On the Effects of Consistency in Data Operations in Wireless Sensor Networks

Kewei Sha and Weisong Shi
Wayne State University
{*kewei, weisong*}*@wayne.edu*

Abstract

In battery powered systems such as wireless sensor networks, energy efficiency is one of the most important system design goals. In this paper, energy efficiency is examined from the perspective of data consistency, which includes both temporal consistency *and* numerical consistency*, and considers* the application specific requirements of the data *and* data dynamics in the data field*. We first formally define the energy-efficiency problem with the goal of delivering a minimum number of messages under the constraint of data consistency. Then, we give the formal definition of the data consistency. To achieve both consistency and energy efficiency, we propose a data collection protocol named* Alep*, which adapts the data sampling rate to the data dynamics in the data field and keeps lazy when the data consistency is maintained. From the results of a comprehensive simulation we find that the proposed approach indeed reduces the number of delivered messages by more than* 20%*, and improves the accuracy of the sampled data.*

1 Introduction

With the development of technologies in micro-sensor and wireless communication, wireless sensor networks (WSN) have become a very hot research field in last five years [2]. Micro sensors such as Motes are developed to make WSN applications possible; TinyOS [3, 5] is designed to provide system support for operating sensors; and lots of efficient protocols are proposed to make the sensor system workable. Thus, Applications such as habitat monitoring [15], and environment sampling [1] have been launched, showing the promise of wide deployments of WSN.

Because of the special characteristics of WSN such as limited power supply, restricted computing and storage capability, previous efforts in WSN are mainly focus on designing an energy efficient sensor system. These approaches including [7, 11, 16, 17], achieve energy efficiency by taking energy-efficient paths or increasing the sleep time of sensors. Several recent work from database filed tries to achieve energy-efficient by adapting the sample rate [4, 7, 8] and filtering unnecessary sampled data [10, 14]; however, a model to measure the quality of collected data is missed in their work. We argue that a model for the quality of collected data, such as a data consistency model, is essential in WSN applications. Furthermore, the energy efficiency problem should be revisited by considering data consistency.

We model data consistency, including *temporal consistency* and *numerical consistency*, based on two factors, *specific application requirements to the sampled data* and *the feature of data dynamics* in the sensor field, and we examine the effect of consistency in data operations from the angle of energy efficiency in a scenario of a data collection application. Having known that the major goal of a WSN is to collect consistent data and energy is mostly consumed in the data transmission and idle listening, we use the number of delivered messages to evaluate energy efficiency property. Thus, we first model the energy efficient data collection problem with the goal of delivering a minimum number of messages under the constraints of the data consistency. Then, an *adaptive, lazy, energy-efficient* data collection protocol named *Alep* is designed to support the goal of data consistency and to take advantage of data consistency. The basic idea of our protocol is three-fold: (1) adapting the data sampling rate of each sensor to the data dynamics in the data field based on a reinforce learning strategy; (2) reducing the number of total transmitted messages by dropping the data when data consistency is maintained; (3) reducing the number of total transmitted messages by aggregating and delaying the data reporting as much as possible.

The contributions of this paper are listed as three aspects. First, consistency requirements and data dynamics and their relation with energy consumption of WSN applications are analyzed. A formal model for data consistency in WSN is proposed. To our knowledge, we are the first to consider the formal model for data consistency in WSN. Second, an adaptive lazy protocol is proposed to reduce the number of delivered messages and to save energy. Finally, a comprehensive simulation is designed and implemented based on TOSSIM [5] to validate the effectiveness and efficiency of the proposed protocol by considering both non-aggregation and aggregation cases.

The rest of this paper is organized as follows. We first analyze data consistency requirements from specific appli-

cations and the feature of data dynamics in Section 2. In Section 3 we formally model the data collecting problem and present the formal definition for data consistency and data dynamics. An adaptive lazy energy-efficient protocol for data collection is described in Section 4. The comprehensive performance evaluation for the proposed protocol is reported in Section 5. Finally, related work and conclusion are listed in Section 6 and Section 7 respectively.

2 Consistency Requirements and Data Dynamics

WSNs are mostly application-specific systems that are widely used in various scenarios, and different applications have different requirements to the data consistency. Besides, WSNs are also data-centric systems, so that data consistency is closely related with data dynamics in the data field. In this section, we analyze different data consistency requirements and the feature of data dynamics.

