Opportunistic Source Coding for Data Gathering in Wireless Sensor Networks

Tao Cui (taocui@caltech.edu)

Advisor: Tracey Ho

Department of Electrical Engineering, California Institute of Technology

Problem and Motivation

Wireless sensor networks are emerging as a dominant technology in the coming decades. Recent technological advancements have made possible the deployment of a large number of small, inexpensive, low-power, distributed devices, called sensor nodes, which are capable of sensing, information processing and wireless communication. Due to limited energy and computational power, each single sensor node has only a limited amount of processing capacity. But when deployed in large numbers, they have the ability to measure and monitor a given physical environment in great detail. Sensor networks have broad applications such as environment and habit monitoring (e.g., determining the plant and animal species population and behavior), seismic detection, healthcare applications, and military surveillance and target tracking, etc. Being a pervasive system, sensor networks have the potential to revolutionize the very way we understand and construct complex physical systems, and enable automated and distributed control of environments.

Wireless sensor networks pose numerous diverse research challenges. Our focus is on data gathering, where information sampled at distributed sensor nodes needs to be communicated to central base stations for collection, processing and/or analysis. In view of the severe energy constraints of sensor nodes and the limited transport capacity of multihop wireless networks, an important topic studied by the wireless sensor networks community has been in-network data aggregation. The idea is to pre-process sensor data in the network by sensor nodes endowed with computational power, so as to eliminate data redundancy and reduce expensive data transmission.

In this work we consider data-gathering scenarios where data is sampled at a number of distributed correlated sources and needs to be routed to one or a few base stations or sinks. Data aggregation in this context involves in-network data compression and its interaction with routing, see, e.g., [1],[2]. Much of the existing work on correlated data gathering implicitly assumes routing techniques similar to those in wireline networks, neglecting the characteristics of wireless transmission. On the one hand, wireless transmission is error-prone. Sequential forwarding of packets along a fixed path may incur many retransmissions, and thus exhaust scarce network resources such as energy and capacity. On the other hand, wireless transmission is broadcast in nature. The chance that all the neighboring nodes fail to receive the packet is small (multiuser diversity in packet reception). Moreover, multiple receptions of a packet by different nodes can also be exploited to aid data compression. By leveraging the wireless broadcast advantage and multiuser diversity, we can reduce the number of wireless transmissions needed for data gathering.

We propose a jointly opportunistic source coding and opportunistic routing (OSCOR) protocol for correlated data gathering in wireless sensor networks, which exploits the broadcast nature of wireless transmission. OSCOR broadcasts each packet, which is received by possibly multiple sensor nodes, and opportunistically chooses a receiving neighbor to forward the packet, with the goal of dynamically obtaining a path with highest possible compression and best possible link quality. Opportunistic forwarding with opportunistic compression allows OSCOR to exploit multiuser diversity in packet reception, data compression and path selection, resulting in high expected progress per transmission.

The design of OSCOR involves several challenges. First, sensor nodes need to coordinate wireless transmission and packet forwarding to exploit multiuser diversity in packet reception. Second, sensor nodes need a distributed source coding algorithm that does not require full knowledge of the joint source distributions or too much coordination overhead. Finally, in order to achieve high diversity and compression gain, routing (or more precisely, forwarding decisions) must be based on a metric that is dependent on not only link-quality but also compression opportunities, which is nontrivial because the effect of data compression is not additive along a path and the source distributions are not known *a priori* but are learned online. In this work, we develop practical solutions to these challenging issues.

Background and Related Work

There are two main components in data gathering: routing and source coding. Routing is the process of selecting paths along which to send information and data, while source coding encodes correlated information using fewer bits, reducing wireless transmission and energy consumption. The Lempel-Ziv (LZ) algorithm is a popular algorithm for lossless source coding; variants of LZ are used in gzip and GIF images.

According to the way they perform routing and source coding, existing data gathering schemes proposed in the literature can be classified into four classes:

(1) Distributed Source Coding (DSC) [3]–[6]: If the sources know their joint entropy rates, they can encode/compress data by using distributed source coding (e.g., Slepian-Wolf coding) so as to avoid transmitting redundant information. In [3], it was shown that each source can send its data to the sink along the shortest path without the need for intermediate aggregation. Sources need to coordinate to operate at a certain point within the Slepian-Wolf region such that the total cost is minimized. In [4], a suboptimal hierarchical difference broadcasting scheme is proposed without requiring knowledge of joint entropy of sources. But it works for the single sink case only. The multi-sink scenario is considered in [5], where a suboptimal distributed scheme is proposed which requires information exchange between sources. In [6], we proposed a fully decentralized algorithm without requiring coordination among sources, which works for both the single sink and multi-sink cases. However, this scheme still requires knowledge of the joint entropy of sources for decoding, which is difficult to estimate in practice. Nevertheless, this scheme provides a baseline for evaluating the other schemes.

