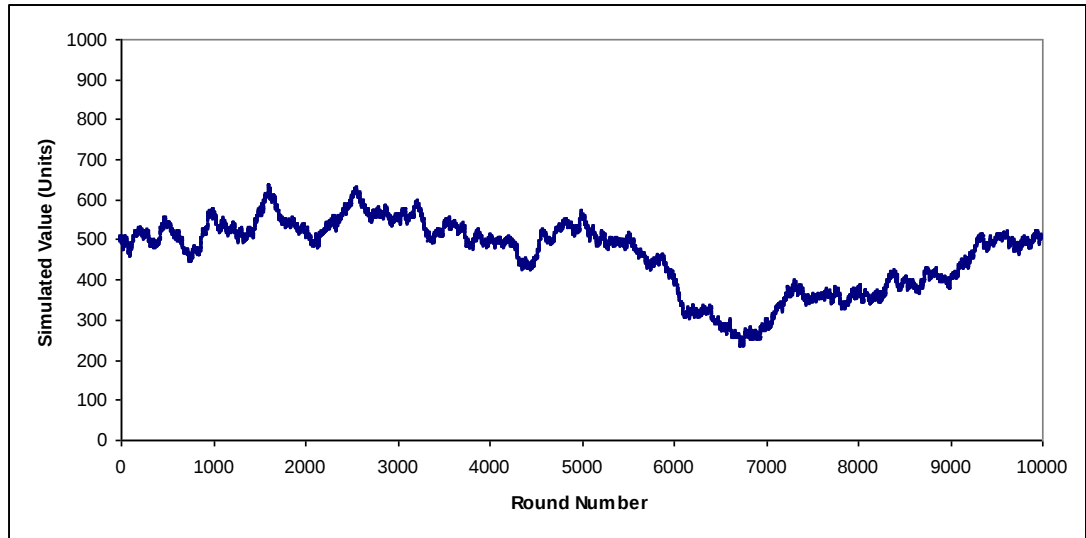


Figure C.5. Simulated values generated using R_2

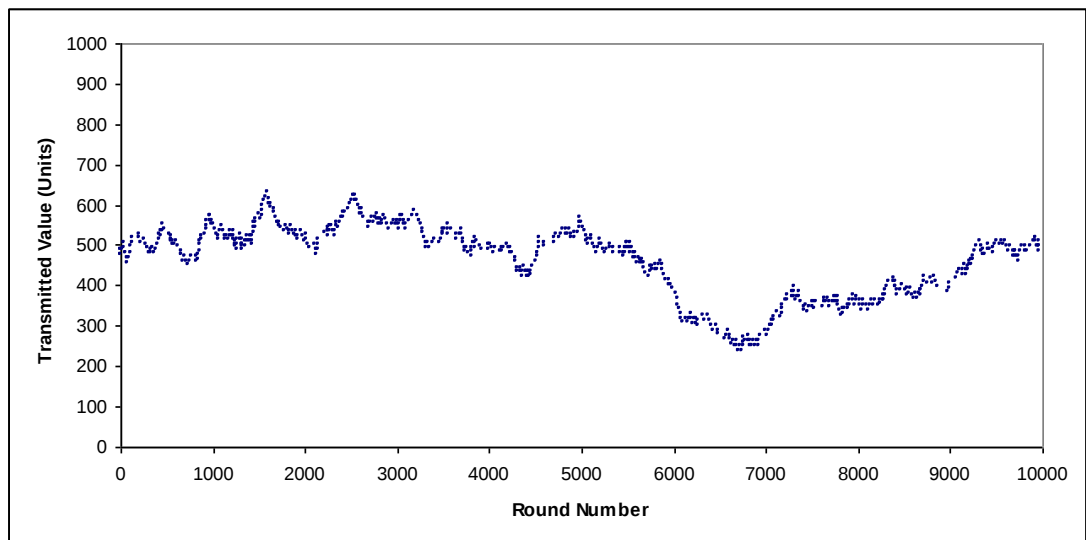


Figure C.6. Transmitted values among the ones in Figure C.5 by the sensor nodes using TEEN

