RIDA: A Robust Information-Driven Data Compression Architecture for Irregular Wireless Sensor Networks

Thanh Dang, Nirupama Bulusu, and Wu-chi Feng

Department of Computer Science,
Portland State University,
PO Box 751, Portland, OR, USA
(dangtx,nbulusu,wuchi)@cs.pdx.edu

Abstract. In this paper, we propose and evaluate RIDA, a novel information-driven architecture for distributed data compression in a sensor network, allowing it to conserve energy and bandwidth and potentially enabling high-rate data sampling. The key idea is to determine the data correlation among a group of sensors based on the value of the data itself to significantly improve compression. Hence, this approach moves beyond traditional data compression schemes which rely only on spatial and temporal data correlation. A logical mapping, which assigns indices to nodes based on the data content, enables simple implementation, on nodes, of data transformation without any other information. The logical mapping approach also adapts particularly well to irregular sensor network topologies. We evaluate our architecture with both Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) on publicly available real-world data sets. Our experiments on both simulation and real data show that 30% of energy and 80-95% of the bandwidth can be saved for typical multi-hop data networks. Moreover, the original data can be retrieved after decompression with a low error of about 3%. Furthermore, we also propose a mechanism to detect and classify missing or faulty nodes, showing accuracy and recall of 95% when half of the nodes in the network are missing or faulty.

Key words: Distributed data compression, Error detection, Wavelet analysis, DCT analysis, Sensor networks, Irregular network

1 Introduction

With the continued development of sensor networking hardware, the ability to deploy large numbers of sensors is becoming possible. Typically, the sensor networks are deployed to gather environmental information over a period of time with the sensors working together to forward data to a central data sink. One of the main challenges with such sensor networking technologies is the need to minimize wireless packet transmissions in order to save power.

There are several basic ways to minimize the amount of traffic generated by the sensor network. Aggregation techniques such as TinyDB [1]and TAG [2] process and consume the collected data within the sensor network, forwarding only a small subset of the data to the sink. Query-based techniques such as directed diffusion aim to filter the data within the network to only what the application requires. Low-level networking techniques have been proposed in order to help route data within the sensor network with the hope of minimizing duplicated packets and minimizing the number of hops needed to deliver the data. Finally, data compression techniques are emerging for such sensor networks [3] [4] [5] [6] [7] [8].

Compression can be applied to a single data stream being generated by a single sensor [9]. The advantage of this approach is that the sensor will typically be generating similar data over time. The drawback, however, is that if the data from a single sensor is lost, then a significant amount of data may be lost. An alternative approach is to cluster the sensors together and compress the data across the sensors one snapshot at a time. The main advantage of this approach is that it is more resilient to transmission errors. At the same time, however, all the data needs to be transmitted at least once in order to be collected.

Correlation of data among sensors is determined not only by spatio-temporal proximity, but other factors as well. Building on this observation, we propose a *cluster-based* and *information-driven* architecture for a wide range of compression algorithms for scalar sensor data for a popular class of network of sensors. The key contributions of this paper are as follows.

- The exploration beyond spatial and temporal correlation of data in sensor networks. The key idea here is that correlation of the data is based on the value of the data itself rather than other factors, which we will show later are irrelevant in some cases.
- The information-driven architecture (RIDA) with a logical mapping framework for various compression and analysis algorithms which builds on the above observation. In this approach, data reported by sensors is observed over a short period of time. After that, the pattern of the data can be used to logically assign sensors with indices such that the correlation of data is utilized. Depending on the underlying compression algorithm, an appropriate logical assignment can be used.
- The design, implementation, and evaluation of different compression algorithms (1D and 2D, DCT-based and wavelet-based) on real sensor data.
- A resiliency mechanism in RIDA for missing and faulty nodes in sensor networks. We address a real practical problem in wireless sensor networks where nodes are frequently missing or faulty.

In the next section, we will review related work. Section 3 will point out some key observations about correlation of sensor readings that drive the design of our architecture. Section 4 will describe the proposed information-driven architecture for compression algorithms for sensor networking, including our proposed resiliency mechanism. Section 5 will describe the experiments that we conducted in order to show the efficiency of our approach. We discuss the limitation of our approach and future work, and conclude in section 6.

2 Related Work

In this section, we review related work on data compression with emphasis on data compression in sensor networks.

2.1 Data Compression

There are two main categories of data compression – *lossless* and *lossy* data compression. Lossless compression algorithms usually generate a statistical model of the data and map the data to bit strings based on the generated model. Meanwhile in lossy compression, data is often transformed into a new space using appropriate basis functions. In the new space, the data information or signal energy is usually concentrated in a few coefficients. Hence, compression can be achieved after quantization and entropy coding. For example, discrete fourier transform (DFT), discrete cosine transform (DCT), and discrete wavelet transform (DWT) are used extensively in most image compression applications (e.g. JPEG,JPEG2000). Audio and video compression also use predictive codecs, where previously decoded data is used to predict the current data and only the difference between the predicted and real data is encoded.

For sensor networks, the sensed data of the environment can also be modeled as an image of a temperature, humidity or light map and a standard image compression technique may be subsequently applied. However, sensor networks have some distinct features such as limited computation, distributed processing, degree of correlation and faulty readings, motivating new compression architectures and techniques tailored to meet their requirements. We briefly review recent work in the next section.

