Using Remote Storage towards Energy Savings in Sensor Networks

Ankur Sahai
University of Southern California
asahai@usc.edu

Abstract

The motes in sensor network applications requiring very high sensing-rates need to transmit the collected data to the base station at frequent intervals in order to avoid an overflow of their limited storage capacity. This process consumes a high amount of energy, magnitude being directly proportional to the distance of the base station from the motes and the duration of transmission. Remote Storage uses some nodes with higher storage capacity distributed over the network to alleviate this problem by having the motes send their data to the nearest such node. Given the pre-configured location of motes in a sensor network and a load balancing parameter, it is possible to achieve energy savings through reducing the radio communication range of the motes, by placing the nodes with higher storage capacities in a specific manner. We use some randomized schemes to find the energy saving configurations. This paper also proposes a communication protocol that adapts to the sensing rate of the motes to increase the life-time of sensor network. The results demonstrate a ten fold energy savings when the sensing-rate adaptive protocol is run on top of an energy saving configuration as compared to a network without remote storage.

1. INTRODUCTION

One can think of several practical applications of sensor network that require the motes to sense the data at very high rates to improve the accuracy of results and may have the base station located very far apart due to inaccessible environmental conditions. Take the example of an underwater deployment of sensor network to monitor increased seismological activity at the ocean bed as a part of a Tsunami Warning system. Clearly, it may not be possible to have the base station set up at greater depths of ocean and such an application has to collect data at a very high rate in order to make accurate predictions that can save hundreds of lives. Due to high sensing rate involved, it becomes necessary for the motes to transmit the data to the distant base station in order to avoid overflow of their limited storage capacities. This is energy intensive process due to the increased radio communication involved over extended periods and is bound to diminish the life-time of the network. Such applications can utilize the Remote Storage architecture to extend their lifetimes.

Remote Storage architecture uses some *special* nodes with higher (~10^6 times) storage capacities than the *ordinary* motes deployed in the sensor network. In this setup, the ordinary motes transmit the data collected to a neighboring such special node. Apart from saving the radio transmission energy, the remote storage architecture provides the opportunity to organize the data at the special nodes (e.g. in a data-centric manner) thus, reducing the cost of queries.

Given a configuration of a sensor network with ordinary motes along with their locations (obtained via GPS), how can one place the special nodes with larger capacity in the network so that there is a quantitative energy saving (with respect to radio communication of sensed data) in this new configuration as compared to a configuration without remote storage? This is a challenging question because assuming a continuous search space and a finite supply of special nodes, there exist infinite possible remote storage configurations. In fact, finding any such configuration is NP-Complete which can be shown by a trivial reduction to the bin-packing problem. Then how does one approach the problem practically? We added two external constraints in the form of load balancing parameters and zones. Load balancing parameter is an upper bound on the number of ordinary motes that can transfer their data to a single special node. Zones are mutually exclusive and exhaustive sets of one or more ordinary motes delegated to a special node. Given a preconfigured sensor network, a load balancing parameter, it is possible to check if a given configuration satisfies a quantitative measure of energy consumed. We call such a configuration an *energy saving configuration*.

We propose a communication protocol for sensor networks that adapts to the sensing rate of the motes which is especially useful for applications that require dynamic sensing rates. Further, data transmission is triggered only when the used storage in the motes approaches a critical limit. This approach saves the energy consumed in transmitting the data periodically over the radio.

It was found that the sensing rate adaptive protocol when run on top of the energy saving configuration gives up to ten folds energy saving over the periodic data-transfer protocol which is run on top of a network without remote storage.

2. RELATED WORK

Storage in sensor networks has received much attention by the sensor network researchers off the late mainly because of two factors: limited availability of storage space in the motes and need for organized data storage for easier querying and retrieval.

Remote storage for sensor networks was first looked into by a research group at UCLA[1]. They used Stargate as the special node. Their results showed that using flash disks on the ordinary motes resulted in a bigger energy saving as compared to remote storage.

Data Centric Storage [3] is an efficient approach that stores data based on its semantics for ease of querying. Data that is closely related semantically is stored in a fashion that enables easier access to set of related data using structures like Geographic Hash table

In a Zone-sharing based approach [2], a set of closely located nodes is grouped together into a zone. In this scenario, the storage functionality of any node in a zone is also distributed among a set of other nodes from the same zone which, helps in reducing the burden of storage from any one node in the network. Efficient usage scenarios for flash-based data storage have been studied. Another paper [4] suggests that using efficient techniques, like having a sleep mode, for flash drives can achieve up to a 100-fold energy savings.

A number of Storage architectures utilizing some of these techniques have also been put-forward. PRESTO (Predictive Storage Architecture for Sensor Networks)[5] uses a mathematical prediction model to predict the change in the collected data and times the queries to the nodes in such a way as to save radio communication energy.

