APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks*

Arati Manjeshwar and Dharma P. Agrawal†
Center for Distributed and Mobile Computing, ECECS Department,
University of Cincinnati, Cincinnati, OH 45221-0030
Email : {amanjesh,dpa}@ececs.uc.edu

Abstract

Wireless sensor networks with thousands of tiny sensor nodes, are expected to find wide applicability and increasing deployment in coming years, as they enable reliable monitoring and analysis of the environment. In this paper, we propose a hybrid routing protocol (APTEEN) which allows for comprehensive information retrieval. The nodes in such a network not only react to time-critical situations, but also give an overall picture of the network at periodic intervals in a very energy efficient manner. Such a network enables the user to request past, present and future data from the network in the form of historical, one-time and persistent queries respectively. We evaluated the performance of these protocols and observe that these protocols are observed to outperform existing protocols in terms of energy consumption and longevity of the network.

1. Introduction

The advancement in sensor technology has made it possible to have extremely small, low powered sensing devices equipped with programmable computing, multiple parameter sensing and wireless communication capability. Also, the low cost makes it possible to have a network of hundreds or thousands of these sensors, thereby enhancing the reliability and accuracy of data and the area coverage. Wireless sensor networks offer information about remote structures, wide-spread environmental changes, etc. in unknown and inhospitable terrains.

There are a number of advantages of wireless sensor networks over wired ones such as ease of deployment (reducing installation cost), extended range (network of tiny sensors can be distributed over a wider region), fault-tolerance (failure of one node does not affect the network operation), self-organization (the nodes can have the capability to reconfigure themselves) But there are a few inherent limitations of wireless media such as low bandwidth, error prone transmissions, collision free channel access requirements etc. Also, since the wireless nodes are mostly mobile and are not connected in any way to a constant power supply, they derive energy from a personal battery. This limits the amount of energy available to the nodes. In addition, since these sensor nodes are deployed in places where it is difficult to either replace individual nodes or their batteries, it is desirable to increase the longevity of the network and preferable that all the nodes die together so that the whole area could be replenished by a new set of tiny nodes. Finding individual dead nodes and then replacing those nodes selectively would require pre-planned deployment and eliminate some advantages of these networks.

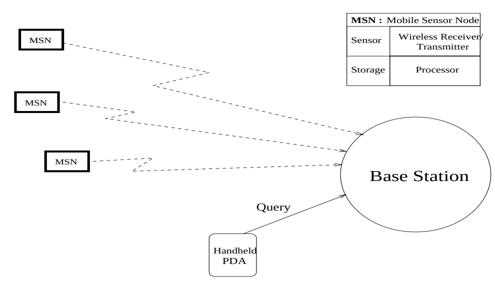

Figure 1. *A model of a Sensor Network System.*

A model of such a sensor network, is shown in Figure 1. Each tiny sensor has a sensing module, a computing module, memory and a wireless communication module with a limited radio range and hence constituting a multi-hop MANET. The only difference here is the presence of a powerful Base Station (BS), which can directly access any or all sensors in the region as well as has adequate storage capacity to hold the data from the sensors. The user would expect to be able to query the network through the

*This work is supported by the Ohio Board of Regents' Doctoral Enhancement Funds
†Responsible for all communications.

BS. Consider the following scenario: Temperature sensors are placed around a factory (such as chemical, automotive, etc.) Typical queries posed by the user include:

- Report immediately if the temperature in north-east quadrant goes below $40°F$
- Retrieve the average temperature in southern quadrant over the last 5 hours
- For the next two hours report if the temperature goes beyond $200°F$.
- Which areas had a temperature between $40°F$ and $200°F$ in the past two hours.

In general, user queries can be broadly categorized into three types:

1. *Historical queries*: This type of query is mainly used for analysis of historical data stored at the BS (base station). For example, "What was the temperature 2 hours back in the northwest quadrant ?"
2. *One-time query*: This type of query gives a snapshot view of the network. For example, "What is the temperature in the northwest quadrant?"
3. *Persistent*: This type of query is mainly used to monitor a network over a time interval with respect to some parameters. For example, "Report the temperature in the northwest quadrant for the next 2 hours".