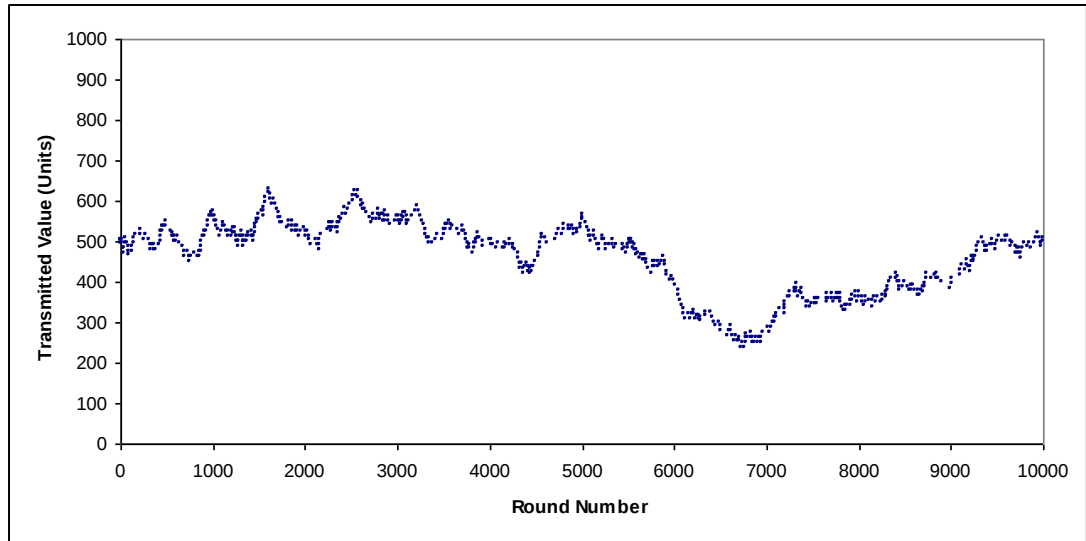


Figure C.7. Transmitted values among the ones in Figure C.5 by the sensor nodes using APTEEN

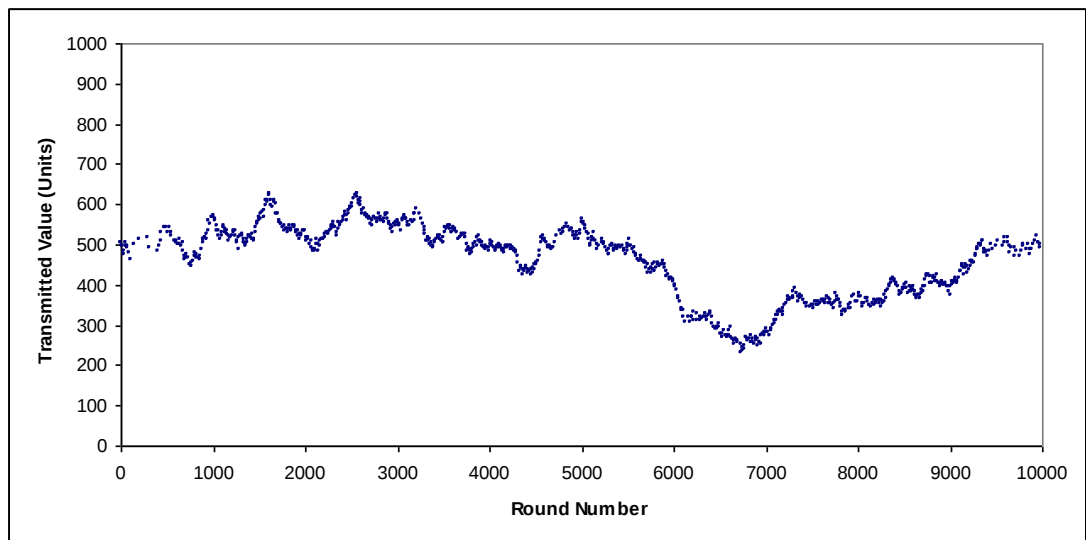


Figure C.8. Transmitted values among the ones in Figure C.5 by the a selected sensor node using the new protocol

