Distributed Network Configuration for Wavelet-Based Compression in Sensor Networks

Paula Tarrío , Giuseppe Valenzise , Godwin Shen , and Antonio Ortega

Departamento de Señales, Sistemas y Radiocomunicaciones,
Universidad Politécnica de Madrid, Madrid, Spain
Dipartimento di Elettronica e Informazione,
Politecnico di Milano, Milan, Italy
Department of Electrical Engineering,
University of Southern California, Los Angeles, California, USA
paula@grpss.ssr.upm.es, valenzise@elet.polimi.it,
godwinsh@usc.edu, ortega@sipi.usc.edu

Abstract. En-route data compression is fundamental to reduce the power consumed for data gathering in sensor networks. Typical in-network compression schemes involve the distributed computation of some decorrelating transform on the data; the structure along which the transform is computed influences both coding performance and transmission cost of the computed coefficients, and has been widely explored in the literature. However, few works have studied this interaction in the practical case when the routing configuration of the network is also built in a *distributed* manner. In this paper we aim at expanding this understanding by specifically considering the impact of distributed routing tree initialization algorithms on coding and transmission costs, when a tree-based wavelet lifting transform is adopted. We propose a simple modification to the collection tree protocol (CTP) which can be tuned to account for a vast range of spatial correlations. In terms of costs and coding efficiency, our methods do not improve the performance of more sophisticated routing trees such as the shortest path tree, but they entail an easier manageability in case of node reconfigurations and update.

Keywords: In-network compression, wavelet lifting, distributed routing algorithms, collection tree protocol, shortest path tree.

1 Introduction

Data gathering in Wireless Sensor Networks (WSN) is generally deemed to be energy-consuming, especially when the density of sensor deployment increases, since the amount of data to be transmitted across nodes grows as well. The observation that, for many naturally occurring phenomena, data acquired at

[*] This work has been partially sponsored by the Spanish Ministry of Science and Innovation, under grant BES-2006-13954.

(spatially) neighboring nodes are correlated has suggested the use of en-route compression techniques, which directly send the acquired data to the sink in a compressed form. Such in-network compression methods and their interaction with routing have been extensively studied in the literature, both from a theoretical point of view [1,2] and in practical scenarios, using Slepian-Wolf coding [3], opportunistic compression along the shortest path tree [4] or wavelet transforms [5,6,7]. Nevertheless, only limited work [8] has been done on understanding the problems associated with distributed node configurations, i.e., how to build in a distributed fashion a routing structure along which to compress and gather data. This appears to be particularly appealing when the number of inter-node (or node-to-sink) messages is to be minimized in order to reduce initialization or reconfiguration times (e.g., in dynamic networks with mobile nodes). In the SenZip architecture [8], this aspect has been considered for the case of a tree-based spatial decorrelating transform, where the tree is built in a distributed way allowing only local message exchanges between nodes. However, a clear understanding of how different distributed routing initialization schemes impact the performance of a distributed transform is still missing.

This work builds on the distributed approach proposed in SenZip in order to analyze the effects of distributed tree-building algorithms on coding and transmission costs. As in SenZip, we focus on the in-network data compression scheme described in [7]. The key point of this algorithm is the computation of a 2D wavelet transform using a lifting scheme, where the lifting is operated along an arbitrary routing tree connecting nodes irregularly spaced in a sensor network. In SenZip, the collection tree protocol (CTP) [9], a tree-based collection service included in TinyOS, is employed to build this tree incrementally. We extend this CTP algorithm in order to fit naturally to a lifting scheme; moreover, we simplify tree management in the case of possible node reconfigurations. The proposed approach is compared with two distributed versions of the shortest path tree routing, which achieve better performance but with a higher reconfiguration complexity. We also compare the costs of transmission when the wavelet transform compression is employed with those incurred in the simple raw data gathering case. We observe that, even in the distributed setting, the gain of using a transform to decorrelate data is significant with respect to just forwarding data from nodes to the sink. Furthermore, we show that the gain of the transform increases when the average depth of peripheral nodes (leaves) of the routing tree with respect to the sink gets larger, i.e., when the average hop length in the tree is smaller (for a fixed density of the nodes in the network), as this leads to more highly correlated data (assuming that data in two nodes tends to be more correlated if the two nodes are closer to each other). Our study suggests that trees that are in general good for data gathering, will also be good when a transform is used, while trees specifically designed to improve transform performance will reduce only minimally the total costs, or will worsen them due to routing sub-optimality.