The protocol should enable strategic distribution of energy dissipation, which in turn increases the overall lifetime of the system. In addition, slightly longer latency for non-critical data is acceptable if that helps increasing node's life. However, queries for time critical data should not be delayed and should be handled immediately.

Traditional routing protocols defined for MANETs are not well suited for wireless sensor networks as mentioned in our earlier paper [12]

An ideal sensor network should have *Attribute based addressing*, and *location awareness*. Another important requirement in some cases is that the sensors should react immediately to drastic changes in their environment, for example, in *time-critical applications*. The end user should be made aware of the ground situation with minimum delay while making efficient use of the limited wireless channel bandwidth. Thus, wireless sensor network needs protocols which are data centric, capable of effective data aggregation, distribute energy dissipation evenly, efficiently use their limited energy to increase the longevity of the network and avoid any single point bottleneck (except the BS).

2. Motivation

An energy-efficient communication protocol LEACH, has been introduced [7] recently which employs a hierarchical clustering done based on information received by the BS. The BS periodically changes both the cluster membership and the cluster-head (CH) to conserve energy. The CH collects and aggregates information from sensors in its own cluster and passes on information to the BS. By rotating the cluster-head randomly, energy consumption is expected to be uniformly distributed. Otherwise, the CHs closest to the BS, end up transmitting majority of data and drainage of power could force them to die much earlier than other nodes. If a CH, for some reason, cannot communicate with its cluster members or the BS, then periodic re-clustering by BS, would enable selection of another active node as the CH. Details of how to form a cluster and how to select a CH for each cluster have been covered in [9] and we assume a similar scheme.

The main problem we see is how to process user's query and how to route needed information. Most current protocols [7] assume a sensor network collecting data periodically from its environment and then respond to a query when it arrives. In LEACH [9], sensed data is sent to CHs periodically, and after aggregation, data is passed on to the BS for storing the information. No particular attention has been given to the time criticality of the target application in sensor networks. Sensor networks should also provide the end user with the ability to control the trade-off between energy efficiency, accuracy and response times dynamically. In our research, we have focussed on developing an efficient routing protocol and a comprehensive query handling mechanism which can best fulfill these needs.

3. Query Handling

The two ways of handling queries are:

- The sensor nodes send a pre-defined set of data regularly to a centralized site (BS) and is stored in a database. The user queries this centralized system, known as the *warehousing* approach [1]. An obvious drawback of this method is that data is sent always, the critical data has to be extracted from the database.
- When a user sends any query, the data satisfying the query is collected on demand. The main drawback of such a method is the unacceptable delay for the queries concerning time critical data.

We need a mechanism where the BS always possess time-critical data so that the queries about such data are not delayed. When a user wants an answer to a non-critical query and BS does not has the BS can send the query to the sensor nodes directly. Therefore we need protocol where nodes not only react to time-critical situations but also send periodic information.

4. Hybrid Networks

In an earlier paper [12], we have described a classification methodology for sensor networks based on their mode of functioning and type of target applications as:

- *Proactive Networks*: The nodes in this network periodically switch on their sensors and transmitters, sense the environment and transmit the data of interest and is employed in *LEACH* [8].
- *Reactive Networks*: In this scheme the nodes react immediately to sudden changes in the value of a sensed attribute beyond a pre-determined threshold value and are well suited for time critical applications as used in *TEEN* [12].

However, both methods have their limitations. In reactive networks, if the thresholds are not reached, the nodes will not communicate and the user will never get any data from the network at all, or will not come to know even if all the nodes die. We propose to combine the best features of proactive and reactive networks by creating a *Hybrid network* with that sends data periodically, as well as responds to sudden changes in attribute values. In section 6, we introduce a protocol for hybrid networks, called *APTEEN*.

