Compression in Wireless Sensor Networks: A Survey and Comparative Evaluation

M. A. RAZZAQUE, Universiti of Teknologi Malaysia
CHRIS BLEAKLEY, University College Dublin
SIMON DOBSON, University of St Andrews

Wireless sensor networks (WSNs) are highly resource constrained in terms of power supply, memory capacity, communication bandwidth, and processor performance. Compression of sampling, sensor data, and communications can significantly improve the efficiency of utilization of three of these resources, namely, power supply, memory and bandwidth. Recently, there have been a large number of proposals describing compression algorithms for WSNs. These proposals are diverse and involve different compression approaches. It is high time that these individual efforts are put into perspective and a more holistic view taken. In this article, we take a step in that direction by presenting a survey of the literature in the area of compression and compression frameworks in WSNs. A comparative study of the various approaches is also provided. In addition, open research issues, challenges and future research directions are highlighted.

Categories and Subject Descriptors: C.2.3 [**Computer-Communication Networks**]: Network Protocols; [**Computer-Communication Networks**]: Distributed Systems

General Terms: Algorithms, Measurement, Performance

Additional Key Words and Phrases: Wireless sensor networks, compression

ACM Reference Format:
Razzaque, M. A., Bleakley, C., and Dobson, S. 2013. Compression in wireless sensor networks: A survey and comparative evaluation. ACM Trans. Sensor Netw. 10, 1, Article 5 (November 2013), 44 pages.
DOI: http://dx.doi.org/10.1145/2528948

5

1. INTRODUCTION

Wireless sensor networks (WSNs) are critically resource constrained by limited power supply, memory, processing performance, and communication bandwidth [Akyildiz et al. 2002]. Due to their limited power supply, energy consumption is a key issue in the design of protocols and algorithms for WSNs. Typically, energy consumption is dominated by radio communication [Pottie and Kaiser 2000; Barr and Asanović 2006]. The energy consumption of radio communication is directly proportional to the number of bits of data, that is, data traffic, transmitted within the network [Heinzelman et al. 2000]. Therefore, using compression to reduce the number of bits to be transmitted has the potential to drastically reduce communication energy costs and so increase network lifetime. Similarly, sampling-level [Candès and Wakin 2008; Haupt et al. 2008] and communication-level [Lu et al. 2010; Tulone and Madden 2006] compression can reduce energy costs in WSNs and increase network lifetime. In most cases, the savings due to compression are greater than linear, since reducing the number of bits transmitted has the knock-on effect of reducing link-level congestion, which in turns reduces

Corresponding author's address: M. A. Razzaque, Faculty of Computing, Universiti of Teknologi Malaysia, Skudai, JB, Malaysia; email: marazzaque@utm.my.

DOI: http://dx.doi.org/10.1145/2528948

the number of collisions and re-tries in the network. Consequently, researchers have been investigating optimal algorithms for compression of sensed data, sampling, and communications in WSNs.

Unfortunately, most conventional compression algorithms are not directly applicable to WSNs. First, in conventional compression approaches, the key objective is to save storage, not energy. In WSNs, energy is more important than memory. Thus, energy saving is the primary evaluation metric. Second, it has been shown [Sadler and Martonosi 2006] that, in terms of energy consumption, transmission of just one byte of data is equivalent to execution of roughly four thousand (Chipcon CC2420) to two million (MaxStream XTend) instructions. These calculations only consider local energy consumption at the compressing node; network-wide energy savings due to compression can further compensate for the energy expense of compression. Thus, compression algorithms with some degree (low or medium) of computational complexity are worth exploring. On the other hand, excessively computationally complex algorithms are not worth pursuing. Finally, conventional compression algorithms, originally designed for desktops or servers, must be restructured to reduce code size and dynamic memory usage due to the limited memory capacity of WSN nodes—typically less than 50 kB for code memory and even less for data memory. Recently, researchers have addressed these challenges by adapting conventional compression techniques and, in some cases, by proposing new approaches.

Compression in WSNs is a very active research area. Papers published in this area are highly diverse in their approaches and implementations. To our knowledge, there are only two articles [Kimura and Latifi 2005; Srisooksai et al. 2012] which provide survey of the area. However, Kimura and Latifi [2005] is out of date and does not report recent, dominant, work in the field, whereas, the more recent survey [Srisooksai et al. 2012] focuses only on pure data compression techniques. It excludes aggregation from the list of data compression techniques due to its route dependency. However, the issue of the interdependency between compression and routing [Scaglione and Servetto 2002; Pattem et al. 2004] is well known in WSNs. A number of papers have reported its effect on compression schemes, including Scaglione and Servetto [2002] for distributed source coding (DSC), Lee et al. [2009] and Quer et al. [2009] for compressed sensing (CS), and Ciancio et al. [2006] and Shen and Ortega [2008a] for transform coding. Moreover, Srisooksai et al. [2012] classified data compression techniques into distributed (exploits spatial correlation) and local (exploits temporal correlation) approaches for dense and sparse networks, respectively. However, in dense networks, spatiotemporal correlation allows use of both the distributed and local approach [Chu et al. 2006; Baron et al. 2005, 2009]. This survey also presents CS as a distributed approach, but CS exploits intra-signal structures (temporal correlation) within a node. Furthermore, distributed CS (DCS) that exploits inter-signal or spatial correlation [Baron et al. 2005, 2009] is missing from the paper. Considering these points, we feel now is an appropriate time to put recent works into perspective and take a holistic view of the field. This article takes a step in that direction by presenting a survey of the literature in the area of compression in WSNs focusing on current, state-of-the-art research. A comprehensive overview of compression techniques in WSNs is provided together with a comparative study of the various approaches. Finally, this work points out open research challenges and recommends future research directions.

Section 2 presents the requirements for compression in WSNs and a brief introduction of different compressions in WSNs. Section 3 provides an overview of existing approaches to compression in WSNs along with a comparative study in Section 4. Open research challenges and suggestions for future research directions are presented in Section 5. Finally Section 6 concludes the work and points to areas of potential future work.

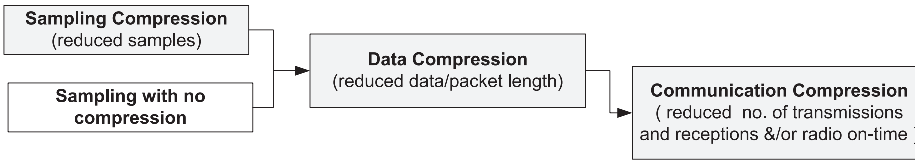


Fig. 1. Hierarchical relationship between types of compression.

2. COMPRESSION IN WIRELESS SENSOR NETWORKS

This section introduces the different types of compression algorithms, examines application requirements for compression, and lists the features of common compression algorithms.

2.1. Types of Compression

In WSNs, the main objective of compression is to reduce energy consumption. Sensing/sampling, computation, and communication are the three operations mainly responsible for energy consumption in WSNs. Any technique that directly or indirectly reduces one or more of these operations, while meeting application requirements (e.g., distortion, complexity), can be considered as compression. Based on this, compression in WSNs can be classified as follows.

Sampling Compression (SC). SC is the process of reducing the number of sensing/sampling operations while keeping network coverage (for spatially correlated sensors) and /or distortion loss within an acceptable margin. A number of research works [Cardei et al. 2005; Subramanian and Fekri 2006] exploit spatially correlated data to reduce the sensing tasks. These works primarily focus on keeping the sensors in a sleep state, while a minimal number of sensors are kept active within a group. These are not the concern of this article. In contrast, CS [Candès and Wakin 2008; Haupt et al. 2008] approaches perform the sampling-level compression by exploiting temporal data correlations at a sensor node.

Data Compression (DC). Data compression is the process of converting an input data stream (the source stream or the original raw data) into another data stream (the compressed stream) that has fewer bits. It can be viewed as the process of discovering structure that exists in the data and eliminating it by using more efficient encoding. All nonrandom data has some structure, and this structure can be exploited to obtain a more compact representation of the data, that is, representation wherein no structure is noticeable. The terms redundancy and structure are used in the professional literature and are interchangeable [Salomon 2007; Sayood 2006]. Most work on WSN compression (e.g., predictive coding, DSC, transform coding) supports data-level compression.

Communication Compression (CC). Typically, this is the process of reducing the number of packet transmissions and receptions, hence reducing the radio on-time of transceivers a WSN. The longer the packet to be transmitted or received, the greater the radio on-time of transceivers [Kimura and Latifi 2005; Salomon 2007; Barr and Asanović 2006]. Hence reduced packet or data size (e.g., data compression) reduces radio on-time and reduces communication cost in WSNs. Aggregation, DCS, and predictive coding support communication-level compression.

Usually, there is a hierarchical relationship between the aforementioned types of compressions (as shown in Figure 1). For instance, a reduced number of samples helps in reducing the data/packet length (data compression), which ultimately reduces the radio on-time of the transceivers (communication compression). It is desirable to have compression techniques, which support these three levels of compressions.

Unfortunately, very few of the existing compression techniques do so. As shown in Figure 1, data compression may operate on compressed samples or noncompressed samples.

2.2. Requirements

WSNs are used in a wide range of applications. This leads to a diverse range of requirements for compression algorithms. For example, mission-critical applications, such as health monitoring, battlefield, and fire rescue, provide real-time user information and so can tolerate only bounded latency and data loss. In contrast, other applications, such as habitat monitoring, may tolerate significant latency and accept certain losses or distortions in the data presented at the sink. Considering this, we classify the requirements of compression in WSNs in two ways: (i) generic and (ii) application specific. In the following, we summarize the key elements of each category and refer to these in later sections of this article when analyzing the compression algorithms.

2.2.1. Generic Requirements. This section summarizes the generic requirements for compression in WSNs.

Computational Complexity and Memory Requirements. Typically, WSN nodes are equipped with limited processing and memory capability. For instance, popular WSN nodes, for example, Mica, TelosB, and Tmote Sky, are equipped with Atmel Atmega128L and Texas Instruments MSP430 micro-controllers (4-8 MHz clock speed), which have instruction and data memories of only 128 and 48 KB, respectively [Wikipedia 2012]. Given these limitations, it is essential to design a low-complexity and small code-size (light-weight) compression algorithm for WSN applications. With these limitations, all but the simplest of data compression schemes can be challenging to implement in WSNs [Barr and Asanović 2006; Kimura and Latifi 2005; Sadler and Martonosi 2006]. Algorithms with asymmetric computational complexity are often desirable, whereby most computation takes place at the decoder (sink), rather than at the encoder (sensor nodes), thus sensors with minimal computational performance can efficiently compress data.

Communication Requirements. Since radio communication consumes a significant amount of node energy [Sadler and Martonosi 2006; Karl and Willig 2005], compression algorithms are typically designed to eliminate or reduce the redundant information exchange between nodes. Unlike conventional communication networks, the purpose of communication in WSNs is not only moving bits from one node another. Rather, a WSN is expects to provide meaningful information and/or actions: "People want answers, not numbers" [Huang 2003]. This means more processing and less communication. So, if possible, compression techniques should minimize communication at the cost of increased computation both at the decoder and encoder.

Redundant Sensing. In some scenarios, the sensing coverage of nodes may overlap, leading to the acquisition, communication, and storage of redundant, perhaps duplicate, information. Compression techniques can be used to identify and exploit this redundancy to reduce the amount of data sensed and transmitted. Typically, these approaches use internode communication to establish sensing schedules with a reduced frequency of observation. The missing data can be imputed at the sink based on known data relationships and/or decompression techniques. WSNs, which employ energy-expensive sensors, benefit most from this form of compression.

On-Route Compression. Conventional compression algorithms compress data at the source and decompress at the destination only. In contrast, some WSN applications require that the data is available at intermediate nodes for en-route in-network processing or transformation, for example, for aggregation or transcoding. Compression schemes allowing on-route compression need to be sufficiently flexible to allow the inspection, modification, addition and/or removal of data at intermediate nodes. On-route compression algorithms can be particularly effective for heterogeneous networks consisting of different types of nodes. Lightweight compression at low-performance nodes can be combined with more powerful compression or processing at higher-performance or mains-powered routing nodes.

Reliability. Reliability in WSNs has two aspects: communication reliability and data reliability [Kim 2004; Brown and Sreenan 2007]. Data reliability can be improved by exploiting spatial redundancy in sensor measurements. Communication reliability can be improved by exploiting measurement redundancy or by adding error checking bits. In contrast, compression techniques aim to reduce redundancy in order to increase energy efficiency. Clearly, there is an interplay or dependency between reliability and compression.

Robustness. Node failure, due to power shortage or physical damage, and link failure, due to unreliable wireless communication, are common phenomena in WSNs. Compression techniques in WSNs need to be robust enough to work properly even if there is a failure. To tolerate node and link failure, redundant deployment is necessary, which clearly conflicts with one of the key requirements (redundancy removal) of compression. For robustness, we need reliable communications or reliable topology or both [Nath et al. 2008], so a trade-off between robustness and energy efficiency in WSNs may be needed.

Scalability. WSN applications range from small numbers of nodes to large numbers (tens to thousands, even hundreds of thousands) [Karl and Willig 2005]. Hence, compression techniques must scale with network size.

2.2.2. Application-Specific Requirements. WSNs have highly diverse applications with diverse requirements. In the following, we briefly describe these diverse and application specific requirements.

Real-Time vs. Non-Real-Time. WSN applications, which provide real-time user data or control solutions, such as in healthcare or intelligent transport systems (ITSs), require bounded latency. Therefore, compression may need to be performed one sample at a time. This can limit the compression ratio achieved. However, spatial correlations can still be exploited. Non real-time compression allows processing of data from several sampling periods in a single batch and transfer in-bulk. This can significantly increase the compression ratio.

QoS-Awareness. Generally, a WSN provides services to its users by providing information about the environment where it is deployed. So, in WSNs, quality of service (QoS) also means quality of information (QoI). In WSNs, what is relevant is the amount and quality of information that can be extracted at given sinks/decoders about the observed objects or environment [Karl and Willig 2005]. Typical QoS/QoI metrics in WSNs include timeliness, reliability, and distortion. The relative importance of these aspects of QoS [Chen and Varshney 2004] is application dependent, for example, timely delivery of compressed data to the sink is more important in real-time applications. Due to the removal of redundancy and approximation in compression, it is often difficult to maintain these QoS/QoI metrics.

Security. Most WSN applications (e.g., Body Sensor Networks) require a certain degree of security [Perrig et al. 2004]. However, security and data compression algorithms may conflict. For example, security protocols require that sensor nodes encrypt sensed data prior to transmission and decryption and authentication is only performed at the base station (sink). In contrast, most data compression protocols (e.g., aggregation, wavelet-transform) process plain text data at intermediate nodes so that energy efficiency is maximized. In addition, lossy compression results in alterations to the sensor data making authentication difficult. Hence, data compression and security protocols should be codesigned so that compression can be performed without sacrificing security.

2.3. Features

A list of typical features of compression for WSNs is provided.

Lossless vs. Lossy. Some compression algorithms are designed to support exact reconstruction of the original data after decompression (lossless). In other cases, the reconstructed data is only an approximation of the original (lossy). Use of a lossy algorithm may lead to loss of information, but generally ensures a higher compression ratio.

Distortion vs. Accuracy. In the case of lossy compression, there is a trade-off between the data rate (R) achieved and the distortion (D) in the reconstructed signal. Mean Square Error (MSE) is a natural distortion metric. However, MSE can be misleading, since different types of distortion may have very different effects on the statistical inferences, which can be drawn after decompression. In addition, the energy consumption of communication should be taken into account. In order to address this issues, previous work has proposed the use of a rate-energy-accuracy (R-E-A) metric [Chen 2006].

Data Aggregation. In some applications, only a summary of the sensor data is required. For example, statistical queries, such as MIN, AVG, MAX, allow for compact responses from the sensors. However, the original sample values cannot be reconstructed from the summarized representation. Aggregation requires in-network processing of sensor data but can greatly reduce communication overhead.

Data Correlation. Since sensor nodes are normally deployed in close proximity, correlations between the sensed values at different nodes is often high (spatial correlation). Furthermore, since sensors observe events in a continuous manner, observed successive discrete signal samples often exhibit high correlation (temporal correlation). WSN compression algorithms typically exploit these correlations in order to improve the compression ratio achieved.

Symmetric vs. Asymmetric. In the case of symmetric algorithms, the computational complexity of compression and decompression are similar. In the asymmetric case, compression and decompression have different computational complexity. Traditional schemes tend to have higher computational complexity on the compression side. In contrast, in WSNs, it is desirable that compression, which is typically performed on the motes, is low complexity and that decompression, which is typically performed at the sink, is high complexity.

Nonadaptive vs. Adaptive. In nonadaptive compression, the compression operations and parameters are fixed. This type of compression is suitable for stationary data, that is, when the statistics of the data do not change with time. In contrast, adaptive or dynamic compression methods monitor the raw data statistics and modify their operation and/or parameters in order to improve performance [Lee and Jung 2010]. This approach is more complex but provides better performance for nonstationary data.

3. SURVEY OF EXISTING COMPRESSION ALGORITHMS IN WSNS

Compression is key in reducing the energy consumption in WSNs. Consequently, a large number of compression techniques have been proposed in the literature. Herein, existing works are categorized based on the compression technique utilized. The following sections summarize text-based compression, data aggregation, distributed source coding, transform-based compression, compressive sensing, and predictive coding and their variants.

3.1. Text-Based Compression

The dictionary-based Lempel-Ziv-Welch (LZW) algorithm [Welch 1984] is a popular lossless compression scheme for text data. It encodes new strings based on previously encountered strings. Research works, which address the use of dictionary/text-based compression in WSNs are few in number. S-LZW [Sadler and Martonosi 2006] is the only work, to our knowledge, which explicitly adopts the LZW concept to reduce data transmission in WSN. S-LZW treats sensed data as strings and divides the strings into fixed-size blocks, with each being compressed using the LZW algorithm. Although S-LZW is appropriate for sensor nodes, it does not take specific advantage of sensor data characteristics, especially the spatial and temporal correlations, which exist in sensed data. Sensor data tends to be repetitive over short intervals. Even sensor data, which exhibits large sudden changes in value, tends to be repetitive over consecutive samples due to the use of high sampling rates designed to allow accurate capture of these sudden changes. S-LZW was optimized for these situations by means of a Mini-Cache (S-LZW-MC) [Sadler and Martonosi 2006]. In this approach, the most important design decision is the size of the mini-cache. Results show that, in most scenarios, S-LZW-MC with 32 mini-cache entries outperforms basic S-LZW.