Basically, the data consistency requirements in WSN consist of two aspects: *temporal consistency* which means that the data should be delivered to sink before it is expected and *numerical consistency* which requires that the collected data should be accurate. Some systems pay more attention to the temporal consistency and others care more about the numerical consistency. For example, in a patient monitoring system, emergency conditions of a patient should be reported to the control panel or caregivers in a limited time. Otherwise, the patient may be in a dangerous condition. Thus, most systems that need quick response or have high real-time requirements usually have high requirements on the temporal consistency. Other systems may have no strict time requirements on the collected data. For instance, a system that is counting the number of passed vehicles in one area may only need the data to be reported every long period, e.g., twice every day. However, these kinds of systems may have high requirements on the accuracy of the collected data, e.g., recording totally 80 and 90 vehicles may differ a lot. Thus in WSN system design, temporal consistency and numerical consistency should both be adjusted carefully in terms of energy-efficiency and application requirements.

The data consistency should also be integrated with the feature of data dynamics. Here, *data dynamics* means the trend and frequency of data changing. Usually, the data dynamics comes from two dimensions, *temporal data dynamics* and *spatial data dynamics*. In the temporal dimension, data changing frequency varies at different time periods. Figure 1 shows the data changing in terms of the time. In the figure, the data changes very fast before time t1 and between time t2 and t3, while it keeps almost stable between time t1 and time t2. Thus, if we keep the constant data sampling rate, the different data consistency will get during different periods with various data dynamics. On the other hand, from the spatial dimension, the data dynamics differs from area to area. An example of data changing differing spatially is shown in Figure 2, where, the data changes quickly in the right part of the sensor field and slowly in the left part. If we use the same data sampling rate in different locations, we will get different data accuracy, i.e., the collected data may be accurate in the area with low data dynamics, but not accurate for the area with high data dynamics. Furthermore, the temporal data dynamics and spatial data dynamics effect the data consistency at the same time. Thus to collect consistent data, the data sampling rate should be adapted to the feature of data dynamics from time to time and from area to area, e.g., it should sample more data when the data dynamics is high and in the area with high data dynamics, while sample less data when data dynamics is low and in the area with low data dynamics.

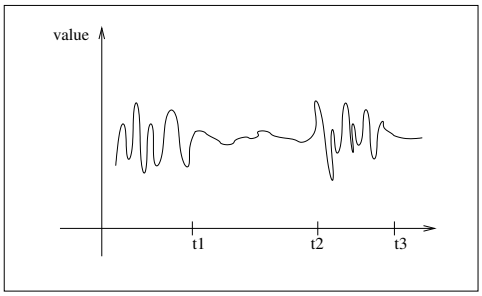

Figure 1. Data dynamics with the time.

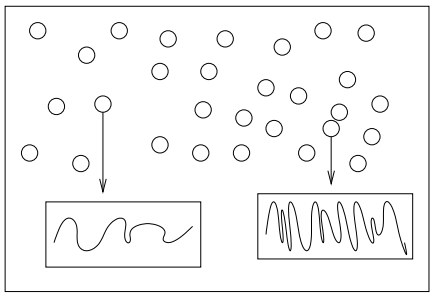

Figure 2. Data dynamics in different location.

Having known that data consistency should take consideration both specific application requirements to data and the feature of data dynamics, next, we explore the effect of data consistency in the data collection in WSN.

3 Formal Consistency Models

We examine the effect of consistency in data operations from the angle of energy efficiency in a scenario of a data collection application. Noting that most energy is consumed in message transmission and idle listening, we want to save energy by reducing the number of delivered messages, which

can not only save energy from sending and receiving messages but also increase possible sleeping time. In this section, we first model the energy efficiency data collection problem integrating data consistency, then we give the formal models for data consistency and data dynamics.

3.1 Problem Definition and System Level Data Consistency

We model the energy-efficient data collecting problem as the problem with the goal of reducing the total delivered messages meanwhile keeping the data consistency. So the problem can be modeled as the following,

$$\begin{array}{ll} obj. & \min \sum_{i=1}^{n} m_i - - - - - - - - - - - - - - - (1) \\ s.t. & T(r_{ijk}) - t(r_{ijk}) \geq 0, - - - - - - - - - (C1) \\ & \sum_{i=1}^{n} \sum_{j=1}^{nm} \sum_{k=1}^{dpm} (r_{ijk} - e_{ijk})^2 \leq C - - (C2) \end{array}$$

where the goal of the model is to minimize the number of delivered messages; the first condition, $C1$, implies the *temporal consistency*, i.e., the message should be delivered to the sink before it is expected, and the second condition, $C2$, shows the *numerical consistency*, i.e., the maximum variance of the collected data should not exceed the upbound of the application consistency requirements to the data as denoted as C, which is specified by applications. n is the total number of sensors; nm is the number of messages sampled at each sensor and dpm is the number of data in each message. m_i is the number of messages delivered at node i; r_{ijk} and e_{ijk} are the real and estimated value of the k^{th} reading in the j^{th} message at node i accordingly. $T(r_{ijk})$ is the expected deadline for reading r_{ijk} while $t(r_{ijk})$ is the time the reading arrives at the sink.

