Congestion-adaptive Data Collection with Accuracy Guarantee in Cyber-Physical Systems

Nematollah Iri†, Lei Yu‡, Haiying Shen†, Gregori Caulfield†
†Department of Electrical and Computer Engineering, Clemson University, USA
‡ Department of Computer Science, Georgia State University, USA
†{niri, gcaulfi, shenh}@clemson.edu, ‡{lyu13@student.gsu.edu}

***Abstract*—Data collection by wireless sensor networks is a fundamental and critical function for cyber-physical systems (CPS) to estimate the state of the physical world. However, unstable network conditions impose great challenges in guaranteeing data accuracy, which is essential for reliable state estimation of the physical phenomena. For underlying sensor networks, without efficiently resolving congestion in data transmission, packet loss at congested nodes can considerably increase the estimation error. However, previous congestion control schemes relying on reducing transmitted data samples also increase the estimation error. Thus, we propose a Congestion-Adaptive Data Collection scheme (CADC) to efficiently resolve the network congestion while guaranteeing the overall data estimation accuracy. CADC mitigates congestion by adaptive lossy compression with guarantee that a given overall data estimation error bound is satisfied. Besides, since a CPS application may have different priorities for different data items, we further propose a weighted CADC scheme such that the data with higher priority has less distortion. Extensive experimental results demonstrate the effectiveness and efficiency of our CADC schemes.**

I. Introduction

Wireless Sensor Networks (WSNs) enable the sensing of physical phenomena in a large scale and become fundamental infrastructures in cyber-physical systems (CPS). The sensor nodes in a WSN sample the physical phenomena such as temperature and light, and transmit the data to the base station (or controllers) for the state estimation of physical phenomena. Such data collection, however, faces great challenges from unstable network conditions. A WSN usually consists of hundreds to thousands of sensor nodes, which generate a tremendous amount of sensed data and deliver it to the base station using multi-hop wireless transmission. The large amount of data and unstable wireless links easily lead to network congestion, which incurs substantial packet loss. The packet loss can significantly increase the estimation error of CPS, though the controllers require guaranteed estimation accuracy of the physical phenomena for reliable control decisions.

To avoid network congestion, many congestion control schemes for WSNs have been proposed. Most schemes [1]–[5] are based on rate control, which reduces the data generation rate at source nodes or compresses the spatio-temporal samples at the intermediate relay nodes. However, by reducing data samples to be transmitted to avoid congestion, these schemes also concurrently increase the estimation error. Therefore, a congestion control scheme should work around a "sweet spot" that avoids the congestion while still guarantees the estimation accuracy of applications.

In this paper, we consider the problem of ensuring required estimation accuracy while reducing congestion. We assume nodes transmit data upwards to the sink through a routing tree [5]. We propose a Congestion-Adaptive Data Collection scheme (CADC), which determines the maximum tolerable distortion of data due to compression at each node to guarantee a given estimation accuracy at the sink. In order to reduce data distortion by compression, each node novelly uses the k-means clustering algorithm for lossy data compression with its maximum tolerable distortion bound. When congestion occurs, by adaptively adjusting the maximum tolerable distortion allowed at sensor nodes, CADC makes best effort to guarantee the given estimation accuracy at the sink and reduce the congestion.

Besides, we also consider the different priorities of data measurements. A CPS application may have different priorities for data items in different value ranges. For example, the safety monitoring system may be more interested in high temperature readings, thus the temperature measurements with higher values are more important and should have lower distortion hence compression degree. To this end, we propose weighted CADC scheme, which assigns weights to the measurements according to their priorities and aims to minimize the weighted estimation error. We conduct extensive simulations to evaluate our CADC schemes in comparison with previous schemes. Experimental results demonstrate the high effectiveness and efficiency of our schemes.

The rest of paper is organized as follows. Section II summarizes the related work. Section III illustrates our system model and objective. Section IV presents our congestion-adaptive data collection schemes in detail. Section V presents the performance evaluation of our schemes in comparison with previous methods. Section VI concludes this paper with remarks on our future work.

II. Related Work

We present previous works for data collection to the sink in WSNs in three categories: data compression, routing and congestion control.

Data compression. Many works have exploited data correlation to compress data in transmission to reduce communication cost. Cristescu *et al.* [6] utilized the Slepian-Wolf coding to compress correlated readings and addressed the problem of finding the optimal rate allocation for each node to minimize total data transmission cost. Silberstein *et al.* [7] proposed CONCH, which exploits the spatio-temporal data

correlation to suppress unnecessary value transmissions in continuous data collection to reduce energy cost. Luo *et al.* [8] proposed to apply compressive sampling theory to sensor data gathering to reduce global scale communication cost. Gupta *et al.* [9] proposed to select a small subset of sensor nodes that may be sufficient to reconstruct data for the entire sensor network within predefined error bound. Wang *et al.* [10] proposed an approximate data collection, in which the network is partitioned into clusters, and cluster heads construct the local estimation model with prespecified error bounds to approximate the readings of sensor nodes in the clusters. The sink then estimates the data based on the model parameters sent by cluster heads.

Routing. Many routing protocols have been proposed for data collection to reduce energy cost or latency. Chang *et al.* [11] addressed the optimal routing problem with the objective to maximize the network lifetime. Park *et al.* [12] developed an online heuristic for the problem of routing message to maximize the network lifetime. Lee *et al.* [13] proposed a collision-free scheduling method for data collection routing to optimize energy consumption and reliability. Han *et al.* [14] addressed the problem of minimizing the expected total transmission power for reliable data dissemination in duty-cycled WSNs. Su *et al.* [15] aimed at achieving optimal rate allocation for data aggregation in WSNs with the goal to maximize resource utilization and proposed a distributed algorithm for joint rate control and scheduling. Wan *et al.* [16] studied the problem of finding the minimum-latency schedule for data aggregation in wireless networks under the interference constraint.

Control congestion. The previous control congestion schemes can be classified into two classes: centralized rate control schemes and distributed rate control schemes. Event-to-Sink Reliable Transport (ESRT) [1] lets the base station adjust the reporting frequency of sensor nodes such that the required information can be obtained with minimum energy considering one-hop communication between nodes and the base station. Bian *et al.* [17] proposed a centralized rate allocation scheme that assigns sending rates to all sensors in the routing tree based on the wireless link characteristics. Zhou *et al.* [2] proposed a source reporting rate control mechanism (PORT), which is aware of transmission cost of the sources, and adjusts the source reporting rates with a guarantee that the sink can still obtain enough information. Paek *et al.* [3] proposed the rate controlled reliable transport protocol (RCRT), where the sink is responsible for congestion detection and rate allocation of sensor nodes based on AIMD (Additive Increase - Multiplicative Decrease). CODA [4] is a distributed rate control scheme for congestion avoidance. In these rate control based schemes, since decreasing data rate reduces the number of spatio-temporal samples, they cannot control the accuracy of the state estimation. To address this problem, Ahmadi *et al.* [5] takes into account the estimation error in the congestion control. Using least-error summarization, their scheme eliminates congestion while incurs the least possible overall error in sensing the physical environment.