5. Sensor Network Model

These tiny sensor nodes have limited energy and memory constraints, and routing protocols that could possibly reduce the routing complexity are desirable. One way of achieving this is to use a topology different from a conventional flat topology and assign the routing responsibilities to just a few nodes and rotate this periodically.In this section, we give a brief introduction to the sensor network model on which we have based our protocols.

We assume that all the nodes in the network are homogeneous and begin with the same initial energy. The BS has adequate power to transmit directly to the sensor nodes, providing a direct path for the down-link. However, the sensor nodes cannot always do this because of their limited power supply, leading to an asymmetric communication. This stringent energy constraints, makes hierarchical clustering to be the most suitable model for Wireless Sensor networks.

The nodes of Figure 3 are grouped into clusters (for example, nodes 1.1.1, 1.1.2, 1.1.3, 1.1.4, 1.1.5 and 1.1) with each cluster having a cluster head (node 1.1 for the example cluster). This cluster head aggregates all the data sent

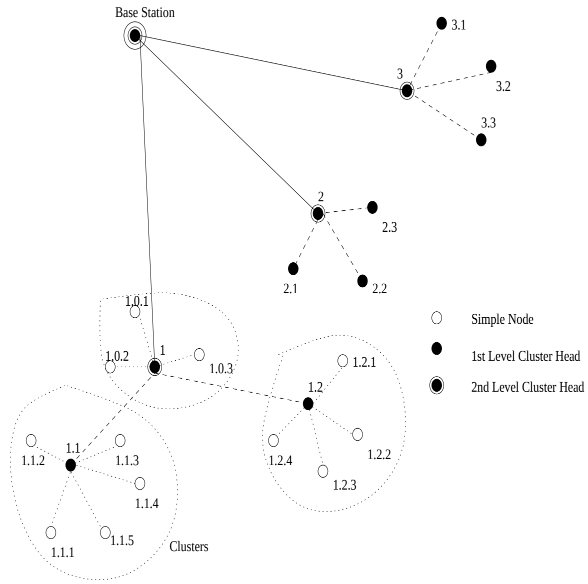

Figure 2. *Hierarchical Clustering*

to it by all its members and forwards it to its upper level cluster head (node 1) and so on till the data reaches the BS. Since the CHs perform functions that consume more energy, and to evenly distribute energy consumption, clusters exist for an interval called the *cluster period* T, and then BS regroups clusters. This happens at a time called the *cluster change time*. The main features of such an architecture are:

- All the nodes need to transmit only to their immediate cluster-head, thus saving energy.
- Only the cluster head needs to perform additional computations on the data such as aggregation, etc. So, energy is conserved.
- The cluster members of a cluster are mostly adjacent to each other and sense similar data and are aggregated by the CH.
- CHs at increasing levels in the hierarchy need to transmit data over relatively larger distances. To distribute this consumption evenly, all nodes take turns becoming the CH.
- Since only the CHs need to know how to route the data towards its higher level CH or the BS, it reduces its routing complexity.

Many protocols have been proposed in literature which use such a hierarchical clustering scheme such as CBPR [11], Scalable Coordination in Wireless Networks [5], LEACH [8] and any of these clustering techniques is appropriate. We have used the second version of LEACH, *leach-c* wherein clusters are formed by the BS based on the information received about sensor's energy and location by the sensors at the end of the cluster change period. In *leach*, the clustering is done by sensor nodes themselves. Even though the performance depends on how far the BS is from the CHs, we found this most suitable for our protocol due to the following reasons:

- Since BS decides the cluster heads, it can appoint a fixed number of nodes as cluster heads.(viz. 5% nodes used as CH in *leach* and *leach-c*.

- Since BS has global information of the network, it can optimally form clusters and evenly distribute the number of nodes in each cluster.