Here the energy efficient data collection problem is modeled in a centralized way, i.e, the data consistency is measured centrally at the sink at system level; however, in WSN, a totally distributed environment, the data transmission decision is made locally at each sensor, so it is better to achieve the global goal locally with local goal and local constraints on data consistency which is depicted in next subsection.

3.2 Model for Individual Sensors

We convert the system level model for problem to the model at individual sensor level, i.e., each sensor intends to reduce the number of delivered messages, and keeps the requirements of data freshness and value accuracy. So the problem model for each individual sensor can be,

$$\begin{array}{ll} obj. & \min m_i - - - - - - - - - - - - - - - - (2) \\ s.t. & T(r_{ijk}) - t(r_{ijk}) \geq t(is), - - - - - (C3) \\ & \sum_{j=1}^{nm} \sum_{k=1}^{dpm} (r_{jk} - e_{jk})^2 \leq C_i - - - (C4) \end{array}$$

where the goal is to minimize the number of delivered messages at each sensor; the first condition implies the *time consistency*, and the second condition shows the *numerical consistency*, C_i is an application-specific consistency threshold at sensor level. m_i is the total number of the messages delivered by one sensor; $t(is)$ is the upbound of the estimated time needed to deliver the message from node i to the sink; r_{jk} and e_{jk} are the real and estimated value of k^{th} reading in the j^{th} message separately. Next, we show that the problem modeled at sensor level is a subset of the problem modeled in system level.

Theorem 1 *Solutions for the problem defined in the sensor level model are solutions of the problem defined in the system level model.*

Proof. First, we show that if the objective of the sensor level model is minimized, the objective of the system level model is also minimized. Assuming S_i is the result for individual model, thus, $S_i = \min m_i$. Assume S is the result for the system model, $S = \min \sum_{i=1}^{n} m_i = \sum_{i=1}^{n} \min(m_i) = \sum_{i=1}^{n} S_i$. Thus, the system objective is the sum of the individual objectives. If the individual objective is achieved, the system objective can be achieved.

Second, we check two conditions in both models. We show that if the conditions hold in the sensor level model, they hold in the system level model as well. For the temporal consistency constraint, we can see that in the sensor level model the temporal consistency constraints are expressed as $T(r_{ijk}) - t(r_{ijk}) \geq t(is)$. If we let $i = s$ in $C3$, we can see that $T(r_{ijk}) - t(r_{ijk}) \geq t(ss)$, which is exactly $C1$, where $t(ss) = 0$, so if $C3$ holds, $C1$ holds.

For the numerical consistency constraint, we can show that if we select small enough value of C_i for each sensor, we can guarantee that if $C4$ holds, $C2$ holds. In $C2$, $\sum_{i=1}^{n} \sum_{j=1}^{nm} \sum_{k=1}^{dpm} (r_{ijk} - e_{ijk})^2 \leq \sum_{i=1}^{n} C_i$. Here, if we have $\sum_{i=1}^{n} C_i \leq C$ holds and $C4$ holds, $C2$ must hold. The easiest way to choose each C_i is to make $C_i \leq \frac{C}{n}$, where n is the number of sensors. Thus if we choose sufficient small value for each C_i in sensor level model, we can guarantee to satisfy the second condition in the system level model. □

From above analysis, we can find that the global optimization problem can be converted to a local optimization problem. Now our aim is to minimize the number of delivered messages and to satisfy the data consistency constraints at each sensor. Actually, consistency requirements should be refined to the sensing data level in a real WSN system.

3.3 Model for Data Items without Aggregation

In the previous model, we specify the data consistency requirement of each sensor. However, in a multimodality application, one sensor may deliver multiple messages with multimodality data, such as temperature, light, pressure, and so on. We argue that multimodality is a common case, and not an abnormal for future WSN applications. Thus, even one delivered message may contain several pieces of sensing data and these data may have different requirements on data consistency; furthermore, the data aggregation functions usually distinguish and operate only on the same type of sensing data. Thus, data consistency constraints should be refined to

the level of each piece of sensing data. Here, we formally model *the data consistency for each piece of data* as follows:

$$Consist(p)_{di} = Acc(p)_{di} \& OnTm(p)_{di}$$

where $Acc(p)_{di}$ specifies the numerical consistency of the di^{th} data of monitoring parameter p, and $OnTm(p)_{di}$ denotes the timeliness property of that data. This model means that the data is consistent if and only if the it maintains numerical consistency and temporal consistency. The models for both consistency are listed as follows.

$$Acc(p)_{di} = \begin{cases} 1 & |EV(p)_{di} - V(p)_{di}| \leq C(p)_{s-bnd} \\ 0 & otherwise \end{cases}$$

where $EV(p)_{di}$ and $V(p)_{di}$ are the estimated value and real value of the di^{th} sensing data for p, and $C(p)_{s-bnd}$ is the numerical consistency bound for p.

$$OnTm(p)_{di} = \begin{cases} 1 & T_{due}(p)_{di} - T_s(p)_{di} \leq ET(p)_{di} \\ 0 & otherwise \end{cases}$$

where, $T_s(p)_{di}$ is the time that the message will be delivered and $T_{due}(p)_{di}$ is the time when the sink expected to receive the data; and $ET(p)_{di}$ is the estimated time to deliver the message from current sensor to the sink.