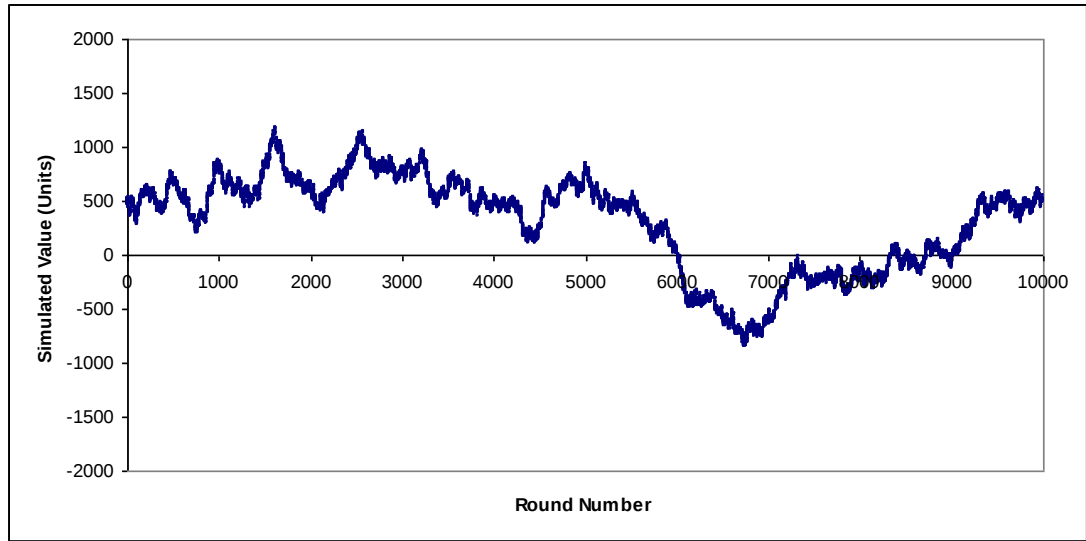


Figure C.9. Simulated values generated using R_3

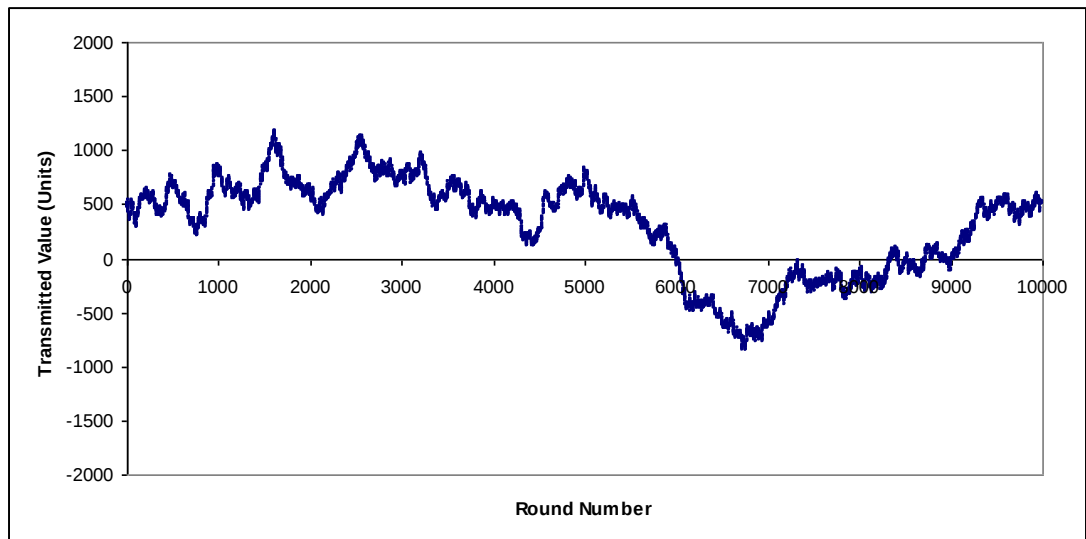


Figure C.10. Transmitted values among the ones in Figure C.9 by the sensor nodes using TEEN