The S-LZW-MC algorithm conserves energy by taking advantage of the characteristic locality patterns of sensor data through use of the Burrows-Wheeler Transform (BWT) [Burrows et al. 1994]. In this approach, BWT is utilized as a data preconditioning step before application of S-LZW. Due to the computational complexity of the method, it does not provide any improvements in energy consumption for nodes with short range radios (CC2420) but does provide savings for nodes with medium and long-range radios at the cost of computational complexity. For structured datasets (e.g., SensorScope [EPFL 2008], Intel Dataset [Intel Berkeley Research Lab 2004]), preconditioning using the Structured Transpose (ST) has been shown to be more effective than using BWT [Sadler and Martonosi 2006]. Use of ST shows reasonable improvements in terms of computational complexity and energy savings compared to basic S-LZW.

In summary, S-LZW and its variants are good compression algorithms for WSNs with very little or zero spatial and temporal data correlations as they are not designed to exploit these correlations during compression.

3.2. Data Aggregation

Data aggregation [Rajagopalan and Varshney 2006; Alzaid et al. 2008] is the simplest in-network processing technique for data and communication compression in WSNs. In certain WSN applications, it is not necessary or efficient for all sensors to transmit the data directly to the sink since data generated by sensors in close proximity is often redundant and spatially correlated. Data aggregation combines or fuses data from nearby sensors into high-quality summary information that is then transmitted to the sink, resulting in conservation of energy and bandwidth. The benefits of aggregation are determined by the distances between the fused data sources relative to that between the sources and the sink and by the size of the summary data relative to that of the original data. For maximum benefit, it is desirable that the aggregator is close to the sources and that the routing paths from the sources to the sink pass through the aggregator. This leads to the research problems of determining the optimal aggregation tree/structure and finding the optimal aggregation function [Ozdemir and Xiao 2009; Karl and Willig 2005].

A large number of papers [Heinzelman et al. 2002; Younis and Fahmy 2004; Lindsey et al. 2002; Madden et al. 2002; Nath et al. 2008], including some good reviews [Rajagopalan and Varshney 2006; Fasolo et al. 2007; Alzaid et al. 2008; Ozdemir and Xiao 2009], have been published on data aggregation in WSNs. In the following, we summarize some key works in this area.

Sensor network architectures (SNAs) play a vital role in determining the performance of data aggregation protocols. Generally, in flat networks, data aggregation is accomplished by data-centric routing and a sink-initiated query message. The Sensor Protocol for Information via Negotiation (SPIN) [Kulik et al. 2002; Krishnamachari and Heidemann 2004] based on push diffusion is one of the earliest works on data aggregation, which shows significant energy savings compared to flooding. A secure version of SPIN is presented in Xiao et al. [2006b]. Global knowledge requirements and the inability to guarantee data delivery are the main disadvantages of SPIN protocols. Two-phase pull-diffusion-based directed diffusion (DD) is another key approach to data aggregation in flat SNAs [Intanagonwiwat et al. 2000]. Use of reliable communication makes reliable DD [Stann and Heidemann 2003] robust at the cost of higher energy consumption. Unlike SPIN, it is not necessary to maintain a global network topology in directed diffusion. However, it is inappropriate for applications, which require continuous data delivery to the sink.

Excessive communication and computation of flat SNAs can be avoided using hierarchical data aggregation [Heinzelman et al. 2000, 2002; Younis and Fahmy 2004]. Generally, in hierarchical data aggregation (e.g., cluster-based, chain-based, and tree-based), data fusion occurs at special designated nodes, reducing the number of messages transmitted [Rajagopalan and Varshney 2006]. Low Energy Adaptive Clustering Hierarchy (LEACH) and Hybrid Energy Efficient Distributed Clustering Approach (HEED) are the two key cluster-based aggregation techniques [Heinzelman et al. 2000, 2002; Younis and Fahmy 2004]. LEACH provides improvements in lifetime and accuracy compared to the direct approach, but it assumes that all sensors are homogeneous in power and capacity, which might not be valid in WSNs. LEACH-Centralized [Heinzelman et al. 2002] overcomes this problem and performs better than LEACH. Unlike LEACH, HEED selects cluster heads based on a combination of node residual energy and proximity to its neighbors. It shows a better network lifetime than LEACH and achieves a geographically well-distributed set of cluster heads. However, the requirement for multiple power levels at sensor nodes is a hindrance to widespread adoption. In cluster-based WSNs, nodes further from a cluster head may require excessive energy in communication. Chain-based aggregation method, like Power Efficient data GAthering protocol for Sensor Information Systems (PEGASIS) [Lindsey et al. 2002], solve this problem by transmitting only to nearest neighbors. PEGASIS is more energy efficient compared to LEACH but suffers due to global knowledge and homogeneity of nodes requirements.

In tree-based SNAs, data aggregation is performed at intermediate nodes in the tree, and aggregated data is transmitted to the root node. Tree-based Tiny AGgregation (TAG) [Madden et al. 2002] uses a generic aggregation service especially designed for TinyOS based WSNs and monitoring applications. It is energy efficient but suffers due to periodicity requirements and lack of robustness. Power Efficient Data gathering and Aggregation Protocol (PEDAP) [Tan and Körpeoğlu 2003] utilizes tree-based SNAs. Minimum-spanning-tree-based PEDAP is a very promising approach that uses load balancing to maximize network lifetime. Even with time complexity of $O(n^2)$, the power-aware version of PEDAP(PA-PEDAP) [Tan and Körpeoğlu 2003] can significantly improve the lifetime of LEACH or PEGASIS. Unfortunately, it relies on centralized operation and global knowledge. The popular directed diffusion method also exploits tree-based SNAs. However, aggregating along a tree is highly vulnerable to node and transmission failures, which are common in WSNs [Madden et al. 2002]. This is because there is only a single path in the tree from a source to the sink node. In order to overcome robustness problems in tree-based aggregations, gossip-based techniques [Boyd et al. 2006; Dimakis et al. 2006] can be used. However these are not energy efficient. In Nath et al. [2008], the Synopsis Diffusion protocol solves these problems through a multipath approach. Use of multipath routing makes the relation between aggregation and the required routing topology loosely coupled, which ultimately makes Synopsis Diffusion robust and energy efficient. A hybrid approach, the Tributaries and Deltas (T and D) protocol [Manjhi et al. 2005] tries to resolve the problems of both tree and multipath structures by combining the best features of both schemes. This algorithm may suffer due to the high overhead incurred in updating the data gathering structure.

Aggregation techniques (e.g., [Zhu et al. 2008]) which exploit correlation can capture more information about the source data than their counterparts, but the overheads involved in acquiring the correlation information is potentially prohibitive. Hence, most existing aggregation schemes do not exploit correlations and fail to maximize their compression ratio. The trade-off between these approaches needs to be understood in order to choose the most effective approach for a given application.

To make aggregation useful in real applications, it is important that data quality requirements are satisfied and the error introduced by aggregation is below a specified threshold. Work on QoS-based aggregation protocols seeks to provide some guarantees on the QoS achieved. The algorithm proposed in Sadagopan and Krishnamachari [2004] and Ordonez and Krishnamachari [2004] tries to maximize the amount of information collected at the sinks subject to constraints on energy, latency, and data flows. In contrast, Application Independent Data Aggregation (AIDA) [He et al. 2004] performs aggregation adaptively so as to control congestion and achieve end-to-end reliability. AIDA can reduce end-to-end delay and transmission energy significantly under heavy traffic conditions compared to a 'no aggregation' scheme. However, the approach may be too complex for resource-constrained sensor nodes. Cappiello and Schreiber [2009] present an aggregation-based compression technique which integrates QoS awareness as well as energy awareness. QoS parameters include accuracy, precision, and timeliness. The initial results are encouraging but are only limited to linear compression algorithms. A recent paper [Jeong et al. 2010], presents a lossless aggregation protocol, called Lump, which employs various properties of packets to not only support QoS but also maximize the Degree of Aggregation (DoA). Since it is a lossless protocol, the DoA is limited.

Table I[1] summarizes the key aggregation protocols. Data aggregation in WSNs significantly reduces energy consumption by only transferring a summary of the sensed values to the sink. As such, the technique sacrifices a lot of information about the measured values. Hence, the technique is limited to applications which can tolerate extreme data loss.

3.3. Predictive Coding

Statistical-model-based sensor data predictions or estimations at the sink or base station are promising ways of compressing data and communications in WSNs. In predictive coding (PC), the inherent temporal correlation between consecutive readings at an individual sensor is used to predict future observations at the sink based on the

[1]Considering the space available, we have excluded the references; please use the references from the discussion of the schemes. This also applies to other tables of this section.

Table I. Summary of the Key Aggregation Protocols

Protocol	Key Features	Advantages	Limitations
SPIN	Push diffusion, flat SNAs, sink driven	Reliable and Secure versions available	Global knowledge needed, no guarantee on data delivery
DD/Reliable DD	Push diffusion, flat/tree SNAs, sink driven	Medium Scalability and high Robustness	High aggregation structural cost
LEACH	Cluster-based, distributed/centralized	Medium aggregation structural cost, Energy efficient	Low Scalability and Robustness
PEGASIS	Chain-based, distributed/centralized	Energy efficient than LEACH	Very low Scalability, low Robustness, High overhead
TAG	Tree-based, sink driven	Energy efficient	Low Scalability and Robustness, High overhead
Gossip-based	Random/Geographic-gossip-based	High Robustness	Not energy efficient
Synopsis Diffusion	Multipath-based, distributed	High Robustness and Scalability	Redundant paths
AIDA	Multipath-based, distributed	Application independent, Adaptive	Low Robustness
T and D	Tree and multipath-based, sink driven	High Robustness and Medium Scalability	High aggregation structural cost

statistical model and recent measurements. Depending on the nature of the sensor data, PC can use parametric modeling or non-parametric modeling. For parametric modeling it is necessary to know (or learn) the statistical parameters, such as mean and variance of the sensor data. On the other hand, non-parametric modeling utilizes regression to represent sensor data, requiring very little prior knowledge about the sensor data. The majority of existing PC schemes [Deshpande et al. 2004; Chu et al. 2006; Lu et al. 2010; Tulone and Madden 2006; Xiao et al. 2006a] are based on parametric modeling, where a predictive model is established for every sensor node during a training phase, and the parameters of the model are passed to the sink. Thereafter, nodes only transmit updates to the sink whenever new data arrives or the difference between the model predicted value and the sensed value exceeds a threshold. Thus, it reduces the number of communications between source nodes and the sink, providing communication-level compression. A typical PC technique consists of the followings.

Statistical Model. The statistical model and its prediction accuracy are the heart of PC [Xiao et al. 2006a]. Key models are mainly autoregression based. Autoregressive (AR) models [Tulone and Madden 2006] are computationally simple and predict future observations as a weighted sum of previous measurements. Autoregressive Moving Average (ARMA) models [Lu et al. 2010] use a similar approach, but the model is more complex, allowing higher accuracy in some situations, at the cost of greater computational complexity. Autoregressive Integrated Moving Average models (ARIMA) [Liu et al. 2005] support modeling nonstationary data as well as stationary data but are even more computationally complex.

Learning Phase. During the learning phase, the system determines the parameters of the statistical model, which can be centralized or distributed. In the centralized case [Deshpande et al. 2004], all sensor nodes send their readings to the sink, or central node, which determines the parameters of the prediction model and transmits them back to the nodes. In the distributed case [Lu et al. 2010; Tulone and Madden 2006], each sensor node calculates their own model parameters and, if necessary, transmits them to the sink.

Model Update. This is done at the sink in one of two ways: (i) pull: the sink requests updates as they are needed [Deshpande et al. 2004]; and (ii) push: the sensor sends an updates as they are needed or become available [Chu et al. 2006]. In lossless applications, sensors transmit all prediction errors, or residues. These prediction errors replace the raw observations and reduce the amount of transmitted data. In lossy applications, updates are only sent when the prediction error exceeds a predefined threshold. Clearly, the lossy approach allows for a greater reduction in the number of communications.

One of the most important early works in PC is the BBQ system [Deshpande et al. 2004]. BBQ uses probabilistic modeling techniques to optimize data acquisition for sensor network queries. The BBQ approach is pull-based, normally employing a complex centralized learning phase that must be rerun if the data statistics change. It uses a dynamic Kalman filter to exploit temporal data correlations. Ken [Chu et al. 2006] addresses the 'SELECT' problem related to sensor data query in WSNs. It is a robust approximation technique that uses replicated dynamic probabilistic models to minimize communication between source nodes and sink. In contrast to BBQ, it is well suited to anomaly and event-detection applications. Moreover, BBQ exploits only temporal correlations at individual nodes, whereas Ken exploits spatiotemporal correlations between nodes. BBQ and Ken are geared toward different application domains and are largely complementary. Unification of these techniques would be a promising approach to data prediction in WSNs. Both BBQ and Ken require heavyweight learning phases, which may not work well for nonstationary data. The Probabilistic Adaptable Query (PAQ) system [Tulone and Madden 2006] provides a method for approximating the values at sensors in a WSN based on time series forecasting relying on AR models built at each sensor to predict local measurements. Unlike Ken or BBQ, PAQ is predicated on using lightweight models that can be learned by the individual nodes in the network and retrained quickly when faced with nonstationary distributions. Along with energy efficiency, the method is effective for outlier detection, adaption to dynamic changes in the data statistics, and tolerance of missing sensor data.

A key trade-off in PC is the accuracy of the prediction model. Accurate models tend to provide high prediction accuracy at the cost of requiring more model parameters. Addition of parameters leads to greater computation complexity in model fitting and greater transmission cost in sharing the models between the sources and sink. Hence, flexible models can be less usable in real applications when the data statistics change frequently. Adaptive Model Selection (AMS) [Le Borgne et al. 2007] takes this trade-off into account by allowing sensor nodes to autonomously and adaptively select the best-performing prediction model. The rationale of this AR-based approach is to only use complex prediction models if they prove to be more efficient both in terms of computation and communication savings. The results demonstrate the potential of AMS. However, the *racing* [Oded and Moore 1997] mechanism, which allows nodes to discard poorly performing models from the set of candidate models, may be a concern in real applications.

A central concern of recent works in the area is to introduce in-network data prediction and aggregation into query processing. ADaptive AGgregation Algorithm for sensor networks with data Prediction (ADAGA-P) [Matos et al. 2010] implements a linear-regression-based data prediction function within an existing in-network data aggregation operator. It employs dynamic adjustment of the regression model and outperforms the previous version, ADAGA [Brayner et al. 2008], in terms of energy savings. As the sinks are responsible for calculating the model coefficients and sending them back to the sensor nodes, energy efficiency is a concern. Moreover, correct synchronization between sensor nodes is required.

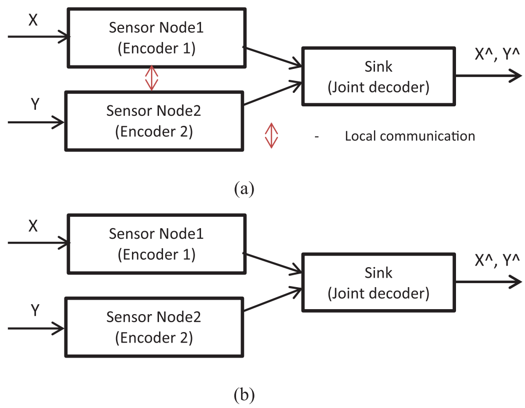


Fig. 2. (a) Joint encoding of X and Y using local communication. (b) DSC based on Slepian and Wolf theorem.

PREdictive STOrage (PRESTO) [Li et al. 2009], a model-driven predictive and two-tier sensor architecture that comprises sensor proxies in a higher tier, each controlling tens of remote sensors in a lower tier. PRESTO proxies and sensors interact and cooperate for acquiring data and processing queries. It relies on an asymmetric prediction technique, and Seasonal ARIMA [Box et al. 1994]. The PRESTO proxy builds the model order and parameters in its initialization phase and distributes them to the responsible sensor nodes. It shows improvements in the energy required for data and query management and in the query latency. The downside of this approach is that spatially-correlated sensors update their parameters almost at the same time causing, high traffic for the entire network. Moreover, it is limited to periodic datasets.

In conclusion, the performance of PC is determined by the effectiveness of the statistical model in terms of its accuracy, parameter size, update rate, and computational complexity. The model needs to be robust enough to handle message loss, especially update message and node failures. Due to the cost of model update and re-training, PC-based compression performs poorly in dynamic networks and environments where frequent updates are necessary.

3.4. Distributed Coding

Distributed source coding (DSC) is an extension of source coding and compression techniques from conventional networks to WSNs. It is asymmetric in nature, as it transfers the computational burden from source nodes to the sink and exploits spatial correlation between adjacent sensors readings. DSC is the compression of multiple correlated sensor outputs where the sensors do not communicate with each other, as shown in Figure 2. Sensors send their compressed data to a central point, or sink, for joint decoding [Pradhan et al. 2002; Xiong et al. 2004]. The theoretical foundation of DSC is based on the Slepian and Wolf [1973] theorem. It shows that the optimal centralized compression efficiency can be achieved by compressing each sensor's data in a distributed manner only using statistical knowledge of the data at the other sensors, but not the actual value of the sensor data.

Slepian-Wolf's foundational work on DSC was only for lossless source coding of discrete sources. For lossy source coding in WSNs, the theory was extended to incorporate a model of the distortion arising in the encoding processing. Lossy distributed compression based on the Slepian-Wolf theorem was first considered by Wyner and Ziv [Kaspi and Berger 1982]. The results show that there is no performance degradation for lossy compression with side information (information from other sources) only available at the decoder (Figure 2(b)) compared to a scheme with side information available at both the encoder and decoder (Figure 2(a)). Rate-distortion extension of the theory provides a tool to characterize the communication required to achieve a given distortion in a network with highly spatially-correlated data [Cristescu et al. 2003].

The published results [Slepian and Wolf 1973; Kaspi and Berger 1982] are solely theoretical. Practical DSC schemes for WSNs involve two key operations: gathering and tracking of correlation knowledge, and code construction [Chou and Petrovic 2003]. Correlation gathering and tracking can be done in a centralized [Chou and Petrovic 2003] or distributed (localized) manner [Yuen et al. 2008]. In the centralized case, an individual node, such as the sink, is responsible for collecting and tracking all of the correlations within the network, whereas, in the distributed case, cluster heads are responsible for gathering and tracking correlation data for a subset of nodes, and a summary is shared with the sink [Yuen et al. 2008]. Encoding can be done in four different ways [Marco and Neuhoff 2004]: No-Slepian-Wolf Scheme (NOSW), Sequential Slepian-Wolf scheme (SEQ), Slepian-Wolf Clustered (CL), and Slepian-Wolf Master Slave (MS).