Similar to the proof in Section 3.2, we can easily prove that if we can guarantee the consistency at each sensing data, we can guarantee the consistency at each sensor and further the consistency at the whole WSN. For example, if we make $C(p)_{s-bnd} \leq \frac{C_i}{nm*dpm}$, where $nm * dpm$ is the total number of sensing data sent at sensor i, the numerical consistency requirement at sensor i will be satisfied.

3.4 Model for Data Items with Aggregation

Data aggregation is a common way in WSN to reduce the number of delivered messages. Having consistency model for single data, we also need to define a consistency model for aggregated data. Similar to the consistency model for a single data, the consistency model for aggregated data is also application-specific and related with different parameters. The difference of two consistency models for single data and aggregated data is that there is an aggregated function operating on a set of data in the case of aggregation. So the data consistency model for aggregated data is defined as follows:

$$Consist(p)_{di} = Acc(p)_{di} \& OnTm(p)_{di}$$

$$Acc(p)_{di} = \begin{cases} 1 & |f(p, ED_{di}) - f(p, D_{di})| \leq C(p)_{a-bnd} \\ 0 & otherwise \end{cases}$$

$$OnTm(p)_{di} = \begin{cases} 1 & T_{due}(f(p, D_{di})) - T_s(f(p, D_{di})) \\ & \leq ET(f(p, D_{di})) \\ 0 & otherwise \end{cases}$$

where f is the aggregation function such as average, sum, count, and so on; p is the specific parameter; D_{di} and ED_{di} are the real and estimated value for the di^{th} data set separately; $f(p, D_{di})$ and $f(p, ED_{di})$ are the real and estimated aggregated value for the di^{th} data set separately; and $C(p)_{a-bnd}$ is the numerical consistency bound for parameter p. T_{due}, T_s and ET have the same meaning as before.

3.5 Model for Data Dynamics

Data consistency reflects the accuracy of the data and the staleness of the data. We envision that the data accuracy is closely related with the data sampling rate. For a series of n sensing data, if we get every piece of data, the accuracy is the best by using reading values as estimation values. If we get readings in a half frequency, the accuracy will decrease since we have to estimate half of the data. On the other hand, the energy is saved from sampling and reporting less data. Thus data sampling rate should be decided by making tradeoff between the data accuracy and energy efficiency, which, we argue that, can be achieved by matching data sampling rate to data dynamics.

To describe data dynamics in the monitoring field, we define a number of windows to observe the data readings. Two parameters, $winSize$ and $winNum$ are defined to model the dynamics of data. $winSize$ denotes the number of readings in one window, and $winNum$ specifies the number of windows in one observation. Thus the total number of readings in one observation is $Num_{rd} = winSize * winNum$. Since data dynamics reflects *the frequency of the data changing*, so we first define the frequency of the data changing as the number of data changing in one observation:

$$Num_{chg} = \{Cnt(i) \| r_{i+1} - r_i| > B \& i \in [0 : Num_{rd}]\}$$

where, $Cnt(i)$ is the number of is satisfying the conditions; r_i and r_{i+1} are the i^{th} and $i + 1^{th}$ readings separately. And $B = C(p)_{bnd}$ is the accuracy bound for this parameter. Based on this definition, we define the data dynamics (DYN) as the average number of changing in one monitoring window.

$$DYN = \frac{Num_{chg}}{Num_{rd}} * winSize$$

From above definition, we can find that data dynamics is defined based on time period, i.e., inside the window of observation. By adjusting the value of $winSize$ and $winNum$, we can get the data dynamics with various sensitivity. Based on data dynamics, it is possible for users to choose suitable data sampling rate to accurately collect data in an energy efficient way, which will be explained in detail in Section 4.

In our design, both concepts of data consistency and data dynamics are data-centric and application-specific. First, both of them are directly related with the value and staleness of sensor reading. Second, the applications can choose suitable data consistency model to meet their specific data consistency requirements by setting specific consistency bounds and choose different values for $winSize$ and $winNum$ to

estimate the data dynamics. Furthermore, our models to calculate the data consistency and data dynamics are full decentralized, i.e., data dynamics and sampling rate is calculated at each sensor, thus it is easy to be applied in WSN.

