A Study on Combined Routing and Source Coding with Explicit Side Information in Sensor Networks

Huiyu Luo and Gregory Pottie
Electrical Engineering Department
University of California at Los Angeles
Los Angeles, Ca, 90095, USA
Email: {huiyu,pottie}@ee.ucla.edu

***Abstract*— This paper studies the problem of combining tree routing and data compression with explicit side information in wireless sensor networks. We first present our network flow and data rate model based on the observation that in many practical situations, side information providing the most coding gain comes from a few nearby sensors. An optimization problem is then formulated and shown to be NP hard. It is subsequently cast as a mixed integer program. For our particular model, we examine several approximation algorithms, and compare their performances through simulations. Improvement over shortest path trees, which completely ignore the source correlation, is achieved only by judiciously merging flows of correlated data based on the coding gain information.**

I. INTRODUCTION

Many researchers have proposed using in-network data fusion to reduce the communication cost in wireless sensor networks. Routing methods that facilitate such fusion are being actively explored [2]–[5]. Two difficulties in developing such schemes are the lack of reasonably practical data aggregation models and the high computational complexity due to the coupling of routing and data processing. For example, many assume that sensors perform the same aggregation function regardless of the origin of fused data. Algorithms so-devised bear strong marks of these simplifying assumptions. In this paper, we attempt to build network models that are computationally useful yet reasonably approximate reality. Several heuristic algorithms are evaluated based on these models.

Source coding in sensor networks is often lossy. Although high resolution lossy coding resembles Slepian-Wolf [6], network coders that approach information theoretic bounds generally employ long blocks of data and have high complexity. In this paper, we assume that only when the side information is available at both the encoder and decoder, can it be used to reduce the data rate. In practice, a lossy encoder (such as the DPCM encoder in [7]) can be implemented at each sensor to compress its measurements using incoming data as side information. Besides local quantization, joint entropy coding using for example a Lempel-Ziv encoder can be conducted when flows with correlated data merge in the network. Communication cost over a wireless channel is abstracted as the edge weight, which is interpreted as the cost (e.g. power) that is required to achieve unit data rate under some given channel transition matrix, modulation, and channel coding schemes. The resulting problem becomes how to design the network routes based on the network topology and source correlation such that the given objective function is optimized.

Recent research has produced many routing algorithms that address the limited power budget of wireless sensor networks. The interaction of routing and data compression using explicit side information is discussed from the viewpoint of information theory in [8]. Clustering methods have been used by some researchers [9], [10] to aggregate data at the cluster head before transmitting them to the fusion center. In [2], a diffusion type routing paradigm that attaches attribute-value pairs to data packets is proposed to facilitate the in-network data fusion. Closely related to our work are the correlated data routing problems studied in [3], [4]. A similar optimization problem is also the subject of [5], where a simplified data model is assumed.

The rest of the paper is organized as follows. In section II, we discuss network flow and data rate models, based on which the optimization problem is formulated, and cast as a mixed integer program in section III. Section IV presents several sub-optimal algorithms. Their average performance is studied through simulations in section V. The paper concludes in section VI.

II. NETWORK MODELS

A. Network flows

The sensor network is modelled as a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. The node set $\mathcal{N}$ includes a set $\mathcal{N}_s$ of n sensors and one fusion center t, and the edge set $\mathcal{A}$ consists of m communication links. $\mathcal{N}_a$ is the set of active sensors that generate data. Both active and nonactive sensors can be relays. If a directed edge (i, j) exists, direct data transmission from node i to node j is allowed. Otherwise, relays have to be used. We assume the network is connected in the sense that there is at least one path from each $i \in \mathcal{N}_s$ to t. Data transmission across $e \in \mathcal{A}$ (or $(i, j) \in \mathcal{A}$) is represented by the flow f_e (or f_{ij}). When there is the need to identify the origin of a flow, we use f^k to indicate that the flow is generated by sensor k. Thus, $f_e = \sum_{k\in\mathcal{N}_a} f_e^k$. A weight c_e is defined on each $e \in \mathcal{A}$ to characterize the cost of communicating across e at unit rate. The objective is to minimize the cost C of transmitting all the data to the fusion center.

$$C = \sum_{e\in\mathcal{A}} c_e f_e \qquad (1)$$

We assume the routing structure is a tree $\mathcal{T} = (\mathcal{N}_T, \mathcal{A}_T)$, on which a unique path exists from each active sensor to t and every $e \in \mathcal{A}_T$ carries nonzero flow.

