

A Data Compression Technique for Sensor Networks with Dynamic Bandwidth Allocation

Song Lin, Dimitrios Gunopulos, Vana Kalogeraki, Stefano Lonardi
Computer Science & Engineering Department
University of California, Riverside
{slin,dg,vana,stelo}@cs.ucr.edu

1. Introduction

With the technical development of low-power embedded communication systems, thousands of sensor nodes would be seamlessly embedded in the physical world and form a wireless sensor network (WSN). These sensors monitor a variety of characteristics of the environment such as sound, temperature, humidity, pressure, noise levels, movement, etc.

A typical problem of WSNs is to let the sink gather historical information from all the sensor nodes. Due to the resource limitations (such as CPU, power bandwidth, etc) of each sensor node, it may be impractical to transmit a full-resolution data feed from each sensor. In order to improve data transmission, several data compression techniques have been proposed [1] [2]. In this paper, we present a new technique to compress the historical information in WSNs. Our technique can efficiently utilize the transmission bandwidth and improve the compression precision further.

In most hierarchical topologies [3], all the sensors in the cluster share the cluster head's communication channel evenly, that is, they utilize the same communication bandwidth, as is shown in Figure 1. If multiple sensors want to communicate with the cluster head, they have to share the communication channel as described in Figure 2. For example, in TDMA scheduling, a different time slot is assigned to each sensor, while in FDMA scheduling, a different frequency slot is allocated to each sensor.

However, in WSNs, as different sensors usually collect different data, the compression qualities of different sensors are usually different if given the same transmission bandwidth. Therefore, the compression qualities at different sensors may not be balanced well. In this paper we address this problem and present a Dynamic Bandwidth Allocation algorithm to solve it.

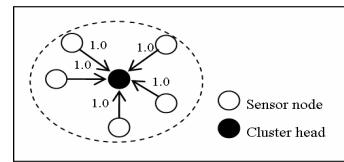


Figure 1. Bandwidth allocation in *LEACH*

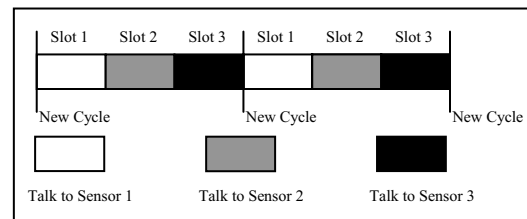


Figure 2. A sample communication channel

1.1. Our Contributions

Our contributions are summarized as follows:

1. We apply the LVQ (Learning Vector Quantization) algorithm to construct the codebook in the Dictionary Lookup Scheme. Our results show that the LVQ learning process can further improve the codebook for high compression precision.

2. We introduce the concept of two-level regression for higher precision compression. We utilize two-level regression to compress the codebook update in order to save more bandwidth while keeping the codebook updated with high precision.

3. We introduce the problem of Dynamic Bandwidth Allocation (DBA) in sensor networks. With our DBA scheme, the data compression qualities for all the sensor nodes are well balanced and maximized.

2. Problem Definition

At each sensor, the collected data is recorded in a one dimensional array that collects the sampled values

with some frequency. As more and more samples are collected, the sensor's memory buffer becomes full. At this point, the time series values in the buffer are compressed and the compression representatives of the collected values are transmitted to the base station.

Formally speaking, our problem can be described as follows:

Given a one dimensional time series array X , the goal is to find a proper encoder function F , $Y = F(X)$ and a decoder function G , so that

$$a) |X| / |Y| \geq R$$

$$b) \|X - G(Y)\| \text{ is minimized.}$$

It is important to note that: In a) the ratio between the cardinality of X and that of Y is bigger than the compression rate R according to application specifications. In b) the distance between the retrieval values from compressed representative and the original values is minimized.

3. The ALVQ algorithm

We will present the ALVQ (Adaptive Learning Vector Quantization) algorithm for compressing the historical information in each sensor node. Our technique is based on the following observation: the historical information collected by a sensor node exhibits similar patterns over time.

The ALVQ framework

1. Create the codebook from training data set at the sensor
2. Transmit the codebook to the base station
3. Let the sensor collect data and fills the local buffer
4. When the buffer is full, compute the codebook update, apply two-level regression to compress the codebook update
5. Compress the collected data with new codebook
6. Transmit the compressed codebook update and the compressed data
7. Flush the buffer and go to step 3

Our ALVQ algorithm works in the following way: First, in the codebook construction, it chooses some data pieces from the training dataset (this step is similar to SBR [2]) as the codebook. Then we perform a LVQ (Learning Vector Quantization) [4] [5] learning process on the codebook, which adjusts the codebook, to be nearer to the optimal codebook. For the codebook updates, ALVQ compresses the codebook update data pieces and transfers the compressed information to the base station. Using a novel 2-level piece-wise regression technique, ALVQ can compress

the update with high precision while saving more bandwidth for data transmission in order to increase the quality of the approximation.

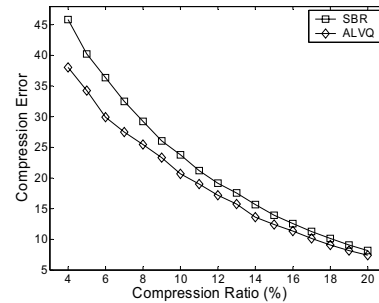


Figure 3. A comparison of ALVQ and SBR

In Figure 3, we show experimentally the approximation error for different compression ratios for SBR and ALVQ (with 30MBs of data files and a 2KB codebook). On average, ALVQ can transmit data with the same comparison error using 25% less bandwidth.