A number of constructive encoding schemes have been proposed [Garcia-Frias and Zhao 2001; Liveris et al. 2002; Pradhan and Ramchandran 2003; Chou and Petrovic 2003; Xiong et al. 2004]. In general, the decoding of a sensor's message relies on the successful decoding of messages from other sensors. For example, if sensor A encodes based on statistical knowledge of the data at sensors B and C, then messages from B and C must be successfully decoded at the destination before sensor A's message can be decoded. Consequently, the loss of a single message may cause decoding failure for multiple other messages, hence the robustness of the schemes. Channel coding is a way to protect against message loss and is well supported by Wyner's realization of the close connection between DSC and channel coding [Xiong et al. 2004]. Hence, most practical proposals for DSC integrate channel coding, such as Turbo codes [Garcia-Frias and Zhao 2001] and LDPCs (Low Density Parity Codes) [Liveris et al. 2002]. Garcia-Frias and Zhao [2001] exploit punctured Turbo codes for compression of correlated binary sources. Unfortunately, the lack of a proper theoretical link between Slepian-Wolf and Turbo code design has, thus far, prevented effective integration of the methods. LDPC codes seem to be more suited for WSN DSC applications [Liveris et al. 2002]. All LDPC code design techniques are applicable to DSC and they perform better than any Turbo coding scheme suggested so far.

Pradhan et al. [1999] present a practical encoding method for distributed compression in an attempt to achieve the bounds predicted by Slepian and Wolf [1973] and Kaspi and Berger [1982]. Distributed source coding using syndromes (DISCUS) [Pradhan et al. 2000, 2002] address the new area of collaborative information communication and processing. Although promising, the correct choice of correlated side information is essential to ensuring the performance of the algorithm and is normally not well known in practice. This limits the feasibility of the approach when applied to real WSNs. Chou and Petrovic [2003] propose a novel approach to reducing energy consumption in sensor networks using a distributed adaptive signal processing framework and algorithm. The algorithm employs a sink-based centralized approach for the correlation gathering and tracking and a modulo-based sequential coding scheme for code construction. This approach enables sensor nodes to blindly compress their readings with respect to one another without intersensor communication. Results show significant energy savings for typical sensor data across a multitude of sensor modalities.

Xiong et al. [2004] presented a sequel to Pradhan et al. [2002], their own work on DSC, and other relevant research efforts ignited by DISCUS. Through analysis and examples, they [Slepian and Wolf 1973; Kaspi and Berger 1982] show that Slepian-Wolf source coding and Wyner-Ziv coding are in fact source-channel coding problems. They also suggest cross-layer design and joint design of distributed source codes, channel codes, and modulation schemes. Paolo et al. [2006] present joint performance analysis of DSC topologies and packet aggregation (PA) with fragmentation schemes. They consider the four coding schemes proposed in Marco and Neuhoff [2004] and their integration with three alternatives aggregation techniques. Expressions for the performance of DSC and PA are derived in terms of packet-loss probability and the average number of transmitted bytes along with energy efficiency. The work concludes that DSC topologies with a master-slave approach and fragmentation of packets exhibit better performance (e.g., robustness).

The distributed framework in Yuen et al. [2008] jointly optimizes rate allocation and transmission in the presence of capacity constraints. During the optimization, it exploits data correlation among the sensor nodes and the effect of location-dependent contention in the wireless channels. To exploit data correlations within sensor nodes, it adopts localized Slepian-Wolf coding, an approximated version of Slepian-Wolf coding. However, the method does not work well in practice as it considers static link capacity and avoids routing issues. Moreover, as it relies on approximated Slepian-Wolf coding, it suffers when the neighborhood size is small (not scalable). In recent work, Hong et al. [2010] present the performance of a DSC-based system (slotted ALOHA) in terms of throughput, delay, and energy efficiency. They provide a closed-form expression for average throughput based on approximations of the average traffic load in each time slot and derive the average delay and energy consumption via Markov Chain analysis. The results show a possible trade-off between average delay and energy consumption for different probability assignment schemes and for fixed and adaptive MAC protocols. They also highlight the importance of cross-layered transmission control for the efficient delivery of DSC messages as a key to the overall success of DSC.

Works on DSC for WSNs directly, or indirectly, inherit from the Slepian-Wolf theorem. Hence, all proposed DSC algorithms require prior knowledge of the data correlations at different sensors, which limits the effectiveness of the methods in real WSN applications. Moreover, lack of robustness and scalability are concern for these proposals.

3.5. Transform-Based Compression

Transform-based compression approaches are very common for image and video signals. Generally, transform-based approaches support lossy compression. Raw data are transformed into a set of coefficients of appropriate basis functions, for example, wavelet functions, which can be used to reconstruct the signal at the receiver. In most cases, a reduced number of quantized and nonzero coefficients are sufficient for recovering an approximation of the original data with low distortion. Entropy coding is typically applied to the coefficients to further reduce data rate. The Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) have been used extensively in image and video compression applications (e.g., DCT is used in JPEG and DWT is used in JPEG2000).

Sensed environmental data, such as temperature, humidity, light, can be modeled as an image map, and standard image compression methods can be applied to the map. However, some of the unique characteristics of WSNs—limited computation, distributed processing, degree of correlation, and faulty readings—make direct implementation of these approaches inefficient. In the following, we briefly review the algorithms for in-network linear transform-based compression in WSNs. Interestingly, transform-based methods have also been used in the training phase of DSC-like algorithms for the purposes of gathering correlation knowledge [Dang et al. 2007].

Transform-based methods can be viewed as data-dependent and structure-dependent techniques, as they exploit statistical correlations in the data and the network's structure, respectively. Most design techniques for transform-based compression can be viewed as either *transform driven* or *routing driven*. Transform-driven approaches [Wagner et al. 2005b, 2006; Ciancio et al. 2006] focus on utilization of a specific transform. Routing and processing strategies are then developed that allow computation of the transform in the network. These approaches are effective from a data de-correlation standpoint. However, the routing and processing strategies may not always be efficient in terms of data transportation cost. For instance, nodes may be required to transmit their data multiple times [Wagner et al. 2005b, 2006], or transmit multiple copies of the same coefficients [Ciancio et al. 2006], or may even be required to transmit data away from the sink [Gastpar et al. 2006; Wagner et al. 2005b, 2006]. These strategies can outperform raw data gathering for very dense networks but can produce considerable communication overhead for small- to medium-sized networks. Routing-driven approaches focus on establishing an efficient routing tree (e.g., shortest-path routing tree) and use transform computations on the routing paths in the tree. These approaches are typically more efficient, since the transforms are computed as data is routed to the sink along efficient routing paths. These transforms can be easily integrated within existing routing protocols, allowing such schemes to be easily applied in WSNs, as demonstrated by the SenZip [Pattem et al. 2009] compression tool.

The *Karhunen-Love Transform* (KLT) [Gastpar et al. 2006] is commonly used for compression and is a key ingredient of many signal-processing and communication systems. The Discrete KLT (DKLT) shows potential for WSN data compression, since it achieves maximum data de-correlation and can be utilized in a distributed fashion [Gastpar et al. 2006]. However, one of the prerequisites for the DKLT is knowledge of the global correlation statistics. In addition, it is being non-unidirectional, that is, data sometimes travels away from the sink, which could be very expensive in terms of communication cost. Thus, a direct implementation of the KLT is unsuitable for practical WSNs applications. To address this problem, a unidirectional tree-based KLT(T-KLT) has been presented [Shen et al. 2009]. The method applies the KLT to data collected at each node and its descendants. This 'whitens', or de-correlates, the data. The coefficients of the transform are then encoded and forwarded to the parent nodes, which applies the inverse KLT to recover the original data. To perform the TKLT, each node must know the second-order statistics of its subtree. This incurs learning costs associated with discovering and disseminating these statistics.

A number of works [Lee et al. 2007; Wang et al. 2009; Dang et al. 2007] have adopted the DCT for data compression in WSNs. The JPEG-based method in Lee et al. [2007] exploits the DCT for energy-efficient communication of images in WSNs. DCT-supported compressed communication is shown to have better time and energy efficiency than uncompressed communication. Wang et al. [2009] adopt the DCT and differential coding to reduce data redundancy. Moreover, Dang et al. [2007] show that the DCT is suitable for smooth signals, whereas wavelet-based transforms are more suitable for piecewise constant data. Generally speaking, DCT-based compression methods improve energy efficiency compared to uncompressed communications at the cost some undesirable side effects, for example, the complexity of de-correlation at block boundaries, blocking artifacts, and difficulties in adapting to data source statistics.

Numerous methods have been proposed to exploit wavelets and their variants in analyzing and compressing sensed data [Wagner et al. 2005b, 2006; Acimovic et al. 2005; Ciancio and Ortega 2004; Ciancio et al. 2006; Ciancio 2006]. The majority of the earlier wavelet-transform-based works on WSNs (e.g., [Servetto 2003; Ciancio and Ortega 2005]) are non-unidirectional and assume a regular-grid placement of sensor nodes. Servetto [2003] used 1D regular-grid wavelet transforms to solve the 2D sensor broadcast problem. The Lifting Scheme based Wavelet Transform (LSWT) [Ciancio and Ortega 2005] exploits the regular-grid nature of some WSNs and employs 1D wavelet decomposition along paths through the 2D measurement field. It minimizes internode communication by transmitting partial coefficients in an forward direction and updates future sensors (e.g., the next sensors in the direction to the sink) until the full coefficients are computed. However, no means for determining the optimal path is given. In real WSNs applications, nodes are seldom placed in regular grids. WSNs with irregularly placed nodes require different algorithms. A version of the lifting algorithm was proposed for applying the wavelet transform by tracing through the path of the minimum spanning tree and performing the wavelet filter [Ciancio and Ortega 2005]. The method implicitly assumes that the path will be long enough to apply wavelet analysis effectively. Moreover, it is not clear how to choose the best path for compression, and spatial correlation is not fully explored. The system described in Ciancio and Ortega [2005] could be extended to use irregular-grid 1D wavelets, using a method similar to the 1D Haar protocol described in Acimovic et al. [2005]. However, the approach would not be capable of fully capturing the higher-dimensional spatial dependencies between the measurements. Wagner et al. [2005a] provide an irregular-grid, fully 2D, distributed wavelet transform for sensor networks based on piecewise-constant multiscale approximation and multiscale routing structures. This work was extended in Wagner et al. [2005b] to develop a fully distributed, irregular-grid wavelet transform and protocol for sensor networks that is capable of piecewise planar multiscale approximation. The paper presents distributed solutions to implementation issues, included mesh building, filter coefficient calculation, and transform coefficient calculation.

The algorithm proposed in Ciancio and Ortega [2004] is one of the first routing-driven transform-based methods to exploit the wavelet transform to de-correlate WSN data in a distributed fashion. Using a flexible means of exploiting trade-offs between processing and communication costs, the method can maximize energy efficiency, as well as network performance, according to given device specifications. This work considers spatially correlated WSN data, not temporal correlations within intra-sensor data. Acimovic et al. [2005] provide adaptive and distributed processing algorithms for large-scale WSNs, where the data-gathering algorithm is selected adaptively based on the properties of the signal field. They claim that wavelet-based processing is well-matched to the challenge of compression of deterministic signals, such as piecewise constant signals, and prediction based on Differential Pulse Code Modulation is optimal for random Gaussian data in correlated fields. Results clearly show the energy efficiency of the distributed de-correlating process as well as en-route in-network transformation and the unidirectionality of the method. Ciancio et al. [2006] consider a slightly different scenario in which a number of compression schemes are available at each node and the objective is to select the best possible on the basis of the expected computation/communication cost trade-off. They addressed scheme assignment in a two-dimensional field assuming that the routing structure is known by using a heuristic extension of dynamic programming based on an optimal solution for a one dimensional network, presented in Ciancio and Ortega [2006]. Their results show that by optimizing compression algorithm selection, overall energy consumption can be significantly reduced compared to the case where data is just quantized and forwarded to the central node. However, the analysis only considers predefined routing topologies, which are not always available in real WSNs. Moreover, independent selection of routing and coding algorithms may not be optimal in all cases.

The key focus of distributed wavelet-based algorithms [Ciancio 2006] is to maximize the data quality at the sink for a given target energy consumption at the nodes. Unlike previous works [Wagner et al. 2006; Acimovic et al. 2005; Ciancio et al. 2006], it considers entropy-based variable-length encoding of DWT coefficients. Along with other improvements (e.g., 2D instead of 1D), the work considers the possibility of using compressive sampling to reduce the overall power consumption. Shen and Ortega

Table II. Summary of the Key Transform-Based Compression Techniques

Technique	Key Features	Advantages	Limitations
KLT	Behaves like PCA(Principal Component Analysis), DKLT and T-KLT suit WSNs	T-KLT has unidirectionality, hence efficient	Global knowledge needed, Scalability
DCT	Exploits cosine function, variants DCT-I to DCT-VIII	Multiresolution	Blocking artifacts
DWT	Exploits wavelets, variants available (e.g., LSWT, 1D, 2D)	Robustness, unidirectionality possible	Scalability

[2008b] present a unidirectional 2D transform for an arbitrary routing tree, allowing the transform to exploit 2D spatial correlations to a greater extent than earlier path-wise transforms (e.g., [Ciancio and Ortega 2004; Ciancio et al. 2006]) without incurring the overhead of more general 2D transforms. The proposed optimization framework exploits the trade-off between higher local costs for more intricate coding in return for a lower final transport cost. The results show the potential of the proposed method, compared to earlier techniques, in terms of transform computation cost and coefficient transport cost. These improvements are mostly due to unidirectional computation of the 2D transform and the effectiveness of unidirectional computation in offsetting excessively high local communication costs, especially in the backward direction. The main objective of a recent work [Shen and Ortega 2010] is to find a general set of en-route in-network (or unidirectional) transforms for given routing trees and schedules in conjunction with a set of conditions for their invertibility. This general set includes a wide range of existing unidirectional transforms and has also inspired new transform designs, which perform better than existing transforms in the context of data gathering in WSNs. The proposed unidirectional, Haar-like transform leads to significant improvements over existing unidirectional transforms.

Quite a few compression frameworks have been proposed using wavelets and their variants for analyzing and compressing sensed data [Ganesan et al. 2005; Xu et al. 2004; Dang et al. 2007]. DIMENSIONS [Ganesan et al. 2005] was one of the first frameworks addressing multiresolution data access and spatiotemporal pattern mining in a sensor network using wavelet compression. Like DIMENSIONS [Ganesan et al. 2005], Wisden [Xu et al. 2004] is a WSN framework for structural monitoring. It employs a wavelet transform-based compression technique to reduce communication in real time. Wagner et al. [2005b, 2006] present a distributed wavelet transform and data harvesting architecture for sensor networks that removes the assumption about the regularity of the grid. The transform sparsifies piecewise-smooth sensor measurement fields.

As summarized in Table II, transform-based compression techniques (e.g., wavelet-based approaches [Wagner et al. 2005b; Ciancio et al. 2006; Shen and Ortega 2008a] and the distributed KLT [Gastpar et al. 2006]) suffer from lack of scalability. This is due to the critical sampling requirement, which causes the cost of data gatherings to scale with the number of sensors and can lead to poor performance in large deployments.

3.6. Compressed Sensing

Three inherent inefficiencies of transform coding motivate the need for alternative compression techniques: First, compressing high-dimensional signal means processing a large number of samples n. Second, the encoder must compute all transform coefficients $\theta(n)$, even though it will discard all but $K(n \gg K)$ of them. Finally, the encoder must encode the indices of large coefficients. This increases the coding rate, since these indices change with each signal. In this context, compressed sensing (CS) has been proposed as a potential alternative, since the number of samples required (i.e., proposed number of sensors that need to transmit data), depends on the characteristics (sparseness) of the signal [Donoho 2006, Candes et al. 2006, Candes and Romberg 2007]. Sparsity arises in WSN data due to spatiotemporal correlations within the sensor readings. The asymmetric computational nature of CS also makes it attractive for WSN data compression. In CS, most computation takes place at the decoder (sink), rather than at the encoder (sensors), thus sensors with minimal computational performance can efficiently encode data.

The CS field (also known as compressive sampling) field has existed for at least four decades, but recently (about 2004) researchers' interest in the field has exploded due to several important results obtained by Donoho [2005, 2006] and Candes et al. [2006]. CS is a novel sensing/sampling paradigm that goes against the traditional understanding of data acquisition. These works on CS milestone showed that if a signal has a sparse representation in one basis, then it can be recovered from a small number of projections onto a second basis, which is incoherent with the first one. A prerequisite for CS is a tractable recovery procedure that can provide exact recovery of a signal of length n and sparsity K. In other words, a signal can be written as a sum of K basis functions from some known basis, where $n \gg K$. CS is promising for many applications, especially in sensing signals that have a sparse representation in some basis. Rather than sampling a K-sparse signal n times, only $M = O(Klogn)$ incoherent measurements are sufficient. Moreover, at the encoder, no manipulation is required for the M measurements except, possibly, some quantization. For more advanced and detailed information on CS theory, readers are referred to Candès and Wakin [2008], Haupt et al. [2008], and Balouchestani et al. [2011] and references therein.

CS exhibits similar benefits to DSC, including a simple encoding process, avoidance of internode data exchange, and decoupling of compression from routing. In addition, CS has two further advantages: graceful degradation in the event of abnormal sensor readings and data reconstruction insensitive to packet loss. In CS, all messages received at the sink are equally important. On the other hand, in DSC, received data is predefined as main or side information. Losing main information causes serious errors at the decoder. These merits make CS a promising solution to the data-gathering problem in large-scale WSNs [Luo et al. 2009]. Research on CS for WSNs is at an early stage. Even though the number of publications in this area is limited, they are quite diverse in terms of the issues studied (e.g., routing, performance). In the following, we briefly summarize the existing works.

CS research for WSNs can be categorized according to the correlations that they exploit: (i) temporal, (ii) spatial, or (iii) spatiotemporal. Most early proposals for CS in WSNs exploit temporal (intra-signal) structures only. They only exploit temporal correlations within multiplesensor readings at a single sensor and do not exploit spatial (inter-signal) correlations amongst nearby sensors. Early CS works on multisensor scenarios consider only standard CS for the joint measurements at single time instances (e.g., [Bajwa et al. 2007]). These schemes ignore intra-signal or temporal correlations. In contrast, spatiotemporal approaches [Vuran et al. 2004; Duarte et al. 2005] exploit the spatial correlation structures within different nearby sensors and the temporal correlation structure of each sensor's time variant readings.

