Data Collection with Accuracy-Aware Congestion Control in Sensor Networks

Yan Zhuang, Lei Yu, Haiying Shen, William Kolodzey, Nematollah Iri, Gregori Caulfield, Shenghua He

Abstract—Data collection is a fundamental and critical function of wireless sensor networks (WSNs) for the cyber-physical systems (CPS) to estimate the state of the physical world. However, unstable network conditions impose significant challenges in guaranteeing the data accuracy that is essential for the reliable estimation of physical states. Without efficiently resolving congestion during data transmission in WSNs, packet loss due to congestion can significantly degrade the data quality. Various congestion control schemes have been proposed to address this issue. Most of them rely on reducing transmitted data samples to eliminate the congestion, which, however, could lead to abysmally high estimation error. In this paper, we analyze the impact of congestion control on the data accuracy and propose a Congestion-Adaptive Data Collection scheme (CADC) to efficiently resolve the congestion under the guarantee of data accuracy. CADC mitigates congestion by adaptive lossy compression while ensuring a given overall data estimation error bound in a distributed manner. Considering that for a CPS application different data items may have different priorities, we also propose a weighted CADC scheme such that the data with higher priority has less distortion. We further adapt CADC to guarantee the accuracy of specific aggregate computations. Extensive simulations demonstrate the effectiveness and efficiency of CADC.

Index Terms—Cyber-physical systems, wireless sensor networks, congestion control, data collection, data accuracy

1 Introduction

Wireless Sensor Networks (WSNs) enable the sensing of physical phenomena in a large scale and have been fundamental infrastructures in cyber-physical systems (CPS), for example, for smart home/building/city [1] and internet of vehicles [2]. The sensor nodes in a WSN sample the physical world, such as ambient sensing signal (e.g, lighting and temperature), and transmit the data to the base station (or controllers) for the further analysis. This data collection task, however, encounters various challenges due to unstable network environment. A WSN typically consists of hundreds to thousands of static or mobile sensor nodes, which generate a tremendous amount of data that need to be delivered to the base station through multi-hop wireless transmission. The large amount of data, low-speed and unstable wireless links and dynamic network topology together can easily cause network congestion for WSNs. The network congestion causes packet loss and thus affect data accuracy and increase the state estimation error of physical world. But in many applications it is critical for controllers to have accurate estimation to make reliable control decisions. Therefore, it is necessary to eliminate the network congestion effectively and efficiently for CPS to guarantee data accuracy.

To address the network congestion, a number of congestion control schemes for WSNs have been proposed. Most schemes [3], [4], [5], [6], [7] [8], [9] either perform rate control to reduce the data generation rate at source nodes or compress the samples at the intermediate relay nodes in a lossy way. By reducing the amount of data to be transferred, these schemes can effectively avoid or mitigate the network congestion. In the meantime, however, the estimation error for the monitored physical state can be largely increased due to the information loss during the congestion control. These congestion control approaches can be paradoxical with regard to the data accuracy since they indeed try to improve the estimation accuracy in a way that degrades data accuracy. Another type of solutions employ redundant network resource to resolve the network congestion [10], [11], [12]. When the congestion happens, the network uses alternative transmission paths that are created by unused/redundant nodes in the network, even at the cost of more transmission hops to the destination. However, the network resource of CPS systems is usually constrained, which in practice may conflict with the assumptions of these solutions, and the data accuracy is not explicitly considered in their approaches. In this paper, we argue that the design of a congestion control scheme should not solely aim at the congestion avoidance and mitigation, but also need to take into account the data accuracy, especially when given the stringent reliability requirement and limited network resources of CPS applications. Otherwise, the data collection can be rendered useless due to accuracy-lossy network congestion control. An ideal congestion control scheme should work around a "sweet spot" that mitigate the congestion while still satisfying the estimation accuracy requirement of applications.

Therefore, a fundamental problem is how the congestion control affects the data accuracy. As we can see from the above discussion, understanding this problem is important and necessary for the design of an efficient congestion control scheme. However, the problem is not examined on

- *Yan Zhuang and Haiying Shen are with department of Computer Science, University of Virginia, Charlottesville, VA, 22903.*
- *Lei Yu is with the School of Computer Science, Georgia Institute of Technology, Atlanta, GA, 30332.*
- *William Kolodzey, Nematollah Iri, Gregori Caulfield and Shenghua He are with Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, 29634.*
- *Corresponding author: Haiying Shen*
  E-mail: yz8bk@virginia.edu, lyu79@gatech.edu, hs6ms@eservices.virginia.edu, {wkolodz, niri, gcaulfi, shenghh}@clemson.edu

Manuscript received April 19, 2017; revised September 17, 2017.

previous works [3], [4], [5], [6], [7] for the congestion control in WSNs. In this paper, we conduct a formal analysis on the impact of congestion control on data accuracy. Our numerical results demonstrate the two-sided effect of congestion control on the data accuracy, manifests its cause, and suggests that the trade-off between congestion mitigation and data accuracy has to be considered for designing a congestion control scheme.

Based on our analysis result, we consider the design of a congestion control scheme that aims to mitigate the congestion while still ensuring required data accuracy by CPS applications. Given that nodes transmit data upwards to the sink through a routing tree [7], we propose a Congestion-Adaptive Data Collection scheme (CADC) with data accuracy guarantee. In CADC, the congestion control is conducted through lossy data compression/aggregation to mitigate the congestion, and a CPS application specifies the upper bound of data estimation error at the sink. During the data collection, a maximum tolerable distortion for the compression and a maximum tolerable error are determined at each node, in order to finally guarantee the given estimation accuracy at the sink. To reduce the data distortion by compression, we propose to use the k-means clustering algorithm to perform lossy data compression within the maximum tolerable distortion bound at the nodes. When congestion occurs, by adaptively adjusting the maximum tolerable distortion and maximum tolerable error allowed at sensor nodes, CADC makes best effort to achieve the required estimation accuracy by the CPS application while mitigating 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 sensitive to high temperature readings, thus the temperature measurements with higher values are more important and should have lower distortion and hence compression degree. To address this issue, we propose to extend CADC with a weighted CADC scheme, which assigns weights to the measurements according to their priorities and aims to minimize the weighted estimation error. Another important issue for the wireless sensor network is its dynamic network topology. Because of the frequent node/link failure in a WSN or node movement if it is a mobile sensor network, the network topology for data dissemination is usually dynamically maintained. Therefore, the adaptivity of congestion control to the dynamic topology changes is an important factor for its performance. We provide a simple but efficient solution for CADC to handle the tree topology changes, which allows CADC to effectively work with both static WSNs and mobile WSNs. Furthermore, we investigate the effectiveness of our solution to a type of aggregate functions over the collected data, since in many scenarios the applications are interested in the error of aggregated results instead of estimation error of overall data. We conduct extensive simulations to evaluate our CADC schemes in comparison with previous schemes. Experimental results demonstrate the high effectiveness and efficiency of CADC.

The rest of paper is organized as follows. Section 2 summarizes the related work. Section 3 defines our system model, analyzes the effects of congestion control on the data accuracy, and introduces our design objective. Section 4 presents our congestion-adaptive data collection schemes in detail. Section 5 presents the performance evaluation of our schemes in comparison with previous methods. Section 6 concludes this paper with remarks on our future work.

2 Related Work

In this section, we present an overview of existing congestion control schemes proposed for WSNs and several works that propose to reduce data transmission rate through data compression in WSNs.

2.1 Congestion Control in WSNs

The control congestion schemes can be classified into two classes: centralized rate control schemes and distributed rate control schemes. Event-to-Sink Reliable Transport (ESRT) [3] 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.* [13] 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.* [4] 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.* [5] 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).