4. Dynamic Bandwidth Allocation in Sensor Networks

Now let us consider the problem in a general case where we want to gather the historical information from all the sensors in the sensor network.

LEACH (Low-Energy Adaptive Clustering Hierarchy) and its variants [3] [4] are among the most popular hierarchical routing algorithms. In LEACH, all the sensor nodes send packages directly to its local cluster head and the cluster head transmits these packages to the sink through multiple hops in the sensor network. Since the communication between the sensor node and the cluster head is wireless, the communication channel of the cluster head is the bottleneck of the total data transmission of all the sensors in the cluster. In LEACH, all the sensors in the cluster share the communication channel of the cluster head evenly (e.g. the TDMA scheduling), that is, they share the same communication bandwidth of transmission data to the cluster head.

As different sensors usually collect different data, it is highly possible that the approximation error in different sensor is different, even given the same compression rate. In applications where similar compression qualities are required for all the sensors, different compression rates should be assigned to different sensors in order to balance the compression qualities among them. As all the sensor nodes share the communication channel of the cluster head, we can take better advantage of it. For those sensors with low

approximation errors, the cluster head assigns less bandwidth to them; for those sensors with high approximation errors, more bandwidth is assigned. Therefore, with different transmission bandwidth, similar compression qualities for different sensors are achieved.

This can be done by changing the channel schedule. For example, in TDMA scheduling, sensors that need more bandwidth can utilize the communication channel longer.

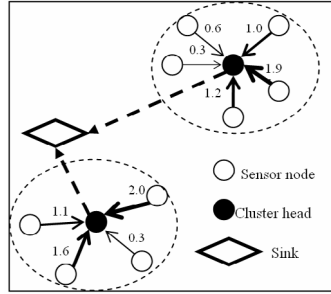


Figure 4. The DBA model

We introduce the Dynamic Bandwidth Assignment (DBA) Algorithm (Figure 4) for sensor information transmission and compression qualities balancing in sensor networks.

The DBA algorithm

1. Embed the sensors into the environment
2. Let the sensors set up clusters as in *LEACH*
3. The cluster head assigns transmission bandwidth evenly to all sensors in the cluster
4. Let the sensor collect and compress the measured data
5. The sensor transmits the compressed data to the cluster head with the bandwidth assigned to it
6. The cluster head transmits data to sink through multi-hops in sensor networks
7. The cluster head reassigns bandwidth to each sensor in the cluster
8. Go to Step 4

After setting up a cluster (with k sensors), the cluster head collects the compression quality of each sensor Q_1, Q_2, \dots, Q_k , and the bandwidth assigned to them B_1, B_2, \dots, B_k based on the last transmission between these sensors and the cluster head. Then we compute their ideal bandwidth A_i as below,

$$A_i = Q_i \frac{B}{Q}, \quad B = \sum_{i=1}^k B_i, \quad Q = \sum_{j=1}^k Q_j,$$

where B is the total bandwidth of the cluster head's channel, Q is the total compression qualities for all the

sensors in the cluster. If the compression qualities for all the sensors in the new transmission are the same as before, ideal bandwidths calculated above fit perfectly for data transmission. In reality, however, this is rarely true. The reason is that sensor data usually changes over time, so does their compression qualities even with the same compression rate. Therefore, for the next data transmitted, we propose to allocate bandwidth to sensor i in a conservative way as

$$B'_i = \alpha B_i + (1 - \alpha) A_i,$$

where $0 \leq \alpha \leq 1$ is a weighting factor. Now the bandwidth of the sensors is allocated dynamically as a combination of the old bandwidth and the estimated ideal bandwidth, which caters to the characteristics of the sensor data.

With the dynamic bandwidth assignment to different sensors, those sensors with high approximation errors can get more bandwidth and those with low approximation errors will get less bandwidth.

5. Conclusions

In this paper, we have presented a new data compression technique, designed for historical information compression in sensor networks. Our method employs the LVQ learning process to construct the codebook and the codebook's updates are compressed to save bandwidth for sensor data transmission. In addition, we have addressed the dynamic bandwidth allocation problem in sensor networks. Our *DBA* algorithm can dynamically adjust the communication bandwidth of different sensors in order to balance data compression qualities at different sensors.

6. References

- [1] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. "Multi-Dimensional Regression Analysis of Time Series Data Streams", In *Proceedings of VLDB*, 2002.
- [2] A. Deligiannakis, Y. Kotidis and N. Roussopoulos. "Compressing Historical Information in Sensor Networks", In *Proceedings of ACM SIGMOD*, 2004.
- [3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocol for wireless sensor networks, in the *Proceeding of HICSS*, Hawaii, 2000.
- [4] A. Manjeshwar and D. P. Agrawal, TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks, in *Proceedings of IPDPS*, 2001.
- [5] T. Kohonen. "Self-Organizing Maps", pringer Verlag, 1995
- [6] T.Kohonen, "Improved versions of learning vector quantization", In *Proceedings of IJCNN*, 1990.