

# A distributed minimum-distortion routing algorithm with in-network data processing

Ramin Khalili

Dept. of Computer Science  
University of Massachusetts, Amherst, MA  
Email: ramin@cs.umass.edu

Jim Kurose

Dept. of Computer Science  
University of Massachusetts, Amherst, MA  
Email: kurose@cs.umass.edu

**Abstract**—In many wired and wireless networks, nodes process input traffic to satisfy a network constraint (e.g., a capacity constraint) and to increase the utility of data in the output flows given these constraints. In this paper we focus on the special case in which data processing is applied to satisfy capacity constraints. This occurs when the sum of the rate of the input traffic at a node exceeds the sum of the capacity of its output links, or in a more general case, when the sum of the input rates is larger than any cut capacity in the network. In this case, nodes process data to decrease the output flow rate. This decrease from input rate to output rate distorts the transmitted data, which we characterize by a distortion metric. We show that the distortion cost of distributively processing input traffic in a network can be written as the sum of the distortion at individual nodes. We present a distributed algorithm for a data-gathering network with many sources and a data sink that routes traffic and performs in-network data processing to minimize the distortion cost. In this algorithm, each node determines its routing table based on gradient information from neighboring nodes.

## I. INTRODUCTION

Increasingly, distributed sensing systems are being developed and deployed that monitor/measure an environment, and report these observations to a central site. Examples include in-situ monitoring of natural habitats [1]–[3], collaborative adaptive sensing of the atmosphere [4], multi-camera video surveillance [5], and measurement/monitoring of engineered systems such as transportation systems [6], and data networks [7]. In many of these systems, a potentially large set of measurements is routed to a central collector. In contrast to traditional networks whose purpose is to forward stored data to the data sink, the purpose of a sensor network is to provide users access to information of interest in the data. For example, in a temperature sensor network, the central sink node might only need receive the local average temperature collected by temperature sensors. In this case, communicating all the stored measurements to the central collector is highly inefficient. Moreover, a large sensor network may be severely constrained in energy, memory, or bandwidth. Therefore, in many wired and wireless sensor networks, nodes process input traffic (rather than just forwarding this traffic) to satisfy a set of

This work was supported in part by the National Science Foundation under award number CNS-0519998 and under the Engineering Research Centers Program, Award number EEC-0313747. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

network requirements (such as capacity or energy constraints) or to increase the utility of the output flows. We refer to this alternative solution as in-network data processing, where the entire network is performing application-specific distributed data processing.

In this paper we focus on the special case in which data processing is performed to satisfy capacity constraints. This occurs when the sum of the input traffic rates at a node is larger than the sum of the capacity of its output links, or in a more general case, when the sum of the input rates is larger than any cut capacity in the network. In this case, nodes must process data to decrease the output flow rate, where the effect of processing is to reduce the rate of input flows to a lower aggregate outgoing rate. However, the decrease from input rate to output rate distorts the transmitted data, which we characterize by a distortion metric. Our definition of distortion is quite general and is not restricted to any specific kind of data processing mechanism. Therefore, our work differs from [8], [9], where in [8], [9] in-network processing is limited to data compression with known correlation structure among input flows.

We first show that the distortion cost of distributively processing input traffic flows in a network can be written as the sum of the distortion increments at individual nodes. This is done by introducing per-node marginal distortion costs that can be calculated locally at the individual nodes. We then propose a gradient-based distributed algorithm for a data-gathering network with many sources and a data sink that optimally routes traffic and performs in-network data processing to minimize the distortion cost. This algorithm requires each node to decide its own routing table based on update information from neighboring nodes. Our work thus differs from [10] in which the goal is to design a joint routing resource allocation mechanism to maximize an overall system utility in the network. Our work is also different from [11]–[15] that establish information-theoretic bounds on the maximum rate at which functions of measurement data can be computed and communicated to the data sink nodes. Random gossiping algorithms [16], [17] are based on information propagation through randomly selected neighbors to compute aggregate functions in the network, where only a restricted class of processing functions are considered, and thus also differ from our work.