Wan *et al.* [6] proposed a distributed rate control scheme, named CODA, for congestion avoidance that consists of three key mechanisms: receiver-based congestion detection, open-loop hop-by-hop backpressure and closed-loop multisource regulation. Brahma *et al.* [9] proposed a distributed congestion control scheme for WSNs, where the network is assumed to be tree structure. It adjusts the traffic in WSNs by assigning a fair and efficient transmission rate to each node. Specifically, the node itself decides to increase or decrease the transmission rate by "observing" the difference between input traffic and output traffic rate. Sergiou *et al.* [10] proposed a distributed hop-by-hop congestion control algorithm, called Hierarchical Tree Alternative Path (HTAP), that resolves congestion by using alternative sub-optimal transmission paths. Based on HTAP, The authors proposed another similar but more dynamic and lightweight scheme called Dynamic Alternative Path Selection Protocol (DAlPaS) [11] for WSNs. It utilizes a soft-stage technique to let each node serve only one transmission flow to reduce the buffer overflow probability and hence the congestion probability in the network. Aghdam *et al.* [8] developed a cross-layer WSN Congestion Control Protocol (WCCP) for multimedia content transmission in WSNs based on Source Congestion Avoidance Protocol (SCAP) and Receiver Congestion Control Protocol (RCCP). At the source node, SCAP detects the network congestion and avoids the congestion by adjusting the sending rate of source node and transmission distribution of packets; at the intermediate node, RCCP detects the congestion by monitoring the queue length of intermediate node and notifies the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2853159, IEEE Transactions on Mobile Computing

source node. This work considers the traffic characteristics and inter-arrival pattern of packets, but does not consider the data accuracy. Chen *et al.* [12] proposed an exponential weighted priority-based rate control (FEWPBRC) using the fuzzy logical controller. The FEWPBRC scheme adjusts the transmission rate of the children nodes according to the output transmission rate of their parent nodes to meet the QoS requirement while minimizing the network resource consumption. It focuses on the multimedia application and QoS requirement on the packet loss and delay, and thus they consider the priority of traffic class and sensor node location. In constract, our work focuses on the data collection application and data accuracy requirement, and the priority is defined based on the numerical value of sensing data.

The main issue of these existing rate control based schemes [3], [4], [5], [6], [8], [9], [12] is that decreasing data rate reduces the number of spatio-temporal samples but their solutions did not consider the accuracy of the state estimation. In this paper, we consider the congestion issue in data collection and aim to design a congestion-adaptive data collection scheme for WSNs with the goal to guarantee the data accuracy. Our work is most related to [7] proposed by Ahmadi *et al.* that took into account the estimation error in the congestion control. Using least-error summarization, their scheme eliminates congestion while incurring the least possible overall error in sensing the physical environment. However, the scheme in [7] 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.

2.2 Data Compression in WSNs

Our congestion control scheme exploits spatial data correlation to effectively compress data to reduce data transmission rate in WSNs. A lot of previous works have exploited data correlation to compress data to reduce the data transmission cost. Cristescu *et al.* [14] 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.* [15] 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.* [16] proposed to apply compressive sampling theory to sensor data gathering to reduce global scale communication cost. Gupta *et al.* [17] proposed to select a small subset of sensor nodes that may be sufficient to reconstruct data for the entire sensor network within a predefined error bound. Wang *et al.* [18] proposed an approximate data collection, in which the network is partitioned into clusters, and cluster heads construct the local estimation model with pre-specified 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. These works focus on reducing the communication cost and energy consumption of data transmission and do not explicitly consider the network congestion especially data accuracy aware congestion control. But in our paper, we utilize data compression to adjust data transmission rate for resolving network congestion, where the data compression ratio is dynamically adjusted based on our congestion control decisions.

This paper is an extension of our previous conference paper [19]. In addition to new experiment results to demonstrate the protocol overhead for the congestion control, we propose several extensions to our previous work: 1) we conduct a formal analysis of the effects of congestion control on the data accuracy. Based on the queue model, we derive a formal relationship between the data accuracy and the lossy-compression based congestion control, and numerically analyze the change of data accuracy with varying compression efficiency under different loads; 2) we consider how the proposed congestion control approach guarantees the accuracy requirement of aggregate functions, since many applications may just require the computation of some aggregation functions over the collected data. We propose to adapt the proposed CADC to accommodate the accuracy requirement of such aggregate computations.

3 System Model And Objective

3.1 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 [20], [21], 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 $\mathcal{T}_i$).

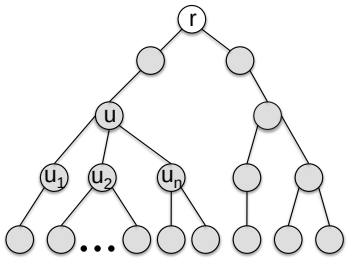


Fig. 1. Routing tree.

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 4.5. 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. We simply assume a reliable wireless medium and a simple CSMA/CA based MAC protocol. Given this system model, the number of raw messages to be delivered to the sink for any subtree is proportional to the size of the subtrees. Congestion occurs at a node when its data transmission rate is lower than the total data arrival rate at it due to insufficient bottleneck resource like egress bandwidth and link availability [7]. Our scheme is agnostic to the nature of the bottleneck resource. One of the main

TABLE 1
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

components in congestion control schemes is congestion detection. For this purpose, we can use a previously proposed congestion detection scheme [7]. 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.

3.2 Motivation and Objective

In this section, we analyze the effects of congestion control on data accuracy and define our congestion control problem for data collection with accuracy requirement.

3.2.1 The Impact of Congestion Control on Data Accuracy

The quality of collected data from wireless sensor network is critical for the controllers to accurately estimate the state of the monitored physical phenomenon. However, congestion control has the two-sided influence on the data accuracy: (1) it can reduce the data loss caused by network congestion and thus improves the estimation accuracy, (2) the ways to mitigate network congestion, such as reducing the source rate at sensor nodes [3], [4], [5], [6], [13] and data aggregation [7], are in the lossy manner and will increase the estimation error.

We validate the two-sided effect of congestion control on data accuracy through a simplified analysis based on $M/M/1/m$ queue model [22]. We assume the queue length is m, service rate is μ and arrival rate is λ. The probability of a packet being dropped, denoted by p_d, is the probability that the queue is full, that is,

$$\begin{aligned} p_d &= \frac{\rho^{m+1}-\rho^m}{\rho^{m+1}-1} \quad \text{if} \quad \rho \neq 1 \\ p_d &= \frac{1}{m+1} \quad \text{if} \quad \rho = 1 \end{aligned} \tag{1}$$

where $\rho = \frac{\lambda}{\mu}$. Each packet only carries one data item.

Consider a set X consisting of n data items $x_1, x_2, \ldots, x_n$ that consecutively arrive at the queue. We consider the data accuracy in two cases: without and with congestion control respectively. For the second case, we assume a congestion control scheme f_{cc} that compresses/aggregates multiple data items to reduce the data arrival rate in a lossy manner. The reconstructed value of a data item x_i is denoted by $f_{cc}(x_i)$. Congestion control through reducing sample rates at sources can be also regarded as this process, where multiple samples obtained at original rate are reduced to one sample.

