Quality of Information based Data Selection and Transmission in Wireless Sensor Networks

Lu Su*, Shaohan Hu*, Shen Li*, Feng Liang†, Jing Gao‡, Tarek F. Abdelzaher*, and Jiawei Han*
*Department of Computer Science, University of Illinois at Urbana-Champaign
†Department of Statistics, University of Illinois at Urbana-Champaign
‡Department of Computer Science and Engineering, State University of New York at Buffalo
Email: {lusu2, shu17, shenli3, liangf, zaher, hanj}@illinois.edu, jing@buffalo.edu

***Abstract*—In this paper, we provide a quality of information (QoI) based data selection and transmission service for classification missions in sensor networks. We first identify the two aspects of QoI, data reliability and data redundancy, and then propose metrics to estimate them. In particular, reliability implies the degree to which a sensor node contributes to the classification mission, and can be estimated through exploring the agreement between this node and the majority of others. On the other hand, redundancy represents the information overlap among different sensor nodes, and can be measured via investigating the similarity of their clustering results. Based on the proposed QoI metrics, we formulate an optimization problem that aims at maximizing the reliability of sensory data while eliminating their redundancies under the constraint of network resources. We decompose this problem into a data selection subproblem and a data transmission subproblem, and develop a distributed algorithm to solve them separately. The advantages of our schemes are demonstrated through the simulations on not only synthetic data but also a set of real audio records.**

***Keywords*-Sensor Networks; Quality of Information; Data Selection; Data Reliability; Data Redundancy**

I. Introduction

The explosive increase in the amount of data collected by all kinds of sensing devices has posed great challenges on designing effective data selection and transmission schemes for sensor networks. Sometimes, data abstraction and compression techniques such as dimensionality reduction [1] and compressed sensing [2] can be used to mitigate this problem. However, in many cases it is desired that the raw data be delivered as is. For example, zoologists may want to use a sensor network to automatically collect high-quality video or audio records of wild animals [3], [4]. In this scenario, it is necessary to select the raw data to be transmitted based on their *Quality of Information* (QoI).

In mission-driven sensor networks, the concept of QoI varies in different contexts. We are interested in the missions that target on classifying or predicting the current or future state of the physical world through combining the information from the sensor nodes. One representative instance is the species classification using the information provided by multiple audio or video sensors [5]–[7]. Other examples include target surveillance and recognition, habitat and environmental monitoring, health care or assisted living, etc [8]–[14]. In this context, the definition of QoI is motivated by the following two observations:

Observation 1: Consider a set of microphone sensors deployed to record bird vocalizations. Suppose one sensor suffers from circuit board noise, and another is located far away from the birds and close to a group of frogs. Intuitively, the data collected by these two sensors should not be forwarded since they contain substantial noise or are irrelevant to the mission.

Observation 2: To achieve data diversity, we would probably not like both of two camera sensors to upload their data if they always take similar pictures. Instead, we would rather allow only one of them to transmit, and save the network bandwidth for a microphone sensor which monitors the same objects, despite the audio data is usually not as informative as video data.

The above intuitions lead to the two aspects of QoI: *Reliability* and *Redundancy*. Essentially, reliability implies the degree to which each individual sensor node contributes to the classification mission, while redundancy represents the information overlap among different sensor nodes. In this paper, we set our goal as providing a data selection and transmission service for classification missions of sensor networks that can optimize QoI, namely, maximize the reliability of sensory data while eliminating their redundancies, under the constraint of network resources. Achieving this, however, is challenging in sensor networks, due to the problems listed below.

- In sensor networks, the sensory data are distributed over a large number of sensor nodes. Furthermore, the network resources cannot afford the delivery of all the raw data, otherwise there is no need to conduct data selection. This eliminates the applicability of centralized solutions working directly on the raw data.
- The reliability and redundancy of a sensor node reflect its relation to the other nodes, and thus cannot be estimated in isolation. Therefore, without obtaining the information from all the sensor nodes, it is hard to precisely estimate the QoI of individual nodes.

- As pointed out in [7], in many applications of sensor networks, the amount of labeled training data is usually small, which can be attributed to the remote, harsh, and sometimes even hostile locales where sensor networks are normally deployed. Without sufficiently large training set, classification algorithms may not be able to describe the characteristics of each class, and thus can potentially make inaccurate predictions on new data.
- The QoI of sensor nodes may be dynamically changing. The dynamics could be resulted from many factors, such as the energy supply of the sensor nodes, the continuous variation of the surveilled environment, and the mobility of the events or even the sensors.
- The data transmission in wireless environment is rather complicated, due to the broadcast nature of wireless communication. How to efficiently utilize wireless spectrum in order to maximize the QoI delivered to the sink remains a problem of great challenge.

The main purpose of this work is to address the above challenges. First of all, we develop a novel online algorithm to estimate the QoI of individual sensor nodes through exploring their clustering results reported to the sink of each mission. Specifically, the reliability of a source node is determined by the level to which its classification or prediction result agrees with those of the majority of other nodes, while the data redundancy between two sensor nodes is measured through investigating the similarity of their clustering results. The algorithm maintains a sliding time window, and the QoI estimates are automatically updated once events within the window are refreshed. Moreover, based on the QoI continuously output by the algorithm, we formulate an optimization framework which aims at maximally utilizing the network resources in order to achieve the optimal aggregate quality of delivered information for a sensor network running multiple concurrent missions. A distributed joint design of data selection and transmission is then proposed to solve this optimization problem.

The rest of the paper is organized as follows. Section II provides an overview of the system model and problem formulation. In Section III, we propose the metrics and methods used to estimate QoI. Section IV and Section V presents our data selection and transmission scheme and its distributed implementation. The proposed scheme is evaluated in Section VI. Section VII concludes the paper.

II. System Overview

Consider a sensor network of N nodes which perform M classification missions concurrently. In each mission, there are multiple source nodes that sense the physical surroundings and a single sink node whose task is to store and process the sensory readings. In addition, some relay nodes are deployed to enable the data forwarding. The source and sink nodes can also help relay the traffic. A sensor node may be associated with multiple missions simultaneously.

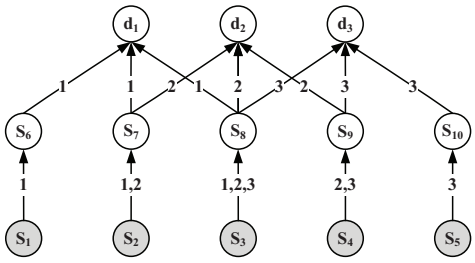

Figure 1. An illustrative sensor network in which 3 concurrent missions are performed.

Figure 1 shows an illustrative sensor network in which 3 concurrent missions are performed. In this scenario, 5 source nodes, i.e., the shaded nodes $s_1, \ldots, s_5$, collect and forward data to 3 different sink nodes, d_1, d_2, d_3, each of which corresponds to a particular mission. Among the source nodes, s_2, s_3, and s_4 serve for multiple missions. In particular, s_2 serves for mission 1 and 2, s_3 serves for mission 1, 2, and 3, and s_4 serves for mission 2 and 3. Nodes $s_6, \ldots, s_{10}$ work as the relay nodes, forwarding data from the source nodes to the sink nodes. The number(s) on each edge indicates the index of the sink node to which the traffic flow on this edge is heading. As can be seen, there may exist flows towards different destinations going through the same edge. For example, the edge connecting node s_3 and s_8 forwards the data to 3 sink nodes since s_3 participates in all the missions.

Based on the above example, we give an overview of the formulation of and solution to the QoI based data selection and transmission problem, which is mathematically formalized as an optimization program $\mathbf{P}$ in Section IV-A. The objective of $\mathbf{P}$ is to select a subset of sensor nodes for each mission so that the aggregate reliability of their data can be maximized, under the network resource constraint and the data redundancy constraint. Suppose in this example, the network resource constraint requires that within a time slot, each link can forward data for only one mission. Thus, the source nodes s_2, s_3, and s_4 need to figure out for which mission they should serve. To achieve this, as suggested in Section III-A, each source should send its clustering result of the detected events to the sink node where the reliability and redundancy of its data can be estimated. Particularly, the metric of data reliability is developed in Section III-C based on the decision aggregation procedure introduced in Section III-B. On the other hand, Section III-D gives the measure of data redundancy. The QoI estimates are then sent back to the source nodes so that they can locally decide who and for which mission would have a chance to collect data based on a distributed algorithm developed in Section IV-B. This algorithm solves $\mathbf{P}$ through decomposing it into a data selection subproblem and a data transmission subproblem that can be tackled separately. In Section V, we provide a detailed description on how the proposed data selection and transmission scheme is implemented in a distributed manner.