(2) Routing Driven Compression (RDC) [1]: In this scheme, the sources do not have any knowledge about their correlations. They send data along the shortest paths to the sink while allowing for opportunistic aggregation wherever the paths overlap. Such shortest path tree aggregation techniques are described, for example, in [1], where the tree is generated greedily.

(3) Compression Driven Routing (CDR) [2]: This was motivated by the scheme in [7]. As in RDC, the sources have no knowledge of the correlations but the data is aggregated close to the sources and initially routed so as to allow for maximum possible aggregation at each hop. Eventually, this leads to the collection of compressed data at a central node, which are sent to the sink along the shortest possible path.

(4) Hybrid Clustering [2]: Sources form small clusters and data is aggregated within them at a cluster head which then sends data to the sink along the shortest path. Opportunistic aggregation is also allowed wherever the paths overlap. It can be considered as a combination of both RDC and CDR. The optimal cluster size depends on the source correlations, which is unknown in advance. This scheme also requires nodes' coordination to find a cluster head.

In [3]–[6], it is assumed that every link in the network is error-free and can transmit information at a rate equal to its channel capacity. In [1], [2], only joint design of source coding and routing is considered on top of the MAC layer and the routing metric is hop distance, which does not take into account the link quality. None of [1]–[6] consider exploiting the broadcast advantage and cooperative diversity of wireless networks.

Uniqueness of the Approach

A. Basic components of OSCOR

OSCOR does opportunistic routing by exploiting the wireless broadcast advantage. As every packet can be received by possibly any node within the sender's communication range, the chance that no node receives the packet is small. An illustrative example is shown in Figure 1, where the black node is the sending node and the number beside each receiving node is the probability that it can successively receive the sent packet. The probability that no node receives the sent packet is only 0.1x0.5x0.8=0.04. Therefore, reliability is improved by exploiting wireless broadcast advantage. Each node keeps a set of its receiving nodes with different priorities that represent the relative cost of forwarding the packet by these nodes. The nodes that actually receive the packet run a protocol to agree upon the highest priority node, which will keep the packet while all the other nodes will drop the packet to prevent unnecessary multiple forwarding of the same packet. If the packet is not received by any node, the source broadcasts the packet again until it is received by at least one node. After an appropriate period of time, the priority of each node is updated by using the information collected in the past such as the compression opportunity and adjacent link quality.

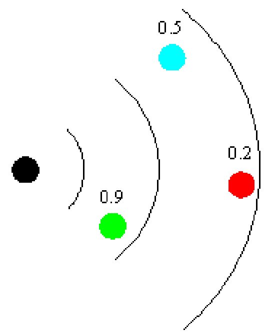
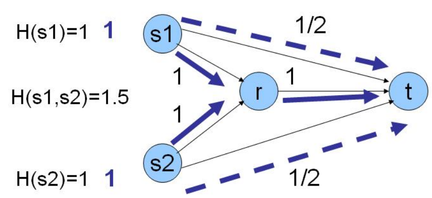


Fig. 1 An illustration of Wireless Broadcast Advantage

OSCOR uses random network coding for data compression. Network coding allows nodes to algebraically combine packets before forwarding them. The use of network coding can significantly improve the ability of the network to transfer information in multicast or lossy settings; practical implementations of such network codes are enabled by distributed random linear network coding [8]. Each coding node forms its output transmissions as a random linear combination of its input (i.e., the received packets) in some finite field. Random linear coding can also be used to perform distributed compression in a network [8]. However, network coding needs a priori knowledge of packets' joint entropies to determine how many coded packets to generate, which may not be available in practice. We solve this issue by combining network coding with Lempel-Ziv coding. The idea is to use Lempel-Ziv to obtain an estimate of the number of coded bits to generate, denoted as n, but the output of the Lempel-Ziv encoder is discarded. Random linear network coding is then applied to generate n coded bits. The coded bits formed by network coding are packetized and sent. This process can also be executed sequentially. We do not use the output of the Lempel-Ziv encoder because Lempel-Ziv is complicated to extend to the network case, where the packets formed by compression of data at a node may be received by different next hop nodes and undergo joint compression with other packets. To recover the original packets, the sink would have to run the Lempel-Ziv decoding algorithm once for each coding step in reverse order. Network coding allows the overall code to be decoded directly, since the data received by the destination is a linear transform of the sources' data. It also provides resilience to packet loss.