In this paper, we consider the congestion issue in data collection and aim to design a congestion-adaptive data collection scheme for WSNs with estimation error bound. Our work is most related to [5]. However, the scheme in [5] is unaware of the accuracy requirements of applications, and the data collection with such congestion control scheme may fail to achieve the required data accuracy. Instead, our scheme aims to ensure the pre-specified error bound when congestion occurs.

III. System Model And Objective

A. System Model

We assume a WSN for data collection, in which N sensor nodes are deployed to monitor a physical phenomenon of the environment and periodically send their sensor readings to a sink. Due to communication limitations of the sensor nodes, they transmit their sensing data in a multi-hop fashion to the sink (denoted by r), which is responsible for collecting and processing the measurements. As shown in Figure 1, we assume a routing tree rooted at the sink as our network layer [18], [19], denoted by $\mathcal{T}_r$. The depth of a sensor node i is defined as the hop distance between node i and the sink, denoted by $h_{i,r}$. Node i is the ancestor of node j if j is in the subtree rooted at i (denoted by $\mathcal{T}_i$).

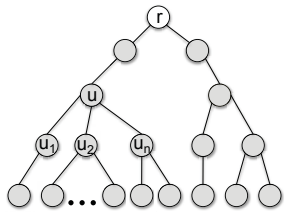


Fig. 1: Routing tree.

TABLE I: Notations

Parameter	Description
$\mathcal{T}_i$	Subtree of the routing tree rooted at node i
r	Sink node
u, u_i, i	Notations for sensor nodes other than the sink
x_i	Data generated at sensor node i
$\hat{x}_i^u$	Value of x_i reconstructed by i's ancestor u based on the received compressed data
e_u	Sum of the errors between received values of data at sensor node u and their actual values
ϵ_u	Maximum tolerable error at node u (upper bound for e_u)
d_u	Sum of the errors between received data at node u and their values after compression at node u
η_u	Maximum tolerable distortion of data due to compression at node u (upper bound for d_u)
w_i	Priority coefficient of x_i generated by node i

To describe our scheme, we first assume that network tree topology is fixed. We will discuss how our scheme adapts to network topology changes in Section IV-E. Data is forwarded along $\mathcal{T}_r$ to the sink. Each node periodically sends its measured data and also forwards its received data from children to its parent. Given this system model, congestion occurs when a node cannot transmit data messages at the rate they are received and generated caused by insufficient bottleneck resource [5]. One of the main components in congestion control schemes is congestion detection. For this purpose, we can use a previously proposed congestion detection scheme [5]. That is, a node compares its output buffer size with a threshold, and it is congested if its buffer size is higher than a threshold.

B. Motivation and Objective

Data quality is critical for the controllers to accurately estimate the state of the physical phenomenon. However,

congestion control has the two-sided influence on the data accuracy: (1) congestion elimination reduces data loss and improves the estimation accuracy, (2) but the ways to control congestion such as reducing source rate [1]–[4], [17] and aggregation [5] increase the estimation error. Thus, the design of congestion control scheme in data-collection networks needs to achieve the trade-off between the two sides while ensures that the estimation error resulting from collected data is within the tolerable range of CPS applications.

With the above motivation, we propose the Congestion-Adaptive Data Collection scheme (CADC), which reduces congestion by reducing the data transmission rate with lossy compression, while still guaranteeing the data accuracy required by CPS applications.

In CADC, when congestion occurs at a node, to reduce the congestion, its children nodes reduce their data transmission rates by lossy compression on the data to be forwarded, which however causes data distortion. Formally, we denote the measurement of sensor node i as x_i, and denote the value of x_i reconstructed by i's ancestor u based on the received compressed data as $\hat{x}_i^u$, which may not equal to x_i due to compression. We define the estimation error and data distortion as follows:

Definition 3.1: *(ESTIMATION ERROR)* Estimation error (error in short) at node u represents the sum of errors between its received data values from its subtree $\mathcal{T}_u$ and their actual values, i.e.,

$$e_u = \sum_{i \in \mathcal{T}_u} (\hat{x}_i^u - x_i)^2. \quad (1)$$

$|\mathcal{T}_u|$ denotes the number of sensors in $\mathcal{T}_u$.

Definition 3.2: *(DATA DISTORTION)* Data distortion at node u_k represents the sum of errors between its received data values from its subtree $\mathcal{T}_{u_k}$ and their corresponding values after compression received by its parent u, i.e.,

$$d_{u_k} = \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})^2. \quad (2)$$

The data accuracy requirement of a CPS application is characterized by the maximum tolerable estimation error at the sink node r, denoted by ϵ_r.

Objective: Our objective is to avoid congestion while ensuring that the resulting estimation error at sink r, denoted by e_r, is less than ϵ_r, i.e., $e_r \leq \epsilon_r$.

IV. Congestion-Adaptive Data Collection

A. CADC Scheme Overview

To achieve the above-stated objective, CADC determines two parameters for each node u, *maximum tolerable error* (ϵ_u) and *maximum tolerable distortion* (η_u). ϵ_u and η_u are the upper bounds for estimation error e_u and data distortion d_u at node u (defined by Definitions 3.1 and 3.2), respectively. That is,

$$e_u \leq \epsilon_u, \ d_u \leq \eta_u. \quad (3)$$

Consider a node u and its children $u_1, \ldots, u_n$ in the routing tree (Figure 1). In CADC, node u uses ϵ_u to determine η_{u_k} of each of its children (u_k). Node u_k compresses its data based on η_{u_k} to reduce its data transmission rate for congestion control. The value of η_{u_k} for each child ensures $e_u \leq \epsilon_u$. Finally, the estimation error at the sink is no more than the fixed maximum tolerable estimation error at the sink ($e_r \leq \epsilon_r$), which means that CADC helps to satisfy the constraint of the desired data accuracy of CPS applications.

CADC dynamically and distributedly determines proper values of maximum tolerable error (ϵ_u) and maximum tolerable distortion (η_u) for every node u based on the network status and a given ϵ_r. During data collection, CADC first determines the initial values of ϵ_u and η_u for each node u, and then dynamically updates them based on the current network congestion status and reduce the congestion accordingly. If a node u is congested, it asks each child u_i to transmit data in a lower rate to avoid congestion through data compression. But if the data compression with such a lower rate incurs data distortion larger than η_{u_i}, node u attempts to increase η_{u_i} to accommodate such compression without violating the constraint of maximum tolerable error ϵ_u. Only if there is no way to achieve the desired η_{u_i} while satisfy ϵ_u, u requests its parent to update ϵ_u. Such parameter update could repeat along the path to the sink to make the error at the sink less than the given error bound ϵ_r. This case means that the overall system is highly congested and $e_r \leq \epsilon_r$ cannot be satisfied in any way. Then, the sink will inform the applications of the off-specification of data.