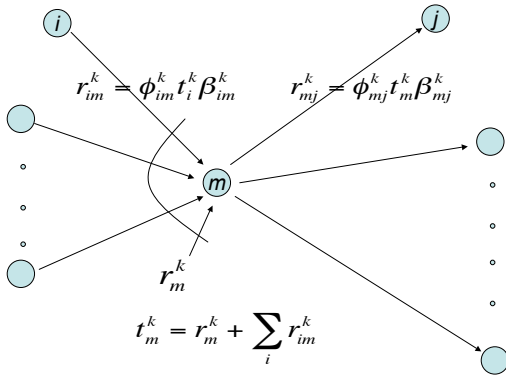


Fig. 1. The node flow  $t_m$  is the sum of the input traffic and traffics routed to  $i$  from other nodes.

The rest of this paper is structured as follows. In the following section we describe the network setting and assumptions for our study. In section II-A, we define the distortion cost metric for simple point-to-point communication. We then show how the per-path distortion cost can be calculated as the sum of the per-node marginal distortion at individual nodes on a path. This is the subject of section II-B. In section II-C, we generalize our definition of distortion cost to a network-wide scenario and present a distributed formulation for our optimization problem. In section III-A we generalize the necessary and sufficient conditions in [18] for an optimal set of routes to minimize the network-wide distortion cost. In section III-B, we propose a distributed algorithm to solve this optimization problem by generalizing Gallager's results [18]. Section IV reviews the related work. We conclude this paper in section V.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

We model the network with a directed graph  $G = (V, L)$  where  $V$  is the set of nodes and  $L$  is the set of directed links in the network. Let  $n$  denote the single sink node in the network and  $1, \dots, n-1$  denote non-sink nodes in  $V$ . In this paper we assume a single type of data and a single sink, but this can be generalized [19]. We denote by  $(m, j) \in L$  a link from  $m$  to  $j$ . We consider that if link  $(m, j)$  exists, link  $(j, m)$  exists too.

Let  $r_m \geq 0$  be the traffic rate, in symbols/sec, entering the network at node  $m$ . Now let  $t_m$  be the total incoming traffic rate at node  $m$  destined for  $n$  (see Figure 1).  $t_m$  includes both  $r_m$  and the traffic from other nodes routed through  $m$  to the data sink.

Let  $\phi_{mj}$ ,  $\sum_m \phi_{mj} = 1$ , be the fraction of traffic  $t_m$  decided to be routed over  $(m, j)$  to downstream node  $j$ . We consider that node  $m$  processes this flow before transmitting it over link  $(m, j)$ , where a symbol unit of this flow produces  $\beta_{mj}$  symbol units of output after processing. We shall refer to the parameter  $\beta_{mj}$ ,  $0 < \beta_{mj} \leq 1$ , as a *shrinkage factor*, which represents the shrinkage effect in data processing. The value of  $\beta_{mj}$  are given. If  $\phi_{mj} t_m$  is the fraction of node flow  $t_m$  to be routed over link  $(m, j)$ , then the output flow would have a rate of  $r_{mj} = \phi_{mj} t_m \beta_{mj}$ . As  $t_m$  at node  $m$  is the sum of

the input traffic and the traffic routed to  $m$  from other nodes in the network, then

$$t_m = r_m + \sum_{i \in V - \{m\}} r_{im} = r_m + \sum_{i \in V - \{m\}} \phi_{im} t_i \beta_{im} \quad (1)$$

for all  $i \in V$ . This equation expresses the balance of flow at each node accounting for shrinkage factors. Note that  $r_{mj} \leq C_{mj}$ , where  $C_{mj}$  is the capacity of link  $(m, j)$ .

Let us denote the set of inputs rates  $\{r_m\}$  as  $r$ , the set of routing variables  $\{\phi_{mj}\}$  as  $\phi$ , the set of shrinkage factors  $\{\beta_{mj}\}$  as  $\beta$ , and the set of total nodes flow rates  $\{t_m\}$  as  $t$ . We are interested in a distributed algorithm in which each node  $m$  in the network decides its own routing variables  $\phi_{mj}$  for each  $j$ . For this purpose we define  $\phi$  in the same manner as [18] to ensure that the set of equations (1) have unique solutions for  $t$  given  $r$ ,  $\phi$ , and  $\beta$ .