With opportunistic routing and forwarding, OSCOR reduces the number of retransmissions and improves reliability and throughput. Paths are not pre-specified but determined dynamically. We will see later that, by coupling routing with opportunistic compression, OSCOR exploits diversity in data compression and path selection so as to route packets over paths with high compression and good link quality.



Probability	0.25	0.25	0.25	0.25
# of Transmissions	2	3	3	3.5
Event	(t, t)	(r, t)	(t, r)	(r, r)

Fig. 2 An example on how OSCOR works

Fig. 2 gives an example on how OSCOR works. There are two sources: s1,s2. Both s1 and s2 send 1 packet each to the destination t. If 1 packet from s1 and 1 packet from s2 are compressed at r, only 1.5 packets need to be sent. Let bi denote the packet sent by source si, i = 1, 2. Link delivery probabilities are shown along the edges of the graph. For OSCOR, with probability 0.25 both b1 and b2 are received by t; with probability 0.25 b1 is received by r only and b2 is received by t; with probability 0.25 b1 is received by t and b2 is received by r only; with probability 0.25 both b1 and b2 are received by r only, where after compression 1.5 packets are needed to deliver. Therefore, the average number of transmissions is 0.25(2 + 3 + 3 + 3.5) = 2.875. For DSC, we can compress the data at s1, s2 such that s1 sends 1 packet and s2 only sends 0.5 packets along their respective shortest paths s1->t and s2->t. If we assume that 0.5 packets require 0.5 transmissions on average, DSC requires 1/0.5 + 0.5/0.5 = 3 transmissions. For RDC, without compression at sources, it requires 1/0.5+1/0.5 = 4 transmissions. Surprisingly, OSCOR outperforms not only RDC but also DSC, the prior state-of-the-art.

B. <![endif]> Forwarding Candidate Set Generation

As mentioned in section A, in OSCOR each node chooses a set of forwarding candidates with different priorities. To increase the opportunity for data compression, each node delays received packets for a period of time Tc before compressing and sending them. Each node i then computes its average compression ratio ρ_i, and estimates the average link packet delivery rate $p_{i,j}$ from i to j and average ACK delivery rate $a_{i,j}$ from j to i. The expected transmission count (ETX) from node i to node j is then estimated as $c_{i,j} = 1/(p_{i,j}a_{i,j})$. To update the forwarding candidate set for each node i, we need to first compute the least average number of transmissions required to transmit a packet from node i to sink t, denoted as w_i, which is also called the expected transmission count discounted by node compression ratio (cETX). Note that ρ_i means that on average each packet received by node i is compressed into ρ_i packets. So, the effect of data compression is not additive along a path, and existing routing algorithms are not directly applicable.

Consider a flow model in which a unit of flow corresponds to one packet per unit time. The total outgoing flow of node i is then equal to ρ_i times of the total incoming flow. Let $f_{i,j}$ denote the flow on edge (i, j). For each node v, we need to solve the following min-cost flow problem:

$$
\begin{aligned}
w_v = \min_f & \sum_{(i,j)\in\mathcal{E}} c_{i,j} f_{i,j} \\
\text{s.t.} & \sum_j f_{i,j} - \rho_i \sum_j f_{j,i} = \begin{cases} \rho_i, & \text{if } i = v, \\ -y, & \text{if } i = t, \\ 0, & \text{otherwise,} \end{cases} \\
& y \geq 0.
\end{aligned}
$$

We find that this problem can be solved efficiently using a modified Dijkstra's algorithm as follows. Let T denote the set of nodes whose w_v is definitively known. Initially, T = {t} where t is the sink node and $w_t = 0$. Add one node to T in each iteration. Initially, $w_v = \rho_v c_{v,t}$ for all nodes v adjacent to t, and $w_v = \infty$ for all other nodes $v \in V$. Do the following:

1) ***loop***
2) *Find v not in $\mathcal{T}$ with the smallest w_v;*
3) *Add v to $\mathcal{T}$;*
4) *Update w_u for all u adjacent to v and not in $\mathcal{T}$:*

$$w_u = \min\{w_u, \rho_u(w_v + c_{u,v})\};$$

5) ***until*** *all nodes are in $\mathcal{T}$.*

Let L(v) denote the forwarding candidate set of node v. For any $u \in L(v)$, it must satisfy the following conditions:

i) The ETX $c_{v,u}$ should be less than or equal to max_retry, the maximum number of retransmissions;

ii) Node u should be closer to sink t than node v, i.e., $w_v > w_u$.

Among those nodes satisfying conditions i) and ii), only the first max_fwd size lowest ($c_{v,u} + w_u$)-value nodes are added into L(v). Condition i) ensures that a packet transmitted by node v can be received with high probability at node u. Condition ii) guarantees that packet is always transmitted towards the sink. Next, all nodes u in the forwarding candidate set L(v) of node v are prioritized according to w_u. The smaller w_u is, the higher priority u has. As we rank the nodes according to w_u, the path with fewer expected number of transmissions is preferable, which may be due to both a shorter distance to the sink and a higher opportunity of data compression on this path. Note that as we adapt ρ_i and $c_{i,j}$ over time, the proposed protocol adapts to network changes such as nodes dying or moving. When $c_{i,j}$ is fixed, nodes initially did not know which path offers more data compression opportunities. With time, nodes learn this through ρ_i, and they will gradually favor paths with more opportunities for data compression. This is in contrast to most existing data gathering schemes where data compression and routing are uncoupled.

Results and Contributions

To evaluate the performance of OSCOR, we develop a packet-level simulator that implements our approach, as well as DSC and RDC. Our simulations are based on the IEEE 802.11b standard. In all simulations, each source transmits 3000 packets. We evaluate the performance of different schemes on a 4 × 4 grid network shown in Fig. 3, where nodes 1, 2, 3, and 4 are sources and node 14 is the sink. The sources' correlation follows the jointly Gaussian data model, where the elements of the covariance matrix $s_{i,j} = \exp(-d_{i,j}/c)$, and $d_{i,j}$ is the distance between nodes i and j and c is a correlation parameter.

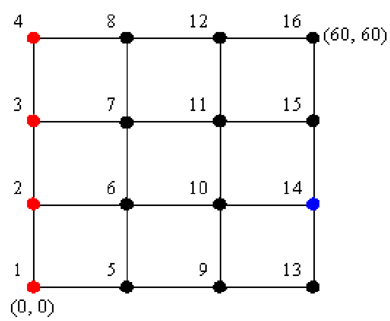


Fig. 3. A 4 × 4 grid network, where nodes 1, 2, 3, and 4 are sources and node 14 is the sink.

Fig. 4 shows the average power consumption per bit versus the correlation parameter c with different schemes. We assume that the sources have perfect knowledge of their joint entropies in DSC. As source correlation c increases, the average power consumption decreases because of higher correlation between the packets from different sources, and DSC outperforms both RDC and OSCOR as it can remove the redundancy in the packets perfectly. When c = 1000, OSCOR reduces the power consumption by 32% as compared with RDC as OSCOR uses opportunistic compression and path adaptation based on compression ratio learning . When c = 1, OSCOR achieves a 16% power saving over both RDC and DSC, which is due to multiuser diversity and spatial reuse with opportunistic routing.

Fig. 5 shows the evolution of compression ratio as a function of rounds with OSCOR. We only show nodes with compression ratio less than 0.95. Nodes 2, 3, and 6's compression ratios reduce gradually with the number of rounds. It is interesting to see that both nodes 7 and 10's compression ratios first decrease and then increase. At first, node 7 is the highest priority node in node 4's forwarding candidate set. Later node 4 finds that its packets have a better chance to be compressed more at node 3. Node 4 then puts node 3 as the highest priority node in its candidate set. The compression ratio at node 7 then increases. Finally, node 4 prefers to send to node 3, node 3 prefers node 6, nodes 1 and 2 both prefer node 6. The same analysis holds for node 10. In RDC, the path is pre-determined and fixed during data-aggregation, and it does not take path adaptation into account.