Bajwa et al. [2006] introduced and analyzed the concept of Compressive Wireless Sensing (CWS) for energy-efficient estimation at the sink of sensor data that is compressible in some basis. Their analysis was based on a function, which depends on the number of sensor nodes and the associated power-distortion-latency trade-offs. Even though CWS is not optimal, it is universal in the sense that it provides us with consistent field estimation, even if little or no prior knowledge of the sensed data is available. Universality comes at the cost of optimality in terms of a less favorable power-distortion-latency trade-off, which is a direct consequence of not having sufficient prior knowledge of the sensed data. CWS uses phase synchronization between the nodes instead of in-network communications and processing. The approach can decrease the latency of data gathering in a single-hop network by delivering linear projections of sensor readings through synchronized amplitude modulated analogue transmissions. However, difficulties in synchronization make it less practical for large-scale sensor networks.

Haupt et al. [2008] describe how CS techniques can be utilized to reconstruct sparse or compressible networked data in a variety of practical settings, such as general multihop networks and WSNs. The central focus of the work is management of resources during the encoding process, which is important as well as challenging. The work presents a procedure based on random gossiping for general multihop networks to exploit CS in storage and retrieval of networked data from multiple points instead of a single sink or fusion centre (FC). A two-step procedure is used to calculate the projections and deliver them to every subset of nodes in the network using gossip techniques or clustering and aggregation. It employs an analogue mechanism similar to the one used in CWS to transmit sensor readings to the FC. This encoding oriented work mainly exploits temporal relationships in calculating projections, not spatial or spatiotemporal.

The key objectives of Compressive Data Gathering (CDG) [Luo et al. 2009] are to compress sensor readings to reduce global data traffic and to distribute energy consumption evenly so as to prolong network lifetime in large-scale WSNs. As in DSC, the decoder exploits the data correlation pattern in this pioneering work. Moreover, compression and routing are decoupled and therefore can be separately optimized. The paper also includes an analysis of the capacity of CDG in WSNs, which shows that CDG can achieve a capacity gain of $\frac{n}{M}$ ($n \gg M$) over baseline transmission. CDG is well suited to large-scale WSNs but suffers in small-scale WSNs where signal sparsity may not be sufficient. CDG works well in networks with stable routing structures, as frequent node failure or dynamic route changes lead to high control overheads that potentially cancel out the gain obtained from compression.

A key focus of CS theoretical developments is to minimize the number of measurements (sampling compression), rather than to minimize the cost of each measurement. To make CS an efficient compression technique for WSNs, an explicit trade-off between measurement cost and reconstruction quality is necessary. Lee et al. [2009] proposed an energy-efficient CS algorithms for WSNs using spatially-localized sparse projections. In order to keep the transmission cost for each measurement low, the method gathers measurements from clusters of adjacent sensors and utilizes localized projection within each cluster. Joint reconstruction provides better performance than independent reconstruction, since it can exploit measurements from multiple clusters. The proposed approach outperforms standard CS techniques for sensor networks. The key to the success of the approach is optimal clustering, which is not a trivial problem.

Event detection is a key application of WSNs. For large-scale WSNs, events are relatively sparse compared to the number of sources. Considering this, Meng et al. [2009] propose a CS method for sparse event detection in WSNs. They show that the number of active (awake) sensors can be greatly reduced. In fact, the number of sensors can be similar to the number of sparse events, which is typically much less than the total number of sources. For signal reconstruction, they consider a fully probabilistic Bayesian framework, which helps in significantly reducing the sampling rate while still guaranteeing a high detection probability. Moreover, use of a marginal likelihood maximization algorithm and a heuristic algorithm for the Bayesian framework leads to higher detection probability than traditional linear programming.

Baron et al. [2009] extended the theory and practice of CS to multi-signal, distributed settings. The paper presents a new theory for Distributed Compressive Sensing (DCS) that facilitates new distributed coding algorithms for multi-signal ensembles. These new compression algorithms rely on a new concept—the joint sparsity of a signal ensemble—and exploit spatiotemporal correlation structures. The work characterizes the fundamental performance limits of DCS for jointly sparse signal ensembles in the noiseless measurement case, for three different modes of CS (i.e., single-signal, joint, and distributed). To demonstrate the potential of the compression framework, detailed examples of three models for jointly sparse signals were presented, and practical algorithms for joint recovery of multiple signals from incoherent projections were developed. For two of the three models, the performance predictions match the results obtained from practical algorithms.

Luo et al. [2010] investigate the benefit of CS in data collection of WSNs. The paper compares a non-CS method (aggregation) with a simple CS algorithm called plain-CS and concludes that in terms of throughput, plain-CS is outperformed by non-CS. The key finding of the work is that applying CS naively may not bring any improvement, and hybrid-CS can achieve significant improvements in throughput as compared with non-CS. Selection of non-CS and CS points within the hybrid-CS scheme is critical in getting the benefit of CS. In a very recent work, Caione et al. [2012] showed that DCS suffers compared to a mixed protocol in large-scale WSNs under real technological constrains. They claimed that CS can be a powerful tool for energy saving in WSN if network size and compression are both taken into consideration in network design.

Thus far, the problems of identifying sparsity requirements, finding the proper basis for random projection calculations, and ensuring local communication have limited the usefulness of CS and DCS in WSNs. In addition, the high decoding complexity could be a problem for real-time time applications in large-scale WSNs.

4. COMPARATIVE STUDY

This section provides a comparison of the performance of each category of compression algorithm described in the previous section. Due to the very limited use of text-based compression in WSNs, it is excluded from the study. Clearly, the proposals within each category are diverse in nature and implementation, making it difficult to come up with a generic and common performance study. However, to take a holistic view of these diverse proposals, it is important to make the comparative study as generic as possible.

This section is structured as follows. First, the assumptions on which the evaluation is based are described. Second, the performance metrics are introduced. Third, expressions for the metrics are derived for each category, and finally, the performance of the approaches is compared with the aid of numerical analysis.

4.1. Assumptions

Herein, we assume a centralized optimal scheduler, which schedules communication in the network. Thus, there are no collisions. WSN topology can play an important role in determining energy efficiency. Naturally, topology varies according to the application. Considering the diversity of WSN applications, it is very difficult to consider all possible topologies and their corresponding performance. In this work, we use a common WSN topology, shown in Figure 3, as a basis and performance metrics as generic as possible so that they apply to all the possible topologies with little or no change. The dependence of the metrics on topology will be discussed in the corresponding section. As shown in Figure 3, each sensor node corresponds to a vertex in the graph G with radius R. Two vertices are connected if and only if their corresponding sensor nodes can communicate directly. Parent nodes can act as aggregation points or transform

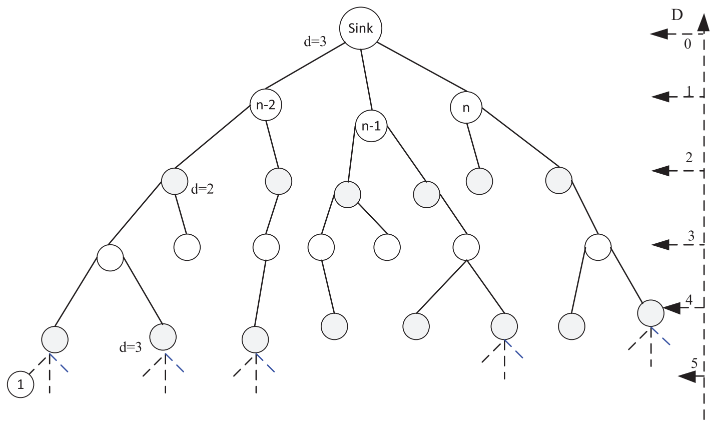


Fig. 3. WSN scenario used for the comparative study.

calculators. The number of nodes n, node degree d (i.e., the average number of child nodes or number of nodes in a cluster for cluster-based topology), average hop count to the sink H, and network depth D (i.e., maximum number of hops to the sink) are the parameters of the network. Within the considered network, we assume that node density is sufficiently high so that there is significant spatial correlation between data collected at neighboring nodes. We also assume that the node-level sampling rate is high enough to maintain intra-signal temporal correlation. Since the network is highly connected, node degree d can be expressed in terms of the number of nodes n [Eschenauer and Gligor 2002].

$$d = \frac{n}{n-1}(\ln(n) - \ln(-\ln(P_c))), \tag{1}$$

where P_c is the probability that the network or graph is connected (P_c is close to 1 for highly connected networks). Based on this, the depth of the network D can be expressed as $D = \frac{ln(n)}{ln(d)}$. These calculations assume a uniform WSN structure, which might not be always true in real life. In real WSNs, d and D might vary within a range.

4.2. Performance Metrics

The following performance metrics are used in the performance analysis.

Compression Ratio (CR). The data compression ratio is the ratio of the uncompressed data size, in bits, b_r to the compressed size b_c, also in bits, and is given by

$$CR = \frac{b_r}{b_c}. \tag{2}$$

The percentage reduction in data size due to compression is given by $(1 - \frac{1}{CR}) \times 100\%$. In case of temporal correlation-based compression (e.g., PC, CS), CR is a node-level parameter, whereas in the case of spatial correlation-based techniques (e.g., DSC, DCS) it can be subnetwork (e.g., cluster) or network-level parameter.

Sampling Ratio (SR). The sampling ratio is the ratio of the number of samples collected when compression is not used, s_r, to the number of samples collected when compression is used, s_c, and is given by

$$SR = \frac{s_r}{s_c}. \tag{3}$$

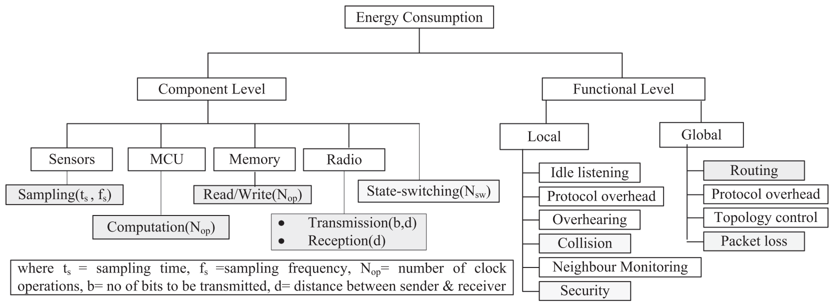


Fig. 4. A typical energy consumption mapping in a WSN.

The percentage reduction in samples is given by $(1-\frac{1}{SR})\times 100\%$. For most compression algorithms, $SR = 1$. However, CS/DCS allows $SR > 1$.

Computational Complexity (CC). Typically, the CC of an algorithm depends on the time and memory space it utilizes. For simplicity, and due to the dominancy of time concerns in assessing the CC of WSNs [Li et al. 2010], we define CC as the computational time complexity. For most MSC platforms (e.g., MSP430 [Polastre et al. 2005]), computational time complexity is directly proportional to the number of clock cycles N_{op} taken to perform the computing task.

Energy Efficiency. Figure 4 shows the various components of energy consumption in WSN nodes [Kamyabpour and Hoang 2010]. In summary, the energy consumption of a node can be expressed as

$$E_{total} = E_{sam} + E_{comp} + E_{sw} + E_{comm}, \tag{4}$$

where E_{sam} is the sampling energy, E_{comp} is the computational energy, E_{sw} is the energy of switching states, and E_{comm} is the communication energy.

The energy cost of sampling is not always insignificant, especially when using power-hungry sensors [MicroDAQ 2010]. Consequently, E_{sam} is highly dependent on the WSN application. In all cases, it is proportional to the total sampling time, which is directly proportional to the number of samples taken. Thus, when applying compression, E_{sam} scales with $SR - 1$.

The energy associated with computation, E_{comp}, is directly proportional to the amount of time that the MCU is on. For modern MCUs which supporting sleep modes, the amount of time that the MCU is on is dependent on the number of clock cycles N_{op} required for the task. The total energy overhead due to the encoding and decoding process is given by E_{coding}.

The switching energy E_{sw} is expended when the radio or MCU switches between states (e.g., sleep, idle, listen/Rx, Tx). Switching energy for the MCU is not significant. On the other hand, the cost of switching the radio [Jurdak et al. 2010] is not negligible. The use of data compression itself does not typically reduce the number of times that the radio must be activated and deactivated, since the compressed source data must still be routed across the network. However, sampling compression reduces the number of radio activations and deactivations by a factor of $(SR - 1)$.

The energy cost of communication E_{comm} is the most important constituent of E_{total}. It is directly proportional to the on time of the radio, both for transmission and reception. It is also depends on the distance between sender and receiver nodes. For a fixed network and for the purposes of the analysis herein, we can note that the energy consumption of communication when using compression E_{comm} scales according to $(CR - 1)$.

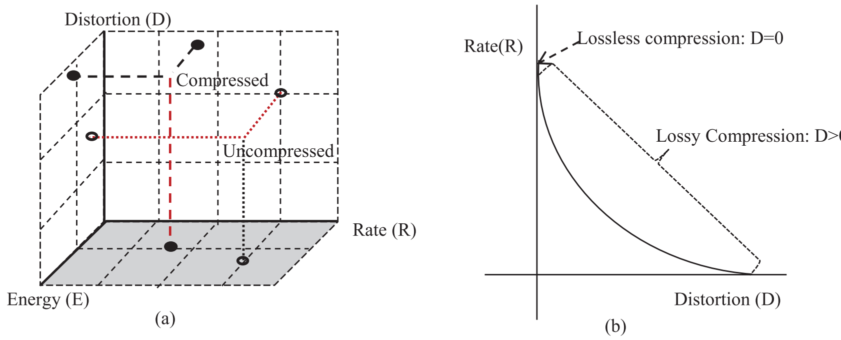


Fig. 5. (a) Rate-energy-distortion relationship. (b) Rate-distortion relationship.

Overall, the energy saving E_{saving} achieved by using compression can be expressed as

$$E_{saving} \approx \left(1 - \frac{1}{SR}\right)(E_{samp} + E_{sw}) - E_{coding} + \left(1 - \frac{1}{CR}\right)E_{comm}, \tag{5}$$

where E_{coding} is the energy required for encoding or processing the compression.

In most deployments, the $(1 - \frac{1}{CR})E_{comm}$ term dominates energy savings [Pottie and Kaiser 2000; Barr and Asanović 2006]. For most compression algorithms (e.g., aggregation, DSC, PC, transform-based), except for CS/DCS ($SR = 1$), Equation (5) can be simplified to

$$E_{saving} \approx \left(1 - \frac{1}{CR}\right)E_{comm} - E_{coding}. \tag{6}$$

Distortion. In lossy compression techniques, distortion measures the difference between the original and reconstructed data. In most cases, distortion is defined as the expected value of the square of the difference between the original and reconstructed signal (i.e., the mean squared error). Figure 5(a) shows the relationship between rate (bits/sample, which is $\propto \frac{1}{CR}$), energy consumption, and distortion in compressed and uncompressed situations, and Figure 5(b) shows the rate-distortion relationship. These figures clearly show that the greater the compression, the higher the distortion. Typically, bounded distortion is desirable in most lossy compression schemes, which makes the R-E-D relationship an optimization problem.

Latency. In WSNs, latency is the average delay incurred in delivering a message from a source to the sink node. Without compression, the main contributor to overall delay is the communication delay T_{comm} of sending the raw information. Typically, for a fixed channel capacity or bandwidth, the latency of communication is directly proportional to the amount of data to be transferred. Hence, when using compression, the latency of the communication is inversely proportional to the compression ratio. However, extra processing delays are incurred both at the encoder $T_{encoding}$ and decoder $T_{decoding}$. Thus, the overall latency T when using data compression can be approximated as

$$T \approx T_{encoding} + \left(1 - \frac{1}{CR}\right) \times T_{comm} + T_{decoding}. \tag{7}$$

For a given MCU, $T_{encoding}$ and $T_{decoding}$ are directly proportional to the number of clock cycles N_{op} needed for the encoding and decoding tasks, respectively, including processing and memory access. When using data compression, these additional processing delays are offset by the resulting reductions in the communication time.

4.3. Performance Metrics for Each Category

For all the calculations, we assume a data set of n_s spatially or temporally-correlated samples, where k is the number of bits per sample in the non-compressed case.

4.3.1. Aggregation.

Compression Ratio. In aggregation, CR is same as the degree of aggregation (DoA), which is defined as the ratio of the number of bits present in all the samples aggregated and the number of bits in the aggregation output. If H_l is the header's bit length of a packet and E_b is the extra bit cost of aggregation, then the CR or DoA is

$$CR_{Ag} = \frac{n_s(k + H_l)}{k + H_l + E_b}. \tag{8}$$

For node-level (temporal) aggregation, n_s is equal to the number of samples generated within the aggregation period, but for spatially distributed signals, n_s depends on the node degree d of the concerned aggregator.

Computational Complexity. Finding an optimal aggregation tree in WSNs and calculating the aggregation function over the collected data at the aggregation points are mainly responsible for the CC of aggregation. For a given and deterministic (as most WSN applications deployments use) aggregation tree, CC depends on the aggregation functions (e.g., *max*, *sum*, *average*, *variance*). For instance, CC for data aggregation based on distributive functions (e.g., *max*, *min*) is of the order $\Theta(D + d_{max}(G))$, where d_{max} is the maximum node degree of the graph G [Li et al. 2010]. So the overall N_{op} in a deterministic aggregation structure is directly proportional to D and d. In the case of dynamic WSNs, the CC of aggregation is dominated by the aggregation structure formation.

Energy Efficiency. Using CR (Equation (8)) and CC in Equation (6), we can determine the approximate energy saving as follows.

$$E_{Ag_{saving}} \approx \left(1 - \frac{k + H_l + E_b}{n_s(k + H_l)}\right) E_{comm} - E_{coding}(D, d). \tag{9}$$

Distortion. Typically aggregation is a lossless compression technique, hence it should be distortionless. As shown in Equation (8), distortion has no direct impact on data aggregation's CR. Distortion may appear due to missing sensor readings (node failure, link failure) or quantization error.

Latency. The delay incurred in the entire data aggregation process is equal to the delay of gathering data from the source that is farthest from the sink. In data gathering, the delay at each hop of the aggregation tree includes transmission delay, contention delay, and aggregation delay. Transmission delays are typically small compared to the delay involved in aggregation. So for the collision-free WSN topology, the main contributor to latency is the aggregation delay comprised of the processing time for aggregation at each node and the time that an aggregation node has to wait for data from downstream nodes to reach it. Thus, the overall latency of aggregation is directly proportional to R (hop count is proportional to R) and d. For centralized aggregation scheduling, the latency bound can be approximated as $23R + d_{max} + 18$ [Huang et al. 2007] and for the distributed aggregation schedule, as $16R + d_{max} - 14$ [Xu et al. 2009].