6. Hybrid Network Protocol: APTEEN

In this section, we introduce a new protocol developed for hybrid networks, called *APTEEN (**A**daptive **P**eriodic **T**hreshold-sensitive **E**nergy **E**fficient Sensor **N**etwork Protocol)*. In APTEEN once the CHs are decided, in each cluster period, the cluster head first broadcasts the following parameters:

***Attributes(A)*:** This is a set of physical parameters which the user is interested in obtaining data about.

***Thresholds*:** This parameter consists of a hard threshold (H_T) and a soft threshold (S_T). H_T is a particular value of an attribute beyond which a node can be triggered to transmit data. S_T is a small change in the value of an attribute which can trigger a node to transmit data again.

***Schedule*:** This is a TDMA schedule similar to the one used in [8], assigning a slot to each node.

***Count Time(T_C)*:** It is the maximum time period between two successive reports sent by a node. It can be a multiple of the TDMA schedule length and it accounts for the proactive component.

In a sensor network, close-by nodes fall in the same cluster, sense similar data and try to send their data simultaneously, causing possible collisions. We introduce a TDMA schedule such that each node in the cluster is assigned a transmission slot, as shown in Fig. 3. In the following section, we refer to data values exceeding the threshold value as critical data.

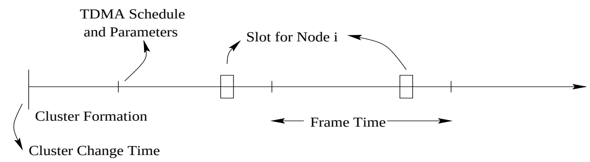

Figure 3. *Time Line for APTEEN*

6.1. Important Features

The main features of our scheme are :

1. By sending periodic data, it gives the user a complete picture of the network. It also responds immediately to drastic changes, thus making it responsive to time critical situations. Thus, It combines both proactive and reactive policies.

2. It offers a flexibility of allowing the user to set the time interval (T_C) and the threshold values for the attributes.

3. Energy consumption can be controlled by the count time and the threshold values.

4. The hybrid network can emulate a proactive network or a reactive network, by suitably setting the count time and the threshold values.

The main drawback of this scheme is the additional complexity required to implement the threshold functions and the count time. However, this is a reasonable trade-off and provides additional flexibility and versatility.

6.2. Query Modeling

To handle queries efficiently in a network, with hundreds and thousands of sensors, we could consider two possible alternatives of a flat topology and a cluster-based approach. In a flat topology, each node satisfying the query conditions has to individually send the data to the requesting node. At best, some intermediate nodes may do some aggregation, as shown in Fig 4. In a hierarchical cluster, only the CH needs to aggregate and so it seems more efficient. This is the scheme used here. If we assume that adjacent nodes

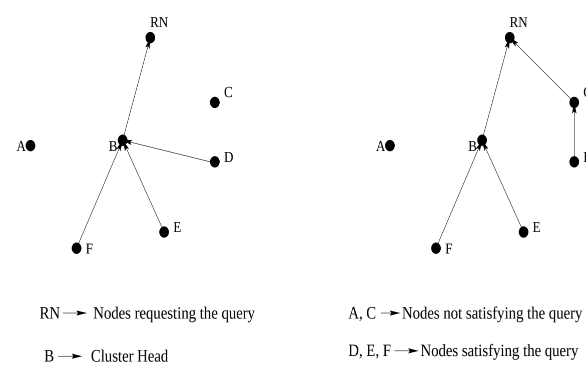

Figure 4. *Comparison of query routing topologies*

can sense similar data, we can form pairs of two nodes and

Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'02)
1530-2075/02 $17.00 © 2002 **IEEE**

make only one node from each pair respond to a query. The other node can go to a "sleep" mode and need not receive the query. Thus, two nodes can alternately take the role of handling queries if there are nodes close enough to form pairs.