The packet drop causes missing data items at the receiver. Here, for simplicity, we do not assume any spatial temporal correlation among data and thus do not consider any sophisticated techniques to estimate the missing values. To count the impact of the missing value on the data accuracy, we simply replace the missing data items by the mean of X $A_X = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Given that, the data accuracy is measured by the sum of each item's square error. In the case without congestion control, x_i is either received as it is with probability $1-p_d$ or replaced by A_X with probability p_d. Then, the expected error for x_i, denoted by $Er_{x_i}^{wo}$, and the total expected error Er_X^{wo} without congestion control are computed as follows:

$$\begin{aligned} Er_{x_i}^{wo} &= p_d(x_i - A_X)^2 + (1-p_d)(x_i - x_i)^2 = p_d(x_i - A_X)^2 \\ Er_X^{wo} &= p_d \sum_{i=1}^{n}(x_i - A_X)^2 \end{aligned} \tag{2}$$

The congestion occurs when packet arrival rate exceeds the service rate, i.e., $\lambda > \mu$. The congestion control scheme reduces λ by aggregating multiple data items into one data item. Let λ' be the arrival rate after aggregation and p_d' be the corresponding packet drop probability. A congestion control scheme adjusts λ' by varying compression ratio or aggregation granularity to adaptively mitigate congestion, thus, x_i's accuracy loss is correlated with λ' and we use $f_{cc}^{\lambda'}(x_i)$ to indicate that. Let g_X be a group of multiple data items that are aggregated into one packet. If the packet is received, the total square error of g_X is $\sum_{x_i \in g_x}(x_i - f_{cc}^{\lambda'}(x_i))^2$; if the packet is dropped, it is $\sum_{x_i \in g_X}(x_i - A_X)^2$. Then, the expected total square error for g_X and X with congestion control are as follows:

$$\begin{aligned} Er_{g_X}^{w} &= (1-p_d')\sum_{x_i \in g_x}(x_i - f_{cc}^{\lambda'}(x_i))^2 + p_d' \sum_{x_i \in g_X}(x_i - A_X)^2 \\ Er_X^{w} &= (1-p_d')\sum_{i=1}^{n}(x_i - f_{cc}^{\lambda'}(x_i))^2 + p_d' \sum_{i=1}^{n}(x_i - A_X)^2 \end{aligned} \tag{3}$$

Based on these results, we study the impact of congestion control on data accuracy in a numerical approach. Suppose that X contains $n = 100$ data items randomly generated from normal distribution $N(50, 100)$. Because $f_{cc}(x_i)$ depends on particular compression/aggregation methods, we simplify $|(x_i - f_{cc}^{\lambda'}(x_i)|$ by a lossy ratio $\alpha(1 - \frac{\lambda'}{\lambda})$ ($\lambda' < \lambda$), that is, $|(x_i - f_{cc}(x_i)| = |\alpha(1 - \frac{\lambda'}{\lambda})x_i|$. We use this lossy ratio to simply model the fact that larger reduction on the arrival rate indicates higher compression ratio and higher accuracy loss. $\lambda' = \lambda$ represents no compression and thus no congestion control. α depends on the capability of compression methods. A compression method having less accuracy loss under the same compression ratio has smaller α. We vary α by 0.1, 0.3, and 0.5

A congestion control scheme reduces λ and thus $\rho = \frac{\lambda}{\mu}$ to mitigate the congestion, so we evaluate the expected total square error of X under different ρ. Initially let $\rho =$

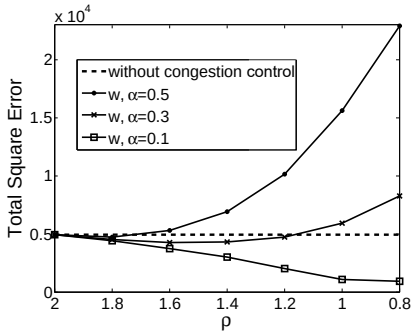


Fig. 2. Congestion control v.s. data accuracy.

2, which indicates a congestion state. Figure 2 shows our numerical results. The horizontal dash line represents the total square error of X when $\rho = 2$ without congestion control. The other three lines shows the effect of congestion control that reduces the data arrival rate at different degrees, represented by $\rho = \lambda/\mu$ from 2 to 0.8. Different α indicates different efficiency for their lossy compression method.

This figure demonstrates the two-sided effect of congestion control. As we can see, the congestion control with the best compression efficiency $\alpha = 0.1$ is able to continuously improve the data accuracy with decreasing ρ, although the congestion actually remains under $\rho > 1$. For the congestion control with the worst compression efficiency $\alpha = 0.5$, it cannot improve the data accuracy at all. The error becomes significantly larger even when the congestion is mitigated under $\rho < 1$. The result for the congestion control with the compression efficiency $\alpha = 0.3$ is more interesting. As we can see, the minimum error occurs around $\rho = 1.6$ and further mitigating congestion to $\rho < 1$ actually increases the error. Therefore, there is a trade-off between resolving congestion and improving data accuracy. The reason for such trade-off is that the compression reduces the data arrival rate in a lossy manner. Different compression efficiency incurs different effects of congestion control on the data accuracy. An efficient congestion control scheme cannot solely depend on compression methods that are expected to achieve high efficiency, even by exploiting spatial temporal correlation among data, since data characteristics varies in different applications and compression efficiency can vary a lot.

Accordingly, the design of a congestion control scheme in data-collection networks needs to handle the trade-off between the data accuracy and the effectiveness of congestion control such that the estimation error resulting from collected data can be constrained into the tolerable range of CPS applications.

3.2.2 Objective

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.,

$$e_u = \sum_{i \in \mathcal{T}_u} (\hat{x}_i^u - x_i)^2. \quad (4)$$

$|\mathcal{T}_u|$ 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 that are sent to its parent u, i.e.,

$$d_{u_k} = \sum_{i \in \mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})^2. \quad (5)$$

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., $e_r \leq \epsilon_r$.

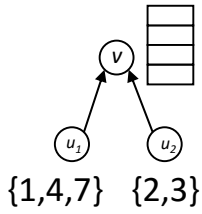
4 Congestion-Adaptive Data Collection with Accuracy Guarantee

In this section, we first provide the overview of our scheme CADC and use an example to explain the idea of its design. Then, we introduce the details of CADC through Section 4.2 to Section 4.4. In Section 4.5 and 4.6, we adapt CADC with the consideration of dynamic network topology and the error bound for aggregate results of sensor data.

4.1 CADC Scheme Overview

Before introducing the proposed CADC scheme, we first simply explain the rationale behind the design of CADC by a toy example in Figure 3.

Figure 3 shows the data transmission from two child nodes u_1 and u_2 to their parent v. We assume a data compression scheme like data summarization [7] is used to avoid the congestion and in such context each data sample to transmit is a tuple $(value, count)$, where the first is the data value and the second is the number of raw sensing readings being summarized and represented by this sample. Suppose that during a time slot node u_1 sends a set of data $\{1, 4, 7\}$ and u_2 sends $\{2, 3\}$ to v, and each data item has $count = 1$. Node v has remaining buffer space to accommodate four samples and no samples are removed from the queue during this time slot. To avoid the congestion as well as the tail drop at v, the child nodes need to reduce their data transmission rates through data compression. As in [7], the data summarization compression here computes the average of consecutive pairs of values. The table in the figure shows two different compression choices: (1) node u_1 summarizes data items (1,1) and (4,1) to (2.5,2); or (2) node u_2 summarizes data items (2,1) and (3,1)



compression	Distortion at u_1	Distortion at u_2	estimation error at v
{(1,1),(4,1)}->(2.5, 2)	4.5	0	4.5
{(2,1),(3,1)}->(2.5, 2)	0	0.5	0.5

Fig. 3. An example to show the rationale behind the design.

to (2.5,2). Both of them can avoid the congestion at node v to the same extent (i.e., the queue length is increased by four samples). However, as shown in the table, they incur different data distortion and thus different estimation error at v. Suppose that a CPS application receives data from node v and has accuracy requirement that is represented by the upper bound to the estimation error. If this bound is no less than 4.5, both of compression choices are acceptable; but the bound is less than 4.5 but no less than 0.5, only the compression at node u_2 satisfies the accuracy requirement.

This example indicates that the congestion control decision has to be aware of data accuracy. Different congestion control solutions that even can mitigate the congestion status at the same level of efficiency may lead to different data accuracy. The congestion control mechanisms that make decisions only based on the congestion status are not suitable for the applications with data accuracy requirements. Therefore, our CADC scheme attempts to enable a node that faces congestion to dynamically guide the rate reduction of the nodes at the lower layers for congestion control according to its data accuracy goal. Suppose that node v has the upper bound 0.5 for the estimation error in Figure 3. Within CADC, v predicts the data distortion allowed at u_1 and u_2 respectively, and based on that, u_1 and u_2 reduce their data transmission rates by data compression to mitigate the congestion at v. With allowing the data distortion 0 and 0.5 at u_1 and u_2 respectively, v can resolve its congestion status and also satisfy its error bound.