**Definition 1** A routing variable set  $\phi$  for network  $G = (V, L)$  is a set of non-negative numbers  $\phi_{mj}$ ,  $m, j \in V$ , satisfying the following conditions :

- $\phi_{mj} = 0$  if  $j = n$ , or  $(m, j) \notin L$ ,
- $\sum_j \phi_{mj} = 1$ , for all  $m$
- for each  $1 \leq m \leq n-1$  there is a routing path from  $m$  to  $n$ , i.e., there is a sequence of nodes  $m, j, o, \dots, l, n$  such that  $\phi_{mj} > 0, \phi_{jo} > 0, \dots, \phi_{ln} > 0$ .

Before going to the details of the routing algorithm, let us consider the in-network computation as represented by  $\beta$ .

### A. Distortion Cost

In this paper we focus in the case that processing reduces the rate of input flows to a node to lower aggregate outgoing rate. For example, let  $X$  (with rate  $r_i$ ) be the input flow at node  $i$  and assume that this node communicates directly to the data sink  $n$  over a link with capacity  $C_{in}$  where  $r_i > C_{in}$ . Node  $i$  generates output flow  $Y$  from  $X$  using the processing function of  $g(\cdot)$  where  $Y = g(X)$ . A symbol unit of flow  $X$  produces  $\beta_{in} (< 1)$  symbol units of  $Y$  after processing and thus the output traffic rate is  $r_{in} = r_i \beta_{in}$ , in symbols/sec, where  $r_i \beta_{in} < C_{in}$  (see Figure 2(a)).

Let us now consider the distortion caused by shrinkage in more detail. The data sink's estimate,  $\hat{X}$ , of the original  $X$  is some function,  $h(\cdot)$ , of the received data  $Y$ , i.e.,  $\hat{X} = h(Y) = h(g(X))$ , with an average distortion of  $\mathcal{D}_{in}(\beta_{in}) = d[X, h(g(X))]$  where  $d(\cdot)$  is a distortion measure and the expectation is with respect to the probability distribution on  $X$  [20].  $\mathcal{D}_{in}(\beta_{in})$  is a measure of the cost of representing  $X$  by  $\hat{X}$ , i.e., the distortion cost of processing a symbol unit of input data to  $\beta_{in}$  symbol units of output data  $Y$  at node  $i$ . For our purpose, it is immaterial what function  $\mathcal{D}_{in}(\cdot)$  is, although we shall make the reasonable assumption that  $\mathcal{D}_{in}$  is decreasing, positive, and convex  $\cup$  in  $\beta$ .

### B. Distortion along a path

Let us now consider the simple scenario in which  $r_i$  crosses a set of nodes in path  $p = (i, i+1, \dots, n-1, n)$  and let  $r_j = 0$  for  $j \neq i$ . Let  $\{\beta_{i(i+1)}, \dots, \beta_{(n-1)n}\}$  be shrinkage factors over