4 ALEP: An *A*daptive, *L*azy, *E*nergy-efficient *P*rotocol

In this paper, we intend to save energy by estimating the value of the sensing data so that to reduce the number of delivered messages. Our proposed *Alep* protocol consists of three components, *adapting the sampling rate based on the data dynamics and resource availability, keeping lazy in transmission based on consistency-guaranteed estimations, and aggregating and using long length packet*. These methods are described in detail in the following subsections.

4.1 Adapting the Sample Rate

We adapt the sampling rate based on the model for data dynamics defined in previous sections. The process of adapting the sampling rate is a process of reinforce learning based on the data reading. Based on the value of DYN, we can define the adaption of the sampling rate as

$$R_{smp} = \begin{cases} \lceil \frac{DYN - Ave_{chg}}{Df_{bnd}} \rceil * R_{cr}, & DYN > Ave_{chg} \\ \frac{Ave_{chg} - DYN}{Df_{bnd}} * R_{cr}, & DYN \leq Ave_{chg} \end{cases}$$

where, R_{smp} is the adapted sampling rate; R_{cr} is the current sampling rate. Ave_{chg} is the normal average changes happen in one window size; and Df_{bnd} bounds maximum difference between the observed value of data dynamics and the normal average changes, i.e., if DYN is larger than Ave_{chg} and the difference exceeds the bound, the sampling rate should be increased; when DYN is much smaller than Ave_{chg}, the sample rate should be decreased. Different applications could define their specific up-bound and low-bound of the suitable sampling rate. However, these bounds cannot exceed the maximum bound and minimum bound. Here we define the maximum bound of the sampling rate as the maximum bandwidth of the sensor and the minimum bound of the sampling rate as the smallest sampling rate that satisfies the Nyquist-Shannon sampling theorem [9].

The sampling rate learns from the previous data dynamics, and uses the most recent data dynamics to estimate the nearest future data dynamics. We believe that in most cases the data dynamics will not change dramatically. The data history is limited by the number of windows and the window size in one observation. We can adjust the length of history based on the window size and the number of windows.

4.2 Keeping Lazy in Transmission

One way to reduce the number of delivered messages is to keep lazy in transmission, i.e., only sending the messages that are necessary to be sent because we think that if the receiver can estimate an accurate enough value for the current reading, the message need not to be sent, i.e., if the data consistency requirement can be hold, the messages are not necessary to be sent.

In this protocol, every sensor caches the last transmitted reading for every parameter for all potential senders that may deliver message to it, and it uses the cached values as the estimation of the current reading. To check the data consistency for this piece of data, the sensor will use the current reading as the real value and the cached value as the estimated value. If the difference between the current reading and the cached value is within the consistency bound, the sender will not send this piece of data, i.e., keeping lazy. For example, in an application which monitors the temperature of a sensor field, when a sensor gets a reading of value 3.7, and the cached last reading is 3.5 which is within the consistency bound of 0.3. So the new reading is not necessary to be sent. When the current data reading is absent, the sensor assumes the value is unchanged so that it keeps silent. This approach has two advantages: easier to estimate the undelivered data locally and only keeping copy of a very small amount of data.

In the case of the aggregated data, every receiver caches a copy of the latest aggregated value calculated from senders. After it applies the aggregation function, it will compare the new calculated value with the cached value. If the difference between them is within the consistency bound, the sender will keep silent. For the aggregated data, the receiver has to wait for the new reading from all the senders for a period of time. If there are still data absent from some senders, the receiver will use the cached data to substitute the current reading and calculate the new aggregated value.

4.3 Aggregating and Delaying Delivery

Another aspect of the Alep protocol is to integrate several pieces of data into one message to reduce the number of messages and delay the delivery when the data temporal consistency is not violated. In our design, each sensor maintains a data queue where the received data are stored. The data in the queue are sorted according to the application specific priority and the requirement of temporal consistency. When there are free space in the queue and the consistency is satisfied, the sensor will keep sleeping instead of sending data to its parent node. The temporal consistency is checked by comparing the estimated time to deliver the message to the sink and the time the data is expected at the sink. In our application, the expected time to deliver the message to the sink can be estimated based on the number of hops to the sink. For example, if we assume it takes T_{dev} to transmit one message from the child to the parent, then we can estimate the time it takes for current sensor to deliver a message to sink is $T_{dev} \times H_{js}$, where H_{js} is the number of hops from the current sensor to the sink. Then the time bound for the data is the sum of the estimated time plus one time slot, which denotes the time between two reporting points according to the TDMA schedule. More discussions of the protocol can

be found at [12].

5 Performance Evaluation

To evaluate the performance of the proposed adaptive, lazy, energy-efficient protocol, we have implemented the protocol in TinyOS using the TOSSIM [5] environment. In the rest of this section, we will describe the simulation setup and the performance metrics first, then present the performance of our protocol in terms of these performance metrics.