To achieve the objective stated above, CADC introduces 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,

$$e_u \leq \epsilon_u,\ d_u \leq \eta_u. \tag{6}$$

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 try to make the error at the sink less than the given error bound ϵ_r. As this error bound is fixed by the application, such case indicates 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 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 5.2.1)?
- How can a node compress its data based on its η_{u_k} while minimizing the data distortion (Section 4.3)?
- How to conduct congestion control and update ϵ_u and η_u to achieve our objective in dynamic network status (Section 4.4)?
- How to adapt to the dynamic network topology (Section 4.5)?
- How to guarantee the accuracy of the aggregate functions with CADC (Section 4.6)?

4.2 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 4.2.1).
- *Priority case*, in which the measurements have different priorities (Section 4.2.2).

4.2.1 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:

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

$$
\begin{aligned}
&= \sum_{i\in\mathcal{T}_{u_k}} ((\hat{x}_i^u - \hat{x}_i^{u_k}) + (\hat{x}_i^{u_k} - x_i))^2 \\
&= \sum_{i\in\mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})^2 + \sum_{i\in\mathcal{T}_{u_k}} (\hat{x}_i^{u_k} - x_i)^2 \\
&+ 2\sum_{i\in\mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})(\hat{x}_i^{u_k} - x_i) \\
&\leq \sum_{i\in\mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})^2 + \sum_{i\in\mathcal{T}_{u_k}} (\hat{x}_i^{u_k} - x_i)^2 \\
&+ 2\sqrt{\sum_{i\in\mathcal{T}_{u_k}} (\hat{x}_i^u - \hat{x}_i^{u_k})^2 . \sum_{i\in\mathcal{T}_{u_k}} (\hat{x}_i^{u_k} - x_i)^2} \\
&= d_{u_k} + e_{u_k} + 2\sqrt{d_{u_k}\cdot e_{u_k}} \qquad (7)
\end{aligned}
$$

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:

$$c_{u_k}^m = \eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k}\cdot\epsilon_{u_k}}. \qquad (8)$$

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

$$\sum_{u_k:u_k is\ child\ of\ u} c_{u_k}^m \leq \epsilon_u \Rightarrow \sum_{u_k}(\eta_{u_k}+\epsilon_{u_k}+2\sqrt{\eta_{u_k}\cdot\epsilon_{u_k}}) \leq \epsilon_u \qquad (9)$$

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 (9) 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 4.4.

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 (9), to guarantee $e_u \leq \epsilon_u$, we let

$$\eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k}\cdot\epsilon_{u_k}} = \alpha_k\epsilon_u \ (0 < \alpha_k < 1) \qquad (10)$$

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

$$e_u \leq \sum_{u_k}(\eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k}\cdot\epsilon_{u_k}}) = \sum_{k=1,\ldots,n} \alpha_k\epsilon_u = \epsilon_u. \qquad (11)$$

We choose α_k by

$$\alpha_k = \frac{|\mathcal{T}_{u_k}|}{\sum_{k=1,\ldots,n}|\mathcal{T}_{u_k}|} \qquad (12)$$

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 (12), we use the same procedure as in [20]. 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 (12), node u needs to determine η_{u_k} and ϵ_{u_k} to satisfy Equation (10). 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 (10), we have $\epsilon_{u_k} = \alpha_k\epsilon_u$. In CADC's congestion control (Section 4.4), 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 $e_r \leq \epsilon_r$.

4.2.2 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.

$$e_u^w = \textstyle\sum_{i\in\mathcal{T}_u} w_i(\hat{x}_i^u - x_i)^2, \qquad (13)$$

$$d_{u_k}^w = \textstyle\sum_{i\in\mathcal{T}_{u_k}} w_i(\hat{x}_i^u - \hat{x}_i^{u_k})^2, \qquad (14)$$

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

$$
\begin{aligned}
c_{u_k}^w &= \sum_{i\in T_{u_k}} w_i((\hat{x}_i^u - \hat{x}_i^{u_k}) + (\hat{x}_i^{u_k} - x_i))^2 \\
&= \sum_{i\in T_{u_k}} w_i(\hat{x}_i^u - \hat{x}_i^{u_k})^2 + \sum_{i\in T_{u_k}} w_i(\hat{x}_i^{u_k} - x_i)^2 \\
&+ 2\sum_{i\in T_{u_k}} w_i(\hat{x}_i^u - \hat{x}_i^{u_k})(\hat{x}_i^{u_k} - x_i)
\end{aligned}
$$

$$\begin{aligned} &\leq \sum_{i\in T_{u_k}} w_i((\hat{x}_i^u - \hat{x}_i^{u_k})^2 + \sum_{i\in T_{u_k}} w_i(\hat{x}_i^{u_k} - x_i)^2 \\ &+ 2.\sqrt{\sum_{i\in T_{u_k}} w_i(\hat{x}_i^u - \hat{x}_i^{u_k})^2 \sum_{i\in T_{u_k}} w_i(\hat{x}_i^{u_k} - x_i)^2} \\ &= d_{u_k}^w + e_{u_k}^w + 2\sqrt{d_{u_k}^w \cdot e_{u_k}^w} \end{aligned} \quad (15)$$

Equation (15) 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 4.3) 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 5. As we can see from Formula (15), 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.

4.3 Data Compression

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) [23] 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

$$C(V) = \sum_{x\in V} (x - c(x))^2, \quad (16)$$

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

$$C(V) = \sum_{x\in V} w(x)(x - c(x))^2, \quad (17)$$

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

In the congestion control (Section 4.4), 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 [23]. 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.

4.4 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. A diagram of our CADC approach is given in Figure 4 and Figure 5 shows the overview of CADC algorithm. CADC involves three procedures: (1) for any node u, the congestion is detected by comparing the data arrival rate r_u^{in} and output transmission rate r_u^o; (2) if the congestion is detected, CADC performs the congestion management to resolve congestion; (3) in the meanwhile, the data accuracy compliance is checked to ensure that the accuracy requirement are not violated. CADC involves distributed coordination between the child nodes and parent node. Basically, for any congested node u, u asks its children to reduce their transmission rate by a certain factor and at the same time checks the data distortion due to compression and adjust the tolerable error bound when necessary to guarantee the accuracy requirement at the base station. The details are presented in the following.

4.4.1 Congestion Control Algorithm

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} ($i = 1, ..., n$) for each child respectively. When node u is congested, u attempts to reduce its input data arrival rate r_u^{in} to less than its output transmission rate r_u^o to avoid the congestion, by asking its children to reduce their transmission rates through data compression. 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. When r_u^{in} becomes less than r_u^o, node u's buffer size will decrease and the congestion can be resolved finally.