The rest of this paper is organized as follows. In Section 2, we provide a brief survey of previous work on en-route data compression, including the 2D wavelet transform of [7] used in our experiments. In Section 3 we present a tree-based collection algorithm to construct the routing tree for compression. In Section 4 we use simulations to study the impact of different distributed tree-building strategies on transmission costs. Our conclusions are in Section 5.

2 Related work

In-network data compression techniques include, among others, the distributed KLT proposed in [10], wavelet based methods [11,6,5,12,7], networked Slepian-Wolf coding [3], and, recently, distributed compressive sensing [13]. In particular we focus on wavelet transforms based on lifting, which can easily be used even in the case of arbitrary node positions. While early approaches required backward data transmission [12] (i.e., away from the sink), these bidirectional data flows (i.e. from and towards the sink) lead to higher transmission costs. Instead unidirectional transforms, where data only flows towards the sink, are preferable [11,14,7]. We focus on the approach in [7], where a unidirectional way along any routing tree, e.g., shortest path tree (SPT), can be computed. The SPT guarantees the best path from a given node to the sink from the routing perspective, but obviously does not ensure that consecutive nodes in a path contain highly correlated data. Conversely a tree that seeks to minimize inter-node distance (e.g., a minimum spanning tree, MST), in order to lower the bitrate, may have a higher transport cost. In [15], this problem is addressed by proposing a joint optimization of routing and compression. A broader perspective is adopted in [16], where each node can collect data broadcasted by its neighbors (i.e. nodes that are spatially close), even in the case they are not directly connected in the tree structure. This enables the use of a routing-optimal tree (SPT) while leveraging the context information at the same time. Very recently, Pattem et al. [8] have demonstrated for the first time an architecture for en-route compression, based on the tree wavelet transform of [7], in which the routing (compression) tree is initialized in a *distributed* way, using CTP. In this paper, our goal is to evaluate different network routing initialization techniques in terms of their impact on overall costs for a given data representation quality.

In order to compress the data gathered by the sensor nodes in a distributed manner we use the 2D unidirectional lifting transforms proposed in [7,16]. These transforms are critically sampled and computable in a unidirectional manner, so that the transmission costs are reduced. Consider a sensor network with N sensor nodes and one sink, where x_n is the data gathered by node n. Let T be the routing tree, with the root of the tree corresponding to the sink (node $N+1$). Let depth(n) be the number of hops from n to the sink in T, with depth$(N+1)=0$. To perform a lifting transform on T, we first split nodes into even and odd sets. This is done by assigning nodes of odd (resp. even) depth as odd (resp. even) in the transform. Data at odd nodes is then predicted using data from even neighbors on T, yielding detail coefficients. Then, even node data is updated using detail coefficients from odd neighbors on T. This can be done over multiple levels of decomposition, either directly on T or on other trees using the notion of "delayed processing" introduced in [16].

3 Distributed Tree Construction Algorithms

We outline in this section three distributed algorithms to initialize the routing tree on which the tree-based wavelet transform will be computed.