In the following, we present the details of the three parts of CADC.

- Given ϵ_r, how to determine the maximum tolerable error (ϵ_u) and distortion (η_u) for every node u to realize our objective (Section IV-B)?
- How can a node compress its data based on its η_{u_k} while minimizing the data distortion (Section IV-C)?
- How to conduct congestion control and update ϵ_u and η_u to achieve our objective in dynamic network status (Section IV-D)?
- How to adapt to dynamic network topology (Section IV-E)?

B. Determination of Maximum Tolerable Error and Distortion

As we can see, in CADC, a fundamental problem is: *Given maximum tolerable error ϵ_u at any node u and current network status, how to determine maximum tolerable errors (ϵ_{u_k}) and maximum tolerable distortions (η_{u_k}) for u's children, $u_1, ..., u_n$, such that $e_u \leq \epsilon_u$.* After the problem solution is found, given a ϵ_r on the sink, the (ϵ_u, η_u) of each of its children u can be determined. Then, the (ϵ_{u_k}, η_{u_k}) of each of u's children are determined and so on. Finally, the (ϵ_i, η_i) of each node i are determined in the top-bottom manner to achieve our objective. In this section, we address this problem in two cases:

- *Non-priority case*, in which all of the sensor measurements are equally important for an application (Section IV-B1).
- *Priority case*, in which the measurements have different priorities (Section IV-B2).

1) Non-Priority Case: The estimation error e_u at node u equals the accumulated errors from each of its children $u_1, ..., u_k, ...u_n$:

$$\begin{aligned} e_u = & \sum_{i \in \mathcal{T}_u} (\hat{x}_i^u - x_i)^2 \\ = & \sum_{i \in \mathcal{T}_{u_1}} (\hat{x}_i^u - x_i)^2 + ... + \sum_{i \in \mathcal{T}_{u_n}} (\hat{x}_i^u - x_i)^2 \end{aligned}$$

The *error contribution* of child u_k, denoted by c_{u_k}, equals $\sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^u - x_i)^2$. Based on the definition of c_{u_k}, we use Cauchy-Schwartz inequality to get:

$$\begin{aligned} c_{u_k} &= \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^u - x_i)^2 \\ &= \sum_{i \in \mathcal{T}_{u_k}} ((\hat{x}_i^u - \hat{x}_i^{u_k}) + (\hat{x}_i^{u_k} - x_i))^2 \\ &= \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})^2 + \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^{u_k} - x_i)^2 \\ &+ 2 \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})(\hat{x}_i^{u_k} - x_i) \\ &\leq \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})^2 + \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^{u_k} - x_i)^2 \\ &+ 2 \sqrt{\sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})^2 . \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^{u_k} - x_i)^2} \\ &= d_{u_k} + e_{u_k} + 2\sqrt{d_{u_k} \cdot e_{u_k}} \end{aligned} \quad (4)$$

where e_{u_k} is the estimation error at child u_k and d_{u_k} is data distortion due to data compression of u_k.

As the maximum tolerable error (ϵ_{u_k}) and maximum tolerable distortion (η_{u_k}) are the upper bounds of e_{u_k} and d_{u_k}, respectively, we define *maximum tolerable error contribution* of u_k:

$$c_{u_k}^m = \eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k} \cdot \epsilon_{u_k}}. \quad (5)$$

To guarantee $e_u = \sum_{u_k : u_k is\ child\ of\ u} c_{u_k} \leq \epsilon_u$, the determination of η_{u_k} and ϵ_{u_k} needs to ensure

$$\sum_{u_k : u_k is\ child\ of\ u} c_{u_k}^m \leq \epsilon_u \Rightarrow \sum_{u_k} (\eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k} \cdot \epsilon_{u_k}}) \leq \epsilon_u \quad (6)$$

In this way, since $d_{u_k} \leq \eta_{u_k}$ and $e_{u_k} \leq \epsilon_{u_k}$, we can achieve that $e_u = \sum_{u_k} c_{u_k} \leq \epsilon_u$. As a result, Formula (6) gives the principle to initialize and update parameters (ϵ_u, η_u) for each node u. We present the parameter initialization below, and present the parameter update in CADC's congestion control in Section IV-D.

Initialization of ϵ_{u_k} and η_{u_k}: With a priori knowledge of network congestion status, we can properly initialize the maximum tolerable error and distortion (ϵ_{u_k}, η_{u_k}) for each node u_k. In the rooting tree for data collection, a subtree with a larger size tend to suffer more congestions because it needs to forward a larger amount of data to the sink. As a result, a larger subtree may introduce higher estimation error into the data to the upper node due to CADC's lossy compression. Thus, the root of a larger subtree needs a larger maximum tolerable error to allow more data compression within the subtree to mitigate the congestions. Based on this rationale, node u initializes the (ϵ_{u_k}, η_{u_k}) for each of its children u_k according to the size of each child's subtree.

Based on Formula (6), to guarantee $e_u \leq \epsilon_u$, we let

$$\eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k} \cdot \epsilon_{u_k}} = \alpha_k \epsilon_u \ (0 < \alpha_k < 1) \quad (7)$$

where $\sum_{k=1,...,n} \alpha_k = 1$ such that

$$e_u \leq \sum_{u_k} (\eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k} \cdot \epsilon_{u_k}}) = \sum_{k=1,...,n} \alpha_k \epsilon_u = \epsilon_u. \quad (8)$$

We choose α_k by

$$\alpha_k = \frac{|\mathcal{T}_{u_k}|}{\sum_{k=1,...,n} |\mathcal{T}_{u_k}|} \quad (9)$$

where $|\mathcal{T}_{u_k}|$ is the size of subtree $\mathcal{T}_{u_k}$, such that any node with a larger subtree size can have a higher maximum tolerable error. To find subtree sizes $|\mathcal{T}_{u_k}|$ in Equation (9), we use the same procedure as in [18]. Particularly, each sensor node sends its subtree size in its packet header. Each parent node sums up subtree sizes of its children and adds one to it to find its own subtree size, with subtree size of leaf nodes being 1.

After α_k (hence $\alpha_k \epsilon_u$) is determined, based on Equation (9), node u needs to determine η_{u_k} and ϵ_{u_k} to satisfy Equation (7). In order to maximize the estimation accuracy, we let every node send raw data without data compression initially, i.e., $\eta_{u_k} = 0$. Later on, CADC adjusts η_{u_k} to avoid congestion when it occurs. With $\eta_{u_k} = 0$ initially, from Formula (7), we have $\epsilon_{u_k} = \alpha_k \epsilon_u$. In CADC's congestion control (Section IV-D), when congestion occurs at node u, if $e_u \leq \epsilon_u$ still can be satisfied by data compression for congestion control, η_u does not need to update and only η_{u_k} needs to update. Therefore, setting ϵ_{u_k} to the possible maximum value ($\epsilon_{u_k} = \alpha_k \epsilon_u$) can avoid frequent updates later on. As a result, we find a solution for the problem indicated at the beginning of this section. Using this solution, given a ϵ_r at the sink, CADC can determine the (ϵ_u, η_u) of each node in the system in the top-down matter to guarantee $e_r \leq \epsilon_r$.

