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

A wireless sensor network is a kind of data-centric, application-specific and recourse constraint distributed system. In these battery powered systems, 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 not only includes temporal consistency *and* value consistency*, but also considers* the application specific requirements of the data *and* data dynamics in the data field. *We first formally model the energy-efficiency problem in the scenario of a passive monitoring application with the goal of delivering minimum number of messages under the constraint of data consistency. Then, we give the formal definition of the data consistency in wireless sensor networks. To achieve this goal, 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 based on PowerTOSSIM, we find that the proposed approach indeed reduces the number of delivered messages by more than* 20%*, extends the lifetime of the wireless sensor network by more than* 50%*, and improves the accuracy of the sampled data. Furthermore, we show that the tradeoff between the energy efficiency and data consistency should be made based on the specific requirements of different applications.*

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 [4]. Micro sensors such as Motes from Intel and Crossbow [2] are developed to make WSN applications possible; TinyOS [7, 9] 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 [25], environment sampling [1] and structure monitoring [27], have been launched, showing the promise of wide applications of WSN.

Because of the special characteristics of WSN such as limited power supply, restricted computing and storage capability, previous proposed protocols in WSN are mainly focus on designing an energy efficient sensor system. Lots of work has been done to extend the lifetime of WSN, including energy-efficient routing protocols [20], energy-efficient MAC protocols [28], and energy efficient data aggregation [11] and clustering [29]. These approaches 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 [8, 11, 13] and filtering unnecessary sampled data [15, 23] to reduce the total data traffic; however, a model to measure the quality of the collected data is missed in their work. As a matter of fact, in data-centric distributed systems such as WSN, we argue that a model for the quality of the collected data such as a data consistency model is essential in WSN applications. In this paper, we first try to model the data consistency, and then examine the effects of data consistency on energy efficiency in data collection in the WSN.

As we argued in this paper, data consistency, including *temporal consistency* and *value consistency*, is considered as the integration of two factors, *specific application requirements to the sampled data* and *the feature of data dynamics* in the sensor field. First of all, most WSN systems are application-specific systems. Thus, different applications may have totally different requirements of data consistency. Moreover, the temporal and spatial data dynamics in various applications also affect the data consistency. Having known that the major goal of the WSN is to collect consistent data and noticing that energy is mostly consumed in the data transmission and idle listening [12], we intend to save energy by reducing the number of delivered messages. Thus, we first model the energy efficient data collection problem with the goal of delivering the minimum number of messages under the constraints of the data consistency. Then, an *adaptive, lazy, energy-efficient* data collection protocol for WSN named *Alep* is designed to reduce the number of messages. 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 save energy. Finally, a comprehensive simulation is designed and implemented based on PowerTOSSIM [24] 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 applications and the feature of data dynamics in Section 2. Section 3 depicts the scenario of a passive monitoring application and the approach to collect data. In Section 4 we formally model the data collecting problem and presents the formal definition for data consistency and data dynamics. An adaptive lazy energy-efficient protocol for data collection is described in Section 5. A comprehensive performance evaluation for the proposed protocol is reported in Section 6. Finally, related work and conclusion are listed in Section 7 and Section 8 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 from the applications as well as 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 *value 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 value 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. In this case, data aggregation is more possible because some aggregation functions need to wait until sufficient data are available. 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 value 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 in the sensor field. In this paper, *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 **??** (a) 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 **??** (b). In the figure, 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. For example, 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.

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 A Typical Application Scenario

The applications in WSN can be classified to three major types, including passive monitoring applications, active query applications, and event-driven applications. In these applications, sampled data form a data stream and are delivered from the source to the sink. Furthermore, these dif-

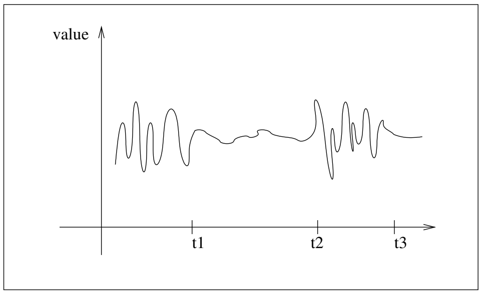

Figure 1. Data dynamics with the time.