2.2 Data Compression in Wireless Sensor Networks

In Distributed Source Coding Using Syndromes (DISCUS), Pradhan *et al* [6] proposed a framework for distributed compression using joint source and channel coding. This approach minimizes the amount of inter-node communication for compression using both a quantized source and correlated side information within each individual node. While it shows an interesting theoretical approach, the choice of the correlated side information is essential to the performance of the algorithm and normally not well known in practice. Unlike this work, we have clearly verified our approach using real data report from sensors at Intel Research Lab at Berkeley.

Based on the recent result of Candes and Tao on near optimal signal recovery from random projections [10], Rabat *et al.* [7][8] propose a distributed matched source-channel communication architecture and reconstruction method from noisy random projections. A similar approach can be found in [8] which uses a gossip communication scheme. Although it is claimed to be universal, there is a trade-off between power-distortion-latency. In addition, they do not consider the correlation of the data itself.

Several methods have been proposed to use wavelets and their variants in analyzing and compressing the sensed data [11] [12][13][14]. Ganesan's DIMENSIONS [11] was one of the first systems addressing multi-resolution data access and spatio-temporal pattern mining in a sensor network. In DIMENSIONS, nodes are partitioned into different clusters and organized in a multi-level hierarchy. Within each cluster, the cluster head performs a two dimensional wavelet transform and stores the coefficient locally. These coefficients are in turn passed to the next level in the hierarchy for wavelet transform at a coarser resolution. While DIMENSIONS shows interesting results, it makes two main assumptions that we do not: (i) nodes are distributed in a regular grid and (ii) cluster heads can always communicate with their parents. Wagner[13][14] proposed an architecture for distributed wavelet analysis that removes the assumption about the regularity of the grid. In addition, an algorithm for performing the wavelet transform by tracing through the path in the minimum spanning tree and performing the wavelet filter along the path is proposed in [12]. It minimizes inter-node communication by transmitting partial coefficients forward and updating future sensors until the full coefficients are computed. It implicitly assumes that the path will be long enough in order to apply wavelet analysis effectively. Furthermore, it is not clear how to choose an optimal path for compression and the spatial correlation is not fully explored.

Few other works in distributed audio and video compression in wireless sensor networks can be found at [15][16][5]. Other approaches [17][18] try to solve multiple goals such as routing, aggregation, indexing and storage, and energy balancing with compression.

Our approach relies only on the sensing data itself. Therefore it does not make any assumptions about regularity of the network [11] or use any further information such as geographical location [13][14] or routing path [12]. In addition, it guarantees the optimal performance of compression algorithms instead of being universal [10][7][8]. We have also implemented and evaluated our architecture using real sensor data to verify that it works within typical sensor environments. Finally we proposed a resiliency mechanism to ensure a robust compression architecture in sensor networks.

3 Understanding Data Correlation

One of the main challenges of transformed data compression is to explore the correlation of data in time, space, or frequency domains. Most existing approaches try to organize sensors into groups based on spatial relationships in order to obtain some correlation of the readings. However, when we observed the readings over time of 54 sensors deployed at Intel Lab at Berkeley, we found out that (i) Sensors in similar environmental conditions that are not necessarily spatially correlated can report correlated data, (ii) Correlation of data may be independent from external factors such as sensor location and environmental conditions.

To illustrate these points, consider the spatial graph of the light sensor readings at night over time as shown in Fig. 1. As you can see, dark areas indicate high light intensity. Hence, sensors nearby opened windows report high readings due to the external light. These readings should be similar to those sensors nearby light sources inside the building. Hence, correlation exists due to the similarity of environmental factors as well as the sources. Spatial correlation can be seen as one specific case of this because nearby nodes can have similar condition. The converse, however, is not always true. In addition, Fig. 2 plots voltage readings of sensors. Intuitively, nodes with similar power level should be similar over time regardless of external environmental and spatial factors.

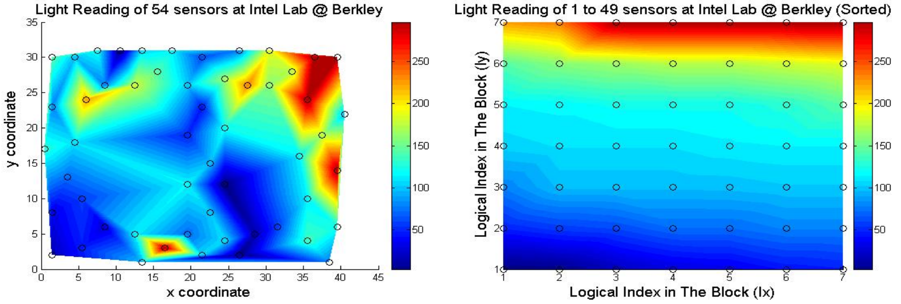


Fig. 1. Nodes nearby open windows and under light bulbs reporting similar reading

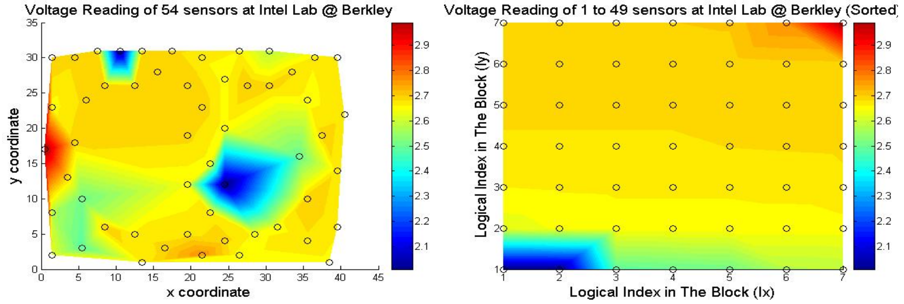


Fig. 2. Correlation of voltage readings is independent of external factors

From these findings, we believe that in order to explore the correlation of data, we should look at the information contained in the data itself rather than considering only attribute meta-data such as location and time. Once the underlying pattern of the data is found, we can assign nodes with appropriate logical indices to ensure the best performance of compression algorithms. The following section describes the information-driven architecture in detail.