B. Source coding with explicit side information

Side information is used to help code the data only when it is available at both the encoder and decoder. Suppose the side information for coding data stream X_i is $\hat{X}_{k_1}, \cdots, \hat{X}_{k_j}$, where $k_1, \cdots, k_j \in \mathcal{H}_i$. $\mathcal{H}_i$ is the set of sensors whose data are correlated with X_i, and $\hat{X}_k$ denotes the coded version of X_k. Under lossy coding, the minimum rate required such that X_i can be recovered subject to some distortion constraint is:

$$f^i = \min_{d(X_i,\hat{X}_i)\leq D} I(X_i, \hat{X}_i | \hat{X}_{k_1}, \cdots, \hat{X}_{k_j}) \tag{2}$$

where $I(\cdot)$ denotes the mutual information. When entropy coding is used,

$$f^i = H(\hat{X}_i | \hat{X}_{k_1}, \cdots, \hat{X}_{k_j}) \tag{3}$$

In either case, the rate depends on what side information is available, hence is a function of $M_i = |\mathcal{H}_i|$ binary variables. ($|\mathcal{S}|$ denotes the number of elements in set $\mathcal{S}$.) For a network of n sensors, M_i can be as large as $(n-1)$. Thus, this description alone requires an exponential amount of information. To simplify the problem, we assume that M_i is relatively small and side information from at most k_s sensors is used. Our analysis focuses on the simplest case $k_s = 1$:

$$f_e^i = \begin{cases} b_0^i & \text{no side information} \\ \min_j b_1^{ij} & f_e^j > 0 \text{ and } j \in \mathcal{H}_i \end{cases} \tag{4}$$

where b_0^i is the rate without side information, and b_1^{ij} the rate when j is the helper. When i produces no data, $b_0^i = 0$.

C. Discussions

Although trees are used as the underlying routing structure, we make no claim of their optimality. In fact, there are instances where other structures outperform trees.

To enforce the implicit chain rule in Eq.s (2) and (3), we label the set of sensors $\mathcal{N}_a$ according to some order such that (1) there is a unique relation $i < j$ defined for any pair $i, j \in \mathcal{N}_a$; (2) if $i < j$ and $j < k$, then $i < k$ for $i, j, k \in \mathcal{N}_a$. We postulate that i can be in set $\mathcal{H}_j$ only when $i < j$.

In many physical situations, high correlation occurs only in a small neighborhood. In others, reconstruction fidelity constraints often permit thinning the number of active sensors, so again only a small number of sensors has high correlation. Coding gain usually saturates as the number of helpers increases due to the correlation of side information. Moreover, determining coding gain and processing side information incurs cost, and the gains of using more than a few sensors are often not enough to be worth it. Therefore, it is generally reasonable to assume small values for M_i and k_s.

The b_1^{ij} in Eq. (4) may have slightly different values depending on whether a lossy or lossless encoder is used. We ignore this subtlety. Also, Eq. (4) implies that different $\hat{X}_j$'s may be used as side information to code X_i as f^i merges with different flows. This is a first order approximation with limited information at hand.

TABLE I

DATA RATE WITH DIFFERENT SIDE INFORMATION.

Side info	none	s_0	s_1	s_2	s_0, s_1	s_0, s_2	s_1, s_2
σ_3^2	5.33	.443	.973	3.43	.424	.431	.943
f^3 (bits)	6.19	4.40	4.96	5.87	4.36	4.38	4.94

We adopt an example from [7] to solidify our data rate model. In Fig. 1, a near-field sensor array records the sound of a tank as it moves by. This array is part of a wireless sensor network (not shown in the figure), and their data are to be transmitted to a fusion center. The variance of measurements at s_3 is listed in Table I when they are quantized alone or with side information using an adaptive DPCM encoder. The data rate is estimated by $0.5 \log(\sigma_3^2/D)$, where $D = 0.001$. Notice the coding gain varies significantly with sensor locations, and saturates as the number of helpers exceeds one. In practice, the cost of processing side information may lead to $\mathcal{H}_3 = \{s_0, s_1\}$ and using only one helper.