4.3.2. Predictive Coding.

Compression Ratio. If the prediction error $r(k)$ is within the range $|r(k)| \leq th_{err}$, then for a lossy scheme, there will be no real communication between the source and sink. However for a lossless scheme, the source nodes will transmit the encoded

$r(k)$ values or the real values to update the model at the sink/sinks. In general, $r(k)$ is assumed to follow a normal distribution with zero mean $N(0, \sigma)$, where σ is the standard deviation. Based on this, the $r(k)$ that is within the range $[-th_{err}, th_{err}]$ is $f(th_{err}, \sigma) = erf(\frac{th_{err}}{\sigma\sqrt{2}})$ [Polastre et al. 2007]. Exploiting $f(th_{err}, \sigma)$ in Equation (2) we define CR_{pc} for both lossless and lossy PC as

$$CR_{pc_{lossless}} = \frac{k}{k\left(1 - erf\left(\frac{th_{err}}{\sigma\sqrt{2}}\right)\right) + erf\left(\frac{th_{err}}{\sigma\sqrt{2}}\right)(k')} \tag{10}$$

$$CR_{pc_{lossy}} = \frac{1}{1 - erf\left(\frac{th_{err}}{\sigma\sqrt{2}}\right)}, \tag{11}$$

where k' is the number of bits per sample transmitted in compression mode and depends on the encoding scheme.

Computational Complexity. In PC, learning and prediction are the main computationally complex operations. CC in PC mainly depends on the order of the statistical model and number of samples. As the order of an AR/ARMA/ARIMA model increases, the number of unknowns as well as the number of equations increases. Hence, the complexity of executing a model parameter estimation process is bounded by $O(m^3 n_{ls})$, where m is the order of the model (which is p for $AR(p)$, $max(p, q + 1)$ for $ARMA(p, q)$, and $max(p, q + 1)$ for $ARIMA(p, d, q)$), and n_{ls} the length of the data record [Deng et al. 1997] or learning samples, which is directly proportional to n. After estimating the model parameters, forecasting requires p, $p+q$, and $p+q$ multiplications and p, $p+q$ and $p+d+q$ additions to calculate the next prediction value for $AR(p)/ARMA(p, q)/ARIMA(p, d, q)$ respectively, where q is the order of MA and d is the differencing times value for ARIMA [Lu et al. 2010; Le Borgne et al. 2007; Liu et al. 2005].

Energy Efficiency. For the given WSN topology, using CR (Equation (11)) and CC in Equation (6), we can approximate the possible energy saving (upper bound as no learning cost is considered), hence the energy efficiency of lossy PC (ARMA based), in the considered WSN by the following equation.

$$E_{PC_{saving}} \approx erf\left(\frac{th_{err}}{\sigma\sqrt{2}}\right) E_{comm} - E_{coding}(p, q). \tag{12}$$

Similarly, we can derive $E_{PC_{saving}}$ for the lossless PC.

Distortion. In lossy PC, certain distortion is allowed to have better savings in energy consumption. As the residue or distortion $r(k)$ in general is assumed to follow a normal distribution with zero mean $N(0, \sigma)$, where σ is the standard deviation, the probability that it will be bounded within the range $[-th_{err}, th_{err}]$ is $erf(\frac{th_{err}}{\sigma\sqrt{2}})$ [Polastre et al. 2007]. As shown in Equation (11), distortion has direct impact on CR, hence on energy efficiency (Equation (12)), so a trade-off between distortion and energy efficiency is necessary.

Latency. In PC (lossy), at the sink, predicted values can be generated almost instantly (only the time required for m sum and product operations, which is negligible for the sink). If $r(k) \leq th_{err}$ then latency will be t_p, the predefined waiting time at the sink to check whether is there a real sensor value update from any source node or not. In the lossless case, it is $t_p + CC_{updt}$, where CC_{updt} is the model update or learning processing time. The value t_p depends on the longest source to sink path delay.

4.3.3. Distributed Source Coding.

Compression Ratio. If $Y_1, \ldots, Y_{n_s}$ are n_s binary sequences/samples of length k correlated such that the Hamming distance between two consecutive sequences is at most t. Since DSC can be viewed as source-channel coding method where a (n_c, k) linear channel code (n_c is the code-word, k is the data-word) C can correct up to $M \geq t$ errors per n_c bit block. DSC uses a total of $n_c + (n_s - 1)(n_c - k)$ bits to encode the n_s samples and is sufficient for perfect reconstruction of all of them at the sink [Gehrig and Dragotti 2005]. Hence, the CR for DSC based on Slepian-Wolf scheme can be expressed as

$$CR_{dsc_{lossless}} = \frac{n_c n_s}{k + (n_s - 1)(n_c - k)}. \tag{13}$$

Considering the rate-distortion function based on Wyner and Ziv [Kaspi and Berger 1982], for Gaussian sources [Scaglione and Servetto 2002] $R(D_s) = \frac{1}{2}log(\frac{\sigma^2}{D_s})$ and $CR_{dsc_{lossy}}$ can be expressed as

$$CR_{dsc_{lossy}} = \frac{H(Y_i)}{\frac{1}{2}log\frac{\sigma^2}{D_s}}, \tag{14}$$

where $H(Y_i)$ is the entropy of the samples.

Computational Complexity. Correlation knowledge gathering and tracking is computationally very expensive, especially for dynamic WSNs where correlation structures may change very frequently. As for PC, the complexity of centralized correlation learning based on linear prediction is $O(m^3 n_{ls})$. Source nodes are only responsible for rate allocation, in general this is not a computationally expensive operation. For instance, for a Modulo-code, and a syndrome code, the CC of encoders or source nodes are $O(1)$ and $O(n_c)$ respectively, whereas the CC of the decoders are $O(log_2 n_c)$ and $O(n_c^2 k)$ respectively, where n_c is the length of the codeword. As the decoder involves a binary-matrix multiplication, complexity is high [Annamalai et al. 2008; Chou and Petrovic 2003].

Energy Efficiency. Like PC, using CR (Equations (13) or (14)) and CC in Equation (6), we can approximate the energy saving (upper bound as no learning cost is considered) for lossy DSC (syndrome code based) as

$$E_{DSC_{saving}} \approx \left(1 - \frac{\frac{1}{2}log\frac{\sigma^2}{D}}{H(Y_i)}\right) E_{comm} - E_{coding}(n_c). \tag{15}$$

Similarly, we can derive $E_{DSC_{saving}}$ for lossless DSC.

Distortion. In lossy DSC, bounded distortion is allowed to have better compression, hence better energy efficiency. This comes at the cost of increased complexity. This complexity occurs in finding the rate needed to encode Y_i under the constraint that the average distortion between Y_i and Y_i' is $E[d(Y_i, Y_i')] \leq D$, assuming the necessary side information is available only at the decoder. As shown in Equation (14), distortion and $CR_{dsc_{lossy}}$ are closely related. In some cases a trade-off between these two might be necessary.

Latency. If correlation knowledge gathering and tracking maintains an up-to-date correlation, then the latency of DSC-based compression depends on the encoding time and the longest source-to-sink path delay and on the computation delay, which is very much similar to the other compression approaches. Decoding in DSC-based compression approaches contributes more to latency compared to its counterparts (e.g., PC, aggregation), as most DSC decoders have a sequential decoding requirement. This latency is very sensitive to the packet losses. Packet drop increases the latency and can even cause decoding failure. Maximum latency is bounded by the communication and computation and processing delay of the furthest node from the sink plus the time for successful reception of all messages $Y_1, Y_2, \ldots, Y_{n-1}$ from nodes $S_1, S_2, \ldots, S_{n-1}$, which are closer to the sink than S_n.

4.3.4. Transform-Based Coding. Due to its performance, we derive metrics for the lifting scheme wavelet transform (LSWT) [Daubechies and Sweldens 1998].

Compression Ratio. In this category CR greatly depends on the level of DWT: the higher the transform level L, the more sensors have low-energy (detailed) data that can be coded using less bits and the better the CR, but at the cost of increased internode communication. For simplicity, we consider a 1-level transform, and after the transform, the dataset n_s is replaced by n_d coefficients (high-pass filter output) and n_{sv} updated source value (low pass filter's output). In lossless compression, these updated datasets are passed through lossless entropy coding, whereas for lossy coding, the contents of the new datasets have to be quantized before entropy coding. Let b_d and b_{sv} be the average bit contents of the coefficients (n_s) and the remaining updated data set (n_{sv}), respectively. Exploiting these values we can define the CR as

$$CR_{DWT} = \frac{n_s k}{b_d n_d + b_s n_{sv}}, \tag{16}$$

where $n_d + n_{sv} = n_s$ and $k \geq (b_d + b_s)/2$. Inclusion of thresholding (i.e., coefficients lower than a certain threshold will be discarded) increases CR but at the cost of increased distortion. Unidirectional- and partial-calculation-based DWT require more transform levels compared to their counterparts.

Computational Complexity. Transform-based compression operates on sampled raw sensor data. Generally, it consists of three steps: LSWT, scalar quantization, and source coding (DSC). Its computation complexity can be expressed as

$$CC_{DWT}(n_s, L) = CC_{LSWT} + CC_{quan} + CC_{DSC}. \tag{17}$$

Computation of the scalar quantization matrix is nontrivial, but it can be reduced to $O(n)$ [Liu and Cheng 2006]. Source coding complexity based on DSC is $O(n)$, and finally the computation complexity of LSWT is $O(n)$, where n is the number of samples and, for the critically sampled case, it is equal to the number of source nodes. So, the overall CC_{DWT} is bounded by $O(n)$.

Energy Efficiency. Using CR (Equation (16)), and CC in Equation (6), we can determine the approximate energy saving of DWT-based compression as follows.

$$E_{DWT_{saving}} \approx \left(\frac{n_s k - b_d n_d - b_s n_{sv}}{n_s k} \right) E_{comm} - 3E_{coding}(n). \tag{18}$$

Distortion. Transform-based lossy compression methods can achieve much higher compression at the cost of signal distortion. Signal distortion induced by the transformed-based lossy data compression is due to quantization and thresholding operations. Depending on the quantizer bit number, signal distortion caused by the LSWT-based lossy data compression typically occurs in the frequency bands corresponding to weak signal components. By selecting different quantizer bit numbers or threshold values, users have the flexibility to decide whether they want to have highly-compressed data with a certain level of signal distortions or higher-quality data with less compression. So, a trade-off between distortion and compression ratio or energy consumption is needed.

Latency. In transform-coding-based compression, along with the common communication and processing latencies, the latency introduced by the encoder in calculating the transform coefficients, averages (low frequency values), and quantized values is significant. It includes processing and local communication latencies. L has an impact on latency, the greater L, the more calculations are needed, hence more delay. The additional requirement of the partial calculation process compared to complete calculation causes little extra latency. The overall latency of DWT is bounded by encoding latency, as both decoding and communication latencies are less than encoding.

4.3.5. Compressed Sensing.

Compression Ratio. In CS/DCS, a temporally- or spatially-correlated signal of length n_s with a K-sparse representation only $M = O(Klogn_s)$ incoherently measured samples are needed to recover the signal with high probability, where $K \ll n_s$. In CS/DCS, this sampling or sensing level compression plays the key role in compression, which can be expressed as

$$SR_{cs} = \frac{n_s}{M}, \tag{19}$$

where $M \approx KClog(n_s)$ for dense RP and C is a some small constant, and for sparse RP $M \approx logn_s$. In particular, as suggested by the "four-to-one" practical rule introduced in Candès and Wakin [2008], $M = 4K$ is generally sufficient for dense RP.

Computational Complexity. In CS/DCS, each source node only needs to compute its incoherent projections (M measurements) of the signal it observes, and no manipulations are required for the M measurements, except possibly for some quantization. CS/DCS exploits a random projection (RP) method [Bingham and Mannila 2001; Duarte et al. 2006; Haupt et al. 2008; Wang et al. 2007] to compute incoherent projections. CS is applicable to temporally-correlated signals where the computational complexity is reduced from $O(n_s)$ to $O(M)$. For spatially-correlated signals, DCS is needed, where specifically sparse RP(SRP) calculation requires pre-processing communication amongst the nodes, which is the main contributor to the overall complexity of DCS. In DCS, SRP-based projections calculation requires an average of $O(\frac{n}{K})$ packets transmission per sensor; hence the average computation cost per sensor is $O(\frac{n}{K})$, whereas the decoding cost of CS/DCS is bounded by $O(n^3)$. For SRP, $\frac{n}{K}$ can be approximated by $log(n)$. CS/DCS requires only $O(Klog(n))$ RPs to obtain an approximation error comparable to the best k-term approximation.

Energy Efficiency. Using SR (Equation (19)) and CC in Eq. (5) and replacing CR by SR (as CR directly proportional to SR), we can determine the approximate energy saving of CS and DCS based compression as follows.

$$E_{CS_{saving}} \approx \left(\frac{n_s - M}{n_s}\right)(E_{samp} + E_{sw} + E_{comm}) - E_{coding}(M) \tag{20}$$

$$E_{DCS_{saving}} \approx \left(\frac{n_s - M}{n_s}\right)(E_{samp} + E_{sw} + E_{comm}) - nE_{coding}(M). \tag{21}$$

Distortion. By definition, CS and DCS are lossy compression techniques, hence they support a certain amount of distortion in reconstruction. The robustness of CS/DCS to quantization and noise [Candès et al. 2006; Haupt and Nowak 2006] helps in keeping distortion bounded to real-world settings. At a higher overall cost, DRPs can provide better distortion or approximation error compared to SRPs. In SRPs, distortion is directly proportional to sparsity, hence the distortion at the decoder depends only on the number of coefficients collected, and not on which sensors are queried. Therefore, distributed DRPs enable efficient and robust approximation with refinable distortion [Wang et al. 2007]. Moreover, DCS is automatically robust to packet loss in WSNs; any loss of measurements leads to a graceful degradation in the approximation error or distortion, hence the reconstruction quality.

Latency. In CS/DCS, decoding is computationally more (time) complex $O(n_s^3)$ (for critical sapling for spatial case $n_s \approx n$) than encoding $O(logn)$ (SRP) or $O(n)$ (DRP). Hence, decoding latency is higher than encoding latency. On the other hand, as decoding is done at the sink, which is computationally more powerful than the source nodes, this reduces decoding as well as overall latency.

4.4. Evaluation

The objective of the evaluation is to study the performance of each compression technique using synthetic as well as real datasets in terms of energy saving and latency.

WSN Scenario. For the evaluation, we consider a clustered WSN topology where n sensors (n_{ch} clusterheads and n_{sn} sensor nodes) nodes are deployed randomly over a planar region A. Both sensors and cluster-heads have sensing capabilities, and their sensing range is r_s, a sensor can communicate with a cluster head if it is within the communication range r_t of the cluster head. For simplicity, we assume $r_t = r_s$, which might be little different in real life [Bai et al. 2010], but its impact on our evaluation is little. Let the average number of sensors connected to a single cluster head or node degree be d. As there are n_{ch} cluster heads and n_{sn} sensor nodes scattered over region A, d is given by [Sevgi and Kocyigit 2008]

$$d = \frac{n_{sn}}{n_{ch}}\left(1 - \exp^{-}\left(\frac{n_{ch}\pi r_t^2}{A}\right)\right). \tag{22}$$

Nodes within each cluster are spatially correlated, and each cluster performs its compression separately and independently of all other clusters except for aggregation. We assume every cluster has the same rate. For separate and independent compression within each cluster and centralized collision free scheduling, cluster-level performance is sufficient to provide relative performance measurements for the various compressions schemes (except aggregation). For aggregation, cluster-level dependency requires all clusters to be considered for calculating latency.

Metrics. Energy efficiency and latency are the two main performance parameters for compression algorithms in WSNs. To evaluate energy savings (energy efficiency) we exploit *CR/SR* as well as CC. In calculating E_{coding}, we disregard the decoding cost, as the high-end sink has sufficient resources.

Parameters Used for Evaluation. A list of parameters used and their corresponding values is given in Table III. Each sensor node's (TelosB) [Polastre et al. 2005] ADC (Analogue to Digital Convertor) output is 12 bits. To accommodate this sample size and for simplicity, we bound the data payload k to 16 bits. Again for simplicity in DSC we consider the codeword (n_c) length is 15 bits and the data payload k_{dsc} length is 11 bits, and this 11 bits is sufficient to represent temperature-like sensor readings with high accuracy. Similarly for representing the coefficients or differences in DWT coding, we consider that 8 bits is sufficient. In calculating the sampling and switching energy, along with the information in Polastre et al. [2005], we have exploited the information in Sensirion [2010] and Jurdak et al. [2010].

Methodology. First, for every value of n, we calculate d (cluster members in a cluster) using Equation (22) and use this to calculate *CR/SR* for each technique. Then we

Table III. Parameters Used for Evaluation

Parameters	Value	Parameters	Value
Node Type	TelosB(8MHz)	Network Size (n)	10–1000
Deployment Area(A)	$500 * 500m^2$	Communication Range(r_t)	50–75m
Node degree(d)	Equation (22)	Hop counts(H)	$H \approx (\frac{ln(n)}{ln(d)})$
ADC output	12 bits	Data Payload(k)	2–2d bytes
Header Length(H_l)	7 bytes	Extra Bits(E_b)	8 bits
Entropy($H(Y_i)$)	15 bits	Codeword(n_c)	15 bits
Dataword(k_{dsc})	11 bits	Syndrome($n_c - k_{dsc}$)	4 bits
Coefficients(n_d)	$d - 1$	Updated Sample(n_{sv})	1
Coefficients(b_d)	8 bits	Updated Sample (b_{sv})	16 bits
Saprsity (K)	n/d	Required Measurements(M)	$3K$
Sensor Type	SHT11	Data Type	Temperature($^{\circ}C$)
Sampling Energy(E_{samp})	300 μJ	Switching Energy(E_{sw})	20.01 μJ
Standard Deviation(σ)	$\pm.5$	Distortion (D_s)	.0001–.045
Error Threshold(th_{err})	$\pm.5$	Tx data rate(R)	250 kbps

find the E_{coding} and E_{comm} using information from Table III in the their respective equations. Finally, using these along with *CR/SR* and information from the Table III in each technique's E_{saving} equation, we calculate the respective savings. In calculating latencies, first we find $H \approx (\frac{ln(n)}{ln(d)})$, and then, using unit distance for each hop, we find the communication delay for each category. Finally, adding the corresponding encoding delay, we get the final latency.