6.3. Modified TDMA Schedule

A best possible pairing of sleeping and idle nodes can be found by the BS using simulated annealing. The nodes which listen for the queries have to be always awake (i.e., in idle state ready to receive any query). Also, these idle nodes will have more data to send if they receive queries, since they might have to send data as well as the queries. Hence, the slots for these idle nodes have to be larger than the slots for the sleeping nodes. By modifying the TDMA schedule, we can have the sleeping nodes send their data first and then the idle nodes. For example, if adjacent node *a* and node *b* constitute sleep/idle pair, they will have their slots at an average distance of half the frame time. So, even though the interval between two successive slots of node *a* is larger because of larger slots for idle nodes, the critical data can still be sensed and transmitted by node *b* without having to wait for node *a*'s next slot. The nodes can change their roles midway between cluster change times, so that sleeping nodes now go into idle mode to handle queries and the idle nodes now go into sleep mode.

The CH aggregates all the data and sends it to its higher level CH (or the BS). Once the BS receives the data from all the CHs, it extracts the queries and the answers from the data and transmits them in down-link mode, directly to the sensor nodes or the user rather than going through the CHs. Different CDMA code is used in each cluster to avoid inter-cluster collision. However, a common CDMA code is employed for the up-link from the cluster heads to the BS and the down-link from the BS to all sensor nodes.

This implies that the BS should not transmit to the nodes when the nodes are transmitting data to their CHs in their slots. So, we need to assign a separate slot for the BS and include it in the TDMA schedule. However, each cluster might have different number of members, leading to different TDMA frame lengths. So, the BS has to calculate the length of the longest TDMA schedule among the clusters and make allowance for the transmitted data from the CHs to reach it, after which it can transmit its own data. Finally, incorporating all these factors, a TDMA schedule can be defined as shown in Fig. 5.

7. Query Routing

Most current protocols developed for queries are suitable for only one of the types of queries. Our model can handle

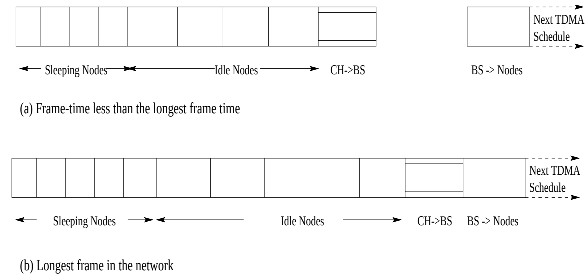

Figure 5. *Different Frame lengths in a network*

all the three types. To our knowledge, this is the first protocol which handles all types of queries efficiently.

Historical Query

The format of the query is as follows:

type 0 // type of query
temp -1 // -1 for "?"
location northwest quadrant
time 120 //in minutes

The node that receives this query transmits it to its CH in its slot. The CH aggregates all the data and transmits it to the BS at the end of the schedule. The BS checks the query type and retrieves the answer from its memory and sends the answer to the nodes directly in its down-link slot. So

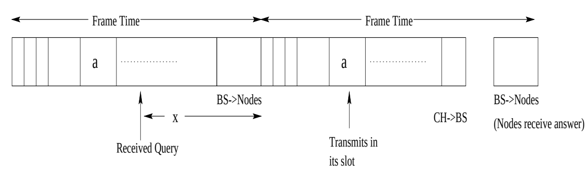

Figure 6. *Handling of History queries*

the node gets the answer to its query in a minimum of x and a maximum of x+*frame-time* where x is the time interval between the arrival of the query and the end of that frame. This is illustrated in Fig. 8.

One-time Query

The format of the query is as follows:

type 1 // type of query
temp -1 // -1 for "?"
location northwest quadrant

The node sends the query to its CH in its slot and the BS receives it at the end of the schedule. If the query is about time critical data, the BS already has this data from the nodes and so it answers such queries immediately in its down-link slot in the same frame. For other queries of this type for which the BS does not have data, it broadcasts the queries to the nodes in its slot. All the idle nodes that satisfy the queries send the data in their slot to the BS via their CHs. The BS station aggregates all the received data and broadcasts the answer. So, if the query is about data which

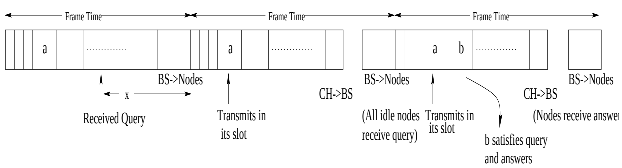

Figure 7. *Handling of One-time queries*

the BS can answer immediately, the delay is same as that for the history queries. For other one-time queries, the response time will be between x+*frame-time* and $x + 2*$*frame-time* and is shown in Fig. 7.