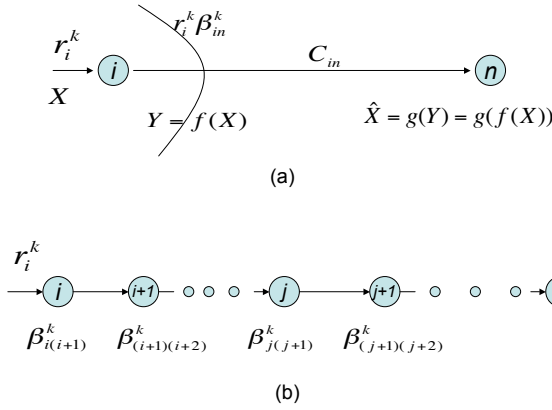


Fig. 2. Per-node and per-path data processing

this path. The distortion cost of distributively processing  $r_i$  over this path is  $D_T = \mathcal{D}_p(r_i \prod_{j=i}^{n-1} \beta_{j(j+1)})$ , where  $\mathcal{D}_p(\cdot)$  reflects the distortion effects of processing at nodes along this path. Now let  $j \in p$  be a downstream node of  $i$  as depicted in Figure 2(b). We define  $\beta'_{(j-1)j} = \prod_{h=i}^{j-1} \beta_{h(h+1)}$  as the shrinkage of a symbol unit of  $t_j$  relative to a symbol unit of  $r_i$ . Now define the per-node marginal distortion cost of processing  $t_j$  from the shrinkage level  $\beta'_{(j-1)j}$  to the new shrinkage level  $\beta'_{j(j+1)}$  as

$$\Delta_{j(j+1)} = \mathcal{D}_{j(j+1)}(\beta'_{j(j+1)}) - \mathcal{D}_{(j-1)j}(\beta'_{(j-1)j})$$

where  $\mathcal{D}_{(j-1)j}(\beta'_{(j-1)j}) = 0$ , for  $j = i$ . The sum of this per-node distortions over all nodes in  $p$  is equal to the distortion cost of distributively processing  $r_i$  over this path, where

$$D_T = \sum_{j=i}^n \Delta_{j(j+1)} = \mathcal{D}_{(n-1)n}(\beta'_{(n-1)n}),$$

and  $\mathcal{D}_{(n-1)n}(\cdot)$  is what we defined as  $\mathcal{D}_p(\cdot)$ .

In section III-A we will show that only nodes that are connected directly to the receiver (for the example in Figure 2, node  $n - 1$ ) need estimate their own distortion measure functions ( $\mathcal{D}_p(\cdot)$ ) and the marginal per-node distortion costs of the other nodes can be calculated based on the update information received from their neighbors.

### C. Network-wide distortion

Now we consider a general network setting in which data enter at different points into the network. We define node  $j$  to be a downstream node of  $m$  if there is a routing path from  $m$  to data sink  $n$  passing through  $j$ . Node  $i$  is defined as an upstream node of  $m$  if  $m$  is downstream from  $i$ . In a general setting, the rate of traffic at  $m$  is expressed by equation (1) where  $r_{im}$  is the input flow received from upstream node  $i$ . These inputs, arriving from different upstream nodes, have different rate and different shrinkage levels as they may have crossed different paths in the network.  $m$  combines these flows together where the resulting has a rate of  $t_m = r_m + \sum_{i \in L-\{m\}} r_{im}$  which is equal to the sum of the input rates. Let  $\beta'_{im}$  be the shrinkage

level of upstream flow  $r_{im}$ . We define

$$\beta'_m = \left( \frac{r_m + \sum_{i \in L-\{m\}} r_{im}}{r_m + \sum_{i \in L-\{m\}} \frac{r_{im}}{\beta'_{im}}} \right), \quad (2)$$

as the shrinkage level of  $t_m$  relative to the fraction of flow  $r = \{r_m\}$  (input traffic before any processing) routed through  $m$  to the destination.

**Definition 2** Let  $\beta'_{im}$  be the shrinkage level of  $r_{im}$  arriving at node  $m \neq n$ . Now let  $r_{mj} = \phi_{mj} t_m \beta_{mj}$  be the flow routed over output link  $(m, j)$ . The shrinkage level of  $r_{mj}$  is  $\beta'_{mj} = \beta_{mj} \beta'_m$  and the marginal distortion of this procedure (processing a fraction  $\phi_{mj}$  of  $t_m$  to  $r_{mj}$ ) is defined as

$$\Delta_{mj} = \mathcal{D}_{mj}(\beta'_{mj}) - \phi_{mj} \sum_i \mathcal{D}_{im}(\beta'_{im}). \quad (3)$$

Let  $\mathcal{N}$  be the set of nodes that have a direct link to data sink  $n$ . We define the distortion metric  $D_T$  as

$$D_T = \sum_{m \in \mathcal{N}} \mathcal{D}_{mn}(\beta'_{mn}) \quad (4)$$

Note that if link flow rates  $r_{mn}$ ,  $m \in \mathcal{N}$ , are equivalent, then  $D_T$  is actually the average distortion cost of distributively processing  $r$  in the network. This is the case in a network with links with the same capacity limit where the flow rates in links connected to the sink node are close to the capacity of those links.