III. Quality of Information

This section starts with a brief introduction on how to preprocess sensory data. Then we define the metrics of the data reliability and redundancy, and elaborate on how they are estimated in each mission of the sensor network.

A. Data Preprocessing

Consider a mission which involves n source nodes, denoted by s_i $(i = 1, 2, ..., n)$. When an event takes place, all the source nodes collect sensory readings about it. Suppose the goal of this mission is to classify the detected events into m different classes. Let $\mathcal{E} = \{e_i |\ i = 1, 2, ..., t\}$ denote the sequence of events (sorted in chronological order) detected by the source nodes. Suppose only a small portion of the events are labeled, and what we need to do is to find out the labels of the rest events. Due to the scarcity of network resources, only a subset of the source nodes can deliver their data to the sink. As previously discussed, without a global view of the information from all the sensor nodes, it is hard to precisely estimate the QoI of each node. A plausible substitution of the raw data is the class labels of the observed events predicted by the sensor nodes. The intuition is as follows. First, if the classification result of a sensor node agrees with those of the majority of others, its data are more likely to be reliable, given the assumption that the majority of the sensor nodes have acceptable classification accuracy. Second, if two sensor nodes always make the same prediction, their information may be redundant.

The challenge of this solution, as aforementioned, is the lack of label information. Without sufficient label information, the classification results of individual sensors cannot be accurate. To tackle this problem, we suggest that each source node locally conduct cluster analysis, which groups data points only based on the similarity of their feature values without any training. The clustering results can provide useful constraints for the task of classification when the labeled data is insufficient, since the data that have similar feature values are usually more likely to share the same class label. Towards this end, we let each of the n source nodes deliver to the sink its clustering result, in which the events in $\mathcal{E}$ are partitioned into m clusters. Thus, there are totally $l = mn$ different clusters generated by the source nodes, denoted by $c_j,\ j = 1, 2, ..., l$. With the clustering results as well as the label information, the sink is now able to estimate the QoI of each source node. The estimation is based on the *Decision Aggregation* procedure proposed in our previous work [7]. We will first provide a brief introduction of this procedure in the next subsection, and then explain in detail how the QoI of each source node is derived accordingly in the rest of this section.

B. Decision Aggregation

The decision aggregation procedure takes as input the clustering results of multiple sensors as well as the label information, and outputs a class label for each event. It first models the relationship between the events and the input clusters as a bipartite graph, called *belief graph*. In belief graph, each input cluster links to the events it contains. Moreover, to integrate label information into the belief graph, one more set of vertices are added to represent the labels of the events. The labeled events are then connected to the corresponding label vertices. Figure 2 provides an example of belief graph involving $n = 3$ sensor nodes and $t = 10$ events. In this case, suppose the mission is to classify the events into $m = 2$ different classes, then there are totally $l = mn = 6$ different clusters.

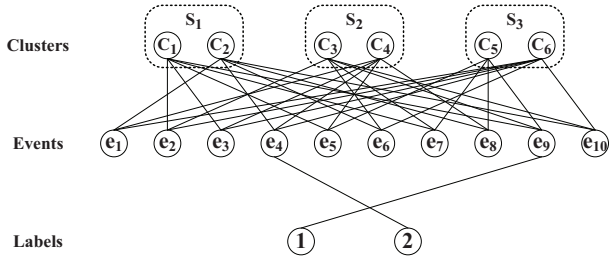

Figure 2. An example of belief graph

The belief graph can be represented by two adjacency matrices: (i) Clustering matrix $A = (a_{ij})_{t\times l}$, where a_{ij} indicates whether event e_i is assigned to cluster c_j. (ii) Groundtruth matrix $Z = (z_{ik})_{t\times m}$, where z_{ik} denotes whether e_i's observed label is k. Then, two sets of probability vectors are defined. First, each event e_i is associated with a m-dimensional probability vector, denoted by $\vec{x}_{i\cdot} = (x_{ik})$. Each element of $\vec{x}_{i\cdot}$, say x_{ik}, indicates the probability of e_i belonging to the k-th class. Second, for each input cluster c_j, a m-dimensional probability vector, denoted by $\vec{y}_{j\cdot} = (y_{jk})$, is also defined. Each element of this vector is the probability that the majority of the events contained in c_j are assigned to a particular class. $\vec{x}_{i\cdot}$ and $\vec{y}_{j\cdot}$ work as the variables in the optimization program solving the decision aggregation problem:

$$\begin{aligned}
\mathbf{DA}:\ \min\quad & \sum_{i=1}^{t}\sum_{j=1}^{l} a_{ij}||\vec{x}_{i\cdot} - \vec{y}_{j\cdot}||^2 + \alpha\sum_{i=1}^{t} b_i||\vec{x}_{i\cdot} - \vec{z}_{i\cdot}||^2 \\
\text{s.t.}\quad & \vec{x}_{i\cdot} \geq \vec{0}, \quad |\vec{x}_{i\cdot}| = 1 \quad \text{for}\ \ i = 1, 2, ..., t \\
& \vec{y}_{j\cdot} \geq \vec{0}, \quad |\vec{y}_{j\cdot}| = 1 \quad \text{for}\ \ j = 1, 2, ..., l
\end{aligned}$$

where $||.||$ and $|.|$ denote a vector's L2 and L1 norm respectively. Besides, $b_i = \sum_{k=1}^{m} z_{ik}$ is a flag variable indicating whether e_i is labeled or not, and α is a predefined parameter. $\mathbf{DA}$ is actually a convex program, which makes it possible to find a global optimal solution. To achieve consensus among the clustering results of multiple sensor nodes, $\mathbf{DA}$ aims at finding the optimal probability vectors of the event nodes ($\vec{x}_{i\cdot}$) and the cluster nodes ($\vec{y}_{j\cdot}$) that can minimize the disagreement over the belief graph, and in the meanwhile, comply with the label information. Moreover, since $\vec{x}_{i\cdot}$ and

$\vec{y}_{j\cdot}$ are probability vectors, each of their components must be greater than or equal to 0 and the sum should equal 1.

C. Data Reliability

As previously discussed, the reliability of a source node can be measured by the level to which its classification or prediction result agrees with those of the majority of other nodes. This can be inferred from the solution of **DA**. In its objective function, the first term ensures that an input cluster has similar probability vector as the events it contains, namely, $\vec{x}_{i\cdot}$ should be close to $\vec{y}_{j\cdot}$ if event e_i is connected to cluster c_j in the belief graph. Let's put it in a more straightforward way. If we fix the values of $\vec{x}_{i\cdot}$ as constants, then the objective function becomes a convex function with respect to $\vec{y}_{j\cdot}$. Its minimum can be obtained by setting the partial derivatives $\frac{\partial f(X,Y)}{\partial y_{jk}}$, $(k = 1, 2, \ldots, m)$ to 0:

$$\vec{y}_{j\cdot} = \frac{\sum_{i=1}^{t} a_{ij}\vec{x}_{i\cdot}}{\sum_{i=1}^{t} a_{ij}}. \tag{1}$$

As one can see, $\vec{y}_{j\cdot}$ is actually the average of the probability vectors of the events that belong to cluster c_j. On the other hand, the second term of **DA**'s objective function puts the constraint that a labeled event's probability vector $\vec{x}_{i\cdot}$ should not deviate much from the corresponding groundtruth vector $\vec{z}_{i\cdot}$, and α can be considered as the shadow price payment for violating this constraint. If the values of $\vec{y}_{j\cdot}$ are fixed, the optimal $\vec{x}_{i\cdot}$ can be derived through the following formula:

$$\vec{x}_{i\cdot} = \frac{\sum_{j=1}^{l} a_{ij}\vec{y}_{j\cdot} + \alpha b_i \vec{z}_{i\cdot}}{\sum_{j=1}^{l} a_{ij} + \alpha b_i}. \tag{2}$$

For an unlabeled event e_i, since its flag variable $b_i = 0$, $\vec{x}_{i\cdot}$ is calculated through averaging the probability vectors of the clusters containing e_i. If e_i is labeled, $\vec{x}_{i\cdot}$ becomes the weighted average of clustering information and label information tuned by the shadow price α. According to Eqn. (2), given that the majority of the sensor nodes are collecting data with acceptable quality, the probability vectors of most of the events should be close to the groundtruth, since the errors of individual sensors can be canceled out by the averaging operation. In other words, x_{ik} should be the largest element of $\vec{x}_{i\cdot}$ if the groundtruth label of e_i is k. Consequently, by Eqn. (1), the elements of $\vec{y}_{j\cdot}$ will be skewed if the majority of the events contained in cluster c_j belong to the same class in groundtruth. In contrast, if c_j's events have diversified groundtruth labels, $\vec{y}_{j\cdot}$'s elements should be evenly distributed.