Persistent Query

The format of the query is as follows:

type 2 // type of query
temp -1 // -1 for "?"
location northwest quadrant
time 120 //in minutes

This type of query is handled almost exactly as the one-time query. The initial delay is the same as that of the one-time query, and then the delay is one frame-time for the duration of the query.

8. Performance Evaluation

8.1. Simulation Environment

We have based the implementation of the queries on the *ns-2* [13] simulator with the *LEACH* extension. The simulation has been performed on a network of 100 nodes and a fixed base station. The nodes are placed randomly in the network. All the nodes start with an initial energy of 2J. Cluster formation is done as in the *leach-c* protocol [8] [9]. However, their radio model is modified to include idle time power dissipation (set equal to 10% of the radio electronics energy) and sensing power dissipation (set equal to 10% of the idle energy). For our experiments, we simulated an environment with varying temperature in different regions. The sensor network nodes are first placed randomly in a bounding area of 100x100 units. The actual area covered by the network is then divided into four quadrants. Each quadrant is later assigned a random temperature between $0°F$ and $200°F$ every 5 seconds during the simulations. It is observed that most of the clusters have been well distributed over the four quadrants.

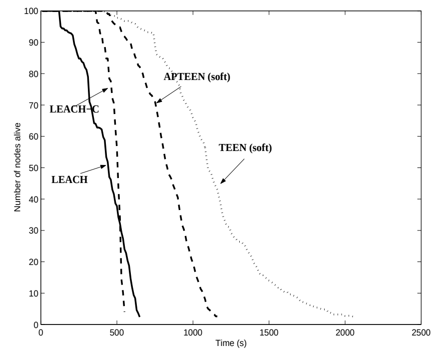

Figure 8. *Comparison of the no. of nodes alive for LEACH, APTEEN and TEEN*

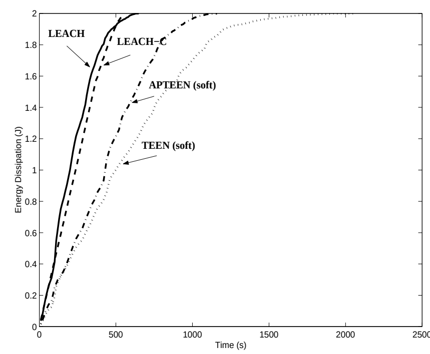

Figure 9. *Comparison of average energy dissipation for LEACH, APTEEN and TEEN*

8.2. Query Generation

For our experiments we assume a *Poisson arrival* process for the arrival of queries at each node, with a mean rate of λ. The type of query (0,1,2) is picked randomly and the duration (for types 0,2) of the query and the location of interest are also decided randomly.

8.3. Experiments

To analyze and compare the effect of queries on our protocol, we use the following metrics:

- *Average energy dissipated*: This metric shows the average dissipation of energy per node in the network a
- *Total number of nodes alive*: This metric indicates the overall lifetime of the network. More importantly, it gives an idea of the area coverage of the network over time.
- *Total number of data signals received at BS*: This metric explains how our protocol is saving energy by not