**Theorem 1**  $D_T$  can be written as the sum of the marginal distortion costs over nodes in the network, i.e. :

$$D_T = \sum_{m,j} \Delta_{mj}. \quad (5)$$

*Proof:* See the technical report [19]. ■

In the next section we propose a distributed algorithm to minimize  $D_T$  in the network.

## III. DISTRIBUTED MINIMUM-DISTORTION ALGORITHM

In this section, we present a distributed algorithm for each node to locally adjust routing variables to converge to the optimal (in the sense of minimizing  $D_T$  in equation (4)) set of routes by generalizing Gallager's result [18]. We first give the necessary and sufficient conditions for minimum distortion set of routes.

### A. Necessary and sufficient conditions for minimum distortion

We first derive the partial derivatives of  $D_T$  with respect to input  $r$  and the routing variable  $\phi$ . Due to space limit, we summarize here; see [19] for more details.

Assume a small increment of  $\epsilon$  in input  $r_m$ . This increment will cause a local increment of  $\mathcal{R}_{mj}$  in the distortion cost of processing traffic  $\phi_{mj} t_m$  at node  $m$ , where  $j$  is a downstream node of  $m$ . It will also cause a change in the shrinkage level of traffic  $t_j$  at node  $j$ . Summing over all these changes and over all adjacent nodes  $j$ , we have :

$$\frac{\partial D_T}{\partial r_m} = \sum_j \phi_{mj} \left[ \mathcal{R}_{mj} + \frac{\partial D_T}{\partial r_j} \right] \quad (6)$$

where

$$\mathcal{R}_{mj} = \begin{cases} \frac{-\beta'_j(1-\beta'_{mj})}{t_j} \frac{\partial D_T}{\partial \beta'_j} & j \neq n \\ \frac{\beta_{mn}(1-\beta'_m)}{\phi_{mn}} \frac{d\mathcal{D}_{mn}(\beta'_{mn})}{d\beta'_{mn}} & j = n \end{cases}. \quad (7)$$

Next we consider  $\frac{\partial D_T}{\partial \phi_{mj}}$ . An  $\epsilon$  increment in  $\phi_{mj}$  causes a local increment of  $\mathcal{H}_{mj}$  in the distortion cost of processing traffic  $\phi_{mj}t_m$  at  $m$ , as well as a change in the shrinkage level of traffic  $t_j$  at node  $j$ , where

$$\frac{\partial D_T}{\partial \phi_{mj}} = \frac{t_m}{\beta'_m} \left[ \mathcal{H}_{mj} + \frac{\partial D_T}{\partial r_j} \right] \quad (8)$$

for

$$\mathcal{H}_{mj} = \begin{cases} \frac{-\beta'_j(1-\beta'_{mj})}{t_j} \frac{\partial D_T}{\partial \beta'_j} & j \neq n \\ 0 & j = n \end{cases}. \quad (9)$$

Finally we consider the marginal term  $\frac{\partial D_T}{\partial \beta'_m}$ . An  $\epsilon$  increment in  $\beta'_m$  will cause a change of

$$\frac{\partial D_T}{\partial \beta'_m} = \sum_j \phi_{mj} \beta_{mj} \left[ \mathcal{A}_{mj} + \frac{\beta'_j t_m}{\beta'_m t_j} \frac{\partial D_T}{\partial \beta'_j} \right] \quad (10)$$

in  $D_T$ , where

$$\mathcal{A}_{mj} = \begin{cases} 0 & j \neq n \\ \frac{1}{\phi_{mn}} \frac{d\mathcal{D}_{mn}(\beta'_{mn})}{d\beta'_{mn}} & j = n \end{cases}. \quad (11)$$

Equation (6), (8), and (10) are the set of equations that describe our system. In the following theorem we show the existence and uniqueness of  $\frac{\partial D_T}{\partial r_m}$  and  $\frac{\partial D_T}{\partial \phi_{mj}}$  given by equations (6) and (8).