An illustrative example may shed more light on this point. Suppose the optimal probability vectors of the events in Fig. 2 are listed in Table I. For the sake of simplicity, in this case we set the elements of $\vec{x}_{i\cdot}$ as binaries. In reality, they cannot be simply 0 or 1, but rather decimal numbers between 0 and 1. For cluster c_1, its probability vector can be derived following Eqn. (1):

$$\vec{y}_{1\cdot} = (\vec{x}_{2\cdot} + \vec{x}_{3\cdot} + \vec{x}_{5\cdot} + \vec{x}_{7\cdot} + \vec{x}_{9\cdot})/5 = (0.8, 0.2).$$

$\vec{y}_{1\cdot}$ is quite skewed since 4 of the 5 events in c_1 have a probability vector of (1,0). Similarly, the probability vectors of other clusters are calculated and shown in Table II. As one can see, some clusters (e.g., c_5 and c_6) have uniform probabilities since they have equal number of events with vector (1,0) and (0,1).

Table I
PROBABILITY VECTORS OF EVENTS

Event	e_1	e_2	e_3	e_4	e_5
Vector	$\vec{x}_{1\cdot}$	$\vec{x}_{2\cdot}$	$\vec{x}_{3\cdot}$	$\vec{x}_{4\cdot}$	$\vec{x}_{5\cdot}$
Value	(1,0)	(1,0)	(1,0)	(0,1)	(1,0)
Event	e_6	e_7	e_8	e_9	e_{10}
Vector	$\vec{x}_{6\cdot}$	$\vec{x}_{7\cdot}$	$\vec{x}_{8\cdot}$	$\vec{x}_{9\cdot}$	$\vec{x}_{10\cdot}$
Value	(0,1)	(0,1)	(0,1)	(1,0)	(0,1)

Table II
CLUSTER AND SENSOR ENTROPIES

Cluster	c_1	c_2	c_3	c_4	c_5	c_6
Vector	$\vec{y}_{1\cdot}$	$\vec{y}_{2\cdot}$	$\vec{y}_{3\cdot}$	$\vec{y}_{4\cdot}$	$\vec{y}_{5\cdot}$	$\vec{y}_{6\cdot}$
Value	(0.8,0.2)	(0.2,0.8)	(0.6,0.4)	(0.4,0.6)	(0.5,0.5)	(0.5,0.5)
Entropy	0.7219	0.7219	0.9710	0.9710	1	1

Sensor	s_1	s_2	s_3
Entropy	0.7219	0.9710	1
Reliability	0.2781	0.0290	0

Guided by the above observation, we propose to estimate the reliability of a source node based on the degree of impurity of the clusters it generates. The smaller the degree of impurity, the more skewed the probability vector. In the above example, c_1 has a pretty low degree of impurity, while c_5 and c_6 are the most impure clusters. In this paper, we use entropy [15], a quantitative representation of random variable uncertainties in information theory, to measure the impurity of clusters. For a cluster c_j, its entropy can be calculated as $-\sum_{k=1}^{m} y_{jk} \log_2 y_{jk}$. The entropy of a source node s_i is defined as the weighted average of its clusters' entropies:

$$-\sum_{c_j \in s_i} \omega_j \sum_{k=1}^{m} y_{jk} \log_2 y_{jk},$$

where the weight ω_j amounts to the ratio of the number of events in c_j over the total event number. The entropies of the six clusters as well as the three sensor nodes are shown in Table II. As one can see, the entropy values can precisely reflect the degree of impurity. However, entropy cannot be directly used as the estimate of reliability. This is simply because larger entropy means higher impurity, which reversely implies less reliability. According to the principle of maximum entropy [15], the largest entropy of a sensor node is reached when the probability vectors of its clusters have equal elements, which are $\frac{1}{m}$. Thus, the largest entropy is $-\log_2(\frac{1}{m}) = \log_2(m)$. Subtracting the entropy of each

node from this number gives the reliability estimate of this node:

$$\log_2(m) + \sum_{c_j \in s_i} \omega_j \sum_{k=1}^{m} y_{jk} \log_2 y_{jk}. \tag{3}$$

The reliability estimates of the three nodes are shown in Table II. As the numbers suggest, now the sensors with impure clusters have lower reliability scores.

In contrast to decision aggregation which works offline on all the collected data, sensor selection calls for a online mechanism which can adaptively determine the set of sensor nodes to transmit based on their recent data's quality of information. The first step to this end is a novel algorithm called Incremental Reliability Estimation (IRE). The IRE algorithm applies a sliding time window of width w to the set of sequentially occurred events $\mathcal{E} = \{e_i |\ i = 1, 2, ..., t, ...\}$, and incrementally outputs the reliability estimate of each sensor node according to the sensory data of the events within the window. Upon the occurrence of v ($1 \leq v \leq w$) new events, the window slides, with the v new events (denoted by $e_{t+i},\ i = 1, ..., v$) added and v outdated events (denoted by $e_{t-w+i},\ i = 1, ..., v$) removed. On the other hand, each of the source nodes updates its clustering result through either re-clustering or incremental clustering [16]. The clustering results of the newly arrived data are then reported to the sink node of each mission where the IRE algorithm is invoked.

The basic idea of the IRE algorithm is as follows. It first calculates the probability vectors of the newly-occurred events through Eqn. (2), and then uses them to update the probability vectors of clusters according to an incremental version of Eqn. (1) as below:

$$\vec{y}_{j\cdot} = \frac{\sum_{i=t-w+1}^{t} a_{ij}\vec{x}_{i\cdot} - \sum_{i=t-w+1}^{t-w+v} a_{ij}\vec{x}_{i\cdot} + \sum_{i=t+1}^{t+v} a_{ij}\vec{x}_{i\cdot}}{\sum_{i=t-w+1}^{t} a_{ij} - \sum_{i=t-w+1}^{t-w+v} a_{ij} + \sum_{i=t+1}^{t+v} a_{ij}}. \tag{4}$$

As can be seen, both the numerator and denominator contain the information from three parts: (i) the events in the original window $(e_{t-w+1}, \ldots, e_t)$ (ii) outdated events $(e_{t-w+1}, \ldots, e_{t-w+v})$ (iii) new events $(e_{t+1}, \ldots, e_{t+v})$. Therefore, the update can be done incrementally through substituting outdated information with new information, without the need of a complete recalculation. Finally, the updated $\vec{y}_{j\cdot}$ are used as the input of Eqn. (3) to derive the reliability score of each sensor node.

In practice, the above computations are conducted via matrix operations, and thus we introduce some matrix notations. Putting the probability vectors of the events in the original window together, we get a probability matrix $X^{(t)}_{w\times m} = (\vec{x}_{(t-w+1)\cdot}, \ldots, \vec{x}_{t\cdot})^T$. Similarly, as shown in Eqn. (5), we define the probability matrices of the outdated events ($X^{\text{old}}_{v\times m}$), the newly-occurred events ($X^{\text{new}}_{v\times m}$), as well as the events in the updated window ($X^{(t+v)}_{w\times m}$).