The data compression ratio is defined as the ratio of the number of compressed samples to the number of available data tuples to be transferred. The data compression ratio is 1

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2853159, IEEE Transactions on Mobile Computing

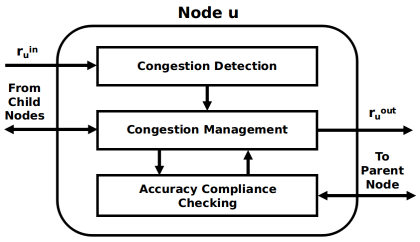


Fig. 4. The diagram of CADC

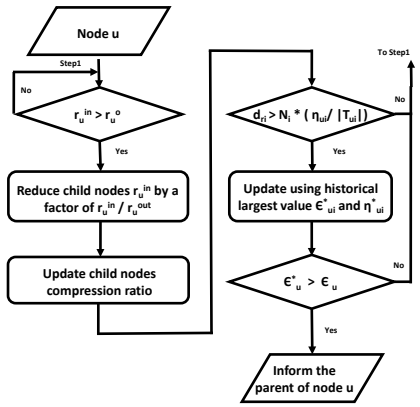


Fig. 5. CADC congestion control algorithm.

if the node sends data without compression. After receiving the congestion notification from u, each child decreases its current data compression ratio by a factor of r_u^{in}/r_u^o. Suppose the set of all available data tuples to be forwarded at node u_i is $< (v_1, n_1), \ldots, (v_i, n_i), \ldots, (v_m, n_m) >$ (following the notation in Section 4.3). Let $N_i = \sum_{j=1}^{m} n_j$, which denotes the total number of sensor readings represented by this set. Node u_i compresses data in compress ratio γ_i by using the k-means clustering with $k = N_i\gamma_i$ (as shown in Section 4.3). The data compression with γ_i will cause data distortion (denoted by d_{γ_i}) that can be calculated by Formulas (16) and (17) in the non-priority and priority cases, respectively. Recall that to ensure the estimator error not larger than the maximum tolerable error at the sink, i.e., $e_r \leq \epsilon_r$, the node u_i needs to ensure maximum tolerable distortion η_{u_i} that is defined over all data readings generated by the sensor nodes in its subtree $\mathcal{T}_{u_i}$. To this end, node u_i needs to compress data with distortion not exceeding $N_i \frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$. This bound is derived as follows: η_{u_i} is the maximum tolerable distortion for all sensor data from the subtree $\mathcal{T}_{u_i}$, and accordingly $\frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$ is the average maximum distortion allowed for each sensor reading from nodes in the subtree; Since the set of data to be forwarded represents N_i readings from the subtree, the total distortion allowed for this set should be $N_i \frac{\eta_{u_i}}{|\mathcal{T}_{u_i}|}$. 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.

4.4.2 Congestion parameter update

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 4.3). 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., to 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. Thus, we let each node maintain 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 (7) based on $\eta^*_{u_i}$. Given d_{γ_i}, node u_i computes its $\eta^*_{u_i}$ and $\epsilon^*_{u_i}$ as follows:

$$\eta^*_{u_i}(t_{\gamma_i}) = \max_{t_{\gamma_i}-w \leq t \leq t_{\gamma_i}} \{\eta^*_{u_i}(t), d_{\gamma_i} \frac{|\mathcal{T}_{u_i}|}{N_i}\} \quad (18)$$

$$\epsilon^*_{u_i} = \sum_{u_{i_k}} (\eta^*_{u_{i_k}} + \epsilon^*_{u_{i_k}} + 2\sqrt{\eta^*_{u_{i_k}} \times \epsilon^*_{u_{i_k}}}) \quad (19)$$

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 (7).

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 (9) in order to ensure $e_u \leq \epsilon_u$, that is,

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

However, the increase of (η_{u_i}, ϵ_{u_i}) to ($\eta^*_{u_i}$, $\epsilon^*_{u_i}$) may violate Formula (9). Note that though u's other children are assigned (ϵ_{u_k}, η_{u_k}) hence maximum tolerable error contribution ($c^m_{u_k}$), 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 $c^m_{u_k}$ of uncongested children and increase the $c^m_{u_k}$ of congested children to satisfy Formula (9). 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 (19), 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 (9), because

$$\begin{aligned} \epsilon_u &\geq \epsilon^*_u \\ &= \sum_{u_k : u_k is\ child\ of\ u} (\eta^*_{u_k} + \epsilon^*_{u_k} + 2\sqrt{\eta^*_{u_k} \times \epsilon^*_{u_k}}) \\ &= \sum_{u_k : u_k is\ child\ of\ u} (\eta_{u_k} + \epsilon_{u_k} + 2\sqrt{\eta_{u_k} \times \epsilon_{u_k}}). \end{aligned} \quad (20)$$

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 (9). 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 (9) 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 can gradually increase its compression ratio by $\gamma_i(t+1) = \gamma_i(t) + \rho$ times, where ρ is a constant value.

4.5 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 (9), 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 4.4.

4.6 Accuracy for Aggregate Functions

In CADC, we measure the estimation error by the sum of square error over all the data items as represented by Formula (3.1). The CPS applications need to specify the maximum tolerable estimation error over the whole data set where each data is from a sensor node. However, instead of the total square error over the sensor measurements received at the sink, many applications may be more interested in the accuracy guarantee of computation results of some types of functions, like sum, average and maximum, which are computed over all the data. We refer to such a function $f : S \to \mathbb{R}$ that is computed over a set of data S and the value is a real number, as an instance of aggregate function. This means that the definition of the maximum tolerable estimation error and data distortion in CADC can be adapted to address the accuracy requirement of the computation of aggregate functions over the collected sensor data.

In this section, we adapt CADC to guarantee the data accuracy for the computation results of a specific type of aggregate functions which we call them *linear decomposable functions*. We assume the domain of a sensor measurement is $\mathcal{X}$. Let $f_n : \mathcal{X}^n \to \mathcal{Y}$ be the function of interest, where n is the number of sensor nodes in the WSN, and $\mathcal{Y}$ is the domain of output; for our application we can assume it is $\mathbb{R}$. We use $f(\cdot)$ instead of $f_n(\cdot)$ for simplicity. Denote $[n] = \{1, ..., n\}$ and let $S = \{i_1, ..., i_k\} \subset [n]$ where $i_1 < i_2 < ... < i_k$. Given $x \in \mathcal{X}^n$, we denote $x_S = [x_{i_1}, x_{i_2}, ..., x_{i_k}]$ where each x_i is a data sample from sensor node i.

Definition 4.1. A function $f : \mathcal{X}^n \to \mathcal{Y}$ is *linearly decomposable* if there exist coefficients $a_i \in \mathbb{R}$ ($1 \leq i \leq k$) such that for any $x \in \mathcal{X}^n$ and partition $\Pi(S) = \{S_1, ..., S_k\}$ of $S \subset [n]$ we have:

$$f(x_S) = a_1 f(x_{S_1}) + a_2 f(x_{S_2}) + ... + a_k f(x_{S_k})$$

Example: Average of measurements is linearly decomposable with $f(x_{S_i}) = \sum_{j \in S_i} \frac{x_j}{|S_i|}$ and $a_i = \frac{|S_i|}{\sum_i |S_i|}$.

It can be shown that the bound for the sum of square error $\sum_{i=1}^n (x_i - \hat{x}_i)^2$ cannot guarantee the accuracy of such type of aggregation functions. Consider $f(x_S) = \sum_{i=1}^n a_i f(x_i)$, $\sum_{i=1}^n (x_i - \hat{x}_i)^2$ cannot provide an error bound for $|f(x_S) - f(\hat{x}_S)|^2 = (\sum_{i=1}^n a_i (f(x_i) - f(\hat{x}_i)))^2$. In order to bound the error, we use the following Inequality

$$b_1^2 + b_2^2 + \cdots + b_n^2 \geq \frac{(b_1 + b_2 + \cdots + b_n)^2}{n}$$

and we can get

$$\left(\sum_{i=1}^n a_i (f(x_i) - f(\hat{x}_i)) \right)^2 \leq n \left(\sum_{i=1}^n a_i^2 (f(x_i) - f(\hat{x}_i))^2 \right)$$

Based on that, suppose $\sum_{i=1}^n a_i^2 (f(x_i) - f(\hat{x}_i))^2 \leq \epsilon$, then $|f(x_S) - f(\hat{x}_S)|^2 \leq n\epsilon$. Therefore, CADC can be easily adapted to work with data accuracy requirement for such type of aggregation functions, by applying $f(\cdot)$ to sensor sample x_i first and replacing (4) and (5) with

$$e_u = \sum_{i \in \mathcal{T}_u} a_i^2 (f(\hat{x}_i^u) - f(x_i))^2. \tag{21}$$

$$d_{u_k} = \sum_{i \in \mathcal{T}_{u_k}} a_i^2 (f(\hat{x}_i^u) - f(\hat{x}_i^{u_k}))^2. \tag{22}$$

We can go through the same procedure as in Section 4.2 with replacing the data sample x_i or its estimation $\hat{x}_i$ with $a_i f(x_i)$ and $a_i f(\hat{x}_i)$ respectively. For data compression at a node, we use the same compression method in Section 4.3 but over the data $a_i f(x_i)$. With setting the maximum tolerable error ϵ for e_r at the sink (defined by (21)) in CADC, the CPS application can have the accuracy guarantee for the aggregation function, i.e., $|f(x_{\mathcal{T}_r}) - f(\hat{x}_{\mathcal{T}_r})| \leq \sqrt{N\epsilon}$ where $\mathcal{T}_r$ represents the set of all sensor nodes and N is the total number of sensor nodes. This means that CADC

can be guided by the accuracy requirements of different computation tasks at the sink.