3.1 Modified Collection Tree Protocol (M-CTP)

The collection tree protocol (CTP) [9] is a tree-based data gathering approach that provides best-effort anycast datagram communication to the sinks in a multihop network. The protocol has an initialization phase in which one or more nodes advertise themselves as sinks (or tree roots) and the rest of the nodes form routing trees to these roots using the expected transmissions (ETX) as the routing gradient. This metric gives an estimate of the number of transmissions it takes for a node to send a unicast packet whose acknowledgement is successfully received. The ETX of a node is the ETX of its parent plus the ETX of the link to its parent and given a choice of valid routes, CTP chooses the one with the lowest ETX value. CTP assumes that the data link layer provides synchronous acknowledgements for unicast packets, so when the acknowledgement of a packet is not received, the packet is retransmitted.

In this work we have used a similar collection protocol, but instead of using the ETX metric, the protocol chooses the links with lowest inter-node distance. In the following, we refer to this modified version of the collection tree protocol as M-CTP. In the initialization phase, the sink broadcasts a 'Hello' packet using a given transmission power. All the nodes within its communication range receive this packet, and label themselves as 1-hop nodes. Then, using the same transmission power as the sink, each 1-hop node broadcasts another 'Hello' packet, which will be received by their neighboring nodes. Among these nodes, those which have not received previously any 'Hello' packet will become 2-hop nodes and they will continue with the initialization procedure in the same way. At the end, every node in the network will know their number of hops to the sink, from which it will be possible to determine the node parity (even or odd), and therefore the role of the node in the tree-based transform[7]. Furthermore, when a node receives a packet from another node, it measures the Received Signal Strength (RSS) of the message and uses this information to estimate the distance to its neighbor. Therefore, at the end of the initialization phase, each node in the network will also have estimates of the distances to its neighboring nodes. Later, when a sensor node collects data from the environment, it just chooses to send this data to the neighbor that is 1-hop closer to the sink than itself. If more than one neighbor is in this situation, the sensor node will send the data to the neighbor which is closer to itself, i.e., the neighbor from which it has the lowest distance estimation. We assume that during the data transmission phase the nodes can adjust their transmission power according to the estimated distance to their parent node in the routing tree, so that they transmit with "just enough" power to reach the parent node reliably. Depending on the transmission power that is used in the initialization phase, the resulting routing tree will be different. If a low transmission power is used, the nodes will be connected to the sink through several short hops; conversely, if the transmission power is higher, nodes will be connected through fewer, but longer, links.

This CTP-based tree construction protocol can be constructed and tuned in a very simple way, without the need of backwards transmissions from children to parents, and therefore maintains a very low initialization cost. The reason to propose this tree construction algorithm instead of using CTP trees directly is twofold. On the one hand, we are interested in using distance (not the ETX) for building the trees, as distance is likely to be more directly related to the data correlation. On the other hand, the parity of each node can be calculated very easily with the proposed algorithm, whereas with the CTP scheme it is more complex to determine, as a given path from a node to the sink can change several times during the initialization process, and this affects the parity of the node and its descendants. Note that node parity is important, since it is used to determine the role of each node in computing the wavelet transform.

3.2 Shortest Path Tree (SPT) and Minimum Distance Tree (MDT)

In order to broaden our evaluation of distributed data compression techniques over different routing trees, we also consider two other trees that can be constructed as well in a distributed way. The first one is a shortest path tree (SPT) that minimizes the sum of the squared distances between any node of the network and the sink. In an ideal propagation environment, the transmission cost for a given link is proportional to the square of the distance between the two nodes, so, under these circumstances, this shortest path tree is optimum in the sense that it minimizes the total transmission cost per bit. The second tree that we consider here is a variation of shortest path tree but, in contrast to the previous one, the objective function to be minimized in this case is the total distance to the sink. We name this tree minimum distance tree (MDT). Assuming that the correlation between sensor data is inversely proportional to the distance between the sensors, this tree is expected to perform well in terms of data compression. Both SPT and MDT can be computed in a distributed manner following the same approach used by CTP (substituting the cumulated ETX to the sink at each node, with the cumulated costs).