2) Priority Case: In this section, we consider the scenario in which the data has different priorities. For example, for a fire detection or cooling application, high temperature values, which may indicate abnormality, have higher priority than low temperature values. High-priority data should suffer less distortion, so that the event can be more accurately modeled and quickly detected. We use *priority coefficients* to show the importance degree of different data items. We assume that the priority coefficient is a function of data value, which is known to all sensor nodes. Approximate values will have the same or close priority coefficients. Then, when a sensor receives a data value, it determines its priority coefficient based on the priority function and the data value. We need to determine maximum tolerable error and distortion with the goal that the higher-priority data has less estimation error in order to achieve more accurate state estimation for CPS control. If priority coefficients are equal for all data, the problem is reduced to the previous non-priority case.

We define *weighted estimation error* and *weighted data distortion* below with the consideration of data priority.

$$e_u^w = \sum_{i\in\mathcal{T}_u} w_i(\hat{x}_i^u - x_i)^2, \quad (10)$$

$$d_{u_k}^w = \sum_{i\in\mathcal{T}_{u_k}} w_i(\hat{x}_i^u - \hat{x}_i^{u_k})^2, \quad (11)$$

where x_i denotes the data measured by node i with priority w_i, $\hat{x}_i^u$ denotes the value of x_i received by node u, and u_k is a child of u. Accordingly, we define *weighted error contribution* of u's child node u_k as $c_{u_k}^w = \sum_{i\in\mathcal{T}_{u_k}} w_i(\hat{x}_i^u - x_i)^2$. Similarly, we have

$$\begin{aligned} c_{u_k}^w &= \sum_{i\in T_{u_k}} w_i((\hat{x}_i^u - \hat{x}_i^{u_k}) + (\hat{x}_i^{u_k} - x_i))^2 \\ &= \sum_{i\in T_{u_k}} w_i(\hat{x}_i^u - \hat{x}_i^{u_k})^2 + \sum_{i\in T_{u_k}} w_i(\hat{x}_i^{u_k} - x_i)^2 \\ &+ 2\sum_{i\in T_{u_k}} w_i(\hat{x}_i^u - \hat{x}_i^{u_k})(\hat{x}_i^{u_k} - x_i) \\ &\le \sum_{i\in T_{u_k}} w_i((\hat{x}_i^u - \hat{x}_i^{u_k})^2 + \sum_{i\in T_{u_k}} w_i(\hat{x}_i^{u_k} - x_i)^2 \\ &+ 2.\sqrt{\sum_{i\in T_{u_k}} w_i(\hat{x}_i^u - \hat{x}_i^{u_k})^2 \sum_{i\in T_{u_k}} w_i(\hat{x}_i^{u_k} - x_i)^2} \\ &= d_{u_k}^w + e_{u_k}^w + 2\sqrt{d_{u_k}^w \cdot e_{u_k}^w} \end{aligned} \quad (12)$$

Equation (12) is derived with the assumption that the priority coefficient of data x_i at parent u and child u_k remains the same in CADC. This is reasonable because the compression method in CADC (Section IV-C) constrains the distortion of data in compression, and the data will most likely have the same or close priority coefficient after compression, which is confirmed in our experiments in Section V. As we can see from Formula (12), it has the same form as the non-priority case. Thus, in the priority case, we can use the same principle for determining the (weighted) maximum tolerable error and distortion, and choose the same initial values.

C. Data Compression Scheme

To reduce the congestion, the nodes compress the received data based on their maximum tolerable distortion (η_u) before transmitting the data to their parents. Sensor readings may be redundant because nodes in the same neighborhood can have approximate readings in WSNs. Unlike the previous compression methods that do not focus on minimizing data distortion in compression, our data compression scheme aims to select most representative data samples that minimize the data distortion. Accordingly, we use the *k-means clustering algorithm* (k-means in short) [20] for data compression. Given a set of data points V in real d-dimensional space $\mathbb{R}^d$ and an integer k, k-means clustering is to partition the points into k clusters; each with a center (i.e., cluster head) not necessarily belonging to the set of points, with the goal of minimizing the mean squared distances of each point to its nearest cluster head. Formally, it is to minimize

$$C(V) = \sum_{x\in V} (x - c(x))^2, \quad (13)$$

where $C(V)$ is the cost of clustering and $c(x)$ is the center of the cluster that data x belongs to. $C(V)$ actually reflects the data distortion.

Thus, in CADC, to compress the data, a node conducts the k-means clustering on its received and generated data and sends the values of cluster heads and corresponding cluster sizes to its parent. CADC represents data in the form of tuples $< (v_1, n_1), \ldots, (v_i, n_i), \ldots, (v_m, n_m) >$, where v_i is the sample value, and n_i is the number of sensor readings (each from a sensor node) with value v_i. $n = 1$ if the data represents a single sensor reading. For example, if a node receives a set of sensor readings $\{(3,1),(4,1),(6,1),(8,1),(10,1),\ (12,1)\}$ from 6 nodes, with 2-means clustering, this dataset is partitioned to two clusters $\{3,4,6\}$ and $\{8,10,12\}$, with centers equal to 4.33 and 10, respectively. Then, the compressed dataset is represented by $\{(4.33,3),(10,3)\}$.

In order to apply the k-means clustering method to the priority case, we modify the cost function $C(V)$ for k-means clustering to

$$C(V) = \sum_{x\in V} w(x)(x - c(x))^2, \quad (14)$$

where $w(x)$ is the priority coefficient of data x. Thus, data with higher priority will have less distortion.

In the congestion control (Section IV-D), CADC uses the k-means clustering algorithm for data compression through two methods under the constraint that the data distortion after compression (i.e., cost of clustering) $C(V)$ is less than a given bound. In the first method, a node needs to reduce the available data into k samples with a given value of k. For this purpose, we can directly use an existing k-means clustering algorithm such as Lloyd's algorithm [20]. In the second method, a node needs to find minimum k for data compression. For this purpose, we can simply enumerate all possible values of k from 1 to the total number of data points. For each value of k, we use Lloyd's algorithm to find k clusters and the cost of clustering. Once the cost of clustering becomes no more than the given bound, the algorithm returns current k and cluster heads.

D. Congestion Control

In this section, we introduce the procedure of congestion control in CADC, including adaptive adjustments of the maximum tolerable error and distortion (ϵ_u, η_u), and the corresponding congestion control.

