An Energy-Efficient Framework for Multi-Rate Query in Wireless Sensor Networks

Yingwen Chen[1,2], Hong Va Leong[2], Ming Xu[1], Jiannong Cao[2],
Keith C.C Chan[2], and Alvin T.S Chan[2]
[1] *School of Computer, National University of Defense Technology, 410073 Changsha, China*
ywch_nudt@hotmail.com, xuming64@public.cs.hn.cn
[2] *Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong*
{csychen, cshleong, csjcao, cskcchan, cstschan}@comp.polyu.edu.hk

Abstract

Minimizing the communication overhead is always a hot topic in wireless sensor networks. In a multi-rate query system, data sources disseminate the data streams to users at the frequency they request. However, sending data in different frequency to individual users is very costly. We address this problem by broadcasting a single consolidated data stream, aiming at reducing the amount of transmitted data. Taking into account the data correlation, we can re-construct the data streams at lower frequencies from the consolidated stream at a higher frequency. In this paper, we propose an energy-efficient framework to process multi-rate queries and investigate rate conversion mechanism. We evaluate both the accuracy and energy efficiency by simulation. Simulation results indicate that with a reasonable level of tolerance, the performance gain is significant. As far as we know, this is the first energy-efficient solution for multi-rate query in wireless sensor networks.

1. Introduction

A wireless sensor network consists of a collection of communicating nodes, each incorporated with sensors collecting real-time data to the sink node. Sensor nodes are battery-powered and energy is the most crucial resource. Many existing research works address the problem of minimizing energy consumption by minimizing the communication overhead. Since sensor nodes possess local computation abilities, part of the computation can be off-loaded from the sink node[1]. In general, performing data operations inside the network, such as eliminating irrelevant records and aggregating raw data, can reduce energy consumption and improve sensor network lifetime significantly. This is referred to as in-network data processing[2][3][4] and data aggregation[5][6].

In a multi-rate query system, a data source serving multiple sink nodes with queries demanding varying data rates needs to send data in different frequency to individual nodes. This is costly, since the sink nodes in general consume data at different moments and most of the data sent by the data source could not be shared across the sink nodes. This new problem is different from the one addressed in in-network processing and aggregation. Observing the correlation among data streams from the data source to different sinks, it is possible to construct a consolidated stream to represent those multiple data streams. We address this interesting problem by broadcasting the single consolidated streaming data series, aiming at reducing the amount of transmitted data, and hence energy consumption.

The contribution of the paper is threefold. First, we propose an energy-efficient framework to process multi-rate queries and investigate rate conversion mechanism between arbitrary frequencies. Second, we analyze analytically the performance on communication cost with our energy-efficient strategy. Third, we conduct simulation studies to evaluate the energy efficiency and accuracy of our strategy. Our simulation results indicate that we can achieve an average saving of up to 30%~45% of communication cost, at an average relative error below 5%.

The remainder of this paper is organized as follows. Section 2 introduces the multi-rate query problem. In Section 3, we propose our energy-efficient framework and describe the frequency conversion mechanism. Section 4 presents some analytical results on the query strategy. In Section 5, we conduct simulated experiments to evaluate the performance. Finally, we conclude the paper briefly and outline some of our future research directions.

Proceedings of The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)
0-7695-2687-X/06 $20.00 © 2006 IEEE

2. Multi-Rate Query in WSNs

In WSNs, the sink nodes may query the data at different frequency according to different requirements. For example, if the WSN is used for collecting the temperature of the environment, application 1 might need the newest temperature every 3 minutes, and application 2 might need the newest temperature every 5 minutes. This will result in multi-rate queries in WSN, for which there are two queries, demanding data at time 0, 3, 5, 6, 9, 10, 12, 15, 18, 20 and so on.

A multi-rate querying system is illustrated in Fig.1. There are m sink nodes S_i (i=1..m) requesting the streaming data series from a source node G at different frequencies f_{ri} (i=1..m). Without loss of generality, we can always find an appropriate time unit such that all frequencies can be represented as integers unless the frequencies are irrational numbers. Intuitively, the source node G disseminates the data along the path to each querying sink node at the corresponding frequency separately. We call this kind of data dissemination strategy the **Native Strategy** (or **N-Strategy**).