When the network configuration has to be built from scratch, both SPT/MDT and M-CTP can be seen as *greedy* procedures, as each node selects its parent *locally*, i.e. making a decision only on the basis of its neighbors' distances or accumulated costs. Specifically, in the initialization phase with SPT/MDT, each node n chooses a parent p_n which minimizes the sum of the costs from the parent to the sink, $C(p_n)$, plus the cost from node n to p_n, $C(n)$. When the costs are non-negative (as is the case of distances), minimizing *greedily* $C(p_n)+C(n)$ for each node n guarantees a globally optimal solution, as in fact this is nothing but the well-known Dijkstra algorithm. On the other hand, M-CTP simply minimizes the number of hops, by choosing closest matches when multiple equivalent choices are available, without any claim of optimality in the sense of total costs. The fundamental distinction between STP/MDT and M-CTP occurs when some tree *reconfiguration* is needed. Consider, without loss of generality, a simple example where a node n

with parity l in the tree runs out of battery and is no longer operational. Thus its "orphan" children need to be re-connected to other nodes. Note that in order to perform the wavelet transform this also implies that the parity of the children nodes needs to be updated to keep the transform consistent (the new configuration will be forwarded to the sink for the transform to be invertible, but we neglect this cost since it cannot be avoided with any of our routing schemes). Let $\rho(n)$ denote the set of children of node n, which have parity $l+1$ in the tree. When n dies, a reconfiguration procedure is launched. In the case of M-CTP, children nodes will have to look for a new parent node having depth in the tree greater than or equal to l using search procedure described in Section 3.1. If the new parent has level l, the configuration update is done; otherwise, if the level of the new parent is $l' > l$, each child $m \in \rho(n)$ set its parity to $l'+1$ and the set of descendants $\rho(m)$, $\rho(\rho(m))\dots$, $m \in \rho(n)$ will modify their parity accordingly as well. Thus, in the worst case, a reconfiguration with M-CTP involves a propagation through descendants of parity level information, but *no other topological changes* in the set of descendants are carried out. In the case of SPT/MDT, on the contrary, each child node $m \in \rho(n)$ needs to look for a new candidate parent node p_m that minimizes $C(m) + C(p_m)$. But once the parent node is found, the descendants of $m, t \in \{\rho(m) \cup \rho(\rho(m)) \cup \dots\}$, will not simply update their parity as in the M-CTP case, since each node t is supposed to be connected to the sink through the *minimum cost path* which minimizes $C(t) + C(p_t)$, where as before $C(p_t)$ denotes the cost of the path from the parent of t up to the sink. Therefore, a reconfiguration in the SPT/MDT case will in general entail a global re-computation of costs for the descendants of the modified node and in general this will lead to an overall modification of the tree. On the other hand, the proposed M-CTP algorithm can substantially reduce reconfiguration complexity, and thus it is a valid practical alternative to optimal SPT/MDT trees when network configuration changes frequently over time.

4 Results and Discussion

4.1 Description of Experimental Setup

In order to analyze precisely how the total energy consumption in the network depends on the different parameters we have run a set of simulations. In each simulation a network composed of 100 nodes (99 sensor nodes and one sink node) is deployed randomly in a 100 m^2 room. Three different data fields with different correlations (high, medium and low, denoted, H, M, L, respectively) are generated over the same region using a separable second order autoregressive filter. The communication channel is modeled using the lognormal shadowing path loss model [17], which establishes for each link a relation between the transmitted power (P_{TX}), the received power (P_{RX}) and the distance between transmitter and receiver (d):

$$P_{RX}(dBm) = P_{TX}(dBm) + A - 10\eta \log \frac{d}{d_0} + N. \quad (1)$$

A is a constant term that depends on the power loss for a reference distance d_0, η is the path loss exponent and $N \sim \mathcal{N}(0, \sigma^2)$ is a zero-mean gaussian noise