Consider an arbitrary node u and its children $u_1, ..., u_n$ with maximum tolerable error and distortion, ϵ_{u_i} and η_{u_i} ($i = 1, ..., n$). When node u is congested, to avoid the congestion, u needs to reduce its input data arrival rate r_u^{in} to less than its output transmission rate r_u^o by asking its children to reduce their transmission rates through data compression. When $r_u^{in} < r_u^o$, node u's buffer size will decrease and the congestion can be resolved finally. Node u first computes the ratio of r_u^{in} to r_u^o, $\frac{r_u^{in}}{r_u^o}$, and then sends a congestion notification with this ratio to each of its children. Since r_u^{in} is the sum of data transmission rates of all u's children, decreasing each child's current transmission rate by a factor of at least $\frac{r_u^{in}}{r_u^o}$ can reduce r_u^{in} to less than r_u^o. To do this, each

child decreases its current data compression ratio by r_u^{in}/r_u^o times. The data compression ratio is the ratio of the number of compressed samples to the number of available data tuples to be transferred. The compression ratio is 1 if the node sends data without compression.

We use γ_i' and γ_i to denote the previous and new data compression ratio of node u_i: $\gamma_i = \gamma_i'/\frac{r_u^{in}}{r_u^o}$. For the dataset of available data tuples to be forwarded at node u_i $< (v_1, n_1), \ldots, (v_i, n_i), \ldots, (v_m, n_m) >$, we use $N_i = \sum_{j=1}^{m} n_j$ to denote the total number of data readings in the dataset. Node u_i compresses data with compress ratio γ_i by using the k-means clustering with $k = N_i\gamma_i$ (Section IV-C). Such data compression with γ_i will lead to data distortion (denoted by d_{γ_i}) computed by Formulas (13) and (14) in the non-priority and priority cases, respectively. Recall that to ensure $e_r \leq \epsilon_r$, node u_i has maximum tolerable distortion, η_{u_i}, defined over all the $|\mathcal{T}_{u_i}|$ data readings from its subtree $\mathcal{T}_{u_i}$. Thus, node u_i needs to compress data with distortion not exceeding $N_i\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$, where $\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$ is the average maximum distortion allowed for each data reading from each node in the subtree $\mathcal{T}_{u_i}$. $N_i\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$ means the distortion allowed for the dataset with N_i readings. Then, if $d_{\gamma_i} \leq N_i\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$, u_i just sends the compressed samples to the parent. If $d_{\gamma_i} > n_i\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$, which means that the compression ratio required to reduce the congestion cannot satisfy the distortion constraint hence $e_r \leq \epsilon_r$, the parameters (ϵ_u, η_u) for congestion control then must be updated. Next, we explain how to update parameters to ensure $e_r \leq \epsilon_r$ while reduce congestion in this case.

When $d_{\gamma_i} > N_i\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$, node u_i tries to compress data as much as possible with data distortion not exceeding data distortion constraint $N_i\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$ by finding the minimum number of cluster heads for k-means clustering under the constraint (Section IV-C). This data compression makes data distortion smaller than d_{γ_i}, so the compression ratio is still larger than γ_i required to avoid congestion. Such data compression can mitigate congestion but cannot eliminate it. In order to avoid the subsequent congestions, i.e., achieve γ_i, u_i requests its parent to increase its maximum tolerable distortion (η_{u_i}) such that $d_{\gamma_i} \leq N_i\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$. In this case, $\eta_{u_i} \geq d_{\gamma_i}\frac{|\mathcal{T}_{u_i}|}{N_i}$. To avoid frequent such requests and parameter updates, η_{u_i} can be set to the historically largest value.

To do that, each node maintains two parameters: maximum necessary distortion ($\eta_{u_i}^*$) and maximum necessary error ($\epsilon_{u_i}^*$). $\eta_{u_i}^*$ keeps track of the maximum distortion required to remove congestion within a fixed time window. If no congestion occurs in a time window, $\eta_{u_i}^* = 0$. $\epsilon_{u_i}^*$ is derived based on Formula (4) based on $\eta_{u_i}^*$. Given d_{γ_i}, node u_i computes its $\eta_{u_i}^*$ and $\epsilon_{u_i}^*$:

$$\eta_{u_i}^*(t_{\gamma_i}) = \max_{t_{\gamma_i}-w\leq t\leq t_{\gamma_i}} \{\eta_{u_i}^*(t), d_{\gamma_i}\frac{|\mathcal{T}_{u_i}|}{N_i}\} \quad (15)$$

$$\epsilon_{u_i}^* = \sum_{u_{i_k}} (\eta_{u_{i_k}}^* + \epsilon_{u_{i_k}}^* + 2\sqrt{\eta_{u_{i_k}}^* \times \epsilon_{u_{i_k}}^*}) \quad (16)$$

where t_{γ_i} is the current time, w is the time window, and $\eta_{u_{i_k}}^* + \epsilon_{u_{i_k}}^* + 2\sqrt{\eta_{u_{i_k}}^* \times \epsilon_{u_{i_k}}^*}$ is the upper bound of u_{i_k}'s error contribution $c_{u_{i_k}}$ according to Formula (4).

Node u_i asks its parent u to update its η_{u_i} and ϵ_{u_i} to $\eta_{u_i}^*$ and $\epsilon_{u_i}^*$, respectively, such that the desired data compression ratio γ_i can be achieved to avoid congestion. At node u, the parameters (ϵ_{u_i}, η_{u_i}) always need to satisfy Formula (6) in order to ensure $e_u \leq \epsilon_u$, that is,

$$\sum_{u_k:u_k is\ child\ of\ u} (\eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k} \cdot \epsilon_{u_k}}) \leq \epsilon_u \Rightarrow \sum_{u_k} c_{u_k}^m \leq \epsilon_u$$

However, the increase of (η_{u_i}, ϵ_{u_i}) to ($\eta_{u_i}^*$, $\epsilon_{u_i}^*$) may violate Formula (6). Note that though u's other children are assigned (ϵ_{u_k}, η_{u_k}) hence maximum tolerable error contribution ($c_{u_k}^m$), they may generate no or a little error if they experience no or little congestion, i.e., ($\eta_{u_k}^*$, $\epsilon_{u_k}^*$) are 0 or small values. Thus, node u can reduce the $c_{u_k}^m$ of uncongested children and increase the $c_{u_k}^m$ of congested children to satisfy Formula (6). Accordingly, node u first attempts to change (ϵ_{u_k}, η_{u_k}) to ($\eta_{u_i}^*$, $\epsilon_{u_i}^*$) for each of its children. It then calculates its ϵ_u^* based on updated $\eta_{u_k}^*$ and $\epsilon_{u_k}^*$ by Equation (16), and then compare ϵ_u^* and ϵ_u to decide the next step as follows:

(1) If $\epsilon_u^* \leq \epsilon_u$, it means that updating each child u_k's parameters with $\epsilon_{u_k} = \epsilon_{u_k}^*$ and $\eta_{u_k} = \eta_{u_k}^*$ can guarantee Formula (6), because

$$\begin{aligned} &\epsilon_u \geq \epsilon_u^* \\ &= \sum_{u_k:u_k is\ child\ of\ u} (\eta_{u_k}^* + \epsilon_{u_k}^* + 2\sqrt{\eta_{u_k}^* \times \epsilon_{u_k}^*}) \\ &= \sum_{u_k:u_k is\ child\ of\ u} (\eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k} \times \epsilon_{u_k}}). \end{aligned} \quad (17)$$

Therefore, u then updates each child u_k's parameters with $\epsilon_{u_k} = \epsilon_{u_k}^*$ and $\eta_{u_k} = \eta_{u_k}^*$ Consequently, node u_i has $\eta_{u_i} = \eta_{u_i}^*$, allowing the data compression with ratio γ_i at u_i.