4 RIDA: Robust Information-Driven Architecture

4.1 Key Assumptions

We are aware that our approach is only suitable for some types of sensor networks, which are characterized by the following assumptions. The network is fixed and can be partitioned into clusters. We also assume that the communication between any two nodes in a cluster takes one hop. This assumption can be relaxed in a hierarchical network topology. Furthermore, significant changes in the environment do not occur at high frequency eg. several times a day is reasonable. In addition, we only consider compression for scalar data. Finally, we assume the existence of cluster formation and synchronization protocols.

4.2 Overview

The system architecture consists of three main components; *information-driven logical mapping*, *resiliency mechanism*, and *compression algorithms*. In information-driven logical mapping, nodes within a cluster exchange their readings over a short period of time. During this period, each node learns the pattern of data of the whole cluster. A information-based logical mapping is designed allowing nodes to choose logical indices for themselves. The intuition is that nodes with correlated data should have logical indices near each other. Several mapping schemes will be discussed in more detail later.

The resiliency component involves detecting, isolating, and classifying faulty and missing nodes during the compression and decompression steps. The detail of this mechanism will be discussed in section 4.5. After the mapping is done, the data can be processed using logical indices. The compression algorithms block includes different compression techniques, which can be easily adapted to the architecture. Section 4.4 outlines the integration of two most popular data compression algorithms to the architecture. In general, nodes first broadcast their readings to the cluster so that each node has a snapshot of the data within each epoch. Individual node performs the data transformation and quantization itself. The coefficient the node has is the one having the corresponding index as the logical index. The node only sends its coefficient back to the server if it is non zero. At the sink or back-end server, original data can be reconstructed by decompression from the nonzero coefficients, classification of the missing data, and remapping to physical map of nodes.

4.3 Information-Driven Logical Mapping

The logical mapping gives nodes indices that can be used for data manipulation. This powerful idea keeps the architecture independent from other information such as nodes' locations while still preserving the advantages, such as data correlation, of having that information. The mapping can be formalized as follows.

$M : (N, N^n) \rightarrow L$

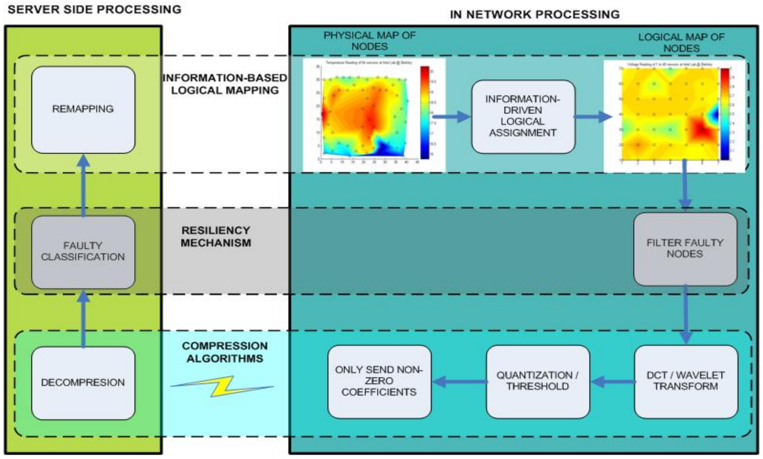


Fig. 3. Detailed System Architecture

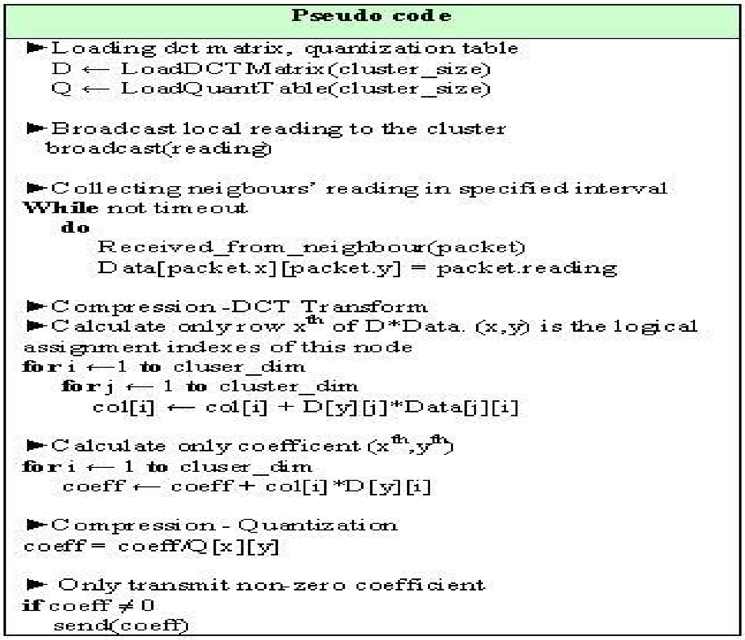
$M(d(s), D) = l$

Where: L is the logical index space. N is the natural set representing the value of sensor data. M notates the mapping from a sensor s to a logical index l such as (x, y) in 2D mapping. It uses only the value of the sensor data $d(s)$ and values of other sensors in the cluster D to determine l. The mapping can be application and algorithm specific. As a first step, we simply sort the data and index the nodes in sequence based on the order of the sorted data.