Figures 6, 7, and 8 present the results for the energy savings, learning cost, and latencies for each category of compression algorithm. Figure 6 shows results for the two different values (50 m and 75 m) of r_t. Ideally, the TelosB mote can communicate up to 100 m, but in noisy and obstacled environments range can be quite low. So, we produced the results based on ranges from 50–75 m. The savings are presented in terms of percentage of the communication cost of noncompressed mode with respect to n the number of nodes in the network or network size. As the energy savings within a bounded distortion D_s greatly depend on the respective *CR* and/or *SR*, so their trends in the graph almost follow the nature of the corresponding *CR/SR*, as computational cost is negligible compared to communication [Raghunathan et al. 2002]. As shown in Figure 6, at lower values of n, most of the schemes—especially aggregation, $DSC_{lossless}$, DCS, and transform-based coding—suffer greatly, and DCS suffers the most. For instance, for DCS up to n = 50, there are no savings, rather a small loss occurs (we rounded the loss to zero savings). This is because a lower value of n means lower node density and very low or no spatial correlation and no sparsity (for DCS) to be exploited, hence there is no scope for compression and energy saving. This clearly shows that these schemes, especially DCS, transform-based coding, and $DSC_{lossless}$, are not scalable in sparsely dense WSNs. As n increases, it increases node density and the spatial correlation amongst the nearby nodes. This ultimately increases the corresponding *CR* and the energy savings. But slowly and after $n = 600$, these become steady as d and H, the two key parameters of *CR* are almost steady (hence keep the size of cluster almost fixed but increases the number of clusters). H is kept constant at 4 in the network under consideration. As our CR is related to a cluster or node, hence fixed size clusters after $n = 600$ produce steady results. Higher node density or cluster size can cause radio interference amongst the nodes, off setting the benefit of compression, so a trade-off between these can be useful. On the other hand, for DSC and PC, there is no direct relations between n and *CR* (as shown in Equations (10), (11), (13), and (16) except for indirect relations in calculating E_{comm} through H. In the case of PC, as it exploits

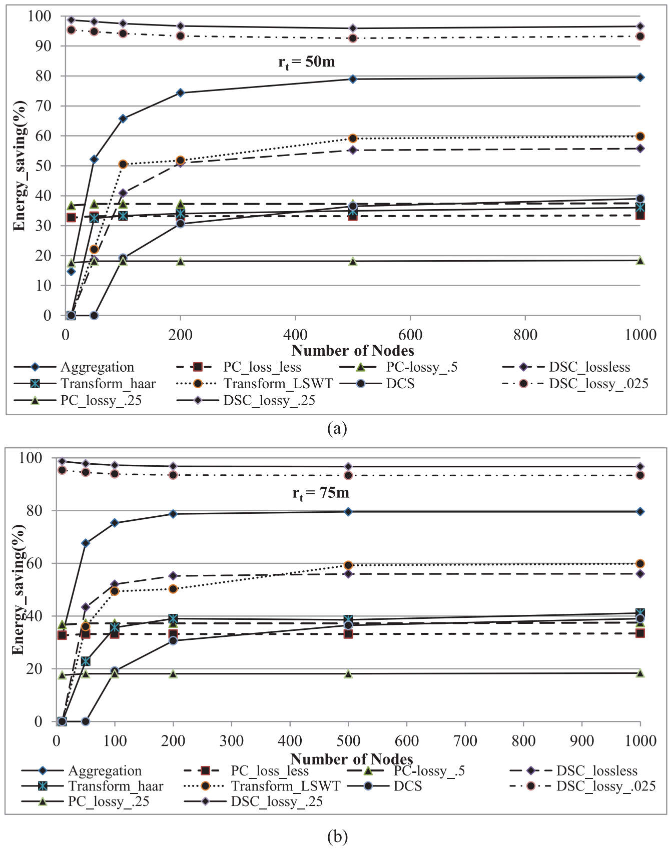


Fig. 6. Network Size (n) vs. energy saving for various compressions: (a) $r_t = 50\ m$; (b) $r_t = 75\ m$.

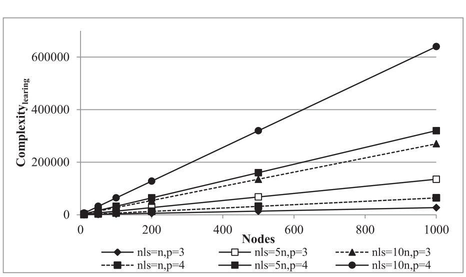


Fig. 7. Network size (n) vs. learning cost in predictive coding and DSC.

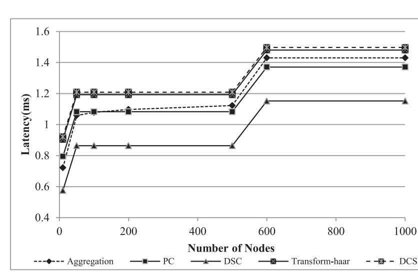


Fig. 8. Network size (n) vs. latencies in compression approaches.

temporal correlation within intra-node signals, there is no direct link to node density and spatial correlation. This is why energy savings for PC are invariant to n and, for DSC, increase slowly as n increases. As shown in Figures 6(a) and 6(b), the impact of communication range r_t on energy saving is not clearly visible except for aggregation. This is because in the aggregation scheme, node degree or child nodes are important, whereas in other schemes, spatially-correlated nodes are important. For a fixed area, increased r_t may include more nodes into the cluster but this does not necessarily mean that they will be correlated.

As shown in Figure 6, aggregation and DSC_{lossy} outperform the others. In aggregation, each cluster head forwards only one packet instead of d full packets (noncompressed) or $d-1$ packets with reduced payloads (e.g., DWT, DSC, PC), and employs a simple encoding scheme. This why it has higher CR and greater energy savings. However, aggregation is unable to provide individual sensor readings. In this analysis, we have excluded the cost of determining the optimal aggregation tree (e.g., obtaining the Minimum Steiner Tree is an NP-Complete problem [Akkaya et al. 2008]) assuming that the WSN is static. In dynamic WSN, it could offset the benefits of aggregation. On the other hand, DSC_{lossy} gains this savings at the cost of distortion. Increasing D increases CR, as the bit contents of the signal are reduced drastically based on Equation (1), hence the energy saving. For instance, if D_s increases from .025 to .25, energy saving increases from 95.38% to 98.68%. PC- (both lossless and lossy), DCS-, and DWT-based compression suffers compared to the others. Increased th_{err} allows PC (lossless) to encode correlated signals with less bits, increasing the CR and energy savings. In the lossy case, as th_{err} increases, more samples are discarded, which increases CR more and yields greater energy savings. For example, in lossy PC, th_{err} increases from .25 to .5, results in energy savings increasing from 16.7% to 31.4%. Higher values of th_{err} allow greater energy savings but at the cost of increased distortion. The prohibitive learning cost of PC/DSC (Figure 7), which linearly increases with n as well as exponentially with the order $m = max(p, q)$ of the prediction model could limit the use of PC and DSC in WSNs, especially in dynamic networks where frequent learning or updates might be needed. In DWT, CR depends on H, which increases slowly with n. Moreover, computationally DWT is more expensive, as it includes transformation and quantization as well as DSC. Threshold-based DWT can improve this saving by discarding the transform coefficients, which are lower than the threshold. In SRP-based DCS (spatial), if the cluster size $d < 4$ (WSNs with $n < 50$) and M becomes close to d, then there are some energy losses (20%, not shown in the graph) instead of savings. This is due to local communication cost. On the other hand, as soon as $d > 4$ (WSNs with $n > 50$), the cost of local communications is compensated by savings due to less measurements M. If we consider spatiotemporal DCS, then better savings are possible due to temporal decorrelations. It can produce savings even when $d < 4$ (WSNs with $n < 50$), but this is at the cost additional latency. Unlike DSC and PC, transform-based and CS/DCS compression do not require learning or global correlation knowledge except for the purpose of local communication (included in the calculations). Hence these schemes do not suffer in dynamic WSNs. Nevertheless, the very high decoding complexity $O(n^3)$ of CS/DCS is a hindrance to the use of CS/DCS in large-scale WSNs. Use of special hardware support, such as DSP (Digital Signal Processor) [Texas Instruments 1994], can mitigate this somewhat.

Figure 8 shows the latencies (excluding decoding and retransmissions) for different compression approaches with respect to n on millisecond (ms) scales. As lossless and lossy versions of PC and DSC display similar latencies, we present them as generic cases. As shown in Figure 8, latency increases slowly with n (upto 500) as H and d increases slowly with n. Increased H and d requires more communications and processing, hence the latency after $n = 500$ sharply increases. As shown Figure 8, the trend is that as the node density increases, latencies also increase and become almost steady after $n = 600$. This is because d and H, the two key parameters of CR, are almost steady (hence keep the cluster size almost fixed but increasing the number of clusters). H is constant at 4, hence fixed size clusters after $n = 600$ are producing steady results. We have disregarded the decoding delays for all the schemes. However, in large-scale WSNs, this could be very significant for CS/DCS due to the high decoding time complexity. Also in DSC, the impact is higher for longer codewords n_c, as it follows $O(n_c^2 k)$.

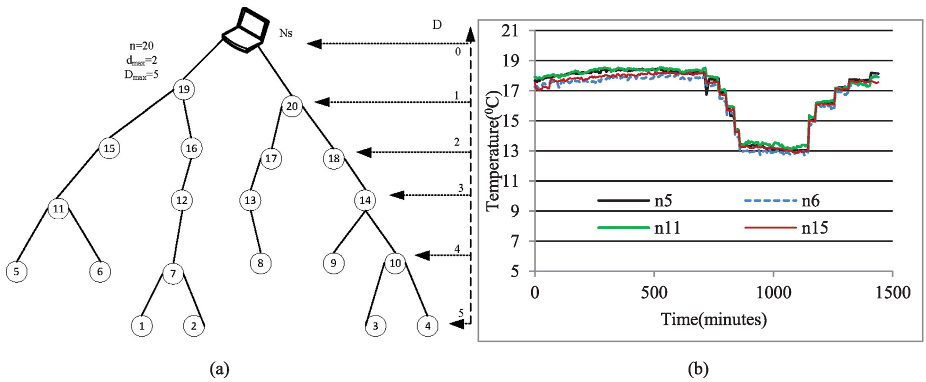


Fig. 9. Dataset one: (a) Network used in dataset one. (b) Snapshot of correlation between nodes 5, 6, 11, and 15.

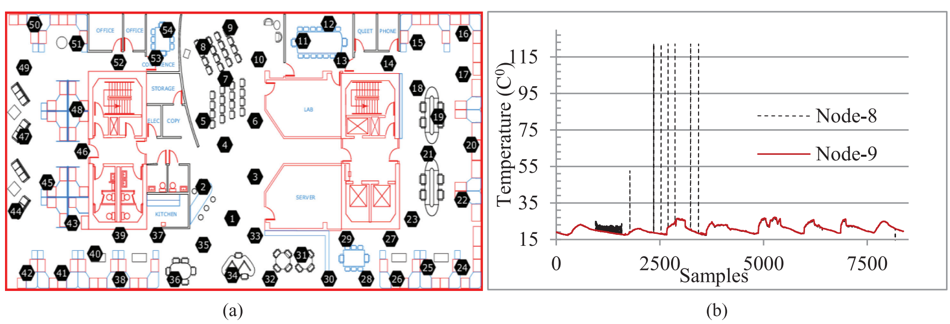
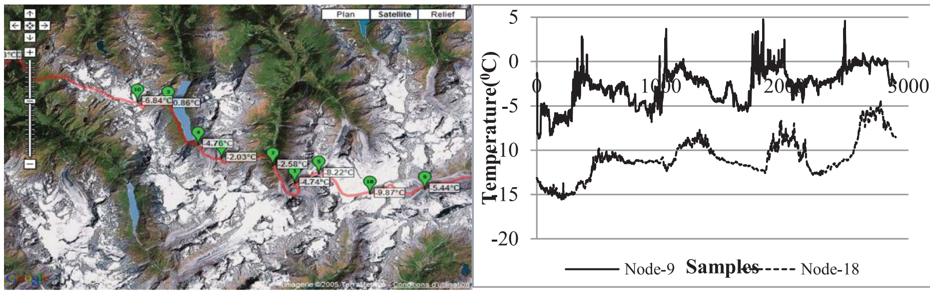


Fig. 10. Dataset two: (a) Network used in dataset two. (b) Snapshot of correlation between nodes 8 and 9.

As shown in Figure 8, latency-wise, all the compression techniques except DSC show similar performances. This is because the communication latency, the main contributor to overall latency, is similar for all techniques. The variations shown in the figure are mainly due to their differing computational time complexities. In the case of aggregation, latency is due to each hop as every aggregation point or cluster head must wait for its child node. In transform coding and DCS, higher latencies are due to their higher computational complexities and local communication. Even though PC shows better performance compared to aggregation, transform coding, and DCS, it suffers compared to DSC as it has higher computational complexity. In DSC the main contributing factors are reduction in bit content and simple encoding.

Numerical Analysis. In this section, we apply the compression schemes to three real-life sensor datasets and perform numerical analysis. Dataset one is generated from our own lab WSNs deployment, the second and third are from the Intel Lab Data [Intel Berkeley Research Lab 2004] and the Sensorscope PDG deployment [EPFL 2008]. To ensure diversity in the datasets, we have included datasets for indoor (first two) and outside environment monitoring (third). Figures 9, 10, and 11 present the network scenarios and snapshots of these datasets. The WSN for dataset one consisted of 20 source nodes (TelosB) and one sink. For simplicity, a constant hop distance of 3 m was used. The environmental temperature was sampled by every node every five minutes. The deployment operated for a month. The total number of samples gathered was 8,640 per node and 172,800 for the whole network. In dataset two, data was collected from 54 sensors deployed in the Intel Berkeley Research lab between February 28, and



(a) (b)

Fig. 11. Dataset three: (a) Scenario used in dataset three. (b) Snapshot of correlation between nodes 9 and 18.

April 5, 2004 [Intel Berkeley Research Lab 2004]. Mica2Dot [Mica2Dot 2004] sensors with weather boards collected time-stamped topology information, along with humidity, temperature, light, and voltage values once every 31 seconds. In the span of 38 days, around 2.3 million readings were collected from these sensors. In dataset three, environmental data were collected from Patrouille des Glaciers (PDG), Switzerland between April 16–20, 2008. Shockfish TinyNode [TinyNode 2008] based ten weather stations collected weather-related data (ambient temperature, wind-speed) every 2 mins, and each node collected on average 3,000 samples within the five-day period. As shown in Figures 9, 10, and 11, sensor readings in datasets one and two (excluding few outliers) are very strongly spatially- and temporally-correlated, but not in dataset three. This is expected, as indoor environments are generally controlled and show stationary statistics, but outdoor environments like PDG, Switzerland do not. This is why all of the compression schemes discussed earlier are suitable for datasets one and two but most are not suitable for dataset three, since the data lacks correlation, making compression less effective and introducing higher distortion in reconstruction [EPFL 2008]. So, we focus on the first two sets for the analysis. However, dataset three provides a clear indication that sensor readings collected in dynamic environments may not always be compressible with bounded distortion.

For the learning phase of DSC and PC, we exploit two days/week (one in a weekday and one in a weekend) data, which means for the dataset one, we need eight days of readings (40,320 samples) and for dataset two, we need 12 days of readings (726,315 samples, approximately). Analysis of the datasets gives an overall network-wide spatial correlation coefficient of 0.915 and a data sparsity of $K/n \approx 0.065$ (based on SRP) for dataset one and 0.95 (approximately) and 0.033 for dataset two.

The performance, which can be obtained by applying the algorithms discussed in Sections 3.2–3.6, was predicted by means of the equations described in Sections 4.3–4.7. The performance metrics were calculated based on node characterization information [Polastre et al. 2005; Goh and Venkat 2006; Mica2Dot 2004; Sensirion 2010]. The parameters used are listed in Table IV. The results approximated for each algorithm category are given in Table V.

In these (static) deployments, all compression schemes achieve handsome energy savings over uncompressed operations. As shown in Table V (subscript 1 for dataset one and 2 for dataset two), these approximated results very much follow the results in Figures 6 and 8, except for DCS. Unlike Figure 6, here DCS shows energy savings as the model considered spatiotemporal correlation rather than only spatial. Aggregation and DSC_{lossy} show the most energy savings. Due to the inclusion of learning cost, the energy saving (both the lossy and lossless) of DSC and PC suffers somewhat compared to Figure 6. On top of learning cost, the smaller cluster size ($d+1$, lower decorrelation

Table IV. Parameters Used for the Numerical Analysis

Dataset	Parameters	Value	Parameters	Value
One	n	20	n_{ls}	40,320
One	Maximum d	2	Maximum H	5
One	One clk cycle cost	.675 nJ	Transmit-cost(1 bit)	260 nJ
One	Receive-cost(1bit)	270 nJ	N_{op}-16bit Math	219
One	N_{op}-16bit Matrix	945	N_{op}-Floating Point	786
Two	n	54	n_{ls}	726,315
Two	Maximum d	4	Maximum H	6
Two	One clk cycle cost	3 nJ	Transmit-cost(1 bit)	2,100 nJ
Two	Receive-cost(1bit)	781 nJ	N_{op}-16 bit Math	266
Two	N_{op}-16bit Matrix	1,488	N_{op}-Floating Point	1,654

Table V. Performances of the Numerical Analysis

Approach	CR_1	$E_{saving1}(\%)$	$Latency_1(ms)$	CR_2	$E_{saving2}(\%)$	$Latency_2$
Aggregation	3.4	70.3	10.21	4.5	77.1	81.6
$PC_{lossless}$	1.99	19.5	5.59	1.99	19.5	44.18
PC_{lossy}	2.83	34.2	5.59	2.83	34.2	44.18
$DSC_{lossless}$	1.95	18.5	4.9	2.23	24.8	38.73
DSC_{lossy}	37.6	66.94	4.9	37.6	66.94	38.73
Transform-based	2.17	54	5.952	2.49	59.1	47.1
DCS	1.34	23.53	5.85	1.69	37.12	46.22

scope) reduces energy savings, especially for $DSC_{lossless}$, and $PC_{lossless}$. As expected, due to the waiting time in each aggregation hop, it shows highest latency, and the others show increased latency compared to Figure 8. This is due to the increased hop counts (H) and hop distance (3 m instead of 1 m). Dataset two performs better than data set one in terms of CR, hence in E_{saving} for aggregation, transform coding, DCS and $DSC_{lossless}$ but suffers in latency as it exploits a radio with lower data rate and MCU, which requires more of clock cycles (approximated) and has more hop counts (H) compared to set one. Improvement in energy saving comes due to little higher correlation and node degree d. On the other hand, both show the same results in terms of energy saving for PC and DSC_{lossy}, as we have considered the same sensor data with same th_{err} and D_s.