Algorithm 1 Incremental Reliability Estimation

Input: The clustering results of the sensor nodes for the v new events, associated with their labels;
Output: The estimate of each node's reliability;

1: Generate A^{new}, B^{new}, C^{new}, D^{new}, and Z^{new}.
2: $D^{(t+v)} \leftarrow D^{(t)} - D^{\text{old}} + D^{\text{new}}$
3: $F^{(t)} \leftarrow {A^{(t)}}^T X^{(t)} - {A^{\text{old}}}^T X^{\text{old}}$
4: $Y^{\text{temp}} \leftarrow Y^{(t)}$
5: Initialize $Y^{(t+v)}$ randomly.
6: **while** $||Y^{(t+v)} - Y^{\text{temp}}|| > \epsilon$ **do**
7: $\quad Y^{\text{temp}} \leftarrow Y^{(t+v)}$
8: $\quad X^{\text{new}} \leftarrow (C^{\text{new}} + \alpha B^{\text{new}})^{-1}(A^{\text{new}} Y^{(t)} + \alpha B^{\text{new}} Z^{\text{new}})$
9: $\quad Y^{(t+v)} \leftarrow {D^{(t+v)}}^{-1}(F^{(t)} + {A^{\text{new}}}^T X^{\text{new}})$
10: $\widetilde{X}^{\text{new}} \leftarrow \text{label}(X^{\text{new}})$
11: $\widetilde{Y}^{(t+v)} \leftarrow {D^{(t+v)}}^{-1}({A^{(t)}}^T \widetilde{X}^{(t)} - {A^{\text{old}}}^T \widetilde{X}^{\text{old}} + {A^{\text{new}}}^T \widetilde{X}^{\text{new}})$
12: $\vec{e}^{(t+v)}_{\text{cluster}} = -\text{sum}(\widetilde{Y}^{(t+v)} .* \log_2(\widetilde{Y}^{(t+v)}), 2)$
13: $\vec{\omega}^{(t+v)}_{\text{cluster}} = \text{sum}({A^{(t+v)}}^T, 2)/w$
14: $\vec{p}^{(t+v)} = \log_2(m) - \text{sum}(\text{vec2mat}(\vec{\omega}^{(t+v)}_{\text{cluster}} .* \vec{e}^{(t+v)}_{\text{cluster}}, m), 2)$
15: **return** $\vec{p}^{(t+v)}$

$$\overbrace{\underbrace{\vec{x}_{(t-w+1)\cdot}, \ldots, \vec{x}_{(t-w+v)\cdot}}_{X^{\text{old}}} | \underbrace{\vec{x}_{(t-w+v+1)\cdot}, \ldots, \vec{x}_{t\cdot}}}^{X^{(t)}} | \overbrace{\vec{x}_{t+1\cdot}, \ldots, \vec{x}_{t+v\cdot}}^{X^{\text{new}}} \tag{5}$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{X^{(t+v)}}$$

Then, we generate a clustering matrix $A = (a_{ij})$ (recall that a_{ij} indicates whether event e_i is assigned to cluster c_j.) corresponding to each of the above probability matrices. They are denoted by $A^{(t)}$ ($i = t-w+1, \ldots, t$), A^{old} ($i = t-w+1, \ldots, t-w+v$), A^{new} ($i = t+1, \ldots, t+v$), $A^{(t+v)}$ ($i = t-w+v+1, \ldots, t+v$), respectively. In the following notation definitions, we will use the same correspondence between the superscripts (i.e., "new", "old", "(t)", "$(t+v)$") and the range of event index i.

The detailed steps of IRE are shown in Algorithm 1. Line 8 displays the derivation of X^{new} through Eqn. (2), where $B = \text{diag}\big\{(b_i)\big\}$ and $C = \text{diag}\big\{(\sum_{j=1}^{l} a_{ij})\big\}$. Note that the superscript "new" implies $i = t+1, \ldots, t+v$. Additionally, here $Y^{(t)}_{l\times m} = (\vec{y}^{(t)}_{1\cdot}, \ldots, \vec{y}^{(t)}_{l\cdot})^T$ is the probability matrix of all the clusters at time t. The matrix form of Eqn. (4) is given at line 9. In this equation, $D^{(t+v)}$ and $F^{(t)}$ are constant matrices. In particular, $D = \text{diag}\big\{(\sum_i a_{ij})\big\}$ corresponds to the items in the denominator of Eqn. (4), and $D^{(t+v)}$ is calculated at line 2. $F^{(t)}$ represents the information of remaining events in the time window, and is generated at line 3. In the while loop from line 6 to line 9. X^{new} and $Y^{(t+v)}$ are repeatedly updated by each other, until no notable change occurs at $Y^{(t+v)}$. The convergence is guaranteed by the theory of coordinate descent [17], due to the convexity of $\mathbf{DA}$.

At line 10, the derived X^{new} is put into a function of label$(\cdot)$, which converts each probability vector of X^{new} into a binary vector. Precisely, it sets the largest element of a probability vector to be 1, and other elements to be 0. The modified matrix $\widetilde{X}^{\text{new}}$ is further used to calculate the final probability vector of clusters $\widetilde{Y}^{(t+v)}$ at line 11. With $\widetilde{Y}^{(t+v)}$,

the algorithm is able to derive the estimate of reliability. It first calculates the entropy as well as weight of each cluster at line 12 and 13, respectively. Here $.*$ denotes the operation of element-wise multiplication between two matrices, while the function sum(A,d) sums along the dimension of A specified by scalar d. Finally, the reliability estimates of the sensor nodes are determined according to Eqn. (3) via a matlab function vec2mat($\vec{v}$,m) that converts the vector $\vec{v}$ into a matrix with m columns.

D. Data Redundancy

The data redundancy between two sensor nodes can be measured through investigating the similarity of their clustering results. The comparison of clustering results can be achieved using *Similarity Matrix* [18]. The similarity matrix of a sensor node is defined as $M = (m_{ij})$, where m_{ij} equals 1 if event e_i and e_j are put into the same cluster by this node, and 0 otherwise.

Table III
SIMILARITY MATRIX OF s_1

Event	e_1	e_2	e_3	e_4	e_5
e_1	1	0	0	1	0
e_2	0	1	1	0	1
e_3	0	1	1	0	1
e_4	1	0	0	1	0
e_5	0	1	1	0	1

Table IV
SIMILARITY MATRIX OF s_2

Event	e_1	e_2	e_3	e_4	e_5
e_1	1	0	1	1	1
e_2	0	1	0	0	0
e_3	1	0	1	1	1
e_4	1	0	1	1	1
e_5	1	0	1	1	1

The similarity matrices of sensor s_1 and s_2 in the previous example for the first 5 events are shown in Table III and Table IV, respectively. To quantitatively measure the difference between two similarity matrices M_1 and M_2, four numbers are defined as follows:

- f_{00}: number of event pairs belonging to different clusters in both M_1 and M_2.
- f_{01}: number of event pairs belonging to different clusters in M_1 and the same cluster in M_2.
- f_{10}: number of event pairs belonging to the same cluster in M_1 and different clusters in M_2.
- f_{11}: number of event pairs belonging to the same cluster in both M_1 and M_2.

Two measures based on the above quantities are widely used: Rand statistic and Jaccard coefficient [18]. Their definitions are shown below:

$$\text{Rand} = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} \qquad \text{Jaccard} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} \tag{6}$$

For the above two matrices, the quantities are $f_{00} = 2$, $f_{01} = 4$, $f_{10} = 2$, and $f_{11} = 2$. Thus, the Rand statistic is $(2 + 2)/10 = 0.4$, while the Jaccard coefficient is $2/(4 + 2 + 2) = 0.25$. In reality, there is no need to exhaustively check each pair of the sensor nodes to find redundancy. By adding some simple rules, the searching space can be dramatically reduced. For example, we may only need to compare the clustering results of the nodes in close proximity and equipped with the same types of sensors. Moreover, the sliding window limits the size of the similarity matrices. To further mitigate the storage and computation overhead, we can randomly sample the elements of a matrix instead of maintaining its entirety. Additionally, at each time when the window slides, for each matrix only the entries involving the newly-occurred events are updated, with others remaining unchanged.

IV. DATA SELECTION AND TRANSMISSION

With the previously defined Quality of Information, we are now ready to formally formulate the QoI based data selection and transmission problem, which is denoted by $\mathbf{P}$. In this section, we will first introduce the objective function as well as the constraints of $\mathbf{P}$, and then propose a distributed algorithm to solve it. The notations defined in this section may have been used in preceding parts of the paper. However, they can be easily distinguished from the context, and thus we believe no confusion would occur.