5.1 Simulation Setup and Evaluation Metrics

In our simulation, 121 nodes are distributed in a circle area, with the base station located at the center of the circle area. All these nodes are connected to for a balanced tree with height of four, i.e., the depth of the tree, where all the internal nodes have three children. The sensors periodically collect data from its children and report the readings to its parent based on a TDMA schedule.

Each sensor node acts as a multiple functional sensor, which can sample three parameters: Temperature as `Temp`, Pressure as `Press`, and Rain-index as `Humid`. To evaluate the proposed protocol in different data dynamics environments, we intentionally make these three parameters have different dynamic characteristics. For example, for the perspective of temporal, the reading always changes fast for `Temp`, relatively stable for `Press`, while fast at first then stable for `Humid`. To simulate spatial data dynamics, we intentionally separate the whole area according to three tree of the root. The reading changes fast in the left subtree area, relatively stable in the right subtree area, and fast at first period then becomes stable in the middle subtree area.

The goal of the *Alep* protocol is to save energy by reducing the number of delivered messages while satisfying the data consistency requirements. Thus, we use two metrics to evaluate our approach. To measure the reduction of the number of delivered messages, we count the total number of messages that have been sent at each sensor.

To answer the question of the effect of reduced messages to the data consistency, we propose a new performance metric called *data inconsistency factor (DIF)*, which is defined as the total variance between the gathered data in the sink and real data, i.e., $V = \sum_1^n (d_{rcv} - d_{fld})^2$, where, V is the value of variance; d_{rcv} and d_{fld} are the reading value received at sink and the real value sampled at data field separately. The more accurate the data, the smaller the variance.

5.2 Number of Delivered Messages

Usually collecting more data is a way to improve data consistency; however, by adapting the sampling rate to fit the feature of data dynamics and keeping lazy when data is in the range of consistency, data accuracy can be improved without significantly increase the number of delivered messages. Moreover, in some cases when data dynamics is low, data consistency can be kept even by delivering fewer number of messages. In this section, we show the number of messages delivered at each sensor using different approaches.

Figure 3 and 4 list the number of delivered messages at each sensor without and with aggregation respectively. The x-axis is the ID of sensors, and the y-axis denotes the number of delivered messages. Note that the y-axis of the two figures are at different scales. As a matter of fact, the number of delivered messages for all approaches reduces significantly when aggregation is used. From the two figures, we can see that *Simple* delivers the maximum number of messages and *Lazy* transfers the minimum number of messages in both cases of with and without aggregation.

These three approaches have totally different performance in terms of the number of delivered messages. In the case of without data aggregation shown in Figure 3, the sensors are classified to four types based on the layer in the tree using *Simple*, i.e., sensors in the same layer using *Simple* deliver the same number of messages. However, using *Alep* and *Lazy*, the sensors transmit different number of messages because of the various data dynamics in the different areas. For example, among sensors located at layer 3, sensors with ID between 13 and 21 transfer 140 messages because the high data dynamics of the monitoring area, while the sensors with ID between 31 and 39 only deliver 41 messages because the low data dynamics of the monitoring area, which is fewer than $\frac{1}{3}$ of that in the high dynamics area. The similar results exist in the case with data aggregation in Figure 4, where all the sensors deliver the same number of messages using *Simple*, while the sensors using *Alep* and *Lazy* located at different areas transmit different number of messages, i.e., the sensors located at high dynamics area deliver 57 messages but the sensors located at low dynamics area only send 9 messages.

Comparing with *Lazy*, we observe that the sensors using *Alep* send more number of messages than using *Lazy* at the area with high data dynamics (e.g., node 13 – 21) but send fewer number of messages than that of using *Lazy* at the area with low data dynamics (e.g., node 31 – 39). This is because the sampling rate is increased much in the area with high data dynamics and decreased a lot in the area with low data dynamics. From above analysis, we conclude that *Lazy* can always reduce the number of delivered messages, and *Alep* usually does not increase the number of delivered messages and reduce the number of delivered messages a lot when the data dynamics is low.

5.3 Data Inconsistency Factor

From the above section, we can see that *Lazy* and *Alep* significantly reduce the number of delivered messages. However, delivering fewer message means that there are more data estimated at the sink, which may result in the degradation of the data consistency. In this subsection, we examine the effect of unsent messages to the data accuracy. We use data inconsistency factor as the metric to measure the effect.

Figure 5 reports the relationship between DIF and differ-

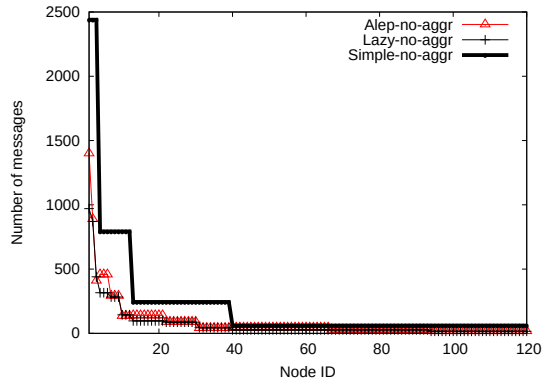

Figure 3. Number of delivered messages without aggregation.