5 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)* [7]. 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* [3]. 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.

5.1 Experimental Setup

We implemented the above four schemes in the simulation, which operate on the same routing tree in order to perform comparable experiments. The simulation constructs a random routing tree for the WSN with average 4 children for each node. The size of Rx and Tx buffer for senor nodes are 15 and 10 data samples respectively. The sensor reading for each node is randomly generated following a Gaussian distribution [24], [25] with mean $\mu = 50$ and variance $\sigma^2 = 5$. In the priority case, the value of sensor readings decides the priority coefficient. The entire range of sensor readings is divided into several ranges, e.g., $(-\infty, \mu/5)$, $[\mu/5, \mu/4)$, $[\mu/4, \mu/3)$,..., $[\mu/2, \mu/1)$, $[\mu, 2\mu)$, $[2\mu, 3\mu)$, ..., $[4\mu, 5\mu)$, $[5\mu, +\infty)$. Each is associated with a priority coefficient in $\{20, 30, 40, 50, 60, 70, 80, 90, 100\}$ respectively. We assume that there is no non-congestion-induced loss for all links to emulate a reliable wireless medium and CSMA/CA. The measurement results for each scheme are the average values over 100 runs.

5.2 Validity of CADC

In this simulation, the network size is set to 800. At the default, the maximum tolerable error at the sink is set to 2000; otherwise, it varies from 500 to 6500 with 500 increase in each step. The time window size w in (18) is set as constant 30 for simplicity, because that when we vary the network size, we actually affect the frequency of network congestion and equivalently changes the number of historical records that a window size can hold. Besides, the CADC scheme is evaluated across various network sizes in $\{100, 200, 400, 600, 800, 1000\}$. In the evaluation, the k-means clustering and adaptive summarization method are used as the compression schemes, which are referred as CADC-k-means and CADC-AdpSum, respectively. Moreover, CADC is also evaluated in both the non-priority (using the suffix '-N') and priority cases (using the suffix '-P').

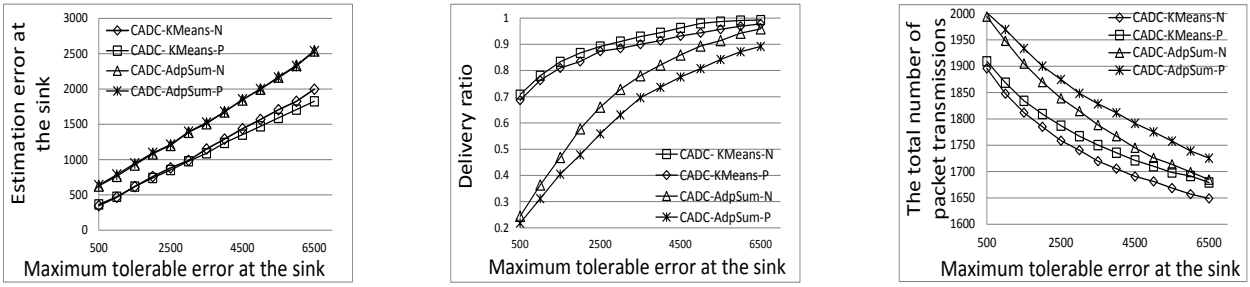
5.2.1 Estimation Error Incurred At the Sink

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 6(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. Figure 6(a) also demonstrates that in either priority case or non-priority case, the k-means compression method incurs less error compared with the adaptive summarization method. The reason of k-means is that for a given compression ratio, k-means scheme finds the best representative data points among the data set by clustering, which leads to less information loss but at higher computation cost. Note for each data point in the figure that is the average of 100 experiments, its standard deviation ranges from 14.4 to 61.8 which mostly is two orders of magnitude smaller than the corresponding data point.

5.2.2 Data Delivery Ratio and Network Overhead

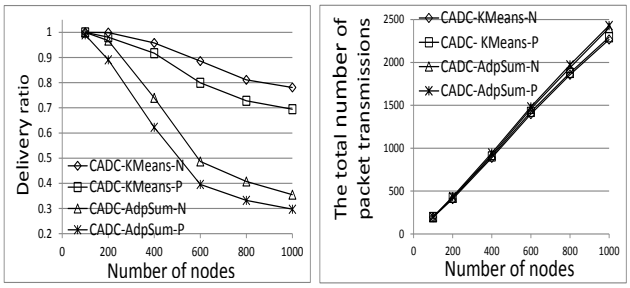
Due to network congestion, packets carrying data tuples may be dropped and the sensor readings they represent are lost in the transmission. The effectiveness of congestion control is indicated by the delivery ratio of sensor readings. A sensor reading is delivered as long as it can be represented by the data received at the sink. We are also interested in the total number of packets that are actually transmitted in the network, indicating the network overhead of data collection. Figure 6(b) depicts the relationship between data delivery ratio and the maximum tolerable error at the sink (ϵ_r). The increasing ϵ_r indicates the higher compression ratio for each node in CADC scheme, which mitigates the congestion and thus reduces the number of missing sensor readings caused by congestion. Therefore, data delivery ratio goes up gradually with ϵ_r. Figure 6(c) shows that the network overhead descreases with ϵ_r ,which is because that higher compression ratio leads to smaller number of total packets transmitted in the network. In terms of priority and non-priority cases, as shown in the two figures, the priority case has less delivery ratio and higher network overhead than non-priority case, because more data packets are transmitted in the priority case under weighted estimation error

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2853159, IEEE Transactions on Mobile Computing



(a) Error at the sink (b) Delivery ratio (c) Network overhead

Fig. 6. Performance vs. the maximum tolerable error.



(a) Delivery ratio (b) Network overhead

Fig. 7. Performance vs. the number of nodes.

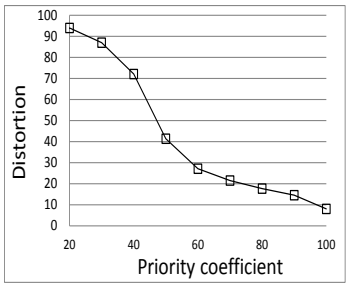


Fig. 8. Distortion vs. priority coefficient.

given the same maximum tolerable error and thus more data packets are dropped due to congestion. Furthermore, Figure 6(b) and 6(c) also demonstrate that k-means has a higher data delivery ratio and a lower network overhead than adaptive summarization in both non-priority and priority cases. This is because k-means is able to achieve higher compression ratio than adaptive summarization under the same distortion bound.

Figure 7(a) and 7(b) show the performance of CADC in term of delivery ratio and network overhead under different network sizes. The network size configuration changes from 100 to 1000. With the increasing network size, the number of data packet to be transmitted also increases, leading to more transmission overhead inside the network, as shown in Figure 7(b). Compared with a small-scale network, a larger network has the higher probability to incur the congestion due to the large amount of packet transmission, and thus has lower delivery ratio, as indicated in Figure 7(a). We can also observe that compared with CADC-AdpSum scheme, the CADC-k-means has superior performance on reducing network overhead and achieving higher delivery ratio. Also, the performance under non-priority and priority setting has the similar trend to Figure 6(b) and 6(c): non-priority case has higher delivery ratio and lower network overhead than corresponding priority case, due to the same discussed before.