A. Problem Formulation

Objective Function

The objective function of $\mathbf{P}$ is the aggregate data reliability of all the missions. In this formulation, we assume that the information collected in different missions are independent of one another. To simplify the presentation, we assume the reliability measures for different missions are normalized and of the same scale. One can easily prioritize the missions by associating the reliability of each mission with a weight parameter. Within each mission, after removing the information overlap among different sensor nodes via the data redundancy constraint, the data reliability of a sensor set can be approximated as the summation of individual sensors' reliability. Guided by this intuition, we specify the objective function as $\sum_{k=1}^{M}\sum_{i=1}^{N} p_i^k x_i^k$. In this function, p_i^k is the reliability estimate of the sensor node s_i with regard to the k-th mission. $p_i^k = 0$ if s_i is not a source node of mission k. $x_i^k \in \{0, 1\}$ is a variable indicating whether sensor s_i is selected to collect data for mission k.

Network Resource Constraint

The network resources could be bandwidth, energy, storage, and many others. For the sake of simplicity, in this paper we focus on bandwidth, which may be the most difficult one to handle. The proposed framework, however, can be easily extended to other resources. In wireless networks, the neighboring links may contend for bandwidth due to the broadcast nature of wireless transmission. The contention relations among the links can be captured by a conflict graph [19], based on the network topology. In the conflict graph, each vertex represents a link, and an edge between two vertices implies the contention between the two corresponding links, i.e., they cannot transmit at the same time. Given a conflict graph, we can identify all its independent sets of vertices that have no edges between each other. The links in an independent set can transmit simultaneously.

Suppose $\mathcal{L}$ is the set of all the links in the sensor network, and let $\mathcal{I}$ denote the collection of independent sets. We represent an independent set, I_q $(q = 1, 2, ..., |\mathcal{I}|)$, as a $|\mathcal{L}|$-dimensional bandwidth vector, which is c^q. In c^q, an element $c^q_{i,j} = b_{i,j}$ if $(i,j) \in I_q$ and 0 otherwise, where $b_{i,j}$ denotes the bandwidth capacity of link $(i,j) \in \mathcal{L}$. The feasible bandwidth region Π at the link layer is defined as the convex hull of these vectors:

$$\Pi := \{c \mid c = \sum_{q=1}^{|\mathcal{I}|} \alpha_q c^q,\ \alpha_q \geq 0,\ \sum_{q=1}^{|\mathcal{I}|} \alpha_q = 1\}.$$

Let $c^k_{i,j}$ denote the amount of bandwidth of link (i,j) allocated to the flow of mission k. Then $c_{i,j} = \sum_{k=1}^{M} c^k_{i,j}$ is the aggregate bandwidth of link (i,j). According to the above definition, the bandwidth vector of all the links $c = (c_{i,j})$ should satisfy $c \in \Pi$. Suppose r^k_i is the data collection rate of source node s_i for mission k. r^k_i implies s_i's demand for bandwidth, and $r^k_i = 0$ if s_i is not a source of mission k. Once s_i is allowed to collect data, to prevent its queue from overflow, for each mission the summation of the bandwidth allocated to s_i's incoming flows and its demanded bandwidth should not exceed the aggregate bandwidth for its outgoing flows. This motivates the network resource constraint for bandwidth resource:

$$r^k_i x^k_i \leq \sum_{j:(i,j)\in\mathcal{L}} c^k_{i,j} - \sum_{j:(j,i)\in\mathcal{L}} c^k_{j,i}.$$

Data Redundancy Constraint

In Section III-D, we discuss the redundancy measures of sensor pairs. By setting up a threshold for the redundancy measure, we can identify the redundant node pairs and thus effectively control the redundancy level in the network. To capture the redundancy relations among the sensor nodes, we first take a graph expansion on the network topology. In particular, for each source node s_i, if it serves for mission k, we create a virtual node denoted by $s_{i,k}$. $s_{i,k}$ is connected to s_i through a virtual link with a capacity of r^k_i and works as a virtual source of mission k. In $\mathbf{P}$, each virtual source $s_{i,k}$ is associated with a previously defined indicator variable $x_{i,k}$.

Figure 3 shows the expanded topology of the sensor network in Fig. 1. In this case, we simply assume that in each mission all the source nodes are redundant with each other. For example, the source node s_1, s_2, and s_3 are redundant in mission 1, and thus they should not collect data simultaneously for mission 1. Equivalently, in the expanded topology this implies that the virtual link a, b, and d cannot transmit at the same time. This kind of confliction among the virtual links can be modeled by constructing a bipartite graph called data redundancy graph. As shown in Fig. 4, in the data redundancy graph a set of auxiliary nodes (black nodes) are created and connected to the virtual source nodes that are redundant with one another. With this graph

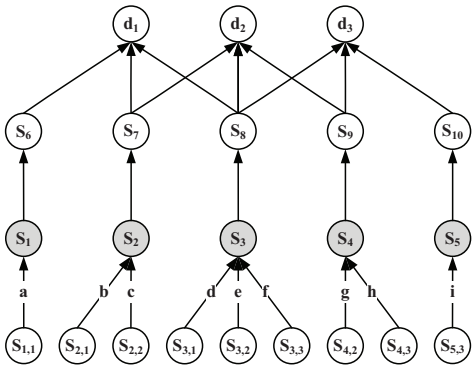

Figure 3. Expanded topology

transformation, the problem of finding a subset of source nodes which can simultaneously collect data becomes the problem of identifying a matching, i.e., a set of links without common nodes, of the data redundancy graph.

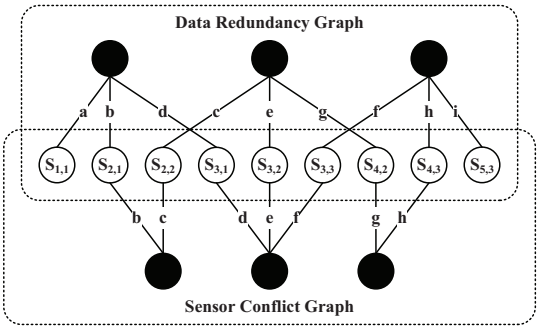

Figure 4. Data redundancy graph and sensor conflict graph

The virtual nodes connecting to the same source may also have conflicts, since the sensing devices on a node may not be able to serve multiple missions at the same time. To tackle this problem, we build another bipartite graph called sensor conflict graph. Similar to the data redundancy graph, the conflicting virtual nodes are connected to the same auxiliary nodes as drawn in Fig. 4. Again, matching algorithms can be used to find the conflict-free node set. Similar to what we did when formulating the network resource constraint, the feasible bandwidth region of the virtual links can be derived through combining the bandwidth vectors of all the matchings in the data redundancy graph and sensor conflict graph.

Putting the above objective function and constraints together gives us the complete optimization program:

$$\mathbf{P}:\ \max \quad \sum_{k=1}^{M}\sum_{i=1}^{N} p^k_i x^k_i \tag{7}$$

$$\text{subject to} \quad r^k_i x^k_i \leq \sum_{j:(i,j)\in\mathcal{L}} c^k_{i,j} - \sum_{j:(j,i)\in\mathcal{L}} c^k_{j,i} \tag{8}$$

$$c \in \Pi$$

The data redundancy constraint is not explicitly present

in $\mathbf{P}$, since it has already been captured in Π which covers the feasible bandwidth region of not only the real links but also the virtual links.

B. Distributed Algorithm via Dual Decomposition

$\mathbf{P}$ is actually a mixed integer program since the feasible values of $x = (x_i^k)$ are restricted to be 0 or 1, making it difficult to find the optimal solution. Furthermore, solving $\mathbf{P}$ directly requires global coordination of all the nodes, which is impractical in a distributed environment such as sensor networks. To address these challenges, we first relax $\mathbf{P}$ into a convex problem, and propose a distributed solution through dual decomposition.