From the basic rule of information theory, the total amount of information is proportional to the number of samples and the number of bits coding the sample[7]. Under the same coding system, a data series at higher frequency (with smaller intervals) contains more information than one at lower frequency. Taking advantage of the data correlation between data series at different frequency, data series at lower frequency could be constructed from data series at higher frequency. It is obvious that N-Strategy is inefficient because the source node propagates the data series regardless of the data correlation between them. Since wireless communication in WSNs is of a broadcast nature, transmitting data at a consolidated frequency can potentially cut down the total amount of transmitted data, leading to savings in energy consumption. Taking Fig.1 as an example, if data series at frequency f_{r2} can be reconstructed from data series at frequency f_{r1} within acceptable error, source node G only needs to disseminate the data to K at frequency f_{r1}. When node K forwards the data to S_1 at frequency f_{r1}, node L can also receive the data at frequency f_{r1}. Node L can then reconstruct the data series at frequency f_{r2} from the received data series and forwards them to S_2. Thus the transmission overhead of source node G is reduced by avoiding sending the data series individually to S_2. Likewise, the total amount of data transmitted across intermediate nodes is also reduced.

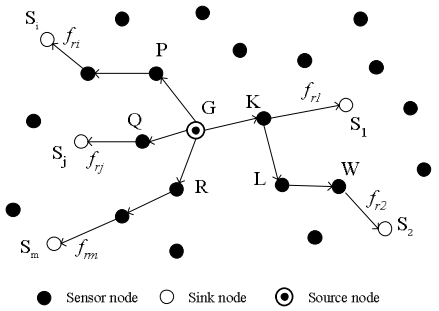

Fig.1. **Multi-rate queries for source node *G* in WSN**

There are two problems that need to be addressed in a multi-rate query system. The first one is how to organize the source node activity in generating a consolidated data stream, with the aim of reducing the amount of transmitted data, hence bandwidth requirement and energy consumption. The second one is how to reconstruct the data streams at the desired frequency from the consolidated stream at a different frequency. We will present the solutions in the subsequent sections.

3. Energy-Efficient Framework

Our energy-efficient framework for multi-rate query in WSNs is built upon a number of components, including query frequency registration, data stream consolidation, data dissemination, data stream reconstruction, and data stream frequency conversion. Query frequency registration allows data sinks to pose their querying requirement to the data source. From the query frequency registered, the source determines the frequency on which the data stream should be generated and then disseminated. The data dissemination process involves the transmission of data streams to their designated destination. Reaching the sink node, the target data stream is reconstructed. Staying in the core is the frequency conversion mechanism, which allows data streams to be converted from one frequency to another. In the midst of data dissemination, forwarding nodes may need to perform frequency conversion, similar to what node L is doing in Fig.1. Our energy-efficient strategy requires the construction of the consolidation stream and the reconstruction of target streams, with proper frequency conversion. We call our strategy the **E-Strategy** in contrast to the intuitive **N-Strategy**.

3.1. E-Strategy

N-Strategy is inefficient because it does not take advantage of the data correlation between data series, even though the data series are transmitted along the same path. In order to make use of the data correlation

between data series, we need the information about the query frequencies on the intermediate node along the path from the source node to the sink nodes. We maintain a list, called *RequestList*, on every node in the network. The list contains the frequencies of all requests passing through that particular node. When the sink node generates a query at a certain frequency, it should register its frequency along the path inclusively to the source node to inform all intermediate nodes of the frequency requested.

```
DataDissemination(MyID)
begin
 RequestF ← FindMax(RequestList);
 if (MyID = SourceID) then
  //broadcast at the requested frequency
  broadcast(Data,RequestF);
 else
  receive(Data);
  ReceivedF ← GetFrequency(Data);
  if (RequestF < ReceivedF) then
   // do down-sampling
   convertFrequency(Data,ReceivedF,RequestF);
   SendF ← RequestF;
  else SendF ← ReceivedF;
  if (myID = SinkID) then
   toApplication(Data);
  else broadcast(Data,SendF);
 end if;
end;
```

Fig.2. **Data dissemination algorithm**

Since all the frequencies of the requested queries are registered in *RequestList* of each intermediate node, it is easy for the intermediate node to determine whether there is bandwidth sharing. In fact, bandwidth sharing happens in those nodes with *RequestList* containing at least two frequencies. As a result, each node can cut down the communication cost by choosing the largest frequency from *RequestList* as the frequency of its consolidated data stream.