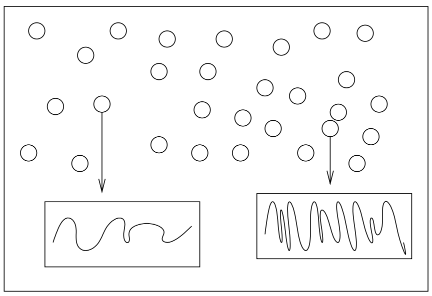

Figure 2. Data dynamics in different location.

ferent applications have different consistency requirements and data dynamics. For example, passive monitoring applications usually collect a large number of data but with looser real-time requirement, furthermore, they usually require a longer system lifetime, while event-driven applications often have higher requirements on the temporal consistency. In this paper, the consistency model and the protocol to save energy and explore the effect of data consistency to the energy efficiency in WSN can be applied in all applications. We choose the passive monitoring application as a scenario to demonstrate the importance of data consistency and the performance of out protocol because it collects largest number of data and is easier to be simulated based on simulators like PowerTOSSIM [9, 24]. Also, this type of applications are the most successfully deployed WSN applications.

In our passive monitoring application, we assume that after sensors are deployed in the sensor field, they are self-organized into a load balanced tree where each sensor has three children as shown in Figure 3. We also assume the ideal case of a load-balanced tree; however, in a real WSN this may not be the case. But this does not hurt our analysis at all. In the tree, the sink acts as the root of the tree. The leaf nodes are responsible for periodically sampling value of monitoring parameters, while internal nodes are responsible for periodically sampling interested parameters, aggregating the data from children and reporting the data to the parent. We assume there is a time synchronization scheme to synchronize all sensors. Each sensor follows a specific wakeup/sleep schedule and serves its children in a TDMA way, i.e., dividing the time slot to several pieces as shown in Figure 4. At the first several periods of time, it gets the data from its children one by one; then it senses the data by itself, and finally transmits the data to its parent. After that the sensor goes to sleep until the next time it wakes up. Thus, the readings at each sensor forming a continuous data steam are transferred to the sink hop by hop.

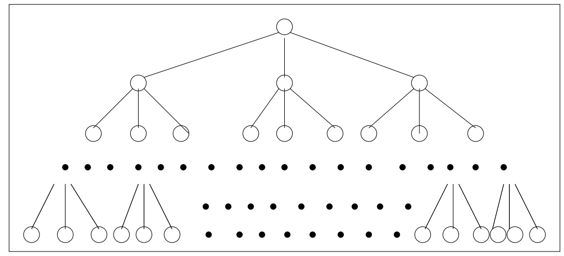

Figure 3. A tree-structured sensor network.

listen to child 1	*listen to child 2*	*...*	*listen to child n*	*sensor data*	*report to parent*

Figure 4. TDMA schedule.

We choose a tree-structure to build our WSN because it is widely used in WSN applications and has the following advantages. First, it is easy to design a TDMA schedule for tree-based structured WSN. Second, each sensor can easily know its distance, the number of hops, to the sink, which is equal to the depth of the sensor in the tree, thus it is easy to estimate the time bound to deliver a message from current node to the sink. Third, hierarchical structures like tree are scalable to be applied in large scale application. However, our protocols and models are not limited to the application of tree-based structure. They can be applied in various applications with different data collecting approaches.

4 Formal Consistency Models for WSN

Our goal of this paper is to examine the effect of data consistency to the data operations in WSN. We argue that energy can be saved by considering the data consistency in WSN. Noting that most energy is consumed in message transmission and idle listening [12], 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. By considering data consistency, we can estimate some data rather than get all the data from the data field to reduce the number of delivered messages. In this section, we first model the energy efficiency data collection problem by considering data consistency, then we give the formal models for data consistency and data dynamics.

4.1 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 modelled 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 *value 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. In the following subsection, we model the problem at individual sensor level.

4.2 Model at Individual Sensors

In the above subsection, we model the energy efficiency data collection problem in a central way, however, due to the inborn distributed feature of WSN, 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 *value 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, we can see that 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 value 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 can have $\sum_{i=1}^{n} C_i \leq C$ holds and $C4$ also 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. As a matter of fact, consistency requirements should be refined to the sensing data level in a real WSN system, as shown in next subsection.

4.3 Data Consistency 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, e.g., a hybrid sensor is capable of sensing 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. However, 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 at each individual sensor should be refined to the level of each piece of sensing data. Here, we formally model *the data consistency for each piece of data*, which is defined as follows:

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

where $Acc(p)_{di}$ specifies the value 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 value 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 value 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 4.2, 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 value consistency requirement at sensor i will be satisfied.

4.4 Data Consistency Model for Data Items with Aggregation

Noting that 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 value consistency bound for parameter p. T_{due}, T_s and ET have the same meaning as that of in the model for single data.