(2) If $\epsilon_u^* > \epsilon_u$, it is obvious that the previous updating solution cannot guarantee Formula (6). Thus, node u attempts to update each child u_k's parameters with $\epsilon_{u_k} = \epsilon_{u_k}^*$ and $\eta_{u_k} = \eta_{u_k}^*$, by requesting its parent node u' to assign ϵ_u^* as u's maximum tolerable error (ϵ_u) so that Formula (6) can be satisfied. Node u' updates maximum tolerable distortion and error of its children in the same way of u updating parameters of u's children by considering two cases $\epsilon_{u'}^* \leq \epsilon_{u'}$ and $\epsilon_{u'}^* > \epsilon_{u'}$. If $\epsilon_{u'}^* > \epsilon_{u'}$, u' will further request update from its parent node. This process can repeat along the path towards the sink until reaching either a node that successfully reassigns these parameters for all its children, or the sink. If the request reaches the sink, the sink informs its application of the lower data accuracy than the specified value.

After congested node u's children decrease their compression ratios to reduce their transmission rates, the input data arrival rate to u starts to decrease until it is not congested anymore. Once the congestion is eliminated, node u notifies its children that it is not congested anymore and they can increase their compression ratios. However, in order to avoid oscillation, children do not abruptly increase their compression ratio to 1 (which means no compression). Instead, a node gradually increases its compression ratio by $\gamma_i(t+1) = \gamma_i(t) + \rho$ times, where ρ is a constant value.

E. Adaptivity to Dynamic Network Topology

The setting of maximum tolerable errors and distortions (ϵ_u, η_u) in CADC depends on the topology of the routing

tree. However, since the routing tree can dynamically change because of common failures of nodes and links in WSNs, CADC needs to adaptively adjust the parameters of (ϵ_u, η_u).

To handle the failures of nodes or links, the routing tree is rebuilt, in which some nodes leave a subtree and join in another subtree along with the subtrees rooted at them. Suppose that node u leaves original parent u' and chooses another node u'' as its new parent because the failure of u' or the link between u and u'. CADC lets the setting of (ϵ_{u_k}, η_{u_k}) remain the same for all nodes in u's subtree $\mathcal{T}_u$. In order to have the same maximum tolerable error at node u in the new subtree $\mathcal{T}_{u''}$, based on Equation (6), u'' needs to increase its maximum tolerable error to $\epsilon_{u''} + \epsilon_u + \eta_u + 2\sqrt{\epsilon_u.\eta_u}$. Thus, u'' requests update from its parent, following the same updating procedure in Section IV-D.

V. Performance Evaluation

In this section, we evaluate the performance of CADC in the non-priority and priority cases through simulations in comparison with previous schemes. In particular, we measured the estimation error incurred at the sink, data delivery ratio and the network overhead under different network conditions. Data delivery ratio is measured by the percentage of nodes whose sensor readings are received by the sink (i.e., represented by compressed samples received by the sink). Network overhead is measured by the total number of packet (i.e., data tuple) transmissions of all nodes in a round of data collection, in which each node generates one sensor reading. We compared CADC with the following data collection schemes with congestion control, which do not have maximum error bound at the sink.

(1) *Spatio-Temporal data collection (ST)* [5]. It uses adaptive summarization as a compression scheme to mitigate congestion while aims to minimize the estimation error. Assume node u_k has m data values. The first level summarization uses every two consecutive values to obtain $\frac{m}{2}$ samples. Continuing this process yields k-th summarization, which computes the average of every 2^k consecutive values to obtain $\lceil \frac{m}{2^k} \rceil$ samples.

(2) *Spatio-Temporal data collection with sorted adaptive summarization (ST-SortAdpSum)*. It is a variant of ST with sorting available data at each node before performing adaptive summarization. It is easy to see that under the same data distortion constraint, sorted adaptive summarization leads to fewer samples (i.e. higher compression) and consequently a lower transmission rate.

(3) *ESRT* [1]. It is a rate based congestion control scheme, which mitigates the congestion by adjusting the reporting rate of sensor nodes.

(4) *Pure congestion elimination (PureElimination)*. It is a congestion control scheme, which just uses lossy compression to mitigate congestion. In particular, we let it use the adaptive summarization method to compress data to the extent that can eliminate congestion.

A. Experimental Setup

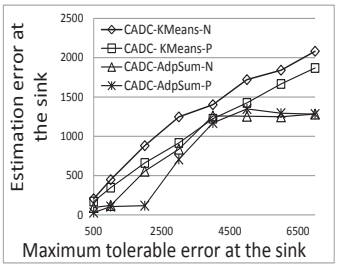
In our simulation, a random routing tree for the WSN was generated. The average number of children for any node was set to 3. The sensor reading of each node was randomly generated, following the Gaussian distribution with mean $\mu = 50$ and variance $\sigma^2 = 5$ [21], [22]. In the priority case, we determine the priority coefficient of value x by the interval x belongs to. The entire range of data value is divided into small ranges $(-\infty, \mu/5)$, $[\mu/5, \mu/4)$, $[\mu/4, \mu/3)$,..., $[\mu/2, \mu/1)$, $[\mu, 2\mu)$, $[2\mu, 3\mu)$, ..., $[4\mu, 5\mu)$, $[5\mu, +\infty)$, each is associated with the corresponding priority coefficient in $\{20, 30, 40, 50, 60, 70, 80,$
$90, 100\}$. We implemented the above four schemes, which operate on the same routing tree in order to perform comparable experiments. We assume there is no non-congestion-induced loss for all links to emulate a reliable wireless medium. The measurement results for each scheme are the average values over 3000 runs.