**Theorem 2** *Let a network  $G = (V, L)$  have input traffic set  $r$  and routing variable set  $\phi$ , and let  $\mathcal{R}_{mj}$  and  $\mathcal{H}_{mj}$  be continuous in  $\beta'_{mj}$  for each  $m$  and  $j$ . Then the set of equations (6) has a unique solution for  $\frac{\partial D_T}{\partial r_m}$ . Moreover,  $\frac{\partial D_T}{\partial r_m}$  and  $\frac{\partial D_T}{\partial \phi_{mj}}$  for  $(m, j) \in L$  are continuous in  $r$  and  $\phi$ .*

*Proof:* See the technical report [19]. ■

Using Lagrange multipliers for the constraint  $\sum_m \phi_{mj} = 1$ , where  $\phi_{mj} \geq 0$ , the necessary conditions with respect to  $\phi$  and for all  $(m, j) \in L$  are

$$\frac{\partial D_T}{\partial \phi_{mj}} \begin{cases} = \lambda_m & \phi_{mj} > 0 \\ \geq \lambda_m & \phi_{mj} = 0. \end{cases} \quad (12)$$

However, as shown by [18], (12) is not a sufficient condition to minimize  $D_T$ . Next we propose a sufficient condition for the optimization problem.

**Theorem 3** *Assume that for each  $(m, j) \in L$ ,  $\mathcal{D}_{mj}(\beta_{mj})$  is convex and continuously differentiable for  $0 < \beta_{mj} \leq 1$ . Let  $\Psi$  be the set of  $\phi$  for which the shrinkage levels are larger than zero. Then (12) is necessary condition for  $\phi$  to minimize  $D_T$  over  $\Psi$  and the sufficient condition is*

$$\frac{\partial D_T}{\partial r_m} \leq \frac{\partial D_T}{\partial r_j} + \mathcal{R}_{mj}. \quad (13)$$

*Proof:* See the technical report [19]. ■

## B. A distributed algorithm for minimum distortion

Based on the sufficient condition given by Theorem 3, we now propose a gradient-based routing algorithm by generalizing [18]. Each node  $m$  must incrementally decrease those routing variables for which the marginal distortion cost  $\frac{\partial D_T}{\partial r_j} + \mathcal{R}_{mj}$  is large and increase those for which this marginal cost is small. The algorithm has two parts: a protocol between neighboring nodes in network to calculate the marginal distortion costs, and an algorithm for modifying the routing variables  $\phi$ . We first discuss the protocol part.

Each node  $m$  can calculate the average shrinkage level ( $\beta'_m$ ) of its own traffic knowing the shrinkage level of each input flow using (2). Thus, if  $m$  has a direct link  $(m, n)$  to the destination, with an appropriate formula for  $\frac{d\mathcal{D}_{mn}(\beta'_{mn})}{d\beta'_{mn}}$ , it can calculate the marginal costs of  $\frac{d\mathcal{D}_{mn}(\beta'_{mn})}{d\beta'_{mn}}$ .

Let us now consider how node  $m$  can calculate  $\frac{\partial D_T}{\partial r_j}$  and  $\frac{\partial D_T}{\partial \beta'_j}$ . As in [18] we assume that the routing variable  $\phi$  is loop free, *i.e.*, there is no  $m$  and  $j$  ( $m \neq j$ ) such that  $j$  is both upstream and downstream for  $m$ . The updating protocol is now as follows : each node  $m$  waits until it has received  $\frac{\partial D_T}{\partial r_j}$  and  $\frac{\partial D_T}{\partial \beta'_j}$  from downstream neighbor nodes  $j \neq n$ . Node  $m$  can then calculate  $\frac{\partial D_T}{\partial r_m}$  from (6) and (7) with the convention that  $\frac{\partial D_T}{\partial r_n} = 0$ .  $\frac{\partial D_T}{\partial \beta'_m}$  can be calculated using (10) and (11) with the convention that  $\frac{\partial D_T}{\partial \beta'_n} = 0$ .  $m$  then broadcasts these values to all of its neighbors except the destination which has no need of the information. It easy to show that this procedure is loop-free if and only if  $\phi$  is loop-free.