3. ENERGY SAVING CONFIGURATIONS FOR REMOTE STORAGE

An energy-saving configuration is mathematically characterized as follows:

$$\sum_{i=1}^{N} | k_i | * d_{Si(avg)} <= | k | * d_{B(avg)} \qquad \ldots(1)$$

for, $\sum_{i=1}^{N} k_i = k$ and i,j $k_i \Lambda k_j = \Phi$

where,

$k_i = \{o/o \in O \ \Lambda \ o \in Z_i\}$, where O is the set of all ordinary motes

$| k_i |$ is the cardinality of the set k_i

Z_i is a zone ***i*** consisting of $| k_i |$ ordinary motes and a unique special node

N is the total number of zones

$d_{Si(avg)}$ is the average distance of ordinary motes from the special node in the zone i

$d_{B(avg)}$ is the average distance of the all the ordinary motes from the base station

```
Given: Time window (W) , Transmit start time (T0) , Current local time (T), Risk factor (Rf)
       Storage capacity (S), Used storage (U), Storage threshold (Smin), k (constant)

If (U(T) >= Rf*S ) SenseRateAdapt

Protocol SenseRateAdapt
begin
  while (U(T) > Smin)
    T0 = T
    Calculate R1 = [U(T0) – U(T0-W)] / (T0-W) bytes/sec
    while (T < = T0 + W)
      Transmit data to the Special node at the rate k*R1 bytes/sec
    do
  do
end SenseRateAdapt
```

Fig 2. Sensing Rate Adaptive Protocol

```
Input:
   Set of n ordinary motes and their positions.
   Set of m (=n) special nodes.
  A load balancing parameter loadBalance
  Area of deployment A= x*y

Problem:
    To find an energy saving configuration

 Algorithm:
EnergySavingConfig := FALSE
While(!EnergySavingConfig)
   Repeat
     Begin
        1. Available: = n; SearchSpace = A
        2. Take a special node and place it at a random location.
        3. Allocate max (Available, loadBalance) nearest nodes to the node just placed
        4. Create a zone consisting of the special node, the allocated ordinary motes and area
        covered
        5. Available := Available – zone(ordinary); SearchSpace := SearchSpace – zone(area)
     End
   Until(Available=0)
   If (EnergySaving(CurrentConfiguration)) then EnergySavingConfig :=TRUE
End While

Output:
    An Energy Saving Configuration
```

Fig 1. A randomized algorithm to find an energy saving configuration

4. ACHIEVING ENERGY SAVINGS USING REMOTE STORAGE

We have algorithms in place that take as input a preconfigured sensor network (without remote storage) and return as output an energy saving configuration for remote storage which consists of the modified network divided into mutually exclusive and exhaustive zones. Each such zone returned by these algorithms has a set of ordinary motes (less than or equal to the load balancing parameter) delegated to a unique special node (and its location) which constitute the zone. It is to be noted that the mapping of the set of Ordinary motes O to the set of special nodes S is a many-to-one function.

How many special nodes are to be deployed and in what configuration? This is a not a trivial problem to analyze because of the following two reasons. First, the sample space is theoretically infinite even over a well-defined region of deployment assuming continuity. Second, how can the nodes be placed practically over such a huge search space. The latter question is a multi-headed one. The first aspect to it is how we place the very first special node. Now, assuming that the first node has been placed, there are still infinite different possibilities for placing the remaining nodes. As has been mentioned before, this is a hard problem. Thus it becomes very important to reduce this problem to a simpler version by introducing some external constraints. This was done by introducing the notion of *zone* and a *load balancing parameter* called ‘loadBalance’.

Once we have these constraints in place, this is how we go about solving the problem. We first place a special node in a random location and delegate to it a minimum of load balancing parameter and the available number of nodes and create a zone corresponding to this. After the creation of a zone it is exclude from the future search space. We repeat this procedure by considering the special nodes one at a time until all the ordinary motes are covered. It is to be noted that at the end of this process the search apace is divided into a set of mutually exclusive and exhaustive zones with each zone having a number (<=load balancing parameter) of ordinary motes delegated to the unique special node in the zone. After this we check if the energy saving condition *(1)* is satisfied. In case it is satisfied we return the configuration and exit. Otherwise, we repeat the above procedure until an energy saving configuration is obtained. Note that at this point we are only concerned with the decision version of the NP-Complete problem. The introduction of load balance parameter not only makes the problem easier to handle in practice but, it offers alternatives to the designer of the network e.g. one may choose a high value for this parameter if the storage capacity of the special nodes is high (or) the network is sufficiently dense (or) lesser special nodes are available.