Convex Relaxation and Dual Decomposition

By allowing x to take any value between 0 and 1, $\mathbf{P}$ can be relaxed into a convex program, denoted as $\widetilde{\mathbf{P}}$. Due to the convexity of $\widetilde{\mathbf{P}}$, strong duality can be achieved (Chapter 5.2.3 in [20]). Therefore, there exists a unique maximizer (x^*, c^*) for $\widetilde{\mathbf{P}}$, which can be attained by a distributed algorithm derived via formulating and solving the Lagrange dual problem of $\widetilde{\mathbf{P}}$. In order to achieve this, we first take a look at the Lagrangian of $\widetilde{\mathbf{P}}$:

$$L(x,c,\mu) = \sum_{k=1}^{M}\sum_{i=1}^{N} p_i^k x_i^k - \sum_{k=1}^{M}\sum_{i=1}^{N} \mu_i^k (r_i^k x_i^k - \sum_{j:(i,j)\in\mathcal{L}} c_{i,j}^k + \sum_{j:(j,i)\in\mathcal{L}} c_{j,i}^k).$$

In $L(x,c,\mu)$, $\mu = (\mu_i^k)$ is the vector of Lagrangian multipliers, corresponding to the network resource constraint (Eqn. (8)). μ_i^k is also interpreted as the "shadow price" of the constraint, which can be understood as the "cost" a node will be charged if it violates the constraint. Furthermore, since

$$\sum_{k=1}^{M}\sum_{i=1}^{N} \mu_i^k (\sum_{j:(i,j)\in\mathcal{L}} c_{i,j}^k - \sum_{j:(j,i)\in\mathcal{L}} c_{j,i}^k) = \sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{j:(i,j)\in\mathcal{L}} c_{i,j}^k(\mu_i^k - \mu_j^k),$$

we reorganize the Lagrangian as follows:

$$\begin{aligned} L(x,c,\mu) &= \sum_{k=1}^{M}\sum_{i=1}^{N} p_i^k x_i^k - \sum_{k=1}^{M}\sum_{i=1}^{N} \mu_i^k r_i^k x_i^k \\ &\quad + \sum_{k=1}^{M}\sum_{i=1}^{N} \mu_i^k (\sum_{j:(i,j)\in\mathcal{L}} c_{i,j}^k - \sum_{j:(j,i)\in\mathcal{L}} c_{j,i}^k) \\ &= \sum_{k=1}^{M}\sum_{i=1}^{N} (p_i^k - \mu_i^k r_i^k) x_i^k + \sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{j:(i,j)\in\mathcal{L}} c_{i,j}^k(\mu_i^k - \mu_j^k). \end{aligned}$$

The dual of the primal problem $\widetilde{\mathbf{P}}$ is:

$$\mathbf{D}: \min_{\mu\geq 0} \quad D(\mu),$$

where the dual objective function $D(\mu)$ is given as

$$D(\mu) := \max_{x\in X,\ c\in\Pi} \quad L(x,c,\mu).$$

In the dual objective function, the Lagrangian multiplier (shadow price) μ serves as the dual variable. Furthermore, $D(\mu)$ can be decomposed into two separate optimization problems: $D(\mu) = D_1(\mu) + D_2(\mu)$. $D_1(\mu)$ and $D_2(\mu)$ are defined below:

$$D_1(\mu) := \max_{x\in X} \quad \sum_{k=1}^{M}\sum_{i=1}^{N} (p_i^k - \mu_i^k r_i^k) x_i^k$$

$$D_2(\mu) := \max_{c\in\Pi} \quad \sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{j:(i,j)\in\mathcal{L}} c_{i,j}^k(\mu_i^k - \mu_j^k)$$

Among them, $D_1(\mu)$ denotes the *data selection problem*, while $D_2(\mu)$ is the *data transmission problem*. In particular, the data selection problem aims at finding the subset of source nodes whose data have the maximum aggregate data reliability, while the data transmission problem aims at scheduling the transmission of the sensory data picked by the data selection problem. In the rest of this section, we will first elaborate on these two problems separately, and then explain how to develop a distributed joint design of them.

The Data Selection Problem

The data selection problem can be further transformed as follows:

$$\begin{aligned} D_1(\mu) &= \sum_{k=1}^{M}\sum_{i=1}^{N} \max_{0\leq x_i^k\leq 1} \Phi(x_i^k) \\ &= \sum_{k=1}^{M}\sum_{i=1}^{N} \max_{0\leq x_i^k\leq 1} (p_i^k - \mu_i^k r_i^k) x_i^k. \end{aligned}$$

In other words, the data selection problem can be solved through separately solving the optimization problem of each source node. since $\frac{\mathrm{d}\Phi(x_i^k)}{\mathrm{d}x_i^k} = p_i^k - \mu_i^k r_i^k$ is a constant, once the value of μ is assigned, the optimal value of x_i^k can be calculated as below:

$$x_i^{k*}(\mu) = \arg\max_{0\leq x_i^k\leq 1} \Phi(x_i^k) = \begin{cases} 1 & \text{if } p_i^k > \mu_i^k r_i^k \\ 0 & \text{if } p_i^k \leq \mu_i^k r_i^k \end{cases}. \quad (9)$$

The result is rather interesting, since x_i^k attains optimum at either 0 or 1, even though we have relaxed its feasible range to be any value between 0 and 1. Therefore, we can directly use x_i^{k*} as the solution to $\mathbf{P}$ without taking the rounding step.

The Data Transmission Problem

We transform the data transmission problem as:

$$\begin{aligned} D_2(\mu) &= \max_{c\in\Pi} \sum_{k=1}^{M}\sum_{i=1}^{N}\sum_{j:(i,j)\in\mathcal{L}} c_{i,j}^k(\mu_i^k - \mu_j^k) \\ &= \max_{c\in\Pi} \sum_{(i,j)\in\mathcal{L}} c_{i,j} \max_{1\leq k\leq M} (\mu_i^k - \mu_j^k), \end{aligned}$$

which can be solved through a joint design of routing and scheduling.

Routing: For each link (i,j), we find the mission k^* that maximizes $\mu_i^k - \mu_j^k$. Then, at the next time slot, the link (i,j) will be dedicated to forward mission k^*'s data.

Scheduling: Let $w_{i,j} = \mu_i^{k^*} - \mu_j^{k^*}$, we target on choosing a bandwidth vector $c^* = (c_{ij}^*)$ such that:

$$c^* = \max_{c \in \Pi} \sum_{(i,j) \in \mathcal{L}} w_{i,j} c_{i,j}. \tag{10}$$

This is actually a linear programming problem, and thus the maximizer can be always found at an extreme point. An extreme point maximizer corresponds to a maximal independent set of the conflict graph. Therefore, this problem is equivalent to the maximum weighted independent set problem over the conflict graph, which is NP-hard. Actually, the conflict graph depends on the underlying interference model. In this paper, we consider node-exclusive interference model, i.e., links that share a common node cannot transmit or receive simultaneously. This model has been widely used in existing work [21]–[23] on network utility maximization. With the node exclusive interference model, the scheduling problem can be reduced to the maximum weighted matching problem, which is polynomial-time solvable. However, the existing polynomial-time solution [24] requires centralized implementation. In [25], a simple distributed approximate algorithm is presented, which is at most a factor of 2 away from the maximum, and has a linear running time $O(|\mathcal{L}|)$. We utilize this algorithm to solve the scheduling problem in a distributed manner.

Actually, the strategy proposed in this paper is a general framework and thus can be extended to other interference models. For any interference model, as long as an appropriate algorithm can be designed to solve the above scheduling problem, it can be integrated with our framework. In addition, the construction of the aforementioned data redundancy graph and sensor conflict graph is independent of interference models. We also use the distributed matching algorithm discussed above to find the subset of source nodes that can collect data simultaneously.

Subgradient Algorithm

We use subgradient method [26] to minimize the dual objective function $D(\mu)$. Specifically, μ is adjusted in the opposite direction to the subgradient:

$$\begin{aligned} \mu_i^k(t+1) &= \left[\mu_i^k(t) - h(t)\frac{\partial D(\mu)}{\partial \mu_i^k}\right]^+ \\ &= \left[\mu_i^k(t) + h(t)(r_i^k x_i^k(t) - \sum_{j:(i,j)\in\mathcal{L}} c_{i,j}^k(t) + \sum_{j:(j,i)\in\mathcal{L}} c_{j,i}^k(t))\right]^+. \end{aligned} \tag{11}$$

In the above formula, the $x_i^k(t)$ and $c_{i,j}^k(t)$ are the maximizers of $D_1(\mu)$ and $D_2(\mu)$, given $\mu(t)$. $h(t)$ is a positive scalar stepsize. Finally, '+' denotes the projection onto the set $\mathbf{R}_+$ of non-negative real numbers.

V. Distributed Implementation

In this section, we describe how the proposed QoI based data selection and transmission scheme can be implemented in a distributed and scalable way. As aforementioned in Section III, every v time slots the events in the sliding window are updated and the source nodes' clustering results of the newly-occurred events are sent to the sink of each mission, where the reliability and redundancy estimates of the sensor nodes are updated. The QoI scores are then sent back to the source nodes. Since both the clustering results and the QoI estimates are numeric data, little communication overhead is incurred during this process.