Now we describe algorithm  $\Gamma$  to modifying the routing variables. To avoid the deadlocks and similar to [18], our algorithm requires a small amount of additional information to maintain the loop-free property : each node  $m$  maintain a set  $\mathcal{B}_m$  of blocked nodes  $j$  for which  $\phi_{mj} = 0$  and the algorithm is not permitted to increase  $\phi_{mj}$  from zero. See [19] for the definition of  $\mathcal{B}_m$  and how to use it to maintain loop free property.

Following [18], on each iteration, the algorithm  $\Gamma$  maps the current routing variable  $\phi$  into a new set of variables  $\phi(1) = \Gamma(\phi)$  using the following mapping structure. For  $j \in \mathcal{B}_m$ ,  $\phi_{mj}(1) = 0$  and  $\delta_{mj} = 0$ . For  $j \notin \mathcal{B}_m$ , we define

$$\delta_{mj} = \min[\phi_{mj}, \eta a_{mj}/t_m] \quad (14)$$

where

$$a_{mj} = \frac{\partial D_T}{\partial r_j} + \mathcal{R}_{mj} - \min_{l \notin \mathcal{B}_m} \left[ \frac{\partial D_T}{\partial r_l} + \mathcal{R}_{ml} \right]. \quad (15)$$

$\eta$  is a scale parameter of  $\Gamma$  and will be discussed shortly. Note that if we multiply both sides of (13) by  $\phi_{mj}$  and sum over  $j$ , then we see from (6) that (13) must be satisfied with equality for  $\phi_{mj} > 0$ . Thus the sufficient condition in theorem 3 is equivalent to  $a_{mj} \geq 0$  for all  $j \neq n$ ,  $(m, j) \in L$  with equality for  $\phi_{mj} > 0$ .

Let's consider that  $i_{min}(m)$  is a value of  $l$  that achieves the minimization in (15), then

$$\phi_{mj}(1) = \begin{cases} \phi_{mj} - \delta_{mj} & j \neq i_{min}(m) \\ \phi_{mj} + \sum_{j \neq i_{min}(m)} \delta_{mj} & j = i_{min}(m). \end{cases}$$

The algorithm reduces the fraction of traffic sent on non-optimal (high distortion) paths and increases the fraction on the best path. The amount of reduction is given by  $\delta_{mj}^k$  which is inversely proportional to the scale factor  $\eta$ . For  $\eta$  very small, convergence of the algorithm is slow but guaranteed as shown in following theorem.

**Theorem 4** For each  $(m, j) \in L$ , assume that  $\mathcal{D}_{mj}(\beta'_{mj})$  has a negative first derivative and non-negative second derivative for  $0 < \beta_{mj} \leq 1$  and that  $\lim_{\beta_{mj} \rightarrow 0} \mathcal{D}_{mj}(\beta_{mj}) = \infty$ . For every positive number  $D_0$ , if  $\phi(0)$  satisfies  $D_T(\phi(0)) \leq D_0$ , then there exists a scale factor  $\eta$  for  $\Gamma$  such that

$$\lim_{l \rightarrow \infty} D_T(\phi(l)) = \min_{\phi} D_T(\phi)$$

where  $\phi(l) = \Gamma(\phi(l-1))$ .

*Proof:* See the technical report [19]. ■

#### IV. RELATED WORK

Joint optimization of source coding and routing for lossy data compression is proposed in [9], where high resolution source coding is assumed. The problem considered in [9] is to find the optimal distributed transmission structure and the rate-distortion allocations in which the total power consumption minimized. [8] proposed a distributed joint source-network coding algorithm that maximizes an aggregate utility measure that is defined in terms of the distortion levels of the data sources. In both of these works it is assumed that the rate-distortion region, and therefore the correlation structure among data sources, is known by the source nodes in the network. Distributed rate-distortion optimization for video compression [21]–[23] is another example in which the in-network processing is limited to source compression algorithm.