The Sensing Rate Adaptive protocol (Fig 2), apart from scaling down the energy consumed by adapting to the sensing rate of motes and triggering the transactions only when the storage reaches a critical limit also has several other useful characteristics. The node failures are detected using the heart-beat messages. In case, of special node failure, all the ordinary motes belonging to the zone of the failed special node are allocated to the zone of their closest special node. For this, the special nodes exchange the data with their neighboring special node by following the sensing rate adaptive protocol. This also provides an opportunity for the special nodes to semantically reorder the data among themselves to organize in a data-centric fashion.

5. ANALYSIS OF TECHNIQUES FOR ENERGY SAVINGS IN SENSORNETS WITH REMOTE STORAGE

The equation (1) is reduced to the following form when the number of ordinary nodes placed under each special nodes are equal i.e. $k_1 = k_2 = \ldots = k_N = k_z$

$$| k_z | * \sum_{i=1}^{N} d_{Si(avg)} <= | k | * d_{B(avg)} \implies \sum d_{Si(avg)} <= (| k | / | k_z |) * d_{B(avg)} \quad \ldots(2)$$

Definition: An Energy Saving Configuration is a Remote Storage configuration that satisfies the Energy Savings equation *(1)*

***Theorem 1**:* Higher the load balancing parameter, higher the energy consumed by the energy saving configuration and vice versa.

Proof: Higher the value of load balancing parameter, more the number of ordinary motes delegated to a special node (in a zone) and thus, lesser the number of zones. At any step during the execution of the algorithm; more number of ordinary motes have to be delegated to a special node due to higher load balancing parameter. This means that (distant) nodes that would not have been allocated to this zone if the value for load balancing parameter had been lower than the current value are now allocated to this zone.

Theorem 2: The algorithm (Fig. 1) always returns an energy saving configuration upon termination

Proof (by contradiction): Let us say that the algorithm returns a configuration that does not satisfy this equation. Then by the design of the algorithm it would not have returned this configuration as the energy saving configuration would have been unsatisfied

6. SIMULATIONS

We designed these simulations in Visual C++ 2005. These simulations were run on the Windows XP machines available in the computer clusters at USC. Synergy remote login was used to access the Visual Studio environment.

6.1 Setup

The simulation was designed as an interactive C program. The user first enters the initial configuration of the network without remote storage. The user is then given a choice to select an algorithm to find the energy saving configuration: a Random Algorithm and a Simulated Annealing Algorithm.

In the case of the Random algorithm, the system picks random locations for the special nodes. In the case of Simulated Annealing algorithm, the system starts with the origin and traverses through the search space by moving to any of the neighboring positions in a random fashion. The probability with which it moves to the neighboring position during any given step is dependent upon two factors: Temperature T (which is the time in this case) and a Transition probability TP. The transition probability is a slightly complicated parameter which depends upon the Energy E of the two states being considered. E at any position P is taken to be the average distance of the special node at P to the nearest *Min(loadBalance, available)* ordinary motes. TP is directly proportional to the temperature 'T' and inversely proportional to the Energy difference between the two states. We have chosen to transition the state for values of TP greater than ½. Two different protocols were run on an energy saving configuration and a configuration without remote storage.

The period for the Periodic Data Transfer (PDT) protocol was chosen to be once every 5 units of time. This time-period was so chosen because of the following reasons. First, the sensing rate of the motes was set to once per unit of time and the amount of data collected was a random number less than or equal to 5 (using rand() function in C). Second, storage capacity of the motes was set to 500. These two put together imply that the storage of the ordinary motes can potentially get filled up in 10 unit of time. In the Sensing-Rate

Adaptive protocol the rate at which the sensed data is transmitted is dynamically controlled by the rate of change of on the ordinary motes. Further, the data transmission is only initiated when the used storage approaches the storage capacity of the mote.

For the Sensing-rate adaptive protocol(SRA), the sampling window was chosen to be 25 units of time. The risk factor that decided how close the used storage is allowed to reach to the storage capacity before the data transmission is triggered was set to 0.8. This was so chosen because, if set to a lower value, it would result in extra transmission overhead and when set to a higher value this would allow the used storage to approach dangerously close to the storage capacity and may even lead to loss of the data due to overflow.

There were other general parameters involved in the design of the protocols. The storage capacities of the ordinary and special node were chosen to be 500 and 100000 units of storage respectively. The data transmission rate was set to 2.5 bits/sec. Energy consumed by the ordinary motes to transmit a single unit of data per unit distance was chosen to be 2 units of energy. The heart-beat period was chosen to be 100 unit of time. There were also some user-induced node failures in place.