Fig.2 describes the algorithm for data dissemination. We can see that the source node simply broadcasts the data at the *largest frequency* of all the queries. However, for other nodes, there may be the case that the frequency of the data series received, *ReceivedF*, is larger than the largest frequency in *RequestList*, *RequestF*, meaning that the incoming data is more than enough. The frequency conversion function is invoked to reconstruct the data series at frequency *RequestF* from the data series at frequency *ReceivedF*. The frequency conversion mechanism is discussed next.

3.2. Frequency Conversion

Frequency conversion is concerned with the problem that given a data series X at frequency f_1, how to determine the value of an unknown data series Y at frequency f_2. The frequency conversion problem is similar in nature with the interpolation problem, which is constructing new data points from a discrete set of known data points.

We adopt interpolation techniques to achieve simple frequency conversion. There are many interpolation algorithms, such as linear interpolation, quadratic interpolation, cubic-spline interpolation and so on. We choose linear interpolation based on two reasons: first, it is the simplest interpolation method, with the least computation over-head; second, our preliminary simulation results show that its accuracy is acceptable, and that the advantage of a few other interpolation mechanisms is not very significant.

In linear interpolation, the values interpolated between two consecutive data samples lie on a straight line connecting them and we can estimate the values $\hat{Y}$ of data series Y by

$$\hat{y}[i] = (x[\lfloor z_i \rfloor + 1] - x[\lfloor z_i \rfloor]) \cdot (z_i - \lfloor z_i \rfloor) + x[\lfloor z_i \rfloor] \quad (1)$$

where $z_i = \frac{i \cdot f_1}{f_2}$, and $\lfloor z \rfloor$ is the floor function, returning the largest integer no larger than z.

If we know the true value of Y, we can use the Average Relative Error (ARE) metric to evaluate the accuracy of interpolation. For a series of length l, ARE is defined as

$$\mathrm{ARE}(Y, \hat{Y}) = (\sum_{i=0}^{l} \frac{|y[i] - \hat{y}[i]|}{y[i]}) / (l+1) \quad (2)$$

3.3. Pragmatic Consideration

From Equation (1), we can observe that if we want to get the i-th value of $\hat{Y}$, we need the $\lfloor z_i \rfloor$-th and $(\lfloor z_i \rfloor + 1)$-th value of X.

Since $\lfloor z_i \rfloor \cdot \frac{1}{f_1} \le \frac{i}{f_2} < (\lfloor z_i \rfloor + 1) \cdot \frac{1}{f_1}$, we need *future value* of X to estimate the current value of Y. This is only possible in a historical system, but not in a real-time system like most sensor network applications. Fortunately, we can still attempt to predict the required future value of X from the historical information of data series X. In particular, we employ the following prediction method for a future value of X:

$$x[\lfloor z_i \rfloor + 1] = \alpha \cdot x[\lfloor z_i \rfloor] + (1 - \alpha) \cdot x[\lfloor z_i \rfloor - 1] \quad (3)$$

Using the frequency conversion mechanism, we can convert the data series between arbitrary frequencies. However, converting data series at lower frequency to

higher frequency brings in a relatively large ARE than the more natural down-sampling operation. That is the reason why we choose the largest frequency to be the frequency of the consolidated broadcasting stream in E-Strategy, in order to reduce the ARE when the intermediate and sink nodes reconstruct the data series at lower frequency.

4. Performance Analysis

We now give some analytical bounds on the performance of N-Strategy and E-Strategy. The greatest performance gain from E-Strategy is due to the ability of sharing the bandwidth as much as possible along the path when disseminating the data series, thereby reducing the precious energy consumed.