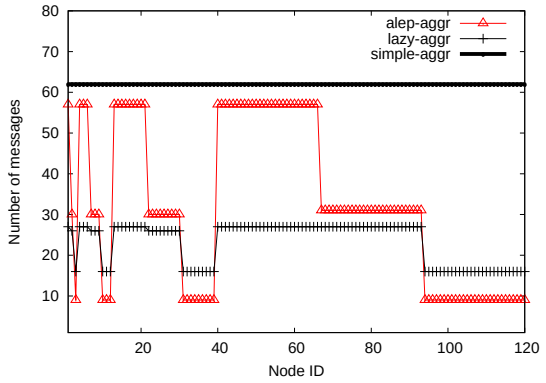

Figure 4. Number of delivered messages with aggregation.

ent monitoring parameters with variant data dynamics. The x-axis is different data types with variant data dynamics and the y-axis represents the calculated DIF of the collected data. Three types of parameters with different data dynamics are monitored, among which `Temp` has relatively higher data dynamics than `Humid` and `Press` while `Press` has relatively lower data dynamics. Furthermore, for each parameter, data dynamics also varies according to different areas, i.e., each parameter has three types of data dynamics, high, high first then low denoted as mix, and low. Thus, there are totally nine sets of data with variant data dynamics.

In the figure, we note that when the data dynamics is high, the value of DIF is larger, e.g., the `Temp high` has larger DIF than `Temp mix` and `Temp low`, and `Temp high` also has larger DIF than `Humid high` and `Press high`. The reason of this is when the data dynamics is high, it is more difficult for the sink to estimate the correct data. From the figure, we also find that *Alep* has much smaller DIF than that of *Simple* and *Lazy* when the data dynamics is high,

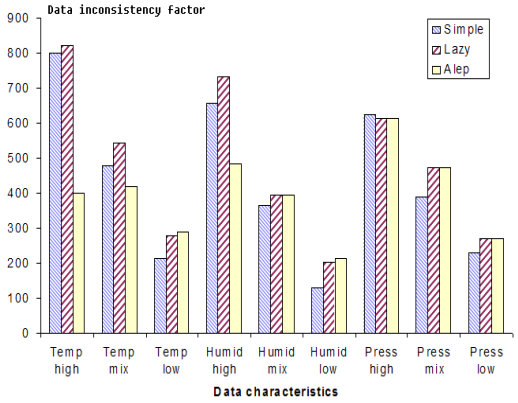

Figure 5. The results of data inconsistency factor.

while it has larger DIF than that of *Simple* and has the same DIF as *Lazy* when the data dynamics is low. This result shows that *Alep* indeed makes the data sampling rate to fit the feature of data dynamics, i.e., when the data dynamics is high, it will use higher sampling rate to gather more data so that to make the variance small. Otherwise, it will sample less data to save energy.

DIF increases very fast with the increasing of data dynamics using *Simple* and *Lazy*, but increases slowly using *Alep*. As a result, *Simple* and *Lazy* may not collect enough accurate data when the data dynamics is high, i.e., DIF exceeds the data consistency requirements of the application. However, *Alep* can keep DIF low by adapting the data sampling rate to data dynamics. We should also notice that *Alep* improves the data accuracy meanwhile somehow reduces the number of delivered messages as shown in Section 5.2.

Comparing *Lazy* with *Simple* in terms of the accuracy of the collected data, *Lazy* has very close value of data variance as *Simple*, however, in Section 5.2 we know that *Lazy* delivered fewer messages than *Simple*, which means that the dropped messages are not necessary to be transferred to the sink. Thus, we conclude that lazy delivering can reduce the number of delivered messages, while the approach of adapting the data sampling rate to data dynamics can significantly improve the data accuracy. We also evaluated the trade off of data consistency and energy efficiency, the results is available at [12].

6 Related Work and Discussions

Having introduced our work, in this Section, we compare our work with previous efforts in terms of energy efficiency design, data consistency, and adaptive design respectively.

Energy efficiency is always one of the major WSN design goals. Thus, energy efficient protocols have been sufficiently explored. Previous work expects to achieve the goal of en-

ergy efficiency by designing energy efficient routing protocols [11], energy efficient MAC protocols [16], energy efficient clustering [17], and other energy efficient approaches. These approaches mainly focus on finding some energy efficient paths, designing better turn on/off schedules, forming energy efficient clusters, and so on.