4.5 Model for Data Dynamics

Data consistency reflects the value difference between the estimated data and the real data and the staleness of the data. From the view of accuracy, 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. In this section, we model the behavior of 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, e.g., if the $winSize$ is seven, then in one monitoring window, the sensor will obtain seven readings, and $winNum$ specifies the number of windows in one observation, e.g., if $winNum$ is four, in one observation there will be four windows. 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} is the i^{th} and $i + 1^{th}$ reading 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. For instance, when we set $winNum$ small, the value of DYN will be calculated with high frequency, i.e., it can be very acute to the data changing. While the value of $winSize$ controls the range of the DYN, e.g., if we set $winSize$ to two, data dynamics can be expressed as above one and below one; however, if we set the value of $winSize$ to four, data dynamics can have four levels, below one, one to two, two to three, and above three. 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 5.

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.

In our models, we use the number of delivered messages to replace the energy consumption in WSN because we believe most energy is consumed in the message transmission and idle listening, which is mostly reflected in the number of the delivered messages. We also simplify our models in calculating the number of delivered message without considering retransmission; however, we believe that the model can be easily extended to a version of considering retransmission by assuming a packet lossy rate. In the next section, we give details of the protocol for data collection.

5 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. Note that the estimated data should satisfy the consistency requirements of applications. Having the models for the energy efficient data collection problem definition, data consistency, and data dynamics, we are now in a position to design a new protocol to reduce the number of delivered messages for data collection. In this section, we will first introduce the rationale of our design, then give the details of the protocol.

5.1 Rationale

As argued in the previous sections, when the data sampling rate is low, we need to estimate more data on the sink side. Thus, the sampling rate affect the data accuracy significantly, because the estimated values may not be accurate enough. There are two extremes between data accuracy and energy-efficiency. For energy-efficiency purposes, we can only gather and deliver very small amount of data. Subsequently, the gathered data cannot satisfy the consistency requirements of the application. On the other hand, if we always keep high sampling rate and deliver a lot of messages to get very accurate data, sensors will run out of energy very quickly. Moreover, considering the limited storage and bandwidth of sensor motes, the data may be over sampled, i.e., the volume of the sampled data exceeds the available resource so that sensors have to drop some important data which ruin the data consistency. Thus, we should make a tradeoff between the energy consumption and the data accuracy as well as find a suitable sampling rate. From our observation, we find that the data dynamics varies from time to time and area to area. Furthermore, we also find that it is easier to get accurate estimation when the data dynamics is low, however it is difficult to get accurate estimation when the data dynamics is high. Thus, within the budget of the available resource, the sampling rate should adapt to the data dynamics in both temporal and spatial ways. When the data dynamics is high, the sampling rate should be raised to improve the data accuracy, otherwise, it should be decreased to reduce the number of delivered energy.

Except adapting the data sampling rate to data dynamics, we can improve the techniques to estimate the next data, so the number of delivered messages can be dramatically reduced using estimated data to replace the sensing data and high data accuracy is kept. Besides, as mentioned in literature [12], sending a message with long length is more energy efficient than sending several messages with short length. Thus, we intend to integrate multiple short messages into one big message.

In summary, 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.

5.2 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. In the model, data dynamics reflects the average number of changes in one monitoring window. Thus 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 [14]. We argue that the data sampled using the sampling rate between the maximum and minimum sampling bounds are meaningful and will not cause the problem of over sampling.

Based on this formula, 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. By adjusting the length of history based on the window size and the number of windows, we can adjust the frequency of sample rate changing and the acuteness of the changing of the environment, e.g., we can change the sampling rate very quickly by setting small value to the number of windows in observations.

5.3 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 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.

5.4 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.

5.5 Discussion

Noting that the proposed protocol is a general protocol for sensor networks, we still have several assumptions. First, the data readings from sensors are accurate, i.e., here we do not consider reading errors. Second, synchronization between sensors are kept by using some mechanisms, e.g., RBS [3]. Third, sensors in the sensor field are static, which is true in most monitoring-based applications. Finally, the sensors in the sensor field are homogeneous, i.e., each sensor has the same physical capacity.

Although our protocol is proposed for a tree-based sensor network, the basic idea of adapting the data sample rate, keeping silent during transmission and merging transmissions can be applied in any protocol to achieve both data consistency and energy efficiency. When there are several parameters being monitored at one sensor, we can assign different priorities to these parameters, however, a good resource allocation algorithm should be designed to appropriately allocate resources to satisfy the consistency requirements of each parameter