5.2.3 Performance in the Priority Case

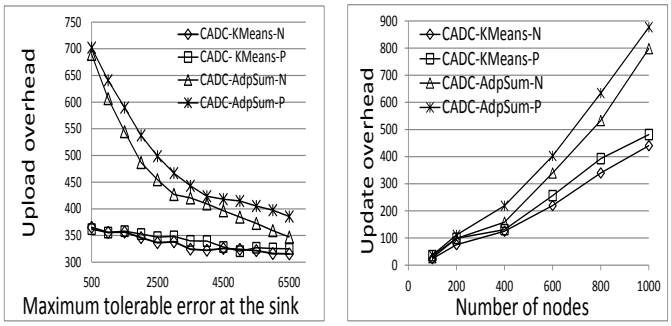
We then validate that in the priority case, CADC indeed incurs lower distortion to high priority data. We measured the average overall distortion incurred to data with different priorities as shown in Figure 8. We see that the experimental results confirm that data with higher priorities does have less 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 4.2.2 that the data compression does not change the priority of data in different hops.

5.2.4 Update Overhead

In CADC, the adaptive adjustments of parameters for congestion control (i.e., the maximum tolerable error and distortion) incur the communication cost between the parent nodes and the child nodes. As described in Section 4.4, when the compression ratio at child nodes required for congestion elimination cannot satisfy the associated distortion constraint, the child nodes request the parameter updates and the parent nodes send the updated values to them. The updates increase the network communication cost, which may interfere the data collection task and degrade the efficiency of congestion control especially when the updates occur frequently. It is expected that such update overhead can be as small as possible for CADC. In this section we measure the update overhead by the number of messages used for parameters update during the congestion control.

Figure 9(a) demonstrates the update overhead under the network size of 800 in different configurations of the maximum tolerable error at the sink (ϵ_r). When the ϵ_r becomes larger, the update overhead for all compress schemes decreases correspondingly. Because the larger ϵ_r allows a higher degree of data compression at each node, the frequent parameters update for each node can be avoided during the congestion control. Therefore the upload overhead for both k-means and adaptive summarization goes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2018.2853159, IEEE Transactions on Mobile Computing



(a) Protocol overhead v.s. Maximum tolerable error (b) Protocol overhead v.s. Network size

Fig. 9. The update overhead of CADC.

down gradually. However, there is a larger upload overhead reduce for adaptive summarization scheme compared with k-means scheme. The reason is that the adaptive summarization scheme generates larger estimation error for a given distortion error bound in a single round of compression. The CADC scheme updates the compression parameters more frequently for AdpSum scheme to find the suitable compression ratio that can meet the error bound at each node, this intensive parameters update activities produce large upload overhead for Adpsum method when ϵ_r is small. In Figure 9(b), unlike the previous evaluation, we measure the update overhead across different network sizes with a fixed ϵ_r of 2000. The overall trend is that the update overheads grows with the the increasing network size, because the larger network size makes it higher probability to cause congestion, leading to more frequent parameters update and higher update overheads in the network. These experiment results indicate that CADC incurs small protocol overhead for congestion control during the data collection. In this update overhead evaluation, the priority cases both for CADC-AdpSum and the CADC-k-means has higher update overhead because of the frequent parameters update. That is, when the priority schemes produce more errors than non-priority cases in a single round of compression, the CADC method has to try different parameters update in order to meet the error bound.

5.2.5 Accuracy for Aggregate Function

In Section 4.6, we propose the adaption of CADC to the accuracy requirement of aggregate functions at the sink instead of the sum of the square error for all sensor data. To evaluate the effectiveness of such adaption, we choose average as the aggregate function and measure its estimation error at the sink in non-priority case. Since the average is computed over the data from all sensor nodes, for each senor data x_i, we have $a_i = \frac{1}{N}$ where N is the network size, and it is used for Formula (21) and (22). The estimation error of the average function is measured by

$$e_{avg} = \sum_{i \in \mathcal{T}_r} |\hat{x}_i - x_i|/N \tag{23}$$

Accordingly, the maximum tolerable error at the sink is set to be a desired upper bound of e_{avg}, instead of the bound of total squared error of all sensor readings given in Definition 3.1. We measure the estimation error with different maximum tolerable error for the average function at the sink, and the results are shown in Figure 10 where

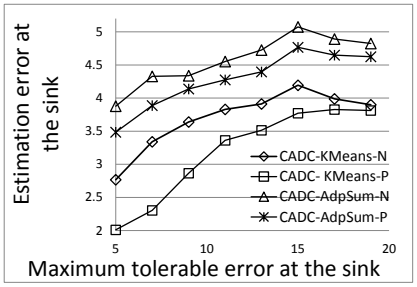
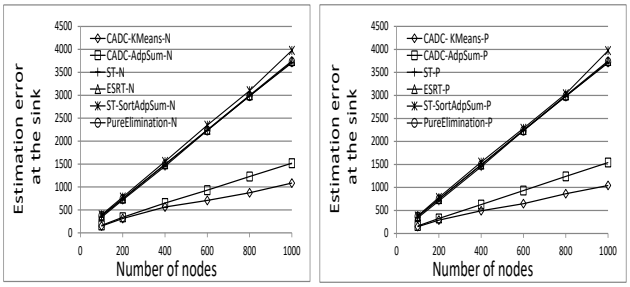


Fig. 10. The estimation error of average aggregate function.



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

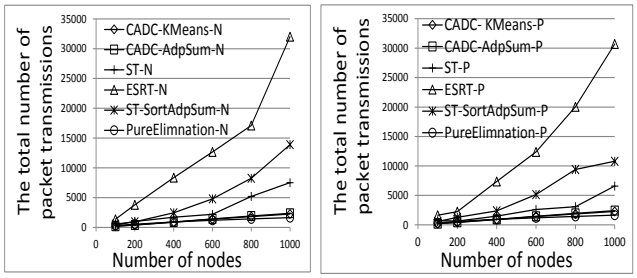
Fig. 11. Performance comparison: error at the sink.

the maximum tolerable error for average aggregation ranges from 5 to 20. As we can see, CADC is also able to achieve the accuracy requirement of average aggregate function, which validates the effectiveness of the adaption proposed for aggregate functions in Section 4.6.

5.3 Performance Comparison

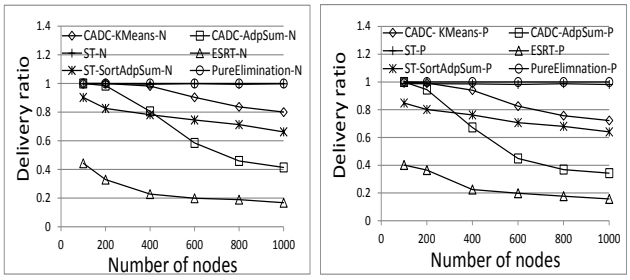
We compared CADC with the ST, ST-SortAdpSum, ESRT, and PureElimination schemes. We varied the number of nodes by $\{100, 200, 400, 600, 800, 1000\}$ and set the maximum tolerable error (ϵ_r) at the sink to 2500. Figures 11(a) and 11(b) show the performance of CADC scheme and other schemes in term of estimation error at the sink in the non-priority and priority cases, respectively. For all of the schemes, the estimation error increases when the network size grows. The CADC-k-mean achieves the smallest estimation error, which is in the range of ϵ_r. The other non-CADC schemes are unable to restrain the estimation error within the range of ϵ_r as they mitigate the congestion without considering any error bound requirement at the sink. We also notice that the CADC-AdpSum incurs larger error than CADC-k-means scheme. The reason is that when the ϵ_r is fixed and the number of nodes is increasing, the CADC-AdpSum is unable to compress the data effectively like CADC-k-means method, which cause higher packet drop and information loss, leading to the larger estimation error at the sink.