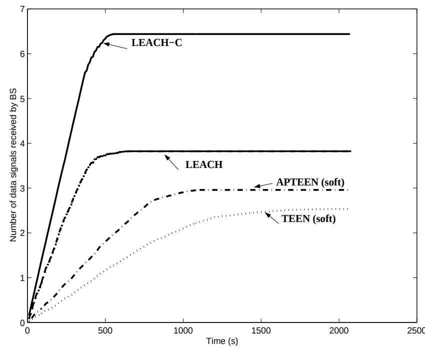

Figure 10. *Total data received at the BS over time*

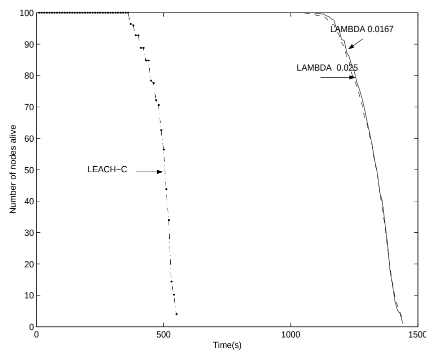

Figure 11. *Effect of queries on APTEEN*

transmitting data continuously, which is not required (neither time-critical nor satisfying any query).

- *Average Delay*: This metric gives the average response time in answering different types of queries.

For all our experiments, the attribute to be sensed is the temperature. The performance of *APTEEN* is studied in the *Soft mode* using both the thresholds. In this mode, as discussed in section 6, once a node senses a value beyond H_T, it next transmits data only when the current sensed value differs from the previous transmitted value by an amount equal to or greater than the soft threshold S_T. The hard threshold is set at $100°F$, the average of the highest and the lowest possible temperatures. The soft threshold was arbitrarily assigned a value of $2°F$ for our experiments. The count time is set equal to 5 times the frame-time. In experiments involving queries, mean arrival rate of the queries at each node, λ, is set at 0.01 and increased gradually to 1.0

8.4. Results

We have simulated different protocols and we observe that our protocol provides lower dissipation value of energy and a higher number of alive nodes at any given time.

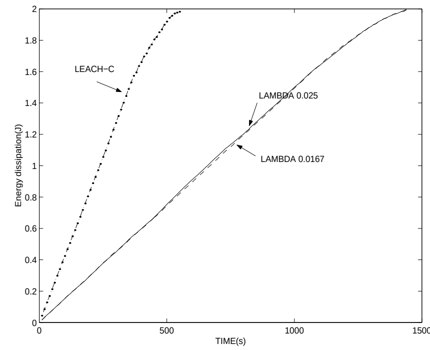

Figure 12. *Effect of queries on energy consumption in APTEEN*

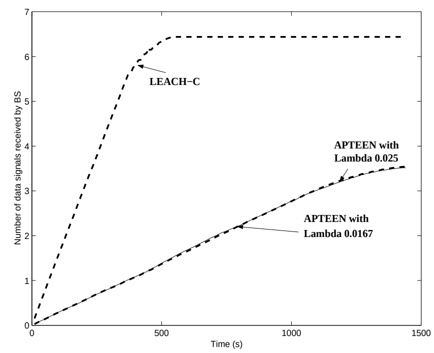

Figure 13. *Effect of queries on the total data received*

Figures 8, 9 and 10 compare APTEEN with TEEN and LEACH (*leach* and *leach-c*) with respect to energy consumption, number of nodes alive and total data signals received at the BS over time, respectively. The performance of APTEEN lies between TEEN and LEACH with respect to energy consumption and longevity of the network. This is expected as *TEEN* only transmits time-critical data while sensing the environment continuously. To overcome the drawbacks of *TEEN* we incorporated the periodic data transmission to form *APTEEN*. APTEEN performs better than LEACH since APTEEN transmits data based on the threshold values unlike LEACH which transmits data at all times. So, based on the application and the energy constraints, we can decide how to select the parameters in our APTEEN protocol. But this energy saving does increase the response time for the queries.