Theorem 1. Using N-Strategy, the upper bound of the dissemination frequency f_{up} of each node is $\sum_{i=1}^{m} f_{ri}$, where f_{ri} (i=1..m) are the requested frequencies contained in *RequestList* of the node. This upper bound is attained if and only if for any pair of data series in the request, there is no point of intersection along their time axes.

Theorem 2. Using N-Strategy, the lower bound of the dissemination frequency f_{low} of each node can be calculated by

$$\sum_{k=1}^{m}((-1)^{k-1} \cdot \sum_{\{F_j\}_{j=1}^{k} \subseteq \{f_{ri}\}_{i=1}^{m}} \gcd(\{F_j\}_{j=1}^{k})) \quad (4)$$

where $\gcd(\{F_j\}_{j=1}^{k}))$ means calculating the *greatest common division* of k frequencies selected in all m frequencies. This holds if and only if for any pair of data series in the request, they have points of intersection along their time axes.

Theorem 3. In the worst case, all the nodes except the source node in the WSNs query the same data source. The upper bound of the total communication overhead in one time unit for N-Strategy is $O(D \cdot (N-1))$, while that of E-Strategy is $O(N-1)$, where D is the diameter of the sensor network and N is the number of sensor nodes.

It is obvious that E-Strategy always outperforms N-Strategy in terms of communication cost. If the multi-rate queries in the network share more paths, there is a greater savings in communication overhead using E-Strategy. Theorem 3 specifies an extreme case that E-Strategy can take full advantage of path sharing, yielding a theoretically perfect performance over N-Strategy.

5. Simulation Studies

In this section, we present the results of our simulation study. We evaluated the communication cost and accuracy of E-Strategy and made a comparison with N-Strategy. We also investigated the effects of the sensor network and query parameters on the performance of E-Strategy.

In our simulation, the sensor nodes are distributed in a region δ, according to the uniform distribution. A communication graph is generated under the assumption that all the nodes have the same transmission range ρ. A summary of the query and sensor network parameters and their default values is presented in Table 1.

Table 1. **Parameters of query and sensor network**

Parameter	Symbol	Default value
Coverage of sensor network	δ	300 by 300
Number of sensor nodes	N	400
Transmission range	ρ	30
Number of sink nodes	m	6
Frequency of the query	f	5-10
Query distance	H	6 hops

In order to ensure that the simulation experiments are repeatable, we use synthetic data. We generate the data source time series with a function of the random-walk series, defined as[8]

$$x[i] = 100 * (\sin(0.1 * RandomWalk[i]) + 1 + i / R) \quad (5)$$

where $i = 0,..., R-1$; $RandomWalk[0..R\text{-}1]$ is a random-walk series; and R is the range of the walk, with a value of 100000. The time unit is chosen as the least common multiplier of all frequencies of the queries launched by the sink nodes, so as to keep the time intervals of all sampled data series integer.

The sink nodes and source node are chosen randomly. Each sink node launches a query to the same source node with an integer frequency. We use Direct Diffusion[8] routing protocol to find the least hop routes for disseminating the data from the source to the sinks. The communication cost is evaluated by number of data packets sent per time unit, and the accuracy is evaluated by the mean of the AREs of all sink nodes.

We generate 30 connected network instances for each simulation and spawn multi-rate queries in each network instance for 100 times. The average performance for the queries in each network topology is measured and the overall performance is obtained as an average over all the 30 topologies.

5.1. Impact of Query Distance

The first set of simulated experiments aims at evaluating the communication cost and accuracy with

different query distance H. The query distance reflects how far it is from the sink node to the source node. It is the number of hops between the sink node and the source node. In this experiment, we fix the number of sensors N to 400 and the number of sink nodes m to 6. The results are depicted in Fig.3 and Fig.4.