Upon receiving the reliability score, each source node can derive x_i^k based on Eqn. (9) for each of the virtual sources connected to it, given the shadow price of the current slot μ_i^k. Subsequently, the neighboring nodes exchange their μ_i^k, and solve the routing and scheduling problem, namely, decide which source (relay) nodes will have chance to sense (transmit) in the next slot, through the strategies as we discussed previously in Section IV-B. Once the work for the current time slot is done, each node updates its shadow price according to Eqn. (11).

VI. Performance Evaluation

In this section, we evaluate our proposed schemes. Results on both synthetic data and recorded audio data are presented and discussed.

A. Experimental Settings

Network Topology: We consider a randomly generated sensor network consisting of 50 sensor nodes. The network topology is shown in Fig. 5. In this network, two missions are being undertaken, and each of them involves 10 source nodes (the nodes with index numbers) and a single sink (the hexagon node). The sensor nodes for the two missions can be distinguished by their colors (red for mission 1 and blue for mission 2). In this experiment, we randomly assign the link capacities and data collection rates of the source nodes. We use synthetic data for mission 1 and real audio data for mission 2.

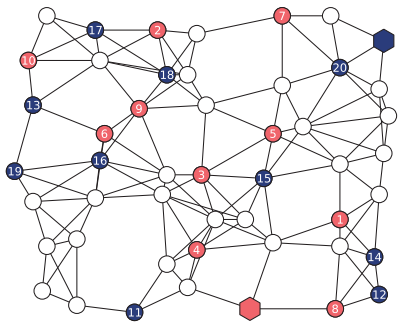

Figure 5. The sensor network used in the experiment.

Synthetic Data: Suppose there are 10 different types of sensors, corresponding to 10 features of the events (e.g., temperature, humidity, etc). We randomly generate events from a Gaussian mixture model with 5 components, each of which corresponds to a class. For the first 7 of 10 source nodes, we randomly assign a subset of the previously defined 10 types of sensors to each of them, and add random Gaussian noise to each type of sensor assigned to this node. We assume that the 8th node is collecting data completely irrelevant to the mission, and thus generate its data from a different distribution. The last two nodes are the duplicates of the first two nodes, with some additional noise added to their data. Therefore, node 1 and 9, node 2 and 10 are two pairs of redundant nodes.

Audio Data: The audio clips we use in this experiment include the sounds of tank, helicopter, and machine gun, corresponding to 3 different classes. We cut the audio clips into pieces with equal time duration, and make a copy for each node. Similar to the synthetic data, we add random noise to the records of the first 7 source nodes with various SNRs. The 8th node is supposed to record fundamentally irrelevant sounds such as crowd talking. The last two nodes are made redundant to node 11 and 12 correspondingly. In the experiment, we extract the MFCC (Mel-Frequency Cepstral Coefficients) features from each audio piece, and feed them as the input to the clustering algorithm.

B. Experimental Results

We evaluate the proposed schemes using the aforementioned data and experimental settings. In this experiment, we set the width w and the stepsize v of the sliding window to be 1000 and 500 time slots. The experiment spans 12500 time slots, and thus the window slides for 25 times.

Table V
NOISE LEVEL

Node	1	2	3	4	5	6	7	8	9	10
Noise (STD)	50	30	1	10	40	5	60	∞	53	31
Node	11	12	13	14	15	16	17	18	19	20
Noise (SNR)	-1	20	10	0	-2	10	5	∞	-1	19

Groundtruth QoI: Table V lists the level of the noise added to each source node. The metric of noise level is the standard deviation (STD) (given Gaussian noise) for mission 1 and signal-to-noise ratio (SNR) for mission 2. For the irrelevant nodes (node 8 and 18), we regard their noise level as infinity. The noise levels of the duplicate nodes (node 9, 10, 19, and 20) are derived through cumulating the noise of the original nodes from which their data are copied and the additional noise injected later. The noise level can be regarded as the groundtruth reliability of the sensor nodes. The groundtruth redundancy, as mentioned in Section VI-A, is as follows: In either mission we set the node pair 1&9 and 2&10 to be redundant, and the other node pairs are irredundant.

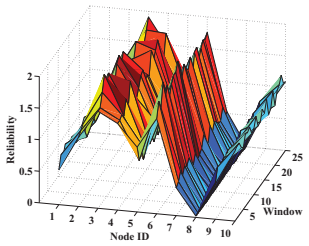

Figure 6. Node-window reliability score on synthetic data

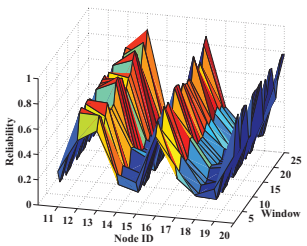

Figure 7. Node-window reliability score on sound data

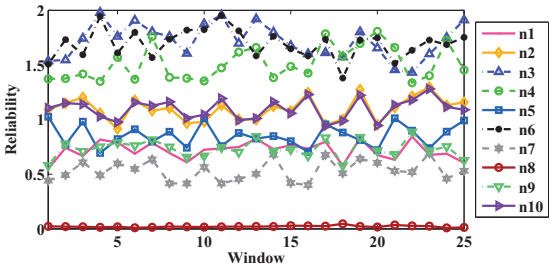

Figure 8. Node reliability evolution on synthetic data

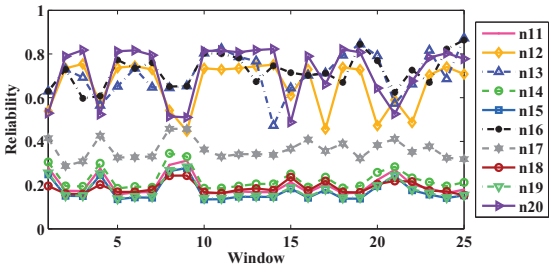

Figure 9. Node reliability evolution on sound data

Next, we use the groundtruth QoI to evaluate the proposed QoI metrics.

Reliability Estimation: We first estimate the data reliability of sensor nodes according to the proposed metric. Figure 6 and 7 show the reliability scores of the source nodes for all the windows in mission 1 and 2. As can be seen, our measures match perfectly with the groundtruth reliability scores, i.e., the levels of noise added to the sensor nodes. For example, node 3 has a noise level of 1 (STD), meaning it is able to capture data with virtually no noise, reflected as the high (peak) reliability measure of node 3 shown in Fig.6. As another example, node 15 has a noise level of -2 (SNR), which means the intensity of the noise added is greater than the original sound captured by the node. Therefore, node 15 would be rather unreliable, reflected as the low reliability measure (valley) at node 15 in Fig.7. Figure 8 and 9 show the per-node reliability traces through the 25 data windows for both missions. As seen, all nodes generally maintain their reliability measures for all the windows. The fluctuations are due to the added noise and the intrinsic nature of the sound data (certain segments of the sounds are indistinguishable).

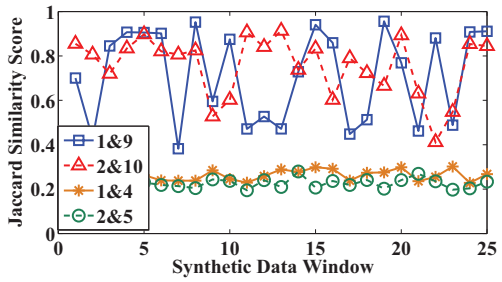

Figure 10. Jaccard coefficient of synthetic data

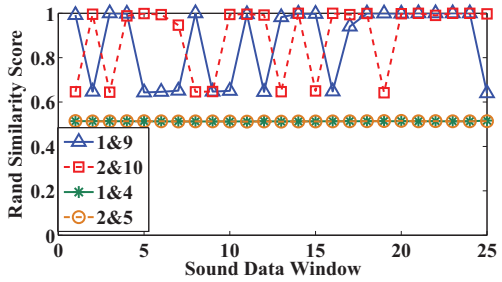

Figure 11. Rand statistic of audio data

Redundancy Estimation: Figure 10 and 11 plot the redundancy measures of the sensor nodes. In particular, we use Jaccard coefficient for synthetic data, and Rand statistic for audio data. In either figure, the measures for two pairs of redundant nodes (node 1&9 and node 2&10) and two pairs of irredundant nodes (node 1&4 and node 2&5) are displayed. As can be seen, there is a clear gap between the measures of redundant and irredundant nodes.