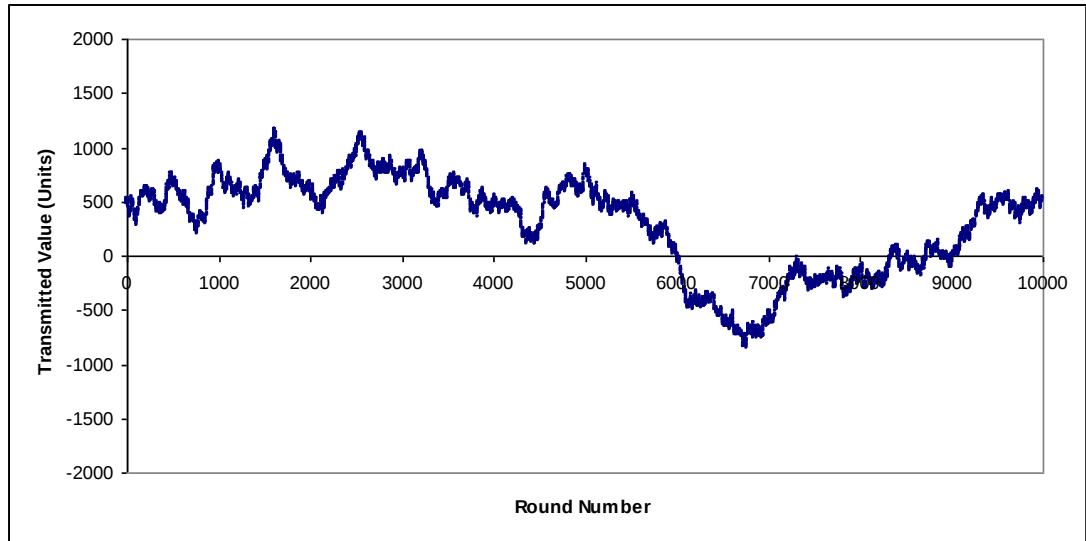


Figure C.11. Transmitted values among the ones in Figure C.9 by the sensor nodes using APTEEN

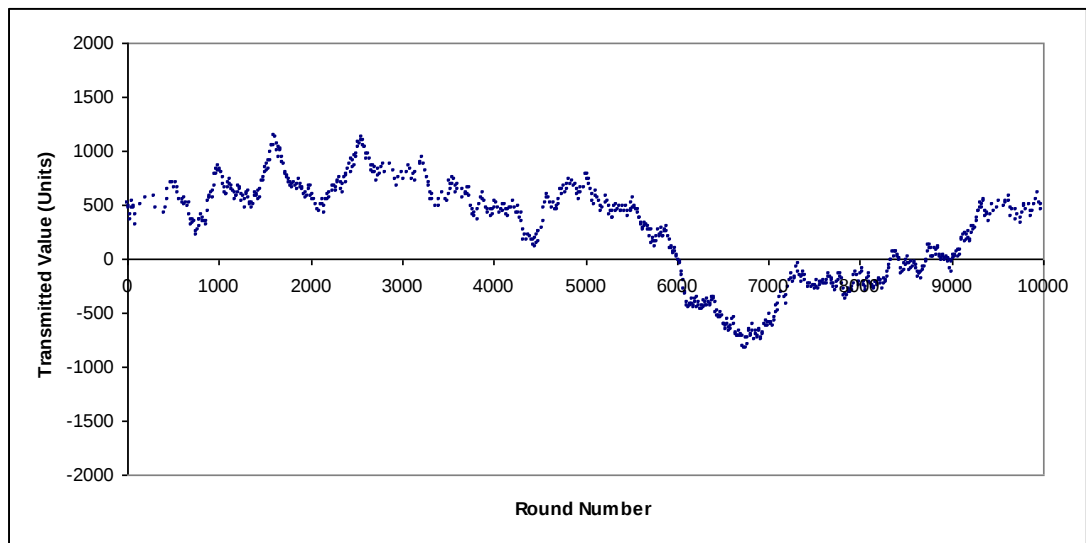


Figure C.12. Transmitted values among the ones in Figure C.9 by the a selected sensor node using the new protocol