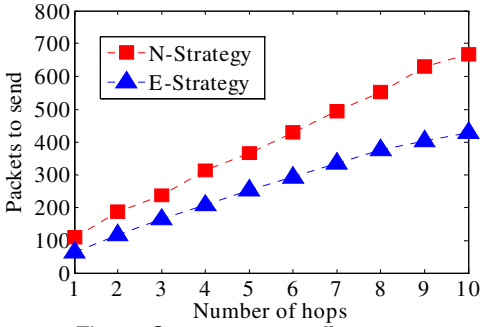

Fig.3. **Cost vs query distance**

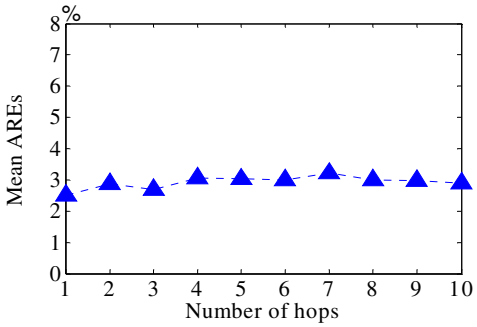

Fig.4. **Accuracy vs query distance**

From Fig. 3, it is obvious that we can benefit a lot in communication cost by adopting E-Strategy. As the query distance H increases, the cost of N-Strategy grows almost linearly with H, faster than that of E-Strategy. That is because the cost of N-Strategy reflects the cumulative overhead of all queries, while the cost of E-Strategy is only a part of that, owing to its bandwidth sharing property. When the average hop of the query distance is getting to 10, E-Strategy leads to a saving of 35% of communication cost over N-Strategy. Fig.4 indicates the tradeoff in accuracy. We can see that using the linear interpolation to convert the frequency generates a very tolerable mean AREs, which is only about 3% of the actual sensor data value. Furthermore, this imprecision is relatively independent on the query distance.

5.2. Impact of Node Density

Since the topology of the sensor network is affected greatly by the node density, we investigate how the node density will affect the performance of the query strategies. In this experiment, we fix the number of hops of the query H to 6, the number of sink nodes m to 6 and vary the number of nodes N, and hence node density. The results are depicted in Fig.5 and Fig.6.

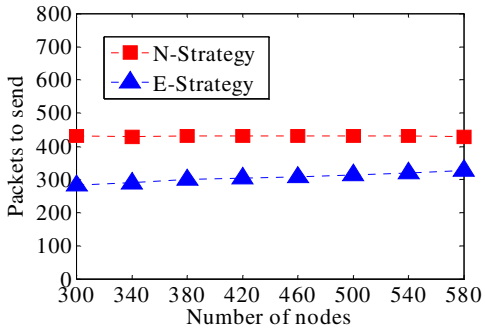

Fig.5. **Cost vs node density**

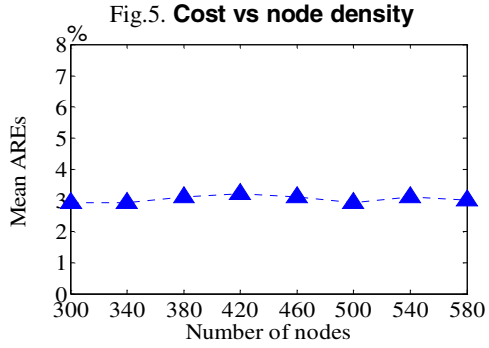

Fig.6. **Accuracy vs node density**

From Fig.5, it is obvious that E-Strategy outperforms N-Strategy in terms of communication cost. N-Strategy is not density sensitive, since the communication cost of N-Strategy is only determined by the number of sink nodes, query frequency, and the query distance. However, E-Strategy is slightly density sensitive, that is, when the node density increases, the communication cost increases slightly as well. That is because when there are more sensor nodes, each node may have more neighbors, which help to form the shortest path from the sink node to the source node, thereby reducing the chance for different sink nodes sharing the same path. This phenomenon does not necessarily mean that the performance of E-Strategy declines in a denser network. It is just because the Directed Diffusion routing protocol we adopt does not favor the route sharing property of E-Strategy. We leave finding the optimal routing protocol for E-Strategy to our future work. When accuracy is concerned, Fig.6 indicates that the mean AREs is again maintained at a comfortable level of about 3%, and is relatively independent of node density.

5.3. Impact of Number of Sink Nodes

The communication cost is closely related to the number of sink nodes and hence the number of queries. Thus, we measure the performance of N-Strategy and E-Strategy with respect to number of sink nodes. In this set of experiment, we fix the number of sensors N to 400 and the query distance H to 6, and varying the

number of sink nodes from 1 to 10. The results are depicted in Fig.7 and Fig.8.