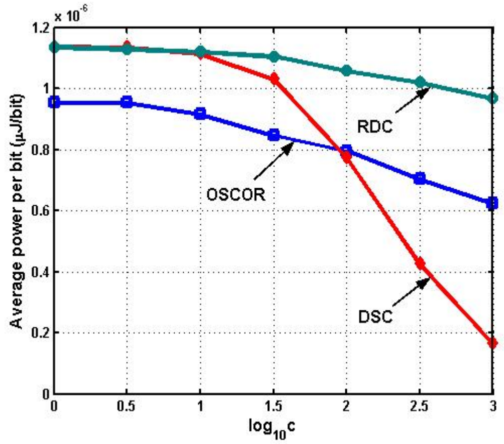


Fig. 4. The average power consumption versus correlation parameter c in the grid network in Fig. 3 with OSCOR, RDC, and DSC.

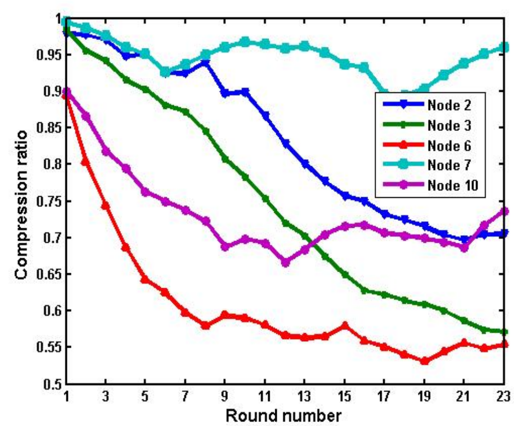


Fig.5. The evolution of compression ratio at nodes 2, 3, 6, 7 and 10 versus round number in the grid network in Fig. 3 with OSCOR.

To summarize, we propose a jointly opportunistic source coding and opportunistic routing (OSCOR) protocol for correlated data gathering in wireless sensor networks. Our main contributions are

• By slightly modifying the 802.11 MAC, we design a low overhead consensus protocol to coordinate wireless transmission and packet forwarding. Although it needs coordination between nodes to choose a single forwarder out of multiple receiving nodes, our protocol is "local" and flexible enough to allow good spatial reuse and to allow easy extension to applications with multicast traffic and multiple sessions.

• We propose a practical distributed source coding scheme that combines and takes advantage of both Lempel-Ziv coding and network coding. Lempel-Ziv coding does not require the knowledge of the statistics of the data, while network coding is well-suited to distributed compression of information in networks.

• We use expected transmission count discounted by node compression ratio (cETX) and expected opportunistic transmission power discounted by node compression ratio (cOETP) along a path as the path metrics for routing. These two path metrics cannot be simply described as the summation of some link metric over the links in a path. So, existing routing algorithms are not directly applicable. We propose modified Dijkstra's algorithms to update the path metrics cETX and cOETP from a node to the sink and select the shortest path, which is used to prioritize the neighboring nodes and update the forwarding candidate set of a node.

With opportunistic routing and forwarding, OSCOR reduces the number of retransmissions and improves reliability and throughput. By coupling routing with opportunistic compression, OSCOR exploits diversity in data compression and path selection so as to route packets over paths with high compression and good link quality. OSCOR also provides a practical distributed source coding scheme that combines and takes advantage of both Lempel-Ziv coding and network coding, and has advantages over the former in terms of robustness to packet loss and one-step decoding.

References

[1] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proc. of International Conference on Distributed Computing Systems*, 2002, pp. 575–578.

[2] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proc. of International Conference on Information Processing in Sensor Networks*, April 2004, pp. 28–35.

[3] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked Slepian-Wolf: Theory, algorithms and scaling laws," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4057–4073, Dec. 2005.

[4] J. Liu, M. Adler, D. Towsley, and C. Zhang, "On optimal communication cost for gathering correlated data through wireless sensor networks," in *Proc. of ACM MobiCom*, 2006, pp. 310–321.

[5] K. Yuen, B. Li, and B. Liang, "Distributed data gathering in multi-sink sensor networks with correlated sources," in *Proc. of IFIP Networking*, May 2006, pp. 868–879.

[6] T. Cui, T. Ho, and L. Chen, "On distributed distortion optimization for correlated sources," in *Proc. of IEEE International Symposium on Information Theory*, Jun. 2007.

[7] A. Scaglione and S. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 149–160, Jan. 2005.

[8] T. Ho, M. M´edard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.