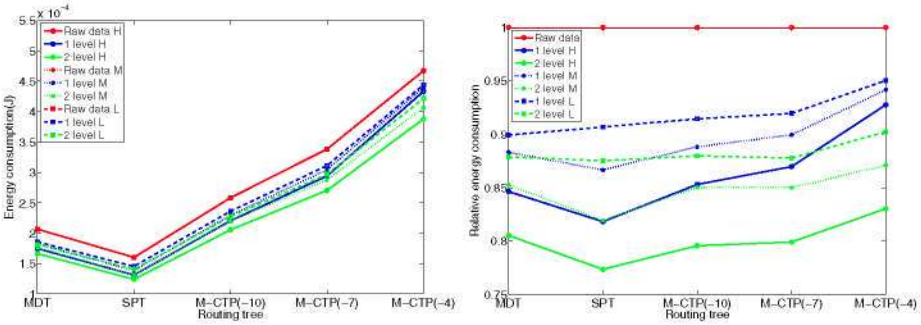

Fig. 1. Total energy consumption (*left*) and relative energy consumption with respect to the raw data case (*right*) for different routing trees. The labels M-CTP(x) correspond to the different M-CTP trees constructed with transmission power x. H, M and L represent the high, medium and low correlation datasets respectively.

with standard deviation σ. In the simulations we have set these parameters to $A = -80$ dB, $d_0 = 1$ m, $\eta = 2$ and $\sigma = 2$.

Different routing trees are constructed for each simulation: the SPT and MDT described in Section 3.2 and various M-CTP trees using the initialization procedure explained in Section 3.1 with different transmission powers. During this phase the nodes measure the RSS of all the packets they receive from their neighbors and they estimate the distances to them using the lognormal model. Then all the nodes send their data towards the sink and the tree-based wavelet transform is applied to reduce the amount of information that is sent over the network. During this data gathering phase the packets are sent with a transmission power that depends on the estimated inter-node distances: $P_{TX}(dBm) = S(dBm) - A + 10\eta \log(\frac{d}{d_0}) + M$, where S is the receiver sensitivity and M is a margin to reduce the packet loss probability (for the lognormal channel $M(dB) = -Q^{-1}(\text{PRP}) \cdot \sigma\sqrt{2}$, where Q^{-1} is the inverse Q function and PRP is the desired packet reception probability, in our simulations PRP=0.99). For the retransmissions it is reasonable to use a higher transmission power so, in the simulations, we have added 0.5 dB for each retransmission. After the data gathering phase, when the sink has received all the information, it computes the inverse transform so the original data is reconstructed (with some distortion).

4.2 Experimental Results

Using the described setup we have evaluated the total energy consumption in the network during the data gathering phase. Note that the cost of the initialization phase, i.e. the cost of constructing the routing tree, is not considered here. Fig. 1 shows the total energy consumption during the data collection phase for the tree routing algorithms described in Section 3. The cost of transmitting raw data across the network is compared against the cases of 1 and 2 levels of wavelet decomposition. For the tunable M-CTP trees, it can be seen that the

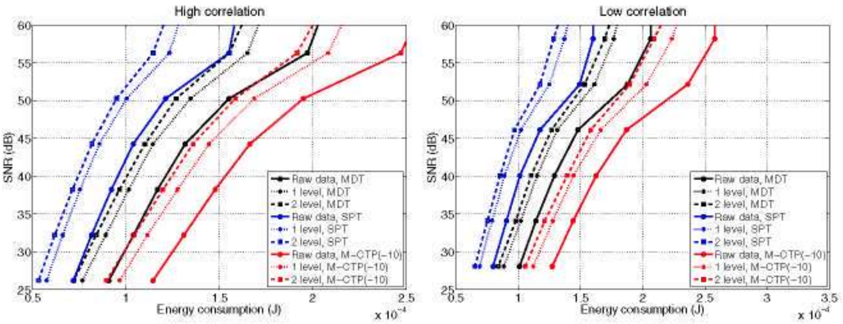

Fig. 2. SNR–Energy consumption for two datasets with different correlation. Each point in the curves represents the average over 50 simulations.