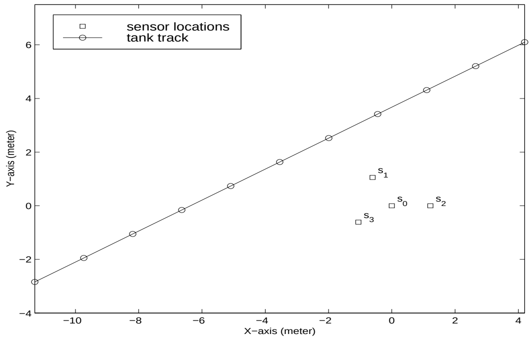

Fig. 1. Near field sensor array configuration

III. PROBLEM FORMULATION

A. NP-completeness

We describe the decision version of our problem in a two-part standard format [11]. The corresponding optimization problem can be easily derived from it.

Combined Tree Routing and Source Coding (CTRSC)
INSTANCE: A digraph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with weight c_e defined on each $e \in \mathcal{A}$, a node $t \in \mathcal{N}$, a set $\mathcal{H}_i$ and the rate f^i as in Eq. (4) defined for each $i \in \mathcal{N}_s = \mathcal{N} \setminus \{t\}$, a number C_0.
QUESTION: Is there a tree $\mathcal{T}$ such that the cost of routing all the data to t using $\mathcal{T}$, is no more than C_0?

Proposition 1: CTRSC is NP-complete.

We recognize that CTRSC is related to but different from the explicit communication problem studied in [3], and it can be reduced to the Steiner tree problem, whose relevance to data-centric routing was noted in [4]. Here, we base our proof on the polynomial transformation of 3-dimensional matching, which is stated as follows [11]:

3-Dimensional Matching (3DM)
INSTANCE: A set $\mathcal{W} \subseteq \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, where $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ are disjoint sets consisting of the same number q of elements.
QUESTION: Does $\mathcal{W}$ contain a subset $\mathcal{V} \subseteq \mathcal{W}$ such that $|\mathcal{V}| = q$ and no two elements of $\mathcal{V}$ agree in any dimension?

Proof: We will only show how to reduce 3DM to CTRSC. The rest is easy. Given a instance of 3DM, create a node for each element in sets $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, and a special node t. For each element $w_i = (x_i, y_i, z_i) \in \mathcal{W}$, form directed edges (x_i, y_i), (y_i, z_i), and (z_i, t). Discard duplicate edges. For example, if $\mathcal{X} = \{x_1, x_2, x_3\}$, $\mathcal{Y} = \{y_1, y_2, y_3\}$, $\mathcal{Z} = \{z_1, z_2, z_3\}$, and $\mathcal{W} = \{(x_1, y_1, z_2)$, (x_2, y_2, z_2), (x_3, y_2, z_3), $(x_2, y_3, z_1)\}$, the network is constructed as in Fig 2. Set the weight of each edge in $\mathcal{G}$ to 1. The set of helping nodes $\mathcal{H}_i$ is $\emptyset$ for $i \in \mathcal{X}$, $\mathcal{X}$ for $i \in \mathcal{Y}$, and $\mathcal{Y}$ for $i \in \mathcal{Z}$. Define the data rate $f^i = 1$ without side information, and 0 otherwise. Set $C_0 = 3q$. Then asking if there exists a routing tree on network $\mathcal{G}$ with cost no more than C_0 is equivalent to asking if there is a matching $\mathcal{V}$ for the original problem. Q.E.D.

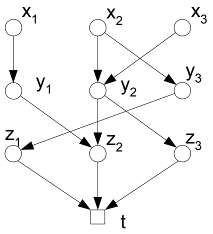

Fig. 2. The network constructed for a 3DM instance.

B. Mixed integer programming

In this section, we develop the mixed integer program, to which standard techniques, e.g. branch and bound, can be applied. The objective is to minimize $\sum_{e \in A} c_e f_e$. To construct a spanning tree $\mathcal{T}$ on $\mathcal{G}$, we use the multi-commodity flow formulation in [12].