Based on Sections 3 and 4, we present a summary of the compression techniques (except text-based because of its limited use in WSNs) in Table VI. We consider characteristics, such as compression and correlation type, complexity (computational), reliability, robustness, scalability, QoS, and security, in summarizing them. The complexity, robustness, and scalability are rated as low, medium, and high. As each of the techniques has number of variants, scale of complexity, robustness and scalability can vary. It is clear from the table that most of these compression techniques suffer in scalability and robustness. Moreover, few address QoS, security, or reliability.

5. OPEN RESEARCH ISSUES AND FUTURE DIRECTIONS

Although the compression techniques presented herein addresses many issues in WSNs compression, there are still some open research challenges. In particular, research is needed in the area of integrating of QoS, reliability, and security with compression. In addition, most previous work views compression from the signal processing perspective only. Hence, research on data compression from the networking protocol perspective in WSNs is limited. Therefore we also briefly consider this viewpoint, examining cross-layer opportunities in particular.

Table VI. Summary of the Key Compression Techniques in WSNs

Char.	Aggregation	PC	DSC	Transform Coding	CS/DCS
Compression	DC and CC	DC and CC	DC and CC	DC and CC	SC, DC and CC
Correlation	spatial (not always)	Temporal	Mainly Spatial	Spatio-temporal	Spatio-temporal
Complexity	High (structural)	Medium (learning)	Medium (learning)	Medium	High (decoder)
QoS/QoI	Addressed	Not yet	Not yet	Not yet	Not yet
Reliability	Possible and addressed	Not yet	Possible and addressed	Not yet	Possible and addressed
Robustness	Low-high	High	Low-Medium	Medium	High
Scalability	Low-high	Low-Medium	Low-Medium	Low	Medium
Security	Addressed	Not yet	Not yet	Not yet	Inherent
Applications	Limited to where aggregation functions applies	Suffers in dynamic applications	Suffers in dynamic applications	Good in dynamic and static environments	Good in dynamic and static environments

Improved Compression. Data sampling and switching of node state, especially in radio, are regular phenomenon in WSN implementations and are not typically inexpensive in terms of energy. For instance, a sampling operation costs (for TelosB) at least 0.3 mJ for temperature (equal to the transmission cost of 1,153 bits) [Polastre et al. 2005; Sensirion 2010], and 0.36 mJ for soil moisture (equal to the transmission cost of 1385 bits) [MicroDAQ 2010]. Unfortunately, existing compression approaches (except CS/DCS) do not consider these two issues, hence their costs. Works on CS/DCS [Baron et al. 2009; Vuran et al. 2004; Duarte et al. 2005] already show that sampling level compression is possible, but is yet to be explicitly explored in WSNs. Typically in WSNs, a sensing operation wakes up the MCU and MCU wakes up the radio [Jurdak et al. 2010]. In PC, if the estimated values are within the error threshold, then there will be no radio transmission. In this situation, switching the radio to the on state immediately after the MCU is a waste of energy. Reactive instead of proactive switching of the radio will reduce the number of switching operations and reduce their energy cost.

The majority of existing compression approaches assume reliable communications, but in reality, WSNs communications are seldom reliable. Moreover, compression schemes often neglect the energy-expense arising from computation of imputation models, that is, evaluation of polynomials, comparisons, and so on, which are usually floating point operations and are therefore relatively costly on tiny sensor hardware [Blaß et al. 2008]. Greater consideration of the effects of unreliable communications on compression is necessary to improve performance.

Existing PC or DSC algorithms use either a centralized or distributed learning phase. In a network, centralized learning is good for nodes closer to sink, while a distributed approach is better for more distant nodes. Hybrid learning may be a good research direction for predictive coding and DSC. Even a combination of reactive and proactive learning could be useful. Due to decoding complexity, CS/DCS suffers in real-time applications in large scale WSNs. Investigating decoding complexity reduction, especially for CS/DCS, could be a fruitful future research direction.

QoS-Awareness. Compression algorithms and frameworks should integrate QoS-awareness so that WSN applications can achieve their objectives. Few papers on data aggregation have considered this issue. To our knowledge, no work explicitly considers QoS in PC, DSC, DCS, and transform-based compression schemes or frameworks. Integration of QoS-awareness in compression schemes or framework could be a potential future direction.

Reliability. There is a clear dependency between reliability and compression, which should be better understood and exploited. Given the limited number of publications on the topic [Iyer et al. 2008; Marco and Neuhoff 2004], there is clearly significant scope for future work in this area.

Scalability. Most of existing compression approaches (e.g., PC, DSC, transform coding) perform poorly in scalability experiments. For instance, DCS suffers in small-scale WSNs due to lack of sparsity and in large-scale WSNs due to high decoding complexity. This issue needs further attention from the researchers.

Security. Security is not considered in most compression schemes, except data aggregation. It is worth noting that the random projections used in CS and DCS inherently provide encryption functionality [Abdulghani and Rodriguez-Villegas 2010]. The randomized measurements themselves look a lot like noise, which is meaningless to an observer who does not know the seed. This inherent encryption in CS and DCS schemes is a real bonus. However, further research is needed in this area.

Cross-Layer Design. Generally, data compression is implemented as an application-layer protocol. However, in some circumstances, application-level implementation of compression is suboptimal. Some compression algorithms reduce the amount of data collected (e.g., CS). To take full advantage of this, nodes should stay off when sensing is not taking place. However, this has an impact on network connectivity, since the radio will be off as well. Optimal operation requires cross-layer or multilayer coordination between application-layer compression and MAC-layer scheduling. The dependency of compression on routing is obvious [Shen and Ortega 2010; Scaglione and Servetto 2002]. Furthermore, incorporation of resource awareness in compression schemes, for example, dependency on remaining energy, requires coordination between application layer compression and the physical layer.

Very little work has been done in cross-layer-based compression [Oldewurtel et al. 2008; Wang et al. 2009]. Exploration of this aspect of compression in WSNs is necessary.

6. CONCLUSIONS AND FUTURE WORK

Development of effective compression algorithms is key to improved utilization of the limited resources of WSNs (energy, bandwidth, computational power). A large number of proposals have addressed this problem. The proposals are diverse and involve various compression approaches. In this work, we have made an effort to put these works into perspective and to present a holistic view of the field. In doing this, we have provided a comprehensive overview of existing approaches, reviewed the current state of the art, and presented a logical classification. Previous works are categorized as involving either aggregation, text-based compression, distributed source coding, transform-based compression, compressive sensing, or predictive coding. Each category has a number of variants, which are presented accordingly. We have analyzed these approaches on the basis of the key performance metrics, that is, compression ratio, computational complexity, energy efficiency, distortion, and latency. Analytical results show that lossy versions of these approaches provide better compression ratios. Hence they achieve higher energy savings than the corresponding lossless versions at the cost of distortion in the reconstructed signals.

Aggregation is the most commonly exploited and easily deployable compression technique. It has a number of variants depending on network topology, such as tree based, chain-based, cluster-based. However, it has limited applications as it is unable to produce original sensor data at sink. Finding appropriate aggregation points is an optimization problem. In the case of unreliable communication, the aggregation point wait time could be prohibitive. Predictive coding is very useful in reducing the amount of data communication but requires learning of data statistics, which can be very expensive in dynamic environments, as the complexity of learning is bounded by $O(m^3n)$. Obtaining the correlation knowledge required by DSC can be as expensive as learning in predictive coding. Lossy DSC can provide very high compression ratios, as well as high energy efficiency, but suffers in dynamic environments and networks. In terms of compression ratio and energy saving, transform-based compression and CS show reasonable performance compared to their counterparts, as these methods do not require any learning of correlation statistics. Hence they are effective in dynamic environments and networks. Transform-based approaches are particularly useful for multimedia communications (e.g., video, images), as specialized compression algorithms are available for this type of traffic. Many CS/DCS approaches operate on analogue signals. The computational complexity arising from use of floating point data as well as matrix calculations could be significant. Moreover, the decoding complexity of CS/DCS can lead to significant delay in large-scale networks. Hence the approach may face scalability problems.

Although the presented approaches and frameworks address many issues associated with data compression in WSNs, some research questions remain relatively unexplored, such as support for and integration of QoS, scalability, reliability, and security. There is significant scope for future work in these areas. Realizing the importance of QoS in WSNs, our future endeavors will focus on developing a compression framework, which integrates QoS-awareness for WSNs. Data compression is common in WSNs, hence integration of QoS awareness will ultimately contribute in developing a QoS-aware data gathering framework for WSNs. The diverse applications of WSNs demand support for a diverse set of QoS requirements. A single compression technique will not be optimal for all applications. Along with QoS awareness, a secondary objective will be to determine the best possible compression technique for a particular application taking into account the limited available resources. We also have the intention to explore the possibilities of cross-layer design of compression approaches in WSNs.

REFERENCES

ABDULGHANI, A. M. AND RODRIGUEZ-VILLEGAS, E. 2010. Compressive sensing: From compressing while sampling to compressing and securing while sampling. In *Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society*. 1127–1130.

ACIMOVIC, J., BEFERULL-LOZANO, B., AND CRISTESCU, R. 2005a. Adaptive distributed algorithms for power-efficient data gathering in sensor networks. In *Proceedings of the International Conference on Wireless Networks, Communications and Mobile Computing*. 946–951.

ACIMOVIC, J., CRISTESCU, R., AND BEFERULL-LOZANO, B. 2005b. Efficient distributed multiresolution processing for data gathering in sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*. 837–840.

AKKAYA, K., DEMIRBAS, M., AND AYGUN, R. S. 2008. The impact of data aggregation on the performance of wireless sensor networks. *Wirel. Commun. Mob. Comput. 8*, 2, 171–193.

AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., AND CAYIRCI, E. 2002. Wireless sensor networks: A survey. *Comput. Netw. 38*, 4, 393–422.

ALZAID, H., FOO, E., AND NIETO, J. G. 2008. Secure data aggregation in wireless sensor network: A survey. In *Proceedings of the 6th Australasian Conference on Information Security (AISC'08)*. Australian Computer Society, Inc., 93–105.

ANNAMALAI, M., SHRESTHA, D., AND TJUATJA, S. 2008. Experimental study of application specific source coding for wireless sensor networks. CoRR abs/0810.3626.

BAI, X., YUN, Z., XUAN, D., JIA, W., AND ZHAO, W. 2010. Pattern mutation in wireless sensor deployment. In *Proceedings of the 29th Conference on Information Communications (INFOCOM'10)*. IEEE Press, 2480–2488.

BAJWA, W. U., HAUPT, J. D., SAYEED, A. M., AND NOWAK, R. D. 2007. Joint source-channel communication for distributed estimation in sensor networks. *IEEE Trans. Inform. Theory 53*, 10, 3629–3653.

BAJWA, W. U., HAUPT, J. D., SAYEED, A. M., AND NOWAK, R. D. 2006. Compressive wireless sensing. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*. ACM, 134–142.

BALOUCHESTANI, M., RAAHEMIFAR, K., AND KRISHNAN, S. 2011. Compressed sensing in wireless sensor networks: Survey. *Canadian J. Multimedia Wirel. Netw. 2*, 1.

BARON, D., DUARTE, M. F., WAKIN, M. B., SARVOTHAM, S., AND BARANIUK, R. G. 2009. Distributed compressive sensing. CoRR abs/0901.3403.

BARON, D., WAKIN, M. B., DUARTE, M. F., SARVOTHAM, S., AND BARANIUK, R. G. 2005. Distributed compressed sensing. Tech. rep., Rice University.

BARR, K. C. AND ASANOVIĆ, K. 2006. Energy-aware lossless data compression. *ACM Trans. Comput. Syst. 24*, 3, 250–291.

BINGHAM, E. AND MANNILA, H. 2001. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 245–250.

BLAß, E.-O., HORNEBER, J., AND ZITTERBART, M. 2008. Analyzing data prediction in wireless sensor networks. In *Proceedings of the 67th IEEE Vehicular Technology Conference*. IEEE.

BOX, G. E. P., JENKINS, G. M., AND REINSEL, G. C. 1994. *Time Series Analysis: Forecasting and Control* 3rd Ed. Prentice-Hall.

BOYD, S., GHOSH, A., PRABHAKAR, B., AND SHAH, D. 2006. Randomized gossip algorithms. *IEEE Trans. Inform. Theory 52*, 6, 2508–2530.

BRAYNER, A., LOPES, A., MEIRA, D., VASCONCELOS, R., AND MENEZES, R. 2008. An adaptive in-network aggregation operator for query processing in wireless sensor networks. *J. Syst. Softw. 81*, 3, 328–342.

BROWN, S. AND SREENAN, C. J. 2007. A study on data aggregation and reliability in managing wireless sensor networks. In *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Systems*.

BURROWS, M., WHEELER, D., AND WHEELER, D. J. 1994. A block-sorting lossless data compression algorithm. Tech. rep., Digital *SR*C Research Report.

CAIONE, C., BRUNELLI, D., AND BENINI, L. 2012. Distributed compressive sampling for lifetime optimization in dense wireless sensor networks. *IEEE Trans. Indust. Inf. 8*, 1, 30–40.

CANDÈS, E. AND ROMBERG, J. 2007. Sparsity and incoherence in compressive sampling. *Inverse Prob. 23*, 969–985.

CANDÈS, E. AND WAKIN, M. 2008. An introduction to compressive sampling. *IEEE Signal Process. Mag. 25*, 3.

CANDÈS, E. J., ROMBERG, J. K., AND TAO, T. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Commun. Pure Appl. Math. 59*, 8, 1207–1223.

CANDÈS, E. J., ROMBERG, J. K., AND TAO, T. 2006. Robust uncertainity principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory 52*, 489–509.

CAPPIELLO, C. AND SCHREIBER, F. A. 2009. Quality- and energy-aware data compression by aggregation in WSN data streams. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications*. IEEE Computer Society, 1–6.

CARDEI, M., THAI, M. T., LI, Y., AND WU, W. 2005. Energy-efficient target coverage in wireless sensor networks. In *Proceedings of the IEEE INFOCOM*. 1976–1984.

CHEN, D. AND VARSHNEY, P. K. 2004. QoS support in wireless sensor networks: A survey. In *Proceedings of the International Conference on Wireless Networks (ICWN'04)*.

CHEN, M. 2006. Data compression for inference tasks in wireless sensor networks. Ph. D. Dissertation, Binghamton University State University of New York.

CHOU, J. AND PETROVIC, D. 2003. A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks. In *Proceedings of the IEEE INFOCOM*. 1054–1062.

CHU, D., DESHPANDE, A., HELLERSTEIN, J. M., AND HONG, W. 2006. Approximate data collection in sensor networks using probabilistic models. In *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*. IEEE Computer Society, 48.

CIANCIO, A. AND ORTEGA, A. 2004. A flexible distributed wavelet compression algorithm for wireless sensor networks using lifting. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*.

CIANCIO, A. AND ORTEGA, A. 2005. A distributed wavelet compression algorithm for wireless multihop sensor networks using lifting. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*. 825–828.

CIANCIO, A. AND ORTEGA, A. 2006. A dynamic programming approach to distortion-energy optimization for distributed wavelet compression with applications to data gathering in wireless sensor networks.

In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'06)*.

CIANCIO, A., PATTEM, S., ORTEGA, A., AND KRISHNAMACHARI, B. 2006. Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*. ACM, 309–316.

CIANCIO, A. G. 2006. Distributed wavelet compression algorithms for wireless sensor networks. Ph.D. Dissertation, University of Southern California.

CRISTESCU, R., BEFERULL-LOZANO, B., AND VETTERLI, M. 2003. Networked Slepian-wolf: Theory, algorithms and scaling laws. *IEEE Trans. Inform. Theory 51,* 4057–4073.

DANG, T., BULUSU, N., AND FENG, W.-C. 2007. RIDA: A robust information-driven data compression architecture for irregular wireless sensor networks. In *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN'07)*. Lecture Notes in Computer Science, vol. 4373, Springer-Verlag, Berlin Heidelberg, 133–149.

DAUBECHIES, I. AND SWELDENS, W. 1998. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl. 4*, 247–269.

DENG, K., MOORE, A. W., AND NECHYBA, M. C. 1997. Learning to recognize time series: Combining ARMA models with memory-based learning. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*. IEEE Press, 246–250.

DESHPANDE, A., GUESTRIN, C., MADDEN, S. R., HELLERSTEIN, J. M., AND HONG, W. 2004. Model- driven data acquisition in sensor networks. In *Proceedings of the 13th International Conference on Very Large Databases (VLDB'04)*. VLDB Endowment, 588–599.

DI PAOLO, L., FISCHIONE, C., GRAZIOSI, F., SANTUCCI, F., AND TENNINA, S. 2006. Performance analysis of distributed source coding and packet aggregation in wireless sensor networks. In *Proceedings of the Global Telecommunications Conference (GLOBECOM)*.

DIMAKIS, A. G., SARWATE, A. D., AND WAINWRIGHT, M. 2006. Geographic gossip: Efficient aggregation for sensor networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*.

DONOHO, D. L. 2005. Compressed sensing. Tech. rep., Standford University.

DONOHO, D. L. 2006. Compressed sensing. *IEEE Trans. Inform. Theory 52*, 4, 1289–1306.

DUARTE, M. F., SARVOTHAM, S., BARON, D., WAKIN, M. B., AND BARANIUK, R. G. 2005. Distributed compressed sensing of jointly sparse signals. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*. 1537–1541.

DUARTE, M. F., WAKIN, M. B., BARON, D., AND BARANIUK, R. G. 2006. Universal distributed sensing via random projections. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN'06)*. ACM, 177–185.

EPFL. 2008. Sensorscope dataset. `http://sensorscope.epfl.ch/index.php/EnvironmentalData`.

ESCHENAUER, L. AND GLIGOR, V. D. 2002. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM, 41–47.

FASOLO, E., ROSSI, M., WIDMER, J., AND ZORZI, M. 2007. In-network aggregation techniques for wireless sensor networks: A survey. *IEEE Wirel. Commun. 14*, 2, 70–87.

GANESAN, D., GREENSTEIN, B., ESTRIN, D., HEIDEMANN, J., AND GOVINDAN, R. 2005. Multiresolution storage and search in sensor networks. *Trans. Storage 1*, 3, 277–315.