Data consistency is a classical problem in computer architecture, distributed systems, and database. A lot of consistency models have been proposed in the research of these fields. However, these models are usually not applicable in WSN. Ramamritham *et al.* propose an idea to maintain the coherency of dynamics data in the dynamics web monitoring application [13]. They model the dynamics of the data items. Our model for data consistency is more general than theirs and applied in different fields. Lu *et al.* propose a spatiotemporal query service in [6] to enable mobile users to periodically gather information and meet the spatiotemporal performance constraints, but they propose neither data consistency models, nor adaptive protocols.

Adaptive approach is always attractive in system design. Several adaptive protocols which adapt cluster formation and duty cycle designing are proposed in literature. Adaptive sampling rate has also been proposed from researchers of database field, sharing the same goal of our Alep protocol. Jain and Chang propose an adaptive sampling for WSN [4]. They employ a Kalman-Filter (KF) based estimation technique and the sensor uses the KF estimation error to adapt the sampling rate. Marbini and Sacks [8] propose a similar approach to adapt the sampling rate as ours; however they do not model the data dynamics and require an internal model, which is usually difficult to find, to compare the sampled data. TinyDB [7] adapts the sampling rate based on current network load conditions, but not based on the data dynamics in the data field.

Filters are used to reduce the size of the data stream. Work by Olston *et al.* uses an adaptive filter to reduce the load of continuous query. Their work focuses on the adaptive bound width adjustment to the filter so that their results are helpful to analyze our lazy approach. Sharaf *et al.* study the trade off between the energy efficiency and quality of data aggregation in [14]. They impose a hierarchy of output filters to reduce the size of the transmitted data. Data prioritization in TinyDB [7] chooses the most important samples to deliver according to the user-specified prioritization function.

7 Conclusions and Future Work

In this paper, we consider the effect of data consistency to data operations in WSN. First, we analyze the data consistency requirements and the feature of data dynamics. Then, we formally model the data collection problem with the goal of delivering a minimum number of messages under constraints of data consistency, and propose a formal definition for data consistency and data dynamics in WSN. Then, $Alep$ is proposed to save energy and keep data consistency.

Considering data consistency in WSN for data quality assurance is an interesting problem, we plan to extend our work in the following two directions. First, implied from the simulation, we plan to design a consistency-driven duty cycle management scheme to take full advantage of $Alep$. Second, we will relax several assumptions made in this paper, and define a more general consistency model for different WSN applications scenarios.

References

[1] M. Batalin et al. Call and responses: Experiments in sampling the environment. *Proc. of ACM SenSys 2004*, Nov. 2004.

[2] D. Estrin, D. Culler, K. Pister, and G. Sukhatme. Connecting the physical world with pervasive networks. *Proc. of IEEE Pervasive Computing*, 1(1):59–69, 2002.

[3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. *Proc. of the 9th ASPLOS'00*, pages 93–104, Nov. 2000.

[4] A. Jain and E. Chang. Adaptive sampling for sensor networks. *Proc. of the 1st international workshop on Data Management for Sensor Networks: in conjunction with VLDB 2004*, Aug. 2004.

[5] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: Accurate and scalable simulation of entire tinyos applications. *Proc. of ACM SenSys 2003*, Nov. 2003.

[6] C. Lu et al. A spatiotemporal query service for mobile users in sensor networks. *Proc. of ICDCS 2005*, June 2005.

[7] S. Madden et al. Tinydb: An acqusitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1), Mar. 2005.

[8] A. Marbini and L. Sacks. Adaptive sampling mechanisms in sensor networks. *Proc. of London Communications Symposium*, 2003.

[9] The nyquist-shannon sampling theorem. *http://ptolemy.eecs.berkeley.edu/eecs20/week13/nyquistShannon.html*

[10] C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. *Proc. of the 2003 ACM SIGMOD International Conference on Management of Data*, June 2003.

[11] K. Seada, M. Zuniga, A. Helmy, and B. Krishnamachari. Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks. *Proc. of ACM SenSys 2004*, Nov. 2004.

[12] K. Sha and W. Shi. On the effects of consistency in data operations in wireless sensor networks. *Technical Report MIST-TR-2005-011*, Wayne State University, Oct, 2005.

[13] S. Shah, S. Dharmarajan, and K. Ramamritham. An efficient and resilient approach to filtering and disseminating streaming data. *Proc. of VLDB 2003*, Sept. 2003.

[14] M. Sharaf, J. Beaver, A. Labrinidis, and P. Chrysanthis:. Balancing energy efficiency and quality of aggregate data in sensor networks. *The VLDB Journal*, 13(4):384–403, 2004.

[15] R. Szewczyk et al. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, June 2004.

[16] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. *Proc. of INFOCOM 2002*, June 2002.

[17] Q. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. *Proc. of INFOCOM 2004*, Mar. 2004.