$$\sum_{j \in \mathcal{O}(t)} g^k_{tj} - \sum_{j \in \mathcal{I}(t)} g^k_{jt} = -1, \quad k \in \mathcal{N}_s \tag{5}$$

$$\sum_{j \in \mathcal{O}(k)} g^k_{kj} - \sum_{j \in \mathcal{I}(k)} g^k_{jk} = 1, \quad k \in \mathcal{N}_s \tag{6}$$

$$\sum_{j \in \mathcal{O}(i)} g^k_{ij} - \sum_{j \in \mathcal{I}(i)} g^k_{ji} = 0, \quad i, k \in \mathcal{N}_s, \ i \neq k \tag{7}$$

$$\sum_{j \in \mathcal{O}(i)} y_{ij} = 1, \quad i \in \mathcal{N}_s \tag{8}$$

$$y_e \geq g^k_e \geq 0, \quad k \in \mathcal{N}_s \tag{9}$$

$$y_e = \{0, 1\}, \quad e \in \mathcal{A} \tag{10}$$

where $\mathcal{O}(i) = \{j \in \mathcal{N} : (i,j) \in \mathcal{A}\}$ and $\mathcal{I}(i) = \{j \in \mathcal{N} : (j,i) \in \mathcal{A}\}$. y_e is the binary variable indicating whether edge e is used to construct the tree. The flow conservation constraints (5)∼(7) postulate that one unit of flow g^k_e is generated at each $k \in \mathcal{N}_s$ and consumed at the fusion center. Summing Eq. (8) over all $i \in \mathcal{N}_s$ gives rise to $\sum_{e \in \mathcal{A}} y_e = n$. Hence, the route is connected and has exactly n edges, which is a spanning tree. Accordingly, g^k_e takes only the value of 1 or 0, which tells whether e carries the flow generated by sensor k. With these in mind, the data flow is formulated as follows:

$$f^k_e = \lambda^k_e b^k_0 + \sum_{j \in \mathcal{H}_k} \lambda^{kj}_e b^{kj}_1 \tag{11}$$

$$g^j_e \geq \lambda^{kj}_e \geq 0, \quad \lambda^k_e \geq 0 \tag{12}$$

$$\lambda^k_e + \sum_{j \in \mathcal{H}_k} \lambda^{kj}_e = g^k_e \tag{13}$$

where $k \in \mathcal{N}_s, j \in \mathcal{H}_k, e \in \mathcal{A}$. Eq. (13) ensures that f^k_e is nonzero only when $g^k_e = 1$. In virtue of minimization, exactly one of λ^k_e and $\lambda^{kj}_e, j \in \mathcal{H}_k$ is 1 and the rest are 0 for each $k \in \mathcal{N}_a$. This gives rise to the data rate function in Eq. (4).

The edges with nonzero flows constitute the solution route. There are $m\,(2n + 2 + \sum_{i=1}^n M_i)$ variables, of which m are binary. The number of constraints is on the same order. The cost of exactly solving it for a large network is rather high. We project its application to be limited to small networks. For instance, it can be used as a sub-algorithm in a clustering approach to sort out intra-cluster routes.

IV. APPROXIMATION ALGORITHMS

In light of the high computational complexity, sub-optimal algorithms are of interest. We examine several heuristics.

A. Shortest path tree

A shortest path tree (SPT) is used to route data to t, and data compression is performed whenever explicit side information is present. We establish a result regarding SPT's worst case performance when the rate model is given by:

$$f^i_e = \begin{cases} b_0 & \text{with side information} \\ \beta b_0 & f^j_e > 0, \ j \in \mathcal{H}_i \end{cases}, \quad 0 \leq \beta \leq 1 \tag{14}$$

Proposition 2: The costs of SPT and optimal solutions satisfy the following relation ($n_a = |\mathcal{N}_a|$):

$$C_{\text{OPT}} / C_{\text{SPT}} \geq \beta + (1 - \beta)/n_a \tag{15}$$

Proof: Denote by $\mathcal{A}_{\text{ST}}$ the set of edges in the Steiner arborescence (ST) [13] that connects all active sensors $\mathcal{N}_a$ to t, and d^{SPT}_i the distance from i to t on the SPT.

$$C_{\text{SPT}} \leq b_0 \sum_{i \in \mathcal{N}_a} d^{\text{SPT}}_i \leq b_0 n_a \sum_{i \in \mathcal{A}_{\text{ST}}} c_e \tag{16}$$

On the other hand, we have the following for a optimal tree.

$$C_{\text{OPT}} \geq \beta b_0 \sum_{i \in \mathcal{N}_a} d^{\text{SPT}}_i + b_0 (1 - \beta) \sum_{e \in \mathcal{A}_{\text{ST}}} c_e \tag{17}$$

These lead to (15). Q.E.D.

It is no surprise that the worst ratio is a strong function of the coding gain. The bound is tight in that there are cases in which the equality in Eq. (15) holds. Such an instance is given in Fig. 3, where the bound is achieved when $\epsilon \to 0$. The average performance will be studied in section V.