B. Validity of CADC

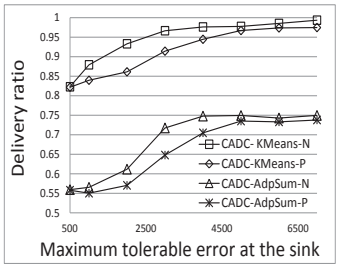
In this test, unless otherwise specified, the network size was set to 20 and the maximum tolerable error at the sink was set to 2000. We varied the maximum tolerable error at the sink from 500 to 1200 with 100 increase in each step. We also evaluated CADC under different network sizes in $\{100, 200, 400, 600, 800, 1000\}$. We used both k-means clustering and adaptive summarization as our compression scheme in CADC, referred to as CADC-Kmeans and CADC-AdpSum, and compared their performance. We tested CADC in both the non-priority and priority cases. In the method names, we use the suffix "-P" for the priority case and use "-N" for the non-priority case.

1) Estimation Error Incurred At the Sink: We first verify the validity of CADC for achieving our primary objective to keep estimation error at the sink below the given assigned maximum tolerable error. Figure 2(a) shows the error incurred at the sink (e_r) versus the maximum tolerable estimation error at the sink (ϵ_r) for different CADC methods. We see that in both the non-priority and priority cases, the error incurred at the sink is lower than the maximum tolerable error. Also, as the maximum tolerable error increases, the incurred error increases. We further observe that in each case, the k-means compression generates higher errors than the adaptive summarization. The reason lies in the fact that k-means achieves higher compression ratio and hence lower transmission rate than the adaptive summarization. Since the samples received at the sink with k-means are less than those with the adaptive summarization, k-means gives higher estimation error. However, this error is still under the maximum tolerable error, and the lower transmission rate and hence energy-efficiency is an evidence of superiority of k-means over adaptive summarization for data compression.

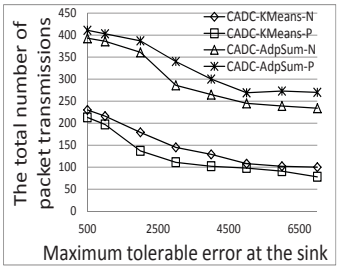
2) Data Delivery Ratio and Network Overhead: Due to network congestion, packets carrying data tuples may be dropped and the sensor readings they represent are lost in the transmission. Figures 2(b) and 2(c) show these two metrics versus the maximum tolerable error at the sink (ϵ_r). We see that as ϵ_r increases, the delivery ratio increases and the number of packet transmissions decreases. With a higher maximum tolerable error at the sink, data is allowed to be compressed with a higher compression ratio, which mitigates the congestion and thus reduces the number of missing sensor readings caused by congestion. A higher compression ratio also reduces the total number of packets transmitted in the network.



(a) Error at the sink



(b) Delivery ratio



(c) Network overhead

Fig. 2: Performance vs. the maximum tolerable error.

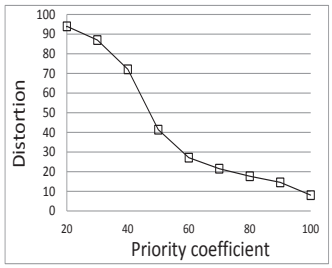
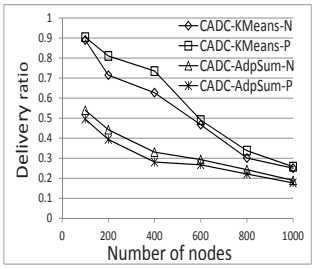
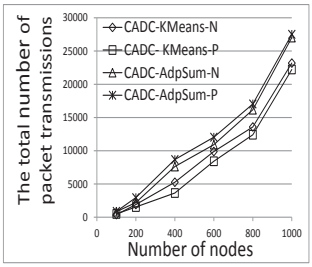


Fig. 3: Distortion vs. priority coefficient.

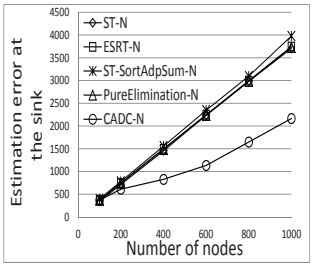


(a) Delivery ratio

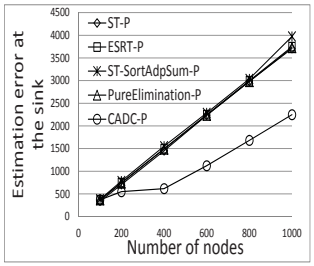


(b) Network overhead

Fig. 4: Performance vs. the number of nodes.



(a) In the non-priority case



(b) In the priority case

Fig. 5: Performance comparison: error at the sink.

From the two figures, we also see that k-means has a higher data delivery ratio and a lower number of total packet transmissions than adaptive summarization in both non-priority and priority cases. This is because k-means can achieve higher compression ratio than adaptive summarization under the same distortion bound. This experimental result supports the superiority of k-means over adaptive summarization.

Figure 4(a) shows that the delivery ratio decreases with the network size. Figure 4(b) shows that the total number of packet transmissions increases with the network size. This is because that a larger network has more data to collect, which leads to more transmissions but also generates a higher probability to cause congestion, leading to delivery ratio decrease. The two figures also show the superiority of CADC with k-means on reducing transmission rate and achieving higher delivery ratio compared to CADC with adaptive summarization.

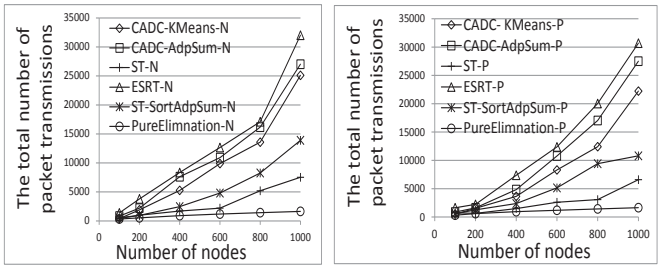
3) Performance in the Priority Case: We then validate that in the priority case, CADC indeed incurs lower distortion to high priority data. We measured the average overall distortion incurred to data with different priorities as shown in Figure 3. We see that the experimental results confirm that data with higher priorities does have less distortion. Recall that when data is transmitted hop by hop along the routing tree from the sensing node to the sink, each forwarding hop may compress the data. After compression in a hop, some data values are changed, and if a value belongs to a different range, its priority coefficient may be changed. In our experiments, most of data has the same priority coefficients at different hops in the forwarding path. This is because the k-means method clusters the most approximate data points, which have same or close priority coefficients. This validates our assumption in Section IV-B2 that the data compression does not change the priority of data in different hops.

C. Performance Comparison

We compared CADC with the ST, ST-SortAdpSum, ESRT, and PureElimination schemes. We varied the number of nodes by $\{100, 200, 400, 600, 800, 1000\}$ and set the maximum tolerable error (ϵ_r) at the sink to 2000. Figures 5(a) and 5(b) show the error incurred at the sink of different schemes in the non-priority and priority cases, respectively. We see that this metric result increases as the number of nodes increases in all schemes. CADC succeeds in constraining the error incurred at the sink below the maximum tolerable error except at the largest network size, which generates high congestion. In this case, the sink alerts the application to increase ϵ_r. The incurred error of CADC is considerably lower than those of other schemes. In other schemes, the incurred error keeps increasing with the number of nodes, and it exceeds the maximum tolerable error when the number of nodes is larger than 500. These schemes produce uncontrollable errors, because they mitigate the congestion without being aware of any error bound requirement at the sink. These experimental results indicate the advantage of CADC in constraining the estimation error below ϵ_r at the sink.