More specifically, the mapping within a cluster has the following steps. The cluster head broadcasts a *begin_mapping* message. Nodes within the cluster send their sensing data to the cluster head. The cluster head receives data from sensors for a short period of time. It then analyses the pattern of the data values and does the mapping accordingly. For example, in 1D sorted mapping, the cluster head sorts all the data values and sensor ids in ascending order and starts assigning indices sequentially. Once this step is done, the cluster head broadcasts the map and waits for all acknowledgements before sending *end_mapping*, which turns sensors into normal sensing mode.

4.4 Data Transformation

Various algorithms can be easily integrated with the architecture. We have adapted the discrete cosine transform, as well as the first and second generations (lifting scheme) of wavelet transform. Again, depending on the underlying compression algorithm, the logical mapping assigns indices to nodes appropriately. This ensures the flexibility of the architecture for a wide range of applications. In addition, since each node only needs to calculate the coefficient corresponding to its index, it does only the necessary operations. For example, in 2D-DCT, a node only multiplies the corresponding row and column in the block instead of doing a matrix calculation for the whole block. Likewise, in DWT, a node with



```
Pseudo code
► Loading dct matrix, quantization table
   D ← LoadDCTMatrix(cluster_size)
   Q ← LoadQuantTable(cluster_size)

► Broadcast local reading to the cluster
  broadcast(reading)

► Collecting neigbours' reading in specified interval
While not timeout
    do
        Received_from_neighbour(packet)
        Data[packet.x][packet.y] = packet.reading

► Compression -DCT Transform
► Calculate only row $x^{th}$ of D*Data. (x,y) is the logical
assignment indexes of this node
for i ←1 to cluser_dim
    for j ← 1 to cluster_dim
        col[i] ← col[i] + D[y][j]*Data[j][i]

► Calculate only coefficent ($x^{th}$,$y^{th}$)
for i ← 1 to cluser_dim
    coeff ← coeff + col[i]*D[y][i]

► Compression - Quantization
coeff = coeff/Q[x][y]

► Only transmit non-zero coefficient
if coeff ≠ 0
   send(coeff)
```

Fig. 4. Pseudo-Code For 2D-DCT

detail coefficient only needs to run the low pass filter with readings of logically nearby neighbors. Fig. 4 shows an example of distributed DCT.

4.5 Error Detection and Classification

Reliability of data is of paramount importance because network nodes fail frequently. Even when nodes have not failed, their operations are typically unstable. Fig. 5 shows the reading history of 54 sensors in a controlled environment. As observed, 53 out of 54 nodes are working. However, the number of nodes reporting data is always around 50% within each epoch. Better design of routing protocols could help increase this rate, but we still have to address the problem of actual faulty and missing nodes. This motivates us to design a simple mechanism to distinguish between missing data and real data at the sink.

All the nonzero data will be projected to an interval (for example [128,255]). The data of different types have different ranges. Although the data value is obtained from the same 10-bit ADC, the ranges of the data are different. Therefore the projection will unify the way we drop coefficients through quantization or thresholding. Missing readings will be set to zero. Hence, we have a set of data from [128,255] for normal data and 0 for missing data of all different scalar types like temperature, humidity, light and voltage. These zero values would result in low values in the reconstructed data. Hence, we can use a threshold to classify them. The threshold we used is 64 which has been shown to classify correctly

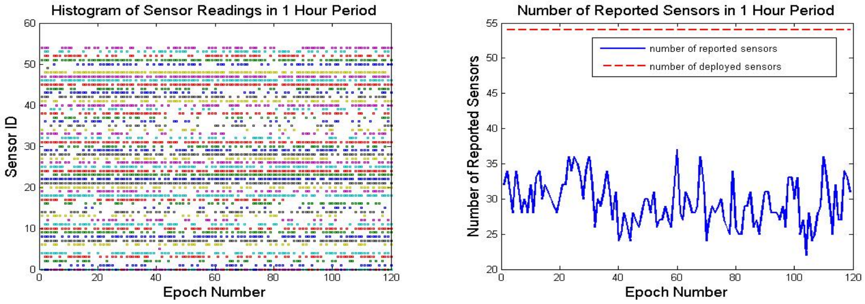


Fig. 5. Reading history over one hour period

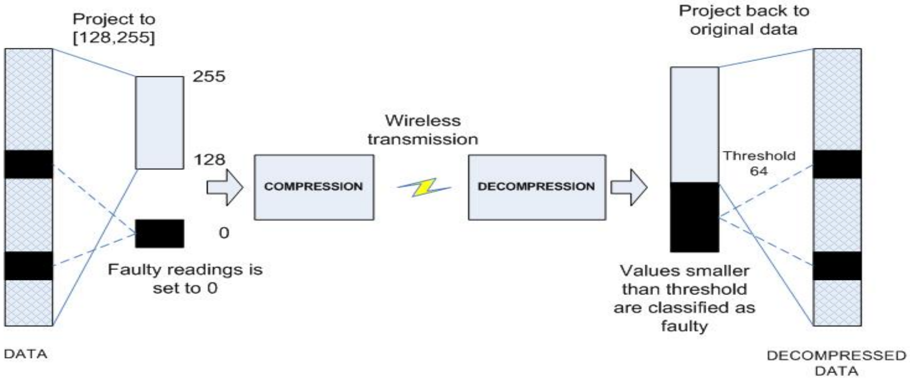


Fig. 6. Resiliency Mechanism

most of the time. Obviously, there is an inherent trade-off in the ability to detect missing readings and the decompression error.