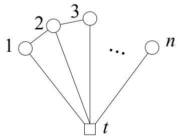

Fig. 3. An instance that achieves the bound: $\mathcal{N}_a = \{1, \cdots, n\}$; $\mathcal{H}_i = \{j : j < i\}$, $c_{it} = 1, i \in \mathcal{N}_a$; $c_{k,k+1} = c_{k+1,k} = \epsilon$, $1 \leq k \leq (n-1)$.

B. Greedy and local search

Given a routing tree $\mathcal{T} = (\mathcal{N}_T, \mathcal{A}_T)$, suppose $i, j, k \in \mathcal{N}_T$ and $(i,k) \in \mathcal{A}_T$. We build a tree $\mathcal{T}'$ by constructing a path from i to j that uses none of the nodes in $\mathcal{N}_T$ as relay and removing edge (i,k). If there is a path from each active sensor to t on $\mathcal{T}'$, a new routing tree is obtained by removing from $\mathcal{T}'$ edges carrying zero flow. We call this new routing tree a neighbor of $\mathcal{T}$. An example is given in Fig. 4, where we assume all sensors are active.

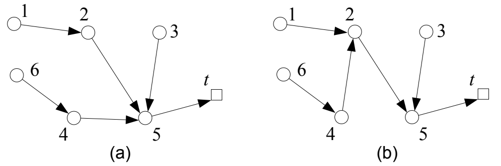

Fig. 4. $i = 4, j = 2, k = 5$, (a) original tree $\mathcal{T}$, (b) neighboring tree.

The greedy and local search starts with a SPT and searches in its neighborhood the tree of minimum cost. Continue with the new tree until there is no cost improvement.

Greedy and Local Search (GLS)

Given a sensor network $\mathcal{G}$, carry out the following steps.

(1) Find the shortest path tree $\mathcal{T}$. Compute its cost C.

(2) Search in $\mathcal{T}$'s neighborhood the routing tree $\mathcal{T}'$ with the smallest cost C'.

(3) If $C \leq C'$, stop the algorithm. Otherwise, replace C by C' and $\mathcal{T}$ by $\mathcal{T}'$. go back to (2).

Each iteration runs in $O(mn)$ time. The algorithm terminates in finite iterations because it visits a routing tree at most once. In our simulations, the number of iterations grows about linearly with the network size.

C. Balanced tree

The balanced tree algorithm is inspired by the idea of balancing SPT and trees of small total weights [14]. It adds the routes of active sensors in successive steps. Each time, the newly added node incurs the smallest cost among the remaining sensors. We state the algorithm as follows:

Balanced Tree Algorithm (BTA)

Given a $\mathcal{G}$ with edge weights and data rates appropriately defined, set $\mathcal{U} = \mathcal{N}_a$, $\mathcal{V} = \emptyset$. Carry out the following steps.

(1) Find the shortest path from $i \in \mathcal{U}$ to t. Find $I = \arg\{\min_{i \in \mathcal{U}} C_i\}$, where $C_i = b_0^i d_i$. Add I to $\mathcal{V}$, and its path to the solution route. Remove I from $\mathcal{U}$.

(2) For each $i \in \mathcal{U}$, determine the smallest cost C_i of routing i to t by adding a path to the existing routes.

(3) Find $I = \arg\{\min_{i \in \mathcal{U}} C_i\}$. Add I to $\mathcal{V}$, and the path of I to the solution route. Remove I from $\mathcal{U}$.

(4) If $\mathcal{U} = \emptyset$, stop the algorithm. Otherwise, return to (2).

Note that both GLS and BTA avoid the worst performance in Fig. 3. The bottleneck of BTA lies in constructing minimum cost paths from $i \in \mathcal{U}$ to existing routes. Using Dijkstra's algorithm, it runs in $O(n_a m \log n)$ for a sparse network, which is generally more efficient than GLS.

D. Clusters

The routing structure in a clustering method is often not a tree but comprises a hierarchy of trees. The network is divided into clusters. The sensors in each cluster transmit to a cluster head, which performs aggregation and subsequently sends data to the fusion center. The computational cost is comparable to that of the SPT if shortest paths are used in each cluster.