Figure 12(a) and 12(b) compare the results between CADC and other methods in term of network overhead across various network sizes in non-priority and priority cases respectively. It is observed that the total number of packet transmissions increases with the number of nodes. The number of packet transmissions of CADC is much lower than that of the other schemes because of the higher data compression in CADC. The PureElimination scheme



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

Fig. 12. Performance comparison: network overhead.



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

Fig. 13. Performance comparison: delivery ratio.

has lowest network overhead because it simply eliminates congestion abruptly.

Figure 13(a) and 13(b) compare the results between CADC with other methods in term of data delivery ratio across various network sizes in non-priority and priority cases. With the increasing number of nodes in the network, the delivery ratio goes down. This is because that as the network size grows, the number of data packet generated from the node also increases, leading to more serious network congestion and dropped packets. In both priority and non-priority cases, the PureElimination and ST have the higher delivery ratios, because they both greatly reduce the number of transmissions by either compressing the packets to half number or eliminating the extra packets directly, which leads to higher delivery ratios but at the same time causes higher estimation errors. The CADC-k-means has higher deliver ratio than ST-SortAdpSum and ESRT. The deliver ratio of CADC-AdpSum becomes worse than CADC-k-means when the network size increases beyond 400. This is because when ϵ_r is fixed and the number of nodes grows, k-means can achieve better compression ratio, leading to higher data delivery ratio than the AdpSum compression.

6 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. In this paper, we formally analyze the impact of congestion control on the data accuracy. Our analysis results demonstrate the two-sided effect of congestion control on the data accuracy and the trade-off between resolving congestion and improving data accuracy. To guarantee the estimation accuracy while controling congestion, we presented a Congestion-Adaptive Data Collection scheme (CADC) with data accuracy guarantee. 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 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. Moreover, CADC is extended with considering the dynamic network topology and the application of aggregate functions in WSNs. 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. Another important factor for congestion control is the underlying MAC protocol, which we did not examine much in this paper. It is worth to note the fact that the level of congestion in a sensor network is a function of the underlying MAC protocol. We will further investigate how CADC works with different behaviors of the MAC protocol and how to further optimize CADC with considering the MAC contention. Besides, we note in this paper we only measure the data accuracy loss due to congestion. We do not consider the accuracy loss due to the link loss in WSNs. Thus, it is worth to analyze the impact of packet loss on the data accuracy and incorporate the data retransmission into the data collection. However, the data retransmission may occur extra bandwidth overhead and aggravate the network congestion. With these considerations, we will analyze the impact of data retransmission on the data accuracy and aim to find the optimal data transmission solution to improve the data accuracy while minimizing the adverse affects like the network congestion.

Acknowledgements

This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, CNS-1249603, and Microsoft Research Faculty Fellowship 8300751. An early version of this work was presented in the Proceedings of IEEE SECON 2015 [19].

References

[1] K. Nellore and G. P. Hancke, "A survey on urban traffic management system using wireless sensor networks," *Sensors*, vol. 16, no. 2, p. 157, 2016.

[2] L. Yan, H. Shen, and K. Chen, "Mobit: A distributed and congestion-resilient trajectory based routing algorithm for vehicular delay tolerant networks," in *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*. ACM, 2017, pp. 209–214.

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

[4] 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.

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

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

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

[8] S. M. Aghdam, M. Khansari, H. R. Rabiee, and M. Salehi, "Wccp: A congestion control protocol for wireless multimedia communication in sensor networks," *Ad Hoc Networks*, vol. 13, pp. 516–534, 2014.

[9] S. Brahma, M. Chatterjee, K. Kwiat, and P. K. Varshney, "Traffic management in wireless sensor networks: Decoupling congestion control and fairness," *Computer Communications*, vol. 35, no. 6, pp. 670–681, 2012.

[10] C. Sergiou, V. Vassiliou, and A. Paphitis, "Hierarchical tree alternative path (htap) algorithm for congestion control in wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 257–272, 2013.

[11] ——, "Congestion control in wireless sensor networks through dynamic alternative path selection," *Comput. Netw.*, vol. 75, no. PA, pp. 226–238, Dec. 2014.

[12] Y.-L. Chen and H.-P. Lai, "Priority-based transmission rate control with a fuzzy logical controller in wireless multimedia sensor networks," *Computers & Mathematics with Applications*, vol. 64, no. 5, pp. 688–698, 2012.

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

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

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

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

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

[18] 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.

[19] N. Iri, L. Yu, H. Shen, and G. Caulfield, "Congestion-adaptive data collection with accuracy guarantee in cyber-physical systems," in *Proc. of IEEE SECON*, 2015.

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

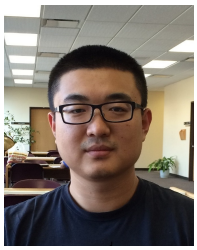
[21] S.Madden, M.J.Franklin, and J.Hellerstein, "TAG: a Tiny AGregation Service for Ad-Hoc Sensor Networks," in *Proc. of OSDI*, 2002.

[22] A. O. Allen, *Probability, Statistics, and Queueing Theory with Computer Science Applications*. San Diego, CA, USA: Academic Press Professional, Inc., 1990.

[23] 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.

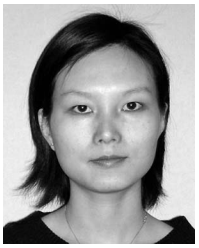
[24] A. Kansal, A. Ramamoorthy, M. B. Srivastava, and G. J. Pottie, "On sensor network lifetime and data distortion," in *Proc. of ISIT*, 2005.

[25] 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.



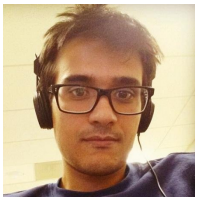
Yan Zhuang is a Ph.D. student in Computer Engineering at University of Virginia, Charlottesville, VA, USA. He received the B.S. degree from Tianjin Polytechnic University, Tianjin, China, and the M.S. degree from the State University of New York (SUNY) at Buffalo, Buffalo, NY, USA, in 2011 and 2014, respectively. His interests are cyber-physical systems and body sensor networks.

Lei Yu is a Ph.D student in Computer Science in Georgia Institute of Technology. He received the BS and MS degree in Computer Science from Harbin Institute of Technology, China in 2004 and 2006, respectively. His research interests include sensor networks, wireless networks, cloud computing and network security.



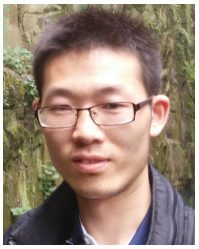
Haiying Shen is an associate professor in the Department of Computer Science at University of Virginia, Charlottesville, VA, USA. Her research interests include distributed computer systems and computer networks, with an emphasis on peer-to-peer and content delivery networks, mobile computing, wireless sensor networks, and grid and cloud computing. She was the Program Co-Chair for a number of international conferences and member of the Program Committees of many leading conferences. She is a Microsoft Faculty Fellow of 2010, a senior member of the IEEE and a member of the ACM. She received the BS degree in Computer Science and Engineering from Tongji University, China in 2000, and the MS and Ph.D. degrees in Computer Engineering from Wayne State University in 2004 and 2006, respectively.

William Kolodzey received a B.S.E degree in Engineering Physics from the University of Michigan in 2009. He is currently pursuing a Ph.D. degree with the Department of Electrical and Computer Engineering at Clemson University. His research interests include machine learning, wireless networks, and data accuracy in large sensor networks.



Nematollah Iri was a Ph.D student in Clemson University, SC, United States during this work.

Gregori Caulfield received the BS degree in Computer Science from Clemson University, SC, United States in 2015. He is currently working in the field of Software Development. His research interests include wireless networks and cloud computing.



Shenghua He received the BS degree in Electronic Science and Technology from Wuhan University of Technology, China in 2012, and the M.S. degree in Electronics and Communication Engineering from Beijing University of Posts and Telecommunications, China in 2015. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering at Clemson University, SC, United States. His research interests include cloud computing, data center networks and mobile computing.