6.2 Results

We used five different energy-saving configurations to study the energy consumed by the protocols over time. For each of these remote storage configurations and their single-tier counterparts (without remote storage) we ran both the protocols to study the energy consumed with respect to time. These are summarized below

Configuration	*I*	*II*	*III*	*IV*	*V*
Area of deployment	100*100	500*500	2500*2500	5000*5000	10000*10000
d_{Bavg}	83.55	241.4	1405.74	3581.78	6340.61
$\|O\|$	10	12	15	20	25
loadBalance	5	3	6	3	8
Number of Special nodes used (=Zones)	2	4	3	7	4
Average E.S.R = $E_{PDT-1Tier}/E_{SRA-2Tier}$	10.25	7.12	8.24	10.72	7.76

As predicted by Theorem 2, it was found that the energy consumed varied directly with the load balancing parameter as is shown in Fig 3. The energy savings ratio (E.S.R) is measured as the ratio of energy consumed by the periodic data transfer protocol run on a network without remote storage to the energy consumed by the sensing rate adaptive protocol when run on a network with remote storage. The fluctuation in E.S.R. is due to the different configurations in which they are run. It is to be noted that these configurations were limited and practical deployment of motes for such an application would cover much larger area having many more motes. It would be interesting to see how the algorithm (fig 1) behaves in such a scenario. The code and related histograms can be found here: http://www-scf.usc.edu/~asahai/694_Node_Placement.cpp

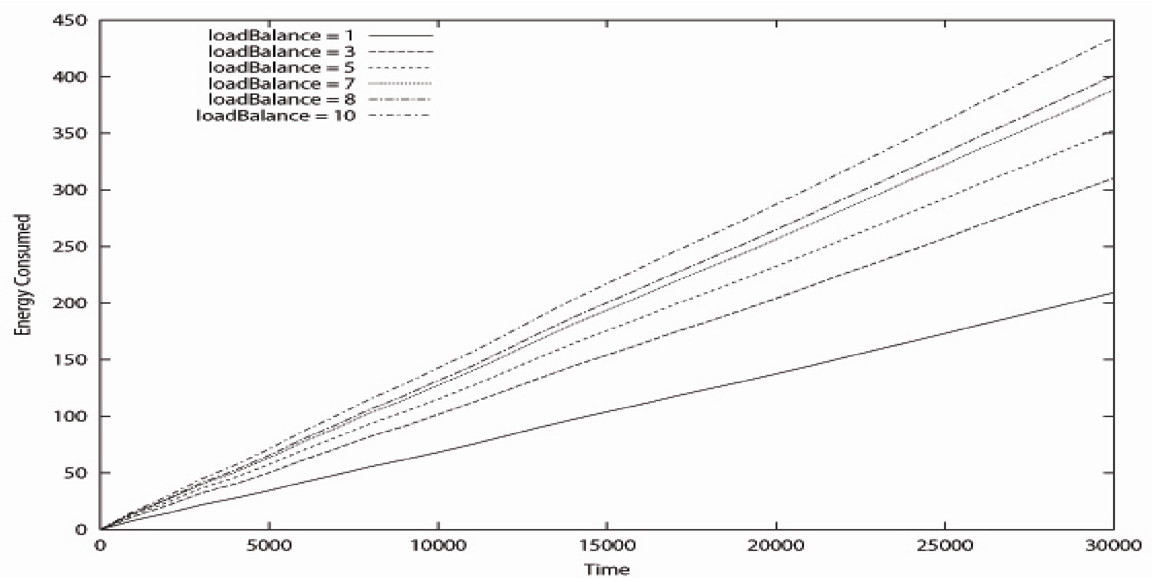

Figure 3. Energy Consumed Vs Load Balancing parameter (on Configuration 1)

6.3 CONCLUSIONS AND FUTURE WORK

We think that it is not enough to only consider the ordinary motes and their special node managers while studying energy saving in the two-tier architecture for remote storage. Instead, one should also take in to account the distance of the special nodes from the Base station in a specific manner as the special nodes have larger but still limited storage space and have to ultimately send their data to the base station. One could associate weights with the distances for example, higher weight for distance of ordinary motes to special nodes and lower weights for distance of special nodes from the base station as they are expected to communicate less frequently with the base station.

The analysis is incomplete without studying the savings in the query costs through a practical implementation of this system. Other protocols could be considered for energy saving in a remote storage sensor network.

REFERENCES

1. Remote Storage for Sensor Networks by Rahul Balani, Chih-Chieh Han, Vijay Raghunathan, and Mani Srivastava

2. An Adaptive Zone-based Storage Architecture for Wireless Sensor Networks by Thang Nam Le, Dong Xuan, Wei Yu

3. Data-Centric Storage in Sensornets by Scott Shenker, Sylvia Ratnasamy, Brad Karp, Ramesh Govindan, Deborah Estrin

4. Ultra-Low Power Storage for Sensor Networks by Gaurav Mathur, Peter Desnoyers, Deepak Ganesan, Prashant Shenoy

5. PRESTO: A predictive Storage Architecture for Sensor Networks by Peter Desnoyers, Deepak Ganesan, Huan Li, Ming Li, Prashant Shenoy