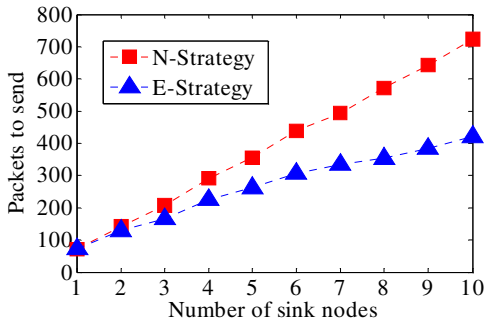

Fig.7. **Cost vs number of sink nodes**

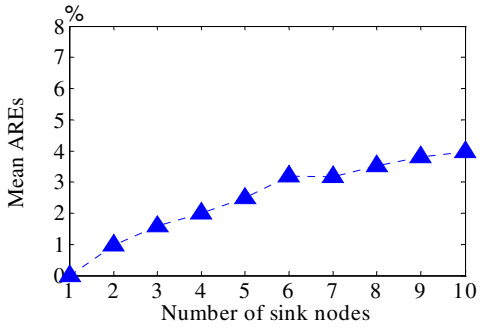

Fig.8. **Accuracy vs number of sink nodes**

From Fig.7, it is obvious that we can again benefit a lot in communication cost by adopting E-Strategy. As the number of sink nodes *m* increases, the cost of N-Strategy increases almost linearly and much faster than E-Strategy. That is because more sink nodes intuitively arouse more queries, hence higher communication overhead. However, by applying E-Strategy, the communication overhead is reduced via bandwidth sharing. When the number of sink nodes gets to 10, E-Strategy leads to a saving of 45% of communication cost over N-Strategy. Unlike the query distance and node density, the number of sink nodes does pose an impact on the accuracy of the reconstructed data series. As evidence from Fig.8, the mean AREs increases with increasing number of sink nodes. This is because more sink nodes implies more varying frequencies, as well as the number of times that frequency conversion needs to be performed. Both factors result in larger mean AREs. However, even when the number of sink nodes becomes 10, the mean AREs is still no more than 5%. In other words, even for a good amount of sink nodes, the mean AREs is still tolerable.

6. Conclusion

Energy consumption is a crucial factor affecting the application and effectiveness of a wireless sensor network. In this paper, we proposed an energy-efficient framework in coping with multi-rate queries in WSNs. Both analytical and simulation results reveal that by tolerating a small degree of imprecision, the E-Strategy can lead to a significant amount of communication cost savings, thereby extending the effective lifetime of WSNs. Our future research work include exploring more accurate frequency conversion method and constructing more effective routing protocol to support our querying strategy.

Acknowledgement

This research is partially supported by a research grant from the Department of Computing, the Hong Kong Polytechnic University and the Doctoral Foundation of National Education Ministry of China under Grant No.20059998022.

References

[1] P. Bonnet, J. Gehrke, and P. Seshadri. Towards sensor database systems. In Proceedings of International Conference on Mobile Data Management, pages 3–14, 2001.

[2] B.J. Bonfils and P. Bonnet. Adaptive and decentralized operator placement for in-network query processing. Telecommunication Systems, 26(2-4):389–409, 2004.

[3] U. Srivastava, K. Munagala, and J. Widom. Operator placement for in-network stream query processing. In Proceedings of ACM SIGMOD, pages 250–258, 2005.

[4] Y. Chen, H.V. Leong, M. Xu, J. Cao, K.C.C. Chan, and A.T.S. Chan. In-network data processing for wireless sensor networks. In Proceedings of International Conference of Mobile Data Management, May 2006.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In Proceedings of International Conference on Mobile Computing and Networking, pages 56–67. 2000.

[6] W. Yu, T.N. Le, D. Xuan, and W. Zhao. Query aggregation for providing efficient data services in sensor networks. In Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems, pages 31–40, 2004.

[7] J. Lesurf. Information and Measurement, Institute of Physics Publishing, London, 2002.

[8] L. Gao, X. S. Wang. Continually evaluating similarity-based pattern queries on a streaming time series. In Proceedings of ACM SIGMOD, 2002.