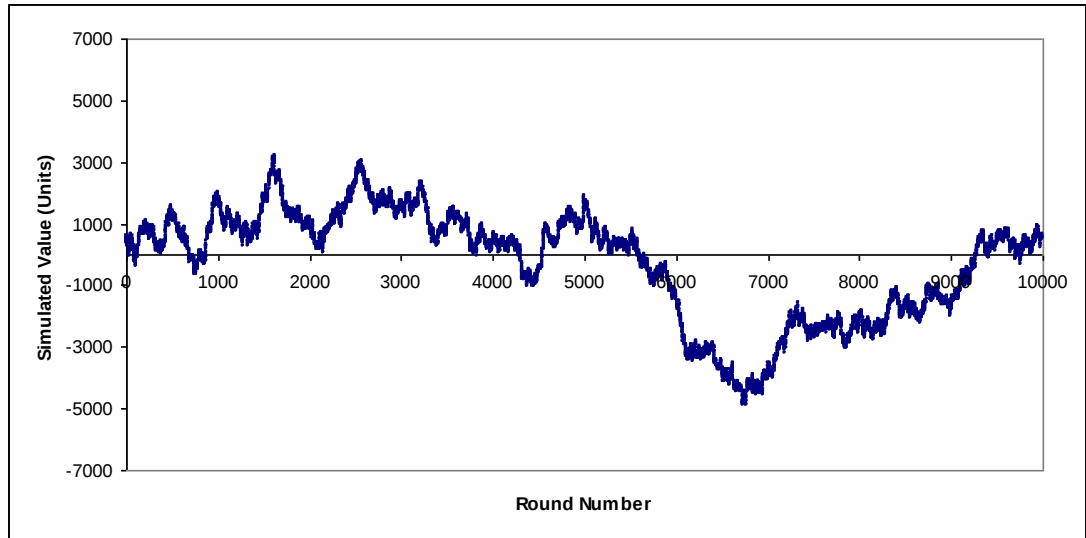


Figure C.13. Simulated values generated using R_4

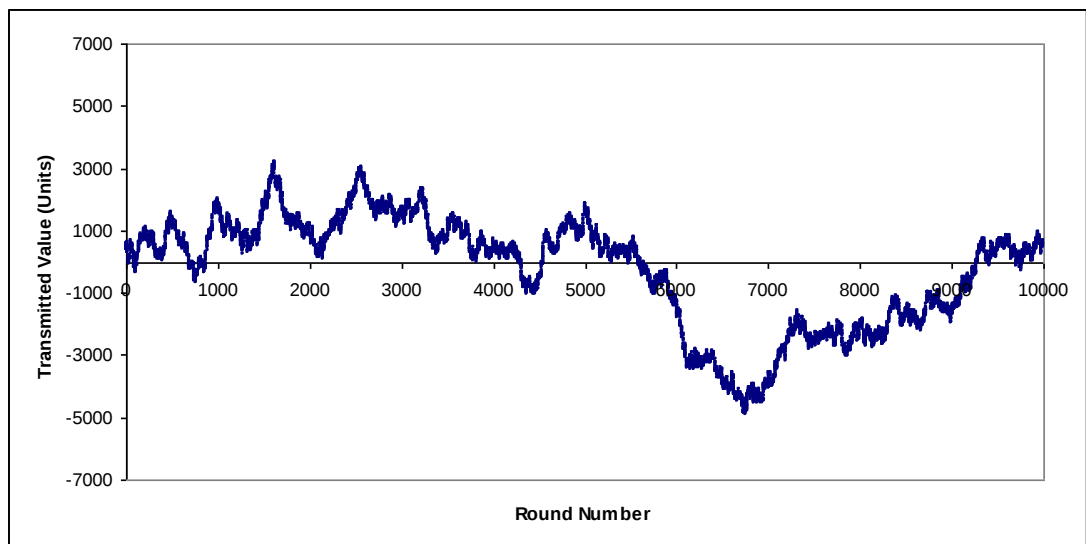


Figure C.14. Transmitted values among the ones in Figure C.13 by the sensor nodes using TEEN