In *leach-c* the queries can be directly asked from the BS and answers are also received directly. Virtually no routing of query is required. For *APTEEN* with query, the delay depends on the frame time. Fig. 14 gives the average delay over λ. As λ increases, the load increases and as expected

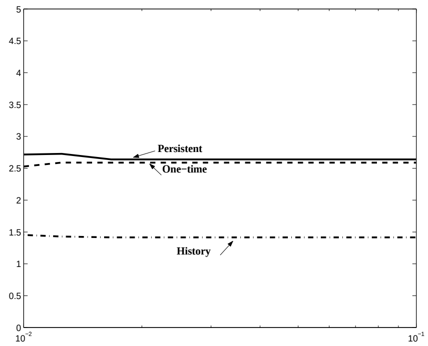

Figure 14. *Variation in response time with* λ

in a TDMA scheme the delay remains almost constant. The discrepancy we see at $\lambda = 0.01$ for persistent queries is due to the fact that a percentage of nodes sleep (taking turns), and it is possible that the nodes that could respond to the query, are asleep at that time. If it had been a one-time query the query would have gone unanswered. Since it is a persistent query, it is repeated by the BS for a duration of the query, and if the nodes satisfying the query wake up, they will send an answer. This delay in response is acceptable as this will not happen to any query requesting time critical data.

9. Related Work

Intanagonwiwat et. al. [10] have introduced a data dissemination paradigm called *directed diffusion* for sensor networks. It is a data-centric paradigm and its application to query dissemination and processing was demonstrated in this work. Estrin et. al. [5] discuss a hierarchical clustering method with emphasis on localized behavior and the need for asymmetric communication and energy conservation in sensor networks.

A cluster based routing protocol (CBRP) has been proposed by Jiang et. al in [11] for MANETs. It divides the network nodes into a number of overlapping or disjoint two-hop-diameter clusters in a distributed manner. Heinzelman et. al. [8] introduce a hierarchical clustering algorithm for sensor networks, called *LEACH*. Bonnet et. al. [1][2] discuss the application of distributed query execution techniques to improve communication efficiency in sensor and device networks.

10. Conclusions

In this paper we have introduced *Hybrid* protocol *APTEEN* which combines the best features of both proactive and reactive networks and to provide periodic data collection as well as near real-time warnings about critical events. We have also demonstrated implementation of a query which is versatile enough to respond to a variety of queries. Even though, our query model is suitable for a network with evenly distributed nodes, it can be extended further to sensor networks with uneven node distributions. We believe we have taken first step in defining an appropriate protocol for upcoming field of wireless sensor networks.

References

[1] P. Bonnet, J. Gehrke, and P. Seshadri. Querying the Physical World. In *IEEE Personal Communications. Special Issue on Smart Spaces and Environments*, October 2000.

[2] P. Bonnet, J. Gehrke, and P. Seshadri. Towards Sensor Database Systems. In *2nd International Conference on Mobile Data Management*, January 2001.

[3] S. Das, C. Perkins, and E. Royer. Performance Comparison of two On-demand Routing Protocols for Ad-Hoc Networks. In *Proceedings of the INFOCOM*, pages 3–12, March 2000.

[4] J. Elson and D. Estrin. An Address-Free Architecture for Dynamic Sensor Networks. Technical Report 00-724, Computer Science Department, USC, January 2000.

[5] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next Century Challenges: Scalable Coordination in Wireless Networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking(MOBICOM)*, pages 263–270, 1999.

[6] D. Estrin, Y. Xu, and J. Heidemann. Geography-informed Energy Conservation for Ad-Hoc Routing. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking(MOBICOM), Rome, Italy*, pages 70–84, July 2001.

[7] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. *"uAMPS ns Code Extensions"*. http://www-mtl.mit.edu/research/icsystems/uamps/ leach.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocols for Wireless Microsensor Networks. In *Proceedings of Hawaiian International Conference on Systems Science*, January 2000.

[9] W. B. Heinzelman. *"Application-Specific Protocol Architectures for Wireless Networks"*. PhD thesis, Massachusetts Institute of Technology, June 2000.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking(MOBICOM)*, pages 56–67, August 2000.

[11] M. Jiang, J. Li, and Y. C. Tay. Cluster Based Routing Protocol. Internet Draft, 1999.

[12] A. Manjeshwar and D. P. Agrawal. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, April 2001.

[13] UCB/LBNL/VINT. *"Network Simulator-ns"*. http://www-mash.cs.berkeley.edu/ns.