energy consumption generally increases with the transmission power; at the same time, the relative cost reduction with respect to simple data gathering (raw bits transmission) tends to get smaller. For the MDT and the SPT the results are better than those obtained with the M-CTP trees, as the former trees have a higher number of (shorter) hops. This supports the thesis that longer paths with shorter hops are better from both a transform coding gain point of view and for routing efficiency. Note that the consumption reduction yielded by wavelet transform with respect to raw data gathering does not change across the different M-CTP trees as significatively as does the total cost. This is partially due to an intrinsic overhead of the adopted transform, which necessarily requires the transmission of non compressed information for 1 or 2 hops. However, we believe that this reduced gain could improve in the case of a denser network[1].

An overall evaluation of in-network compression also needs to consider the quality of the data reconstructed at the sink. We consider in Fig. 2 SNR–Energy consumption curves for the three different routing trees (only the M-CTP tree with a transmission power of -10 dBm is shown in the graph). The SNR here is the reconstruction signal-to-quantization noise. Clearly, to obtain a lower distortion in the reconstructed data, the energy to be consumed is larger (as the number of bits increase). However, we observe that, e.g. for a target distortion of 40 dB, the M-CTP tree yields a cost reduction of about 10% (14%) with a 1 level wavelet transform in the case of data with low (high) correlation; using 2 levels of decomposition we can further gain 5% (6%) for the case of low (high) correlation. These gains are basically the same for the MDT tree, while the SPT produces a further transform gain of about 1-2%. Note that in terms of absolute consumption-distortion performance, the SPT gives the best performance, as it is designed to provide the cost-optimal tree.

[1] We are not considering channel capacity and interference effects, for which having a denser network implies an asymptotically increasing cost due to retransmission and reduced throughput [18].

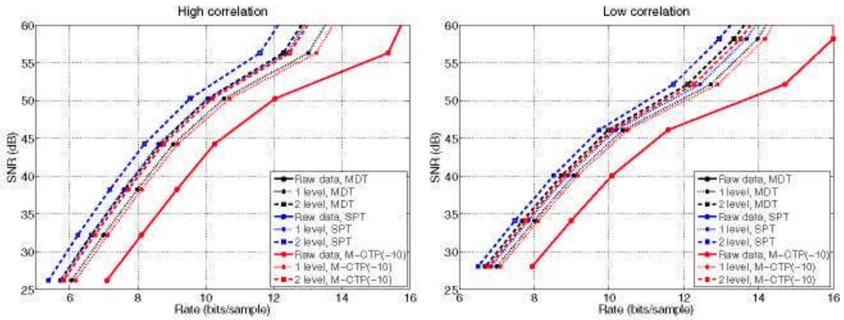

Fig. 3. SNR–Rate curves for two datasets with different correlation. Each point in the curves represents the average over 50 simulations.

In order to separate the specific contribution of the transform to the total cost, we plot in Fig. 3 SNR–Bit rate curves for the same routing trees of Fig. 2 (we only show the performance of one of the M-CTP routing trees for the sake of clarity). Again, the best performance is obtained with the 2-level transform. The marginal gain of using a transform in the MDT with respect to M-CTP is on average about 0.5 dB for both the high and low correlation datasets. The SPT can benefit more from higher correlation, and the gain when the transform is applied increases to 3 dB for the high correlation data (while it stops at 1.5 dB for low correlation). This suggests that a tree which is optimal for routing is in general optimal also for transform coding, even if a sub-optimal tree like the M-CTP can be still preferable in practice when other non-functional requirements (such as the reconfiguration time) come into play.

5 Conclusions

This paper aims at corroborating the *practical* feasibility of transform-based in-network compression using a *totally distributed* initialization approach. Our main contribution is to propose a simple, yet effective distributed tree-construction algorithm based on CTP, which fits particularly well to the 2D wavelet transform adopted in this work by embedding, explicitly, the distance between nodes and thus data correlation. We discuss how to tune, in a practical situation, the initialization power of the M-CTP scheme in order to produce trees with different depth, and we show that deeper trees will lead in general to better performance in terms of transmission costs. The proposed scheme entails lower initialization costs in comparison to distributed versions of optimal routing trees such as the SPT when some nodes need to be reconfigured (which is particularly beneficial in case of highly dynamic networks).