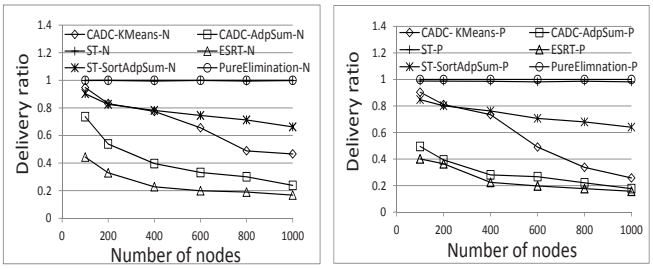
Figures 6(a) and 6(b) show the total number of packet transmissions versus the number of nodes in the non-priority and priority cases, respectively. We see that the total number of packet transmissions increases as the number of nodes increases. The number of transmissions of CADC is lower than that of ESRT because of the data compression in CADC. It is higher than that of ST because the maximum tolerable error in CADC limits the compression degree, while ST does not have estimation error constraint, which results in a high compression ratio. PureElimination has the lowest transmission rate because it eliminates congestion abruptly, while ST mitigates congestion gradually.

Figures 7(a) and 7(b) show the data delivery ratio versus



(a) In the non-priority case (b) In the priority case

Fig. 6: Performance comparison: network overhead.



(a) In the non-priority case (b) In the priority case

Fig. 7: Performance comparison: delivery ratio.

the number of nodes in the non-priority and priority cases, respectively. We see that the delivery ratio in all schemes generally decreases with network size. This is due to the increase in total number of packet transmissions. In both cases, PureElimination, ST and ST-SortAdpSum produce higher delivery ratios than CADC-KMeans and CADC-AdpSum, which have higher delivery ratios than ESRT. These observations can be explained by the experimental results of network overhead (total transmission rate) in Figures 6(b) and 6(a). The total transmission rate has inverse relationship with the delivery ratio. That is, a higher total transmission rate (hence higher network overhead) leads to a higher probability of congestion occurrence and hence lower delivery ratio.

VI. Conclusion and Future Work

In CPS, it is critical to guarantee estimation accuracy of the physical environmental phenomena. Although many congestion control schemes have been proposed to reduce congestion in order to increase estimation accuracy, they also concurrently increase the estimation error due to data sample reduction. Also, none of them can guarantee the data accuracy at the sink. To guarantee the estimation accuracy while control congestion, we presented a Congestion-Adaptive Data Collection scheme (CADC). Based on a given maximum tolerable error bound at the sink, CADC reduces transmission rate of data while keeps the estimation error below the given bound. It novelly uses the k-means clustering algorithm to reduce transmission rate in order to reduce data distortion. CADC also distinguishes data with different importance degrees so that more important data has less distortion, which benefits the accurate environmental phenomena monitoring. Extensive experimental results show the superior performance of our schemes in comparison with previous schemes. In our future work, we will implement CADC and investigate its performance in the real testbed. We will also investigate extending CADC with other types of accuracy measurement, since the CPS applications may have error requirement for the results of specific state estimation functions not limiting to square error over all the data.

Acknowledgements

This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, CNS-1249603, Microsoft Research Faculty Fellowship 8300751.

References

[1] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "Esrt: event-to-sink reliable transport in wireless sensor networks." in *Proc. of MobiHoc*, 2003.

[2] Y. Zhou, M. R. Lyu, J. Liu, and H. Wang, "Port: A price-oriented reliable transport protocol for wireless sensor networks." in *Proc. of ISSRE*, 2005.

[3] J. Paek and R. Govindan, "Rcrt: Rate-controlled reliable transport protocol for wireless sensor networks." *TOSN*, vol. 7, no. 3, 2010.

[4] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, "Coda: congestion detection and avoidance in sensor networks." in *Proc. of SenSys*, 2003.

[5] H. Ahmadi, T. F. Abdelzaher, and I. Gupta, "Congestion control for spatio-temporal data in cyber-physical systems." in *Proc. of ICCPS*, 2010.

[6] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering." in *Proc. of Infocom*, 2004.

[7] A. Silberstein, R. Braynard, and J. Yang, "Constraint chaining: on energy-efficient continuous monitoring in sensor networks." in *Proc. of SIGMOD*, 2006.

[8] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks." in *Proc. of MobiCom*, 2009.

[9] H. Gupta, V. Navda, S. R. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks." *TOSN*, vol. 4, no. 1, 2008.

[10] C. Wang, H. Ma, Y. He, and S. Xiong, "Adaptive approximate data collection for wireless sensor networks." *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 6, pp. 1004–1016, 2012.

[11] J. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks." *TON*, vol. 12, no. 4, pp. 609–619, 2004.

[12] J. Park and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks," *IEEE TOC*, vol. 55, no. 8, pp. 1048–1056, 2006.

[13] H. Lee and A. Keshavarzian, "Towards energy-optimal and reliable data collection via collision-free scheduling in wireless sensor networks," in *IEEE INFOCOM*, 2008.

[14] K. Han, L. Xiang, J.Luo, M. Xiao, and L. Huang, "Energy-Efficient Reliable Data Dissemination in Duty-Cycled Wireless Sensor Networks," in *Proc. of MobiHoc*, 2013.

[15] L. Su, Y. Gao, Y. Yang, and G. Cao, "Towards Optimal Rate Allocation for Data Aggregation in Wireless Sensor Networks," in *Proc. of MobiHoc*, 2011.

[16] P. J. Wan, S. C. H. Huang, L. Wang, Z. Wan, and X. Jia, "Minimum-Latency Aggregation Scheduling in Multihop Wireless Networks," in *Proc. of MobiHoc*, 2009.

[17] F. Bian, S. Rangwala, and R. Govindan, "Quasi-static centralized rate allocation for sensor networks," in *Proc. of SECON*, 2007, pp. 361–370.

[18] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks." in *Proc. of SenSys*, 2004.

[19] S.Madden, M.J.Franklin, and J.Hellerstein, "TAG: a Tiny AGregation Service for Ad-Hoc Sensor Networks," in *Proc. of OSDI*, 2002.

[20] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficientcient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, 2002.

[21] A. Kansal, A. Ramamoorthy, M. B. Srivastava, and G. J. Pottie, "On sensor network lifetime and data distortion," in *Proc. of ISIT*, 2005.

[22] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proc. Of VLDB*, 2004, pp. 588–599.