V. SIMULATIONS

A. Simulation setup

We place $(n+1)$ nodes including t and n sensors in an $n_d \times n_d$ square, where $n_d = \lceil \sqrt{n+1} \rceil$. Supposing $\tilde{x}_i$ and $\tilde{y}_i, i = 1, \cdots, n+1$, are random variables uniformly distributed in $[0,1]$, the coordinates of node i is given by:

$$\begin{aligned} x_i &= [(i \bmod n_d) - 1] + \tilde{x}_i \\ y_i &= \lfloor (i-1)/n_d \rfloor + \tilde{y}_i \end{aligned}$$

Denote by r_c the transmission radius. If two nodes are no more than r_c away from each other, direct communication between the two nodes is allowed. Denote by d_e the Euclidean length of edge e. When $d_e \leq r_c$, the weight c_e is chosen to be proportional to d_e^α, where α is the path loss factor. We use $\alpha = 2$. We assume that all the sensors are active. Any sensor j $(j < i)$ that is no more than r_d away from i has a probability of p_h to be in $\mathcal{H}_i$. The data rate model in Eq. (14) is used.

B. Simulation results

Denote by C_{SPT}, C_{GLS}, C_{BTA}, and C_{Cluster} the routing cost of SPT, GLS, BTA and Cluster heuristics. Define C_{ASP} to be the total cost when an address centric SPT, in which no data compression is performed, is used. The performance ratios of heuristic algorithms to address centric SPT are computed:

$$\mu_s = \frac{C_{\text{SPT}}}{C_{\text{ASP}}},\ \mu_g = \frac{C_{\text{GLS}}}{C_{\text{ASP}}},\ \mu_c = \frac{C_{\text{Cluster}}}{C_{\text{ASP}}},\ \mu_b = \frac{C_{\text{BTA}}}{C_{\text{ASP}}}$$

Fig. 5 plots the performance ratios against network size for $\beta = 0.1$ and 0.6. The wiggling of the μ_c curve is mainly due to the irregular distribution of sensors when $\sqrt{n+1}$ is not an integer. We observe that data centric routing schemes perform better as n increases. This is expected because the rate reduction affects the total cost more as the average distance from sensors to the fusion center increases. The ratio will eventually saturate before reaching β. Our clustering method is based solely on geographic proximity. We notice its inferior performance because the route construction does not take into account the source correlation. Possible improvements include varying cluster sizes and constructing intra-cluster routes based

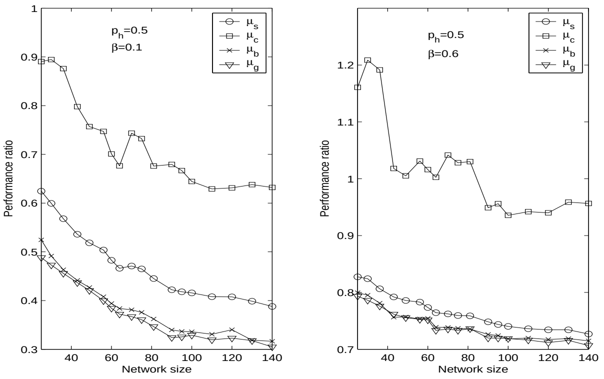

Fig. 5. Performance ratios versus network size when $p_h = 0.5$.

on coding gain. Lastly, it is observed that the gain of data-centric routing decreases as β increases from 0.1 to 0.6. To better illustrate this, in Fig. 6, we plot μ while varying β from 0 to 1 on a 100 node network. For our simulation setup, SPT fares very well. This is because in a SPT, nearby sensors have a good chance of quickly merging their flows, which leads to data compression and cost reduction. This also explains that although the gain over address-centric SPT improves, the gap between μ_s and μ_b, μ_g decreases when p_h increases to 1. Besides, as β tends to 1, GLS and BTA converge to SPT. We also simulated with $k_s = 2$ assuming a uniform rate function for all sensors. Comparing to $k_s = 1$ with the same overall coding gain, the data-centric routing appears less effective. This is understandable since the rate reduction is less drastic, i.e. more side information is required for the same coding gain. Consequently, SPT is closer to the optimal solution.