Convergence: Figure 12 shows the evolution of the objective value (Eqn. (7)), i.e., the aggregate reliability over a period of 1500 time slots or 3 windows. As one can see, within each time window, it converges quickly according to the updated QoI measures and oscillates around the optimal values. This oscillating behavior can be interpreted as due to the process of link scheduling.

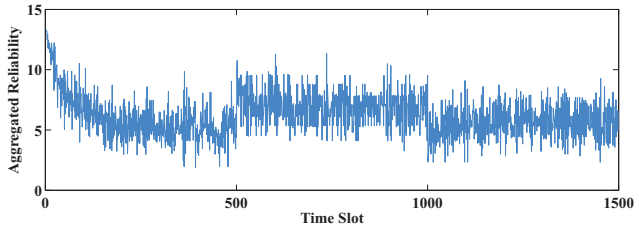

Figure 12. The convergence of the aggregate reliability.

Source Selection: Figure 13 illustrates the selection process of two redundant nodes. In this figure, the vertical axis represents the selection result, where 1 means the corresponding node is selected to collect data for the current time slot and 0 means it is not selected. As can be seen, the two redundant nodes are never selected at the same time.

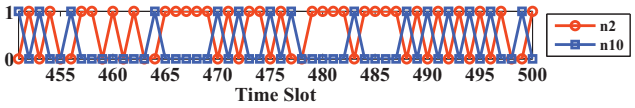

Figure 13. Selection of redundant nodes

Figure 14 and 15 demonstrate the data selection of all the source nodes during the total 25 time window. In the two figures, the darkness of each small rectangle represents the frequency at which that particular node is selected in the corresponding time window. Pure black means the node is selected all the time, while pure white means the node is never selected in any of the time windows.

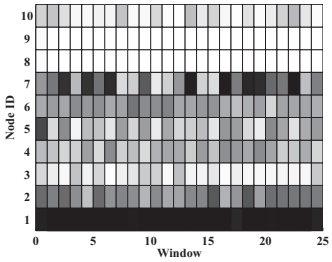

Figure 14. Source Selection of Synthetic Data

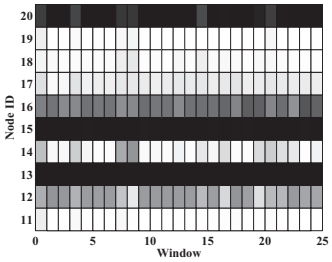

Figure 15. Source Selection of Audio Data

Table VI lists the average selection rate of each source of all the windows (the row named optimal selection). At the first glance, the result does not strictly follow the groundtruth QoI. For example, node 1 keeps working almost all the time, while node 3 has little chance to be selected, although its noise level is much smaller than that of node 1. The problem lies in the network resource constraint. If a node has very little available resource along the path to the sink, it will not be selected frequently even if its data is highly reliable. For comparison, we design a naive baseline scheme in which all the source nodes are treated equally. Specifically, we set the reliability values of all the sources, i.e., p_i^k in the objective function of $\mathbf{P}$, to be equal. The source selection rates of the baseline scheme are listed in Table VI (the row named uniform selection). As the numbers suggest, compared with uniform selection, the proposed optimal selection scheme is more compatible with the groundtruth QoI. For example, node 6 has a rather low noise level, thus our optimal selection scheme picks it nearly half of the time; on the other hand, the uniform selection scheme would only select it one out of ten times. As another example, node 16 has the second highest SNR among all the nodes in mission 2. In our optimal selection scheme, the chance that this node is picked is almost two out of three; but the uniform scheme only selects this node less than 20% of the time.

As a final evaluation measure, we compute the window-averaged total reliability scores of the two selection schemes for both missions. For each window of each node in each mission, we take the product of the node reliability score

Table VI
SOURCE SELECTION

Node	1	2	3	4	5	6	7	8	9	10
Optimal Selection	.99	.57	.09	.33	.31	.43	.6	0	0	.11
Uniform Selection	1	.3	.11	.2	.39	.1	.99	.25	.06	.1
Node	11	12	13	14	15	16	17	18	19	20
Optimal Selection	.01	.39	1	.1	.99	.65	.08	.03	.01	.97
Uniform Selection	.12	.15	1	.53	1	.19	.12	.28	.14	.66

and its selection rate, we then average the products over all the windows, and sum the averages across all the nodes. For the first mission, the uniform selection scheme has a score of 2.7214; our optimal selection scheme achieves a score of 3.5345, an 29.88% advantage. For the second mission, the two scores are 1.8513 and 2.3933, with our optimal scheme claiming a 29.28% advantage.

VII. CONCLUSIONS

In this paper, we identify the two aspects of QoI, data reliability and redundancy, for classification missions in sensor networks, and propose metrics to estimate them. Then, we develop a data selection and transmission service which can maximize the reliability of the delivered data, with data redundancy being removed. The key of the solution lies in the decomposition of the data selection and transmission problem. A distributed algorithm is designed to solve the decomposed problem separately.

ACKNOWLEDGMENT

Research reported in this paper was sponsored by ONR grant N00014-10-1-0172, NSF grants CNS 09-05014 and 10-40380, and the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] I. Fodor, "A survey of dimension reduction techniques," Tech. Rep., 2002.
[2] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.
[3] Y. Yang, L. Wang, D. K. Noh, H. K. Le, and T. F. Abdelzaher, "Solarstore: enhancing data reliability in solar-powered storage-centric sensor networks," in *MobiSys*, 2009.
[4] Y. Yang, L. Su, Y. Gao, and T. F. Abdelzaher, "Solarcode: Utilizing erasure codes for reliable data delivery in solar-powered wireless sensor networks," in *INFOCOM*, 2010.
[5] W. Hu, V. N. Tran, N. Bulusu, C. T. Chou, S. Jha, and A. Taylor, "The design and evaluation of a hybrid sensor network for cane-toad monitoring," in *IPSN*, 2005.
[6] J. Cai, D. Ee, B. Pham, P. Roe, and J. Zhang, "Sensor network for the monitoring of ecosystem: Bird species recognition," in *ISSNIP*, 2007.
[7] L. Su, Y. Yang, B. Ding, J. Gao, T. F. Abdelzaher, and J. Han, "Hierarchical aggregate classification with limited supervision for data reduction in wireless sensor networks," in *SenSys*, 2011.
[8] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. A. Stankovic, T. Abdelzaher, and B. H. Krogh, "Lightweight detection and classification for wireless sensor networks in realistic environments," in *SenSys*, 2005.
[9] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *WSNA*, 2002.
[10] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A line in the sand: A wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, pp. 605–634, 2004.
[11] X. Cheng, J. Xu, J. Pei, and J. Liu, "Hierarchical distributed data classification in wireless sensor networks," in *MASS*, 2009.
[12] E. Miluzzo, C. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. T. Campbell, "Darwin phones: the evolution of sensing and inference on mobile phones," in *MobiSys*, 2010.
[13] M. Keally, G. Zhou, G. Xing, J. Wu, and A. J. Pyles, "Pbn: towards practical activity recognition using smartphone-based body sensor networks," in *SenSys*, 2011.
[14] M. Keally, G. Zhou, G. Xing, and J. Wu, "Exploiting sensing diversity for confident sensing in wireless sensor networks," in *INFOCOM*, 2011.
[15] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[16] J. Beringer and E. Hüllermeier, "Online clustering of parallel data streams," *Data Knowl. Eng.*, vol. 58, no. 2, pp. 180–204, 2006.
[17] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.
[18] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, 2005.
[19] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *MOBICOM*, 2003.
[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[21] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *INFOCOM*, 2006.
[22] X. Lin, N. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.
[23] L. Su, Y. Gao, Y. Yang, and G. Cao, "Towards optimal rate allocation for data aggregation in wireless sensor networks," in *MobiHoc*, 2011.
[24] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Dover Publications, 1998.
[25] J.-H. Hoepman, "Simple distributed weighted matchings," *CoRR*, 2004.
[26] N. Z. Shor, K. C. Kiwiel, and A. Ruszcayǹski, *Minimization methods for non-differentiable functions*. New York, NY, USA: Springer-Verlag New York, Inc., 1985.