In [10] a distributed stream processing systems consisting of a network of servers is considered in which each server provides processing services to the input streams. The network source is considered to be constrained both in computing power at each server and the bandwidth capacity over communication links. [10] proposed a joint routing, source allocation distributed algorithm to minimize a utility loss in the network, thus is relevant to our problem. However, in [10] the utility loss is simply defined as the difference between the overall utility of source data and the utility can be delivered by the network to the destination, where sources are processed independently from each other at the intermediate nodes. Moreover, in-network processing is restricted to be the same over any two distinct paths that have the same starting and ending node.

#### V. CONCLUSION

In this paper we proposed an optimal routing strategy to minimize the distortion cost of processing data in the

network. We first defined the distortion metric for point-to-point communication. We then showed how this definition could be generalized to a network setting. The proposed routing algorithm requires each node to locally compute its own routing variables based on update information from downstream neighboring nodes. This algorithm reduces the fraction of traffic sent on non-optimal (high distortion) paths and increases the fraction on the best path.

#### REFERENCES

- [1] W. Hu, V. N. Tran, N. Bulusu, C. tung Chou, S. Jha, and A. Taylor, "The design and evaluation of a hybrid sensor network for cane-toad monitoring," *Information Processing in Sensor Networks*, April 2005.
- [2] R. Szwedczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Communication of the ACM*, vol. 47, no. 6, pp. 34–40, 2004.
- [3] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," *ACM Sigcomm Workshop on Data Communications in Latin America and the Caribbean*, April 2001.
- [4] "http://www.casa.umass.edu/."
- [5] "http://www.cs.cmu.edu/~vsam/index.html."
- [6] "http://thth.berkeley.edu/tab-db/committeeinfo.php?tcid=10."
- [7] "http://lipmon.sprint.com/index.php."
- [8] T. Cui, T. Ho, and L. Chen, "On distributed distortion optimization for correlated sources with network coding," *International Symposium on Information Theory*, 2007.
- [9] R. Cristescu and B. Beferull-Lozano, "Lossy network correlated data gathering with high-resolution coding," *IEEE/ACM Trans. on Networking*, vol. 14, p. 28172824, June 2006.
- [10] C. Xia, D. Towsley, and C. Zhang, "Distributed resource management and admission control of stream processing systems with max utility," *International Conference on Distributed Computing Systems*, 2006.
- [11] A. Giridhar and P. Kumar, "Computing and communicating functions over sensor networks," *IEEE Journal on Selected Areas of Communication*, vol. 23, p. 755764, April 2005.
- [12] —, "Toward a theory of in-network computation in wireless sensor networks," *IEEE Communications Magazine*, vol. 44, pp. 98–107, April 2006.
- [13] V. Doshi, D. Shah, M. Medard, and S. Jaggi, "Graph coloring and conditional graph entropy," *Asilomar Conference on Signals, Systems, and Computers*, pp. 2137–2141, Oct 2006.
- [14] —, "Distributed functional compression through graph coloring," *IEEE Data Compression Conference*, pp. 93–102, March 2007.
- [15] A. Orliitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Inform. Theory*, vol. 47, p. 903917, March 2001.
- [16] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," *IEEE Symposium on Foundations of Computer Science*, 2003.
- [17] S. Boyd, A. Ghosh, B. Prabhakar, , and D. Shah, "Gossip algorithms: Design, analysis and applications," *IEEE Infocom*, 2005.
- [18] R. G. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Trans. on communications*, vol. 25, no. 1, pp. 73–85, Jan 1977.
- [19] R. Khalili and J. Kurose, "A distributed minimum-distortion routing algorithm with in-network data processing," *In Technical Report 07-67, CS Department, UMASS, Amherst*, 2007.
- [20] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley Sons, New York, 1991.
- [21] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Trans. on Multimedia*, vol. 8, pp. 207–218, April 2006.
- [22] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, pp. 74–90, Nov. 1998.
- [23] M. Dai, D. Loguinov, and H. Radha, "Rate-distortion analysis and quality control in scalable internet streaming," *IEEE Trans. on Multimedia*, Dec 2006.