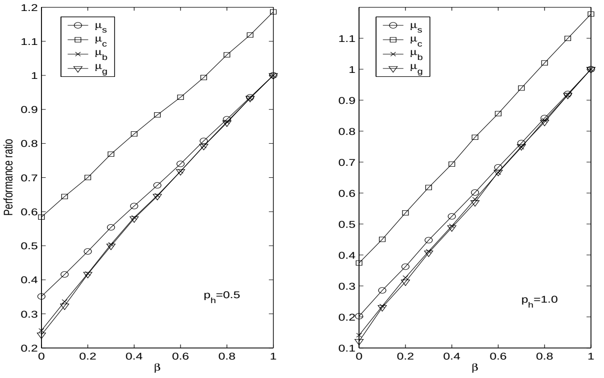

Fig. 6. Performance ratio versus coding gain when $p_h = 0.5$ and 1.0.

Our simulations were conducted on networks where sensors are evenly distributed and high source correlation occurs in a small neighborhood. This leads to fairly good results for SPT. In reality, the rate function and node distribution in sensor networks are far more complicated than what we assumed in simulations. Nevertheless, a few guidelines on constructing good heuristics can be drawn from our experience. (1) The algorithm should utilize the coding gain information whenever it is available. (2) It should be able to adapt to varying coding gain and network topology. In particular, it converges to a SPT when coding gain diminishes, and avoids the worst case scenario in Fig. 3. (3) The algorithm should have low complexity and be easy to implement in a distributed form. Among the heuristics in section IV, BTA appears to be the best candidate.

VI. CONCLUSION

Our study highlights the importance of knowing coding gain in designing data-centric routes. Serious performance loss may occur absent such knowledge, leading naturally to the question of how to obtain such information. One approach is to ask active sensors to periodically circulate short sequences of observations in their neighborhood. Such actions are often required to determine which sensors transmitting to the fusion center in a sampling process [15]. Another approach is to feed back information from the sensors or fusion center that perform data aggregation. If neither are available, simple data descriptions such as the attribute-value pairs in [2] may be used to grossly indicate the level of data correlation.

Our ongoing work continues to explore the potential of heuristic algorithms with focus on the worst case performance bounds and distributed implementations.

ACKNOWLEDGEMENT

The work is supported by the National Science Foundation (NSF) under the Cooperative Agreement #CCR-0121778.

REFERENCES

[1] G. Pottie and W. Kaiser, “Wireless sensor networks,” *Comm. ACM,* vol. 43, no. 5, pp. 51-58, May 2000.

[2] C. Intanagonwiwat, et al. “Directed diffusion for wireless sensor networking,” *IEEE/ACM Trans. Networking,* vol. 11, no. 1, pp. 2-16, Feb 2003.

[3] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, “On network correlated data gathering,” *IEEE Infocom,* Hongkong, March 7-11, 2004.

[4] B. Krishnamachari, D. Estrin, and S. Wicker, “Modelling data-centric routing in wireless sensor networks,” *USC Computer Engineering Technical Report CENG,* 02-14, 2002.

[5] A. Goel and D. Estrin, “Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk, *ACM/SIAM Symposium on Discrete Algorithms,* 2003.

[6] R. Zamir and Toby Berger, “Multiterminal source coding with high resolution,” *IEEE Trans. Inform. Theory,* vol. 45, no. 1, Jan 1999.

[7] H. Luo, Y. Tong, and G. Pottie, “A Two-Stage DPCM Scheme for Wireless Sensor Networks,” submitted to ICASSP 2005.

[8] A. Scaglione and S. Servetto, “On the interdependence of routing and data compression in multi-hop sensor networks,” *ACM/IEEE Mobicom,* 2002.

[9] S. Bandyopadhyay and E. J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” *IEEE Infocom,* 2003.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication procotol for wireless microsensor networks,” *Hawaii Inter. Conf. System Sciences,* Jan. 4-7, 2000.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W. H. Freeman and Company, 1979.

[12] T. L. Magnanti and L. A. Wolsey, “Optimal trees,” Chapter 9 in *Network Models,* Handbooks in Operations Research and Management Science, vol. 7, ed. M. O. Ball et al., Elsevier, 1995.

[13] F. Hwang, D. Richards, and P. Winter, *The Steiner Tree Problem,* North-Holland, 1992.

[14] S. Khuller et al, “Balancing minimum spanning trees and shortest-path trees,” *Algorithmica,* vol. 14, pp 305-321, 1995.

[15] R. Willett, A. Martin, and R. Nowak, “Backcasting: adaptive sampling for sensor networks,” *IPSN,* 2004, Berkeley, Ca, USA.

[16] T. M. Cover and J. A. Thomas, *Elements of Information Theory,* John Wiley & Sons, Inc. 1991.