5 Experimental Design and Analysis

This section describes how the experiments are setup to evaluate the architecture and discusses the results.

5.1 Goals and Metrics

The goals of the experiments in this section are four-fold.

- To understand how flexibly the information-driven architecture can adapt to different underlying algorithms, specifically compression algorithms.
- To understand how different compression algorithms perform on real sensor data with different logical mapping schemes.
- To understand how robust the architecture is to missing sensor data and failures using our proposed resiliency mechanism.
- To understand how much energy and bandwidth is saved in a typical multi hop network using our approach.

To evaluate the first goal, we will show that different compression algorithms such as DCT and DWT can be made distributed and integrated with the architecture. The system only needs to change the logical mapping scheme to apply the underlying algorithms.

The second goal is analyzed by observing the tradeoff between compression ratio and normalized mean squared error (MSE) of the compression algorithms using different mapping schemes. We used two main compression algorithms, DCT and DWT, and two simple mapping schemes, one dimensional ordered and two dimensional ordered mappings. Ideally, we aim for a configuration that results in high compression ratios with low normalized MSE.

To evaluate the third goal, we consider the *accuracy* and *recall* of the classification step against the number of faulty nodes. They can be calculated as:

$accuracy = \frac{TP+TN}{totalnumberofnodes}$

$recall = \frac{TP}{totalnumberofhealthynodes}$

where:

- *TP-True Positive* : Number of correctly classified healthy nodes
- *TN-True Negative* : Number of correctly classified faulty nodes

Therefore *accuracy* indicates how well the system can correctly classify healthy and faulty nodes, while *recall* represents the portion of correctly classified nodes in the set of nodes classified as healthy. Ideally, we wish to see the values of both accuracy and recall as close to 100% as possible.

Finally, we evaluated the energy consumption using PowerTOSSIM. The compression algorithm is implemented for the MicaZ platform and simulated in PowerTOSSIM. The energy consumption can be observed separately by measuring CPU operations and RF transmission. In order to understand how much energy is saved by doing compression in multihop networks, we use the following bench mark.

$c_b = n(t_x + t_r)h$
$c_c = n(t_x + t_r + d) + n'h(t_r + t_x)$
Where
c_b is the cost to transmit raw data back to the server.
c_c is the cost to transmit data back to the server using compression.
n is the cluster size. In the case of missing sensors, n is the number of healthy nodes.
h is the average hop count.
t_r, t_x are transmitting and receiving power for one package.
d is the cost to compress the data.
n' is the number of non-zero coefficients. n/n' is aproximately 20:1 for jpeg.

The energy saving is:

$$r_h = \frac{c_b - c_c}{c_b} = \frac{n(t_x + t_r)h - n(t_x + t_r + d) - n'h(t_r + t_x)}{n(t_x + t_r)h} \tag{1}$$

In the above equations, we do not consider the cost for mapping. However, as we have assumed previously that the frequency of changes in the environment is low, the mapping cost overall is negligible in comparison to the cost of collecting data. In addition, we only consider energy saving for one cluster because the percentage of energy saving in a fixed diameter network is independent of the number of clusters and determined by the hop count. Finally, we also assume there is no transmission loss for compressed data. However, one can expect that because less data is transmitted in the network, the transmission loss is smaller. Hence, in real world applications, we expect to see slightly higher error when loss in transmitting compressed data occurs.

5.2 Experimental Design

The experiments are designed based on the data collected from 54 sensors between February 28th and April 5th, 2004, which has been made available by Intel Berkeley Research Lab [19]. As discussed in the previous section, the number of sensors reporting data within each epoch is only around 50%. Hence, we decided to design two sets of experiments with two sets of data respectively.

The purpose of the first experiment set is to evaluate how different compression algorithms such as DCT and DWT perform on the real data. It also analyzes different to the mapping schemes such as 1D versus 2D and how robust the system is to the number of missing nodes. The raw data set has its missing values filled in via interpolation to create a complete data set. Thus, it creates an ideal sensor network data set, where every node reports readings within each epoch. In order to evaluate how robust the system is against node failures, we randomly insert faulty readings as zero values and perform the classification during the reconstruction phase.

The second set of experiments is to evaluate our approach on the real raw data without any interpolation. This set of data, as you can see from Fig. 5, has about 50% of its readings missing within each epoch. However, our experiments show that the system still achieves a reasonable compression ratio with low error and high detection rate.