References

1. Scaglione, A., Servetto, S.: On the interdependence of routing and data compression in multi-hop sensor networks. Wireless Networks 11(1), 149–160 (2005)
2. Pattem, S., Krishnamachari, B., Govindan, R.: The impact of spatial correlation on routing with compression in wireless sensor networks. In: Proc. 3rd Int. Symp. on Information Processing in Sensor Networks (2004)
3. Cristescu, R., Beferull-Lozano, B., Vetterli, M.: Networked Slepian-Wolf: theory, algorithms, and scaling laws. IEEE Trans. on Inf. Theory 51(12), 4057–4073 (2005)
4. Krishnamachari, B., Estrin, D., Wicker, S.B.: The impact of data aggregation in wireless sensor networks. In: Proc. of the 22nd Int. Conf. on Distributed Computing Systems, pp. 575–578 (2002)
5. Servetto, S.: Distributed signal processing algorithms for the sensor broadcast problem. In: Proc. 37th Annual Conf. Inform. Sciences Syst. (March 2003)
6. Ciancio, A., Pattem, S., Ortega, A., Krishnamachari, B.: Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In: Proc. 5th Int. Conf. on Information Processing in Sensor Networks, April 2006, pp. 309–316 (2006)
7. Shen, G., Ortega, A.: Optimized distributed 2D transforms for irregularly sampled sensor network grids using wavelet lifting. In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, April 2008, pp. 2513–2516 (2008)
8. Pattem, S., Shen, G., Chen, Y., Krishnamachari, B., Ortega, A.: Senzip: an architecture for distributed en-route compression in wireless sensor networks. In: ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (April 2009)
9. TinyOS-2: Collection tree protocol, `http://www.tinyos.net/tinyos-2.x/doc/`
10. Cristescu, R., Beferull-Lozano, B., Vetterli, M.: On network correlated data gathering. In: Proc. 23rd Annual Joint Conf. of the IEEE Computer and Communications Societies, March 2004, vol. 4 (2004)
11. Ciancio, A., Ortega, A.: A dynamic programming approach to distortion-energy optimization for distributed wavelet compression with applications to data gathering inwireless sensor networks. In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (April 2006)
12. Wagner, R., Choi, H., Baraniuk, R., Delouille, V.: Distributed wavelet transform for irregular sensor network grids. In: Proc. IEEE Stat. Sig. Proc. Workshop (July 2005)
13. Duarte, M., Wakin, M., Baron, D., Baraniuk, R.: Universal distributed sensing via random projections. In: Proc. 5th Int. Conf. on Information processing in sensor networks, April 2006, pp. 177–185 (2006)
14. Acimovic, J., Beferull-Lozano, B., Cristescu, R.: Adaptive distributed algorithms for power-efficient data gathering in sensor networks. In: Proc. Int. Conf. on Wireless Networks, Communications and Mobile Computing (June 2005)
15. Shen, G., Ortega, A.: Joint routing and 2D transform optimization for irregular sensor network grids using wavelet lifting. In: Proc. 7th Int. Conf. on Information Processing in Sensor Networks (April 2008)
16. Shen, G., Pattem, S., Ortega, A.: Energy-efficient Graph-based Wavelets for Distributed Coding in Wireless Sensor Networks. In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (April 2009)
17. Rappaport, T.: Wireless communications. Prentice Hall PTR, Englewood Cliffs (2002)
18. Marco, D., Duarte-Melo, E., Liu, M., Neuhoff, D.: On the many-to-one transport capacity of a dense wireless sensor network and the compressibility of its data. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, pp. 1–16. Springer, Heidelberg (2003)