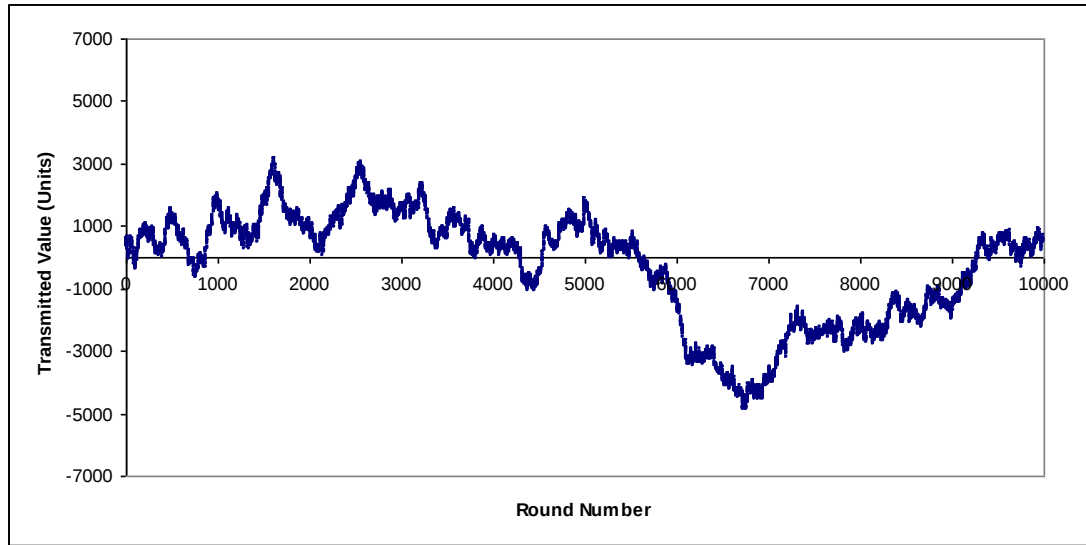


Figure C.15. Transmitted values among the ones in Figure C.13 by the sensor nodes using APTEEN

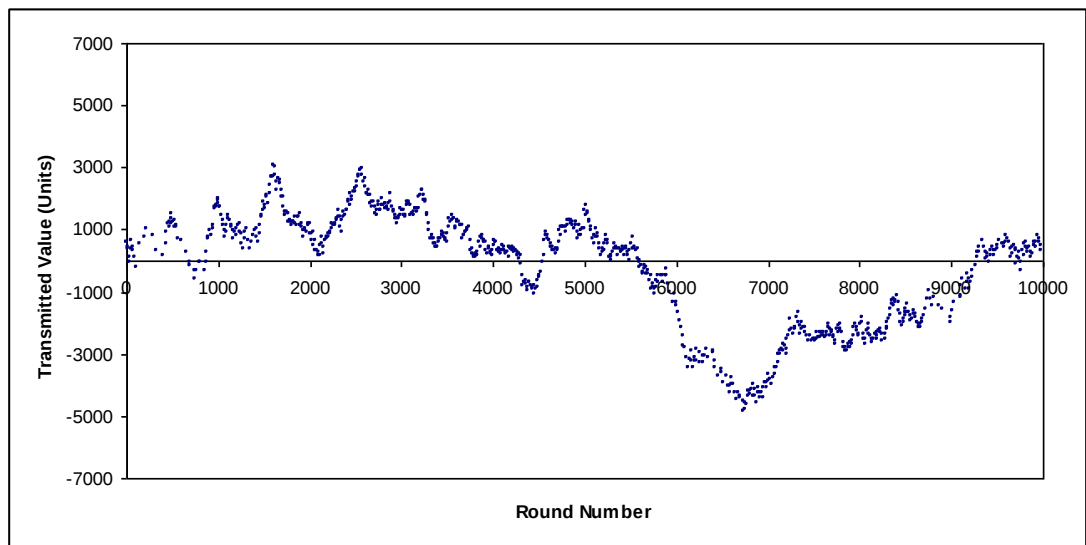


Figure C.16. Transmitted values among the ones in Figure C.13 by the a selected sensor node using the new protocol

BIOGRAPHY

Osman Korkutan was born in 15/05/1980 in Derinkuyu/Nevşehir. He graduated Mersin Science High School in 1998 and Istanbul Technical University Computer Engineering Department in 2003. He worked for Eklips Bilgi İletişim Hizmetleri from June 2003 to August 2004. He has been working for Siemens San. Ve Tic. A.Ş., in Programming and System Engineering Department as a Software Engineer since September 2004.