GARCIA-FRIAS, J. AND ZHAO, Y. 2001. Compression of correlated binary sources using turbo codes. *IEEE Commun. Lett. 5*, 10, 417–419.

GASTPAR, M., DRAGOTTI, P. L., AND VETTERLI, M. 2006. The distributed Karhunen-loève transform. *IEEE Trans. Inform. Theory 51*, 12, 5177–5196.

GEHRIG, N. AND DRAGOTTI, P. L. 2005. Symmetric and a-symmetric Slepian-wolf codes with systematic and non-systematic linear codes. *IEEE Commun. Lett. 5*, 1.

GOH, M. AND VENKAT, K. 2006. Msp430 competitive benchmarking. Tech. rep., Texas Instruments.

HAUPT, J. AND NOWAK, R. 2006. Signal reconstruction from noisy random projections. *IEEE Trans. Inform. Theory 52*, 4036–4048.

HAUPT, J., BAJWA, W., RABBAT, M., AND NOWAK, R. 2008. Compressed sensing for networked data. *IEEE Signal Process. Mag. 25*, 2.

HE, T., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. 2004. AIDA: Adaptive application-independent data aggregation in wireless sensor networks. *ACM Trans. Embed. Comput. Syst. 3*, 2, 426–457.

HEINZELMAN, W. B., CHANDRAKASAN, A. P., AND BALAKRISHNAN, H. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. IEEE, 3005–3014.

HEINZELMAN, W. B., CHANDRAKASAN, A. P., AND BALAKRISHNAN, H. 2002. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel. Commun. 1*, 660–670.

HONG, Y.-W. P., TSAI, Y.-R., LIAO, Y.-Y., LIN, C.-H., AND YANG, K.-J. 2010. On the throughput, delay, and energy efficiency of distributed source coding in random access sensor networks. *Trans. Wirel. Comm. 9*, 1965–1975.

HUANG, G. T. 2003. Casting the wireless sensor net. Tech. rep., MIT Press.

HUANG, S. C.-H., WAN, P.-J., VU, C. T., LI, Y., AND YAO, F. 2007. Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In *Proceedings of the IEEE INFOCOM*. 6–12.

INTANAGONWIWAT, C., GOVINDAN, R., AND ESTRIN, D. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. ACM, 56–67.

INTEL BERKELEY RESEARCH LAB. 2004. Intel lab data. `http://db.csail.mit.edu/labdata/labdata.html`.

IYER, V., RAMMURTHY, G., AND SRINIVAS, M. B. 2008. Training data compression algorithms and reliability in large wireless sensor networks. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'08)*. IEEE Computer Society, 480–485.

JEONG, J., KIM, J., CHA, W., KIM, H., KIM, S., AND MAH, P. 2010. A QoS-aware data aggregation in wireless sensor networks. In *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT'10)*. IEEE Press, 156–161.

JURDAK, R., RUZZELLI, A. G., AND O'HARE, G. M. P. 2010. Radio sleep mode optimization in wireless sensor networks. *IEEE Trans. Mobile Comput. 9*.

KAMYABPOUR, N. AND HOANG, D. B. 2010. A hierarchy energy driven architecture for wireless sensor networks. In *Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*. IEEE Computer Society, 668–673.

KARL, H. AND WILLIG, A. 2005. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons.

KASPI, A. AND BERGER, T. 1982. Rate-distortion for correlated sources with partially separated encoders. *IEEE Trans. Inform. Theory 28*, 6, 828–840.

KIM, S. 2004. Reliable transfer on wireless sensor networks. In *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications (SECON)*. 449–459.

KIMURA, N. AND LATIFI, S. 2005. A survey on data compression in wireless sensor networks. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)*. IEEE Computer Society, 8–13.

KRISHNAMACHARI, B. AND HEIDEMANN, J. 2004. Application-specific modelling of information routing in wireless sensor networks. In *Proceedings of the IEEE International Performance, Computing and Communications Conference*. Vol. 23, 717–722.

KULIK, J., HEINZELMAN, W., AND BALAKRISHNAN, H. 2002. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wirel. Netw. 8*, 2/3, 169–185.

LE BORGNE, Y.-A., SANTINI, S., AND BONTEMPI, G. 2007. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Process. 87*, 12, 3010–3020.

LEE, D.-U., KIM, H., TU, S., RAHIMI, M., ESTRIN, D., AND VILLASENOR, J. D. 2007. Energy-optimized image communication on resource-constrained sensor platforms. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks (IPSN'07)*. ACM, 216–225.

LEE, J.-H. AND JUNG, I.-B. 2010. Adaptive-compression based congestion control technique for wireless sensor networks. *Sensors 10*, 4, 2919–2945.

LEE, S., PATTEM, S., SATHIAMOORTHY, M., KRISHNAMACHARI, B., AND ORTEGA, A. 2009. Spatially-localized compressed sensing and routing in multi-hop sensor networks. In *Proceedings of the 3rd International Conference on GeoSensor Networks (GSN'09)*. Lecture Notes in Computer Science, vol. 5659, Springer-Verlag, Berlin Heidelberg, 11–20.

LI, M., GANESAN, D., AND SHENOY, P. 2009. Presto: Feedback-driven data management in sensor networks. *IEEE/ACM Trans. Netw. 17*, 4, 1256–1269.

LI, X.-Y., WANG, Y., AND WANG, Y. 2010. Complexity of data collection, aggregation, and selection for wireless sensor networks. *IEEE Trans. Comput*.

LINDSEY, S., RAGHAVENDRA, C., AND SIVALINGAM, K. M. 2002. Data gathering algorithms in sensor networks using energy metrics. *IEEE Trans. Parallel Distrib. Syst. 13*, 924–935.

LIU, S. AND CHENG, L. 2006. Efficient data compression in wireless sensor networks for civil infrastructure health monitoring. In *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks*. IEEE Press, 823–829.

LIU, C., WU, K., AND TSAO., M. 2005. Energy efficient information collection with the ARIMA model in wireless sensor networks. In *Proceedings of the IEEE Global Telecommunications Conference*. 5–10.

LIVERIS, A. D., XIONG, Z., AND GEORGHIADES, C. N. 2002. Compression of binary sources with side information at the decoder using LDPC codes. *IEEE Commun. Lett. 6*, 10, 440–442.

LU, J., VALOIS, F., DOHLER, M., AND WU, M.-Y. 2010. Optimized data aggregation in WSNs using adaptive arma. In *Proceedings of the International Conference on Sensor Technologies and Applications*. IEEE Computer Society, 115–120.

LUO, J., XIANG, L., AND ROSENBERG, C. 2010. Does compressed sensing improve the throughput of wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC'10).*

LUO, C., WU, F., SUN, J., AND CHEN, C. W. 2009. Compressive data gathering for large-scale wireless sensor networks. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom'09)*. 145–156.

MADDEN, S., FRANKLIN, M. J., HLLERSTEIN, J. M., AND HONG, W. 2002. TAG: A tiny aggregation service for adhoc sensor networks. In *Proceedings of the 5th Annual Symposium on Operating Systems Design and Implementation (OSDI)*.

MANJHI, A., NATH, S., AND GIBBONS, P. 2005. Tributaries and deltas: Efficient and robust aggregation in sensor network streams. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 287–298.

MARCO, D. AND NEUHOFF, D. L. 2004. Reliability vs. efficiency in distributed source coding for field-gathering sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. ACM, 161–168.

MATOS, T. B., BRAYNER, A., AND MAIA, J. E. B. 2010. Towards in-network data prediction in wireless sensor networks. In *Proceedings of the ACM Symposium on Applied Computing (SAC'10)*. ACM, 592–596.

MENG, J., LI, H., AND HAN, Z. 2009. Sparse event detection in wireless sensor networks using compressive sensing. In *Proceedings of the 43rd Annual Conference on Information Sciences and Systems*. 181–185.

MICRODAQ. 2010. 10hs soil moisture sensor. Tech. rep., http://www.microdaq.com/tandd/sensors/soil-moisture.php.

MICA2DOT. 2004. Mica2dot datasheet. http://www.cmt-gmbh.de/Mica2dot.pdf.

NATH, S., GIBBONS, P. B., SESHAN, S., AND ANDERSON, Z. 2008. Synopsis diffusion for robust aggregation in sensor networks. *ACM Trans. Sen. Netw. 4*, 2, 7:1–7:40.

ODED, M. AND MOORE, A. W. 1997. The racing algorithm: Model selection for lazy learners. *Artif. Intell. Rev. 11*, 193–225.

OLDEWURTEL, F., ANSARI, J., AND MAHONEN, P. 2008. Cross-layer design for distributed source coding in wireless sensor networks. In *Proceedings of the International Conference on Sensor Technologies and Applications*. IEEE Computer Society, 435–443.

ORDONEZ, F. AND KRISHNAMACHARI, B. 2004. Optimal information extraction in energy-limited wireless sensor networks. *IEEE J. Select. Areas Commun. Special Issue on Fundamental Performance Limits of Wireless Sensor Networks*, 1121–1129.

OZDEMIR, S. AND XIAO, Y. 2009. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Comput. Netw. 53*, 12, 2022–2037.

PATTEM, S., KRISHNAMACHARI, B., AND GOVINDAN, R. 2004. The impact of spatial correlation on routing with compression in wireless sensor networks. In *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. ACM, 28–35.

PATTEM, S., SHEN, G., CHEN, Y., KRISHNAMACHARI, B., AND ORTEGA, A. 2009. Senzip: An architecture for distributed en-route compression in wireless sensor networks. In *Proceedings of the Workshop on Sensor Networks for Earth and SpaceScience Applications (ESSA)*.

PERRIG, A., STANKOVIC, J., AND WAGNER, D. 2004. Security in wireless sensor networks. *Comm. ACM 47*, 6, 53–57.

POLASTRE, J., SZEWCZYK, R., AND CULLER, D. 2005. Telos: Enabling ultra-low power wireless research. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN'05)*. IEEE.

POLASTRE, J., SZEWCZYK, R., AND CULLER, D. 2007. Towards energy optimization in environmental wireless sensor networks for lossless and reliable data gathering. In *Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*. 1–6.

POTTIE, G. J. AND KAISER, W. J. 2000. Wireless integrated network sensors. *Comm. ACM 43*, 51–58.

PRADHAN, S. S., KUSUMA, J., AND RAMCHANDRAN, K. 2002. Distributed compression in dense sensor networks. *IEEE Signal Process. Mag. 19*, 51–60.

PRADHAN, S. S., PRADHAN, E., AND RAMCHANDRAN, K. 1999. Distributed source coding using syndromes (discus): Design and construction. *IEEE Trans. Inform. Theory 49*, 626–643.

PRADHAN, S. S., PRADHAN, E., AND RAMCHANDRAN, K. 2000. Distributed source coding: Symmetric rates and applications to sensor networks. In *Proceedings of the Data Compression Conference*. 363–372.

PRADHAN, S. S. AND RAMCHANDRAN, K. 2003. Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE Trans. Inform. Theory 49*, 3, 626–643.

QUER, G., MASIERO, R., MUNARETTO, D., ROSSI, M., WIDMER, J., AND ZORZI, M. 2009. On the interplay between routing and signal representation for compressive sensing in wireless sensor networks. In *Proceedings of the Information Theory an Applications Workshop (ITA)*.

RAGHUNATHAN, V., SCHURGERS, C., PARK, S., SRIVASTAVA, M., AND SHAW, B. 2002. Energy-aware wireless microsensor networks. *IEEE Signal Process. Mag. 19*, 2, 40–50.

RAJAGOPALAN, R. AND VARSHNEY, P. K. 2006. Data aggregation techniques in sensor networks: A survey. *IEEE Comm. Surv. Tutorials 8*, 48–63.

SADAGOPAN, N. AND KRISHNAMACHARI, B. 2004. Maximizing data extraction in energy-limited sensor networks. In *Proceedings of the IEEE INFOCOM*. 1717–1727.

SADLER, C. M. AND MARTONOSI, M. 2006. Data compression algorithms for energy-constrained devices in delay tolerant networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'06)*. ACM, 265–278.

SALOMON, D. 2007. *Data Compression: The Complete Reference* 3rd Ed. Springer-Verlag.

SAYOOD, K. 2006. *Introduction to Data Compression* 2nd Ed. Morgan Kaufmann Publishers Inc.

SCAGLIONE, A. AND SERVETTO, S. D. 2002. On the interdependence of routing and data compression in multi-hop sensor networks. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*. ACM, 140–147.

SENSIRION. 2010. Datasheet sht1x. Tech. rep. http://datasheet.octopart.com/SHT11-Sensirion-datasheet-5323722.pdf.

SERVETTO, S. 2003. Distributed signal processing algorithms for the sensor broadcast problem. In *Proceedings of the 37th Annual Conference on Information Sciences and Systems (CISS)*.

SEVGI, C. AND KOCYIGIT, A. 2008. On determining cluster size of randomly deployed heterogeneous WSNs. *IEEE Commun. Lett. 12*, 4, 232–234.

SHEN, G., NARANG, S. K., AND ORTEGA, A. 2009. Adaptive distributed transforms for irregularly sampled wireless sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*. IEEE Computer Society, 2225–2228.

SHEN, G. AND ORTEGA, A. 2010. Transform-based distributed data gathering. *IEEE Trans. Signal Process. 58*, 7, 3802–3815.

SHEN, G. AND ORTEGA, A. 2008a. Joint routing and 2D transform optimization for irregular sensor network grids using wavelet lifting. In *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN'08)*. IEEE Computer Society, 183–194.

SHEN, G. AND ORTEGA, A. 2008b. Optimized distributed 2D transforms for irregularly sampled sensor network grids using wavelet lifting. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*.

SLEPIAN, D. AND WOLF, J. 1973. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory 19*, 4, 471–480.

SRISOOKSAI, T., KEAMARUNGSI, K., LAMSRICHAN, P., AND ARAKI, K. 2012. Practical data compression in wireless sensor networks: A survey. *J. Netw. Comput. Appl. 35*, 1, 37–59.

STANN, F. AND HEIDEMANN, J. 2003. RMST: Reliable data transport in sensor networks. In *Proceedings of the IEEE International Workshop on Sensor Network Protocols and Applications*. IEEE, 102–112.

SUBRAMANIAN, R. AND FEKRI, F. 2006. Sleep scheduling and lifetime maximization in sensor networks: Fundamental limits and optimal solutions. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*. ACM, 218–225.

TAN, H. O. AND KÖRPEOĞLU, I. 2003. Power efficient data gathering and aggregation in wireless sensor networks. *SIGMOD Rec. 32*, 66–71.

TEXAS INSTRUMENTS. 1994. A parallel approach for matrix multiplication on the tms320c4x dsp. Tech. rep. http://focus.ti.com/lit/an/spra107/spra107.pdf.

TINYNODE. 2008. Tinynode datasheet. http://www.tinynode.com/?q=product/tinynode584/tn-584-868.

TULONE, D. AND MADDEN, S. 2006. PAQ: Time series forecasting for approximate query answering in sensor networks. In *Proceedings of the 3rd European Workshop on Wireless Sensor Networks (EWSN)*. 21–37.

VURAN, M. C., AKAN, O. B., AND AKYILDIZ, I. F. 2004. Spatio-temporal correlation: Theory and applications for wireless sensor networks. *Comput. Netw. 45*, 245–259.

WAGNER, R. S., BARANIUK, R. G., DU, S., JOHNSON, D. B., AND COHEN, A. 2006. An architecture for distributed wavelet analysis and processing in sensor networks. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*. ACM, 243–250.

WAGNER, R. SARVOTHAM, S., AND BARANIUK, R. 2005a. A multiscale data representation for distributed sensor networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05).*

WAGNER, R., CHOI, H., AND BARANIUK, R. 2005b. Distributed wavelet transform for irregular sensor network grids. In *Proceedings of the IEEE Statistical Signal Processing Workshop (SSP).*

WANG, W., GAROFALAKIS, M., AND RAMCHANDRAN, K. 2007. Distributed sparse random projections for refinable approximation. In *Proceedings of the 6th International Conference on Information Processing in Sensor Networks*. ACM, 331–339.

WANG, W., PENG, D., WANG, H., SHARIF, H., AND CHEN, H.-H. 2009. Cross-layer multirate interaction with distributed source coding in wireless sensor networks. *Trans. Wirel. Commun. 8*, 787–795.

WANG, Y.-C., HSIEH, Y.-Y., AND TSENG, Y.-C. 2009. Multiresolution spatial and temporal coding in a wireless sensor network for long-term monitoring applications. *IEEE Trans. Comput. 58*, 6, 827–838.

WELCH, T. A. 1984. A technique for high-performance data compression. *Computer 17*, 6, 8–19.

WIKIPEDIA. 2012. List of wireless sensor nodes. `http://en.wikipedia.org/wiki/List of sensor nodes`.

XIAO, J.-J., RIBEIRO, A., LUO, Z.-Q., AND GIANNAKIS, G. 2006a. Distributed compression-estimation using wireless sensor networks. *IEEE Signal Process. Mag. 23*, 4, 27–41.

XIAO, D., WEI, M., AND ZHOU, Y. 2006b. Secure-spin: Secure sensor protocol for information via negotiation for wireless sensor networks. In *Proceedings of the 1st IEEE Conference on Industrial Electronics and Applications*. 1–4.

XIONG, Z., LIVERIS, A. D., AND CHENG, S. 2004. Distributed source coding for sensor networks. *IEEE Signal Process. Mag. 21*, 5, 80–94.

XU, N., RANGWALA, S., CHINTALAPUDI, K. K., GANESAN, D., BROAD, A., GOVINDAN, R., AND ESTRIN, D. 2004. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*. ACM, 13–24.

XU, X., WANG, S., MAO, X., TANG, S., AND LI, X. Y. 2009. An improved approximation algorithm for data aggregation in multi-hop wireless sensor networks. In *Proceedings of the 2nd ACM International Workshop on Foundations of Wireless Ad Hoc and Sensor Networking and Computing*. ACM, 47–56.

YOUNIS, O. AND FAHMY, S. 2004. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Trans. Mobile Comput. 3*, 366–379.

YUEN, K., LIANG, B., AND LI, B. 2008. A distributed framework for correlated data gathering in sensor networks. *IEEE Trans. Vehicular Technol. 57*, 1, 578–593.

ZHU, Y., VEDANTHAM, R., PARK, S.-J., AND SIVAKUMAR, R. 2008. A scalable correlation aware aggregation strategy for wireless sensor networks. *Inf. Fusion 9*, 3, 354–369.

Received May 2011; revised August 2012, October 2012; accepted October 2012