5.3 Results and Analysis

Logical Mapping Schemes This section discusses several findings on different logical mapping schemes. Basically, there are two logical mapping schemes, 1D content-based and 2D content based mappings, where the data is sorted and indices are assigned based on the order of the data values reported by the sensors. These two mappings are evaluated against two location-based mappings where indices are assigned based on geographical relationships. Nodes which are close together have nearby indices in the block. As we can see in Fig. 7, DCT compression using information-based mappings outperforms those using location-based mappings. With the same compression ratio of 20:1, DCT compression using information-based mappings has a normalized MSE 50% less than location-based mappings. In addition, the 1D transform also gives lower errors in comparison to 2D transform. The normalized MSE is reduced by 30% if we use information-based mappings for voltage. The graph for temperature shows a transition when the compression ratio reaches 25. This is reasonable, because the data set has 49 nodes, so ideally a compression of lower than 25:1 should be considered. Compression ratios of over 25 mean that only one coefficient is left. Therefore, it would be pointless to compare those. This result is even clearer with wavelet transform. One limitation of the 2D transform is that the number of nodes within a cluster must be a square number. Clustering formation is a complex research area and so far no prior work has attempted to constrain the number of nodes in a cluster. In addition, due to the limit on number of nodes within a cluster, we would recommend compression should use 1D mappings.

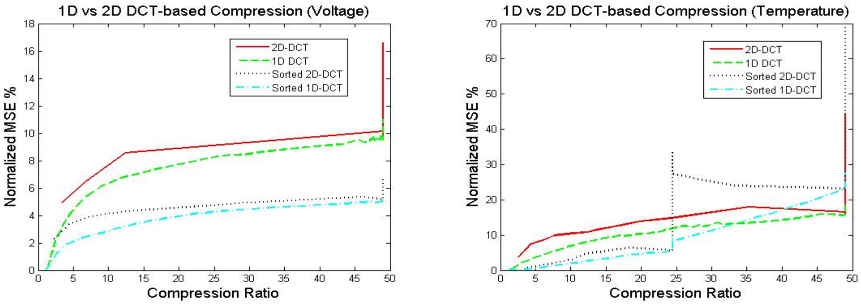


Fig. 7. Location-based mapping versus Content-based mapping and 1D versus 2D Transform

DCT-based and Wavelet-based Compression The wavelet-based compression in general shows much lower error that DCT-based compression. While the DCT-based approach shows an error of around 9%, the Wavelet-based approach has an error of only 3%, which is 67% less. However, due to the limit on the length of the data, wavelets with a high number of coefficients can start to diverge much sooner although they have a lower error with the same small compression ratio.

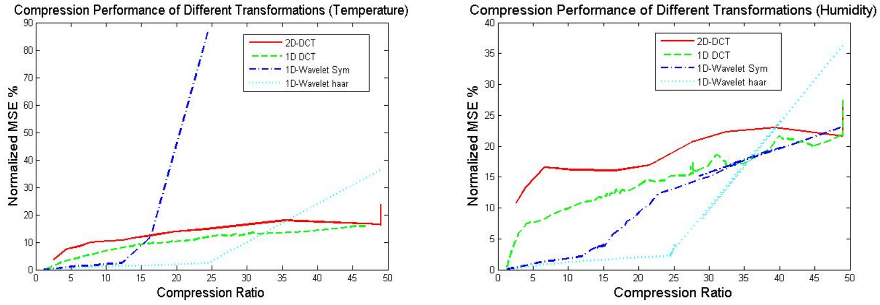


Fig. 8. DCT-based vs Wavelet-based Compression

Error Detection and Classification By the term faulty node, we mean to describe a node that sends odd data or no data at all. This is similar to a missing node, where the node is missing and does not send any data. Hence, we use the term faulty for both. Faulty data is randomly inserted into the data set before compression. The non-zero data is scaled to the [128,255] interval and we use a threshold of 64 to classify faulty data in both cases. When the number of faulty nodes increases from 1 to 30, DCT-based compression error increased dramatically from 6% to 45%. But it becomes stable around 45% when the

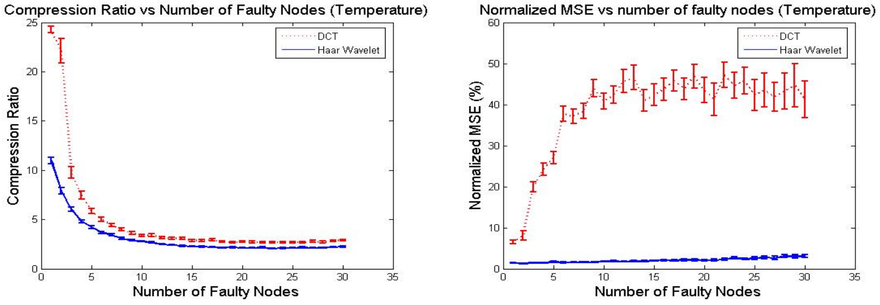


Fig. 9. Compression Performance on Temperature Readings

number of faulty nodes reaches above 10. Likewise, the error in wavelet-based approach only slightly increases from 2% to 4%. The compression ratio also decreases gradually from 10:1 to 3:1. This is reasonable because the nature of DCT-based transform is suitable for a smooth signal whereas Wavelet-based transform is more suitable with piecewise constant data. To our surprise, both DCT and Wavelet have very high accuracy and recall rates even when more than half the network is faulty. Haar wavelets can maintain a performance of up to 97% for both accuracy and recall. DCT is slightly lower but, still above 90% for accuracy and 97% for recall. Both of these values decrease gradually as the number of faulty nodes in the network increases. Similar results can be seen for other types of data such as humidity and voltage.

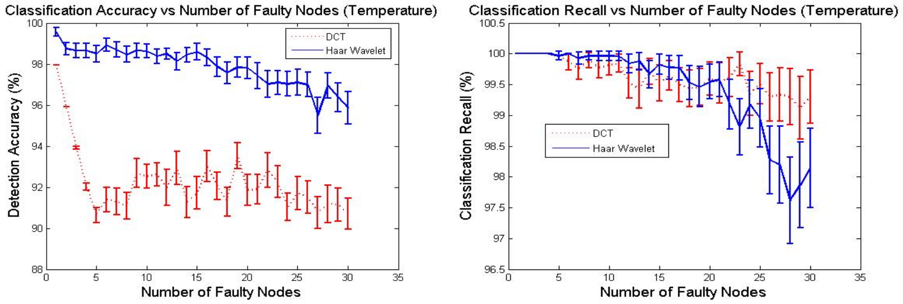


Fig. 10. Classification Accuracy and Recall on Temperature Readings

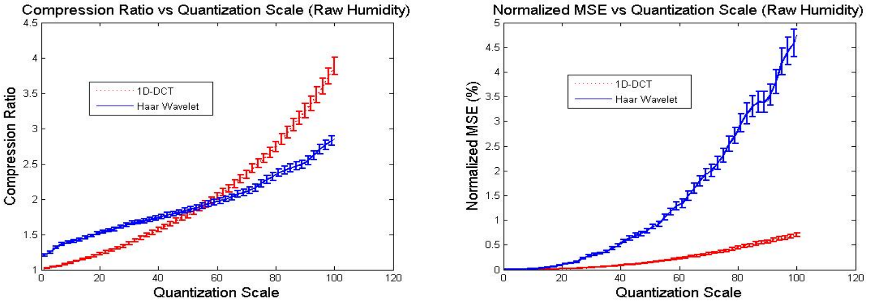


Fig. 11. Compression Performance on Raw Humidity Readings

Performance on Raw Data The data was collected using TinyDB, which queries data among sensors at the same time. However, collected data has a latency and dropping rate. One way to improve it is to design better routing and data aggregation protocols. However, these are still in development. Hence, we applied our system to this real set of data. Surprisingly, we still get the desired results. A compression ratio of 3:1 can be achieved for both DCT and Wavelet with an error less than 5% as shown in Fig. 11. Moreover, around 90% of nodes are still correctly classified and the recall rate is as high as 98%. In both cases, wavelet performs 3% better than DCT as shown in Fig. 12.

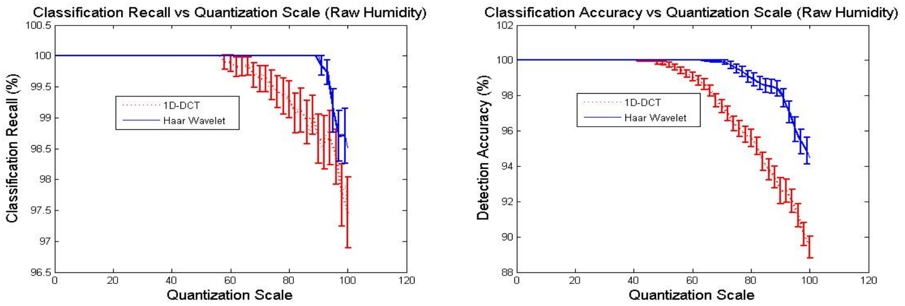
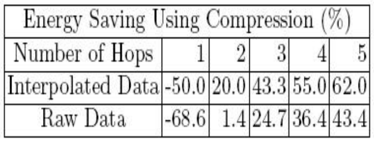


Fig. 12. Classification Accuracy and Recall on Raw Humidity Readings

Energy Consumption As mentioned at the beginning of the paper, one main purpose of data compression is to conserve energy and bandwidth. We have shown how our system can enable various compression algorithms and save a large amount of bandwidth by logically processing the data and sending only a few non-zero coefficients. We also state that the CPU operations consume much less energy than RF transmission. Indeed, Fig. 13 shows that the total energy consumed by the CPU operations including all normal activities and DCT transform is still only 2.5 times less than that of one RF transmission within each epoch. Hence, for multihop networks where the number of RF retransmissions is several times higher, our approach can be expected to save not only a large amount of bandwidth but also a significant amount of energy. Applying Eq. 1, we can know how much energy is saved for multihop networks as shown in Fig. 13. We have seen that different compression algorithms can be easily adapted to our architecture. Moreover, with the introduction of logical mapping, optimal performance can be simply tuned for different applications. In general, due to the limit on the number of nodes within a cluster, 1D mapping and transformation normally gives better performance than 2D mapping and transformation. In addition, Wavelet-based compression gives a lower error bound than DCT-based compression. It is surprising that the wavelet lifting scheme did not perform as well as expected. One of the reasons may once again be the limited length of the signal or the number of nodes within a cluster, correspondingly. Another surprise was that with our resiliency mechanism, the compression system becomes very robust even when half of the cluster is missing. Finally, although DCT transform and wavelet transform require an average amount of work load for Micaz class



Energy Saving Using Compression (%)					
Number of Hops	1	2	3	4	5
Interpolated Data	-50.0	20.0	43.3	55.0	62.0
Raw Data	-68.6	1.4	24.7	36.4	43.4

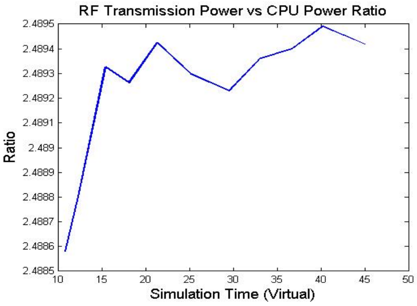


Fig. 13. Energy consumption of RF vs CPU

sensors, we still see that the energy saved by reducing the number of RF transmissions to CPU operations is 2.5. For an average 3-hop network, the energy can be saved by 30%. This ratio will be much higher in multihop networks where the number of RF retransmissions is proportional to the number of hops.

6 Conclusion and Future Work

In conclusion, we have presented RIDA, a novel distributed information-driven architecture for data compression for irregular sensor networks. The key idea is to assign the sensor nodes with logical indices based on the content of the data they report in order to optimally explore the correlation of the sensor data. This approach moves beyond conventional approaches, which have explored how to improve data compression by only exploiting spatial and temporal correlation. We have implemented and evaluated various popular data compression techniques such as DCT-based, Wavelet-based to the architecture. In addition, we also presented a simple method for detecting and classifying faulty nodes. The experimental results on real data show that our architecture can enable high compression ratios, low error and high robustness to faults.

Our current approach is limited to scalar data for environmental monitoring with low changing frequency. We also rely on the clustering structure and assume that the network is fixed. In the future, we would like to investigate further how this approach can be extended to meet the requirements for high rate data compression such as audio and images. Moreover, we would like to consider how it can be adapted to a network of mobile sensors. In addition, we would like to further study several factors that affect compression algorithms such as cluster size, quantization schemes, projection ranges and energy balancing as well as the tradeoff between compression and fault tolerance in sensor networks.

Acknowledgements

The research described in this paper was supported by National Science Foundation grants NSF 05-14818 and 01-21475.

References

1. Madden, S., J.Franklin, M., Hellerstein, J., Hong, W.: Tinydb: An acquisitional query processing system for sensor networks. ACM Trans. Database Syst. **30**(1) (2005) 122–173
2. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tag: A tiny aggregation service for ad-hoc sensor networks. In: OSDI. (2002)
3. Donoho, D.L.: Compressed sensing. In: IEEE Transactions on Information Theory. Volume 52. (2006) 1289–1306
4. Duarte, M.F., Wakin, M.B., Baron, D., Baraniuk, R.G.: Universal distributed sensing via random projections. In: Proceedings of IPSN 2006, Nashville, Tennessee, USA, April 19-21, 2006. (2006) 177–185
5. Gehrig, N., Dragotti, P.L.: Distributed sampling and compression of scenes with finite rate of innovation in camera sensor networks. In: Proceedings of Data Compression Conference, Snowbird, Utah (2006) 83–92
6. Pradhan, S.S., Kusuma, J., Ramchandran, K.: Distributed compression in a dense micro-sensor network. In: IEEE Signal Processing. Volume 19. (2002) 51–60
7. Rabbat, M., Haupt, J., Singh, A., Nowak, R.D.: Decentralized compression and predistribution via randomized gossiping. In: Proceedings of IPSN 2006, Nashville, Tennessee, USA, April 19-21, 2006. (2006) 51–59
8. Bajwa, W.U.Z., Haupt, J., Sayeed, A.M., Nowak, R.D.: Compressive wireless sensing. In: Proceedings of IPSN 2006, Nashville, Tennessee, USA, April 19-21, 2006. (2006) 134–142
9. Sadler, C.M., Martonosi, M.: Data compression algorithms for energy-constrained devices in delay tolerant networks. In: Proccedings of ACM Sensys, Boulder, Colorado (2006)
10. Candes, E., Tao, T.: Near optimal signal recovery from random projections: universal encoding stratergies? In: preprint. (2004)
11. Ganesan, D., Estrin, D., Heidemann, J.: Dimensions: Why do we need a new data handling architecture for sensor networks. (2002)
12. Ciancio, A., Ortega, A.: A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting. In: Proceedings of ICASSP, (Philadelphia, PA)
13. Wagner, R., Choi, H., Baraniuk, R., Delouille, V.: Distributed Wavelet Transform for Irregular Sensor Network Grids. In: IEEE Workshop on Statistical Signal Processing (SSP), Bordeaux, France (2005)
14. Wagner, R.S., Baraniuk, R.G., Du, S., Johnson, D.B., Cohen, A.: An architecture for distributed wavelet analysis and processing in sensor networks. In: Proceedings of IPSN 2006, Nashville, Tennessee, USA, April 19-21, 2006. (2006) 243–250
15. Roy, O., Vetterli, M.: Distributed Compression in Acoustic Sensor Networks Using Oversampled A/D Conversion. In: IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP). Volume 4. (2006) 165–168
16. Gehrig, N., Dragotti, P.L.: Distributed compression in camera sensor networks. In: Proceeding of MMSP, Siena, Italy (2004)
17. A, S.: Routing and data compression in sensor networks: Stochastic models for sensor data that guarantee scalability. In: Proccedings of ISIT2003, Yokohama, Japan (2003)
18. Petrovic, D., Shah, R.C., Ramchandran, K., Rabaey, J.: Data funneling: routing with aggregation and compression for wireless sensor networks. In: Proceedings of SNPA 2003, Seattle, WA (2003) 156–162
19. Lab, I.B.R.: (http://db.lcs.mit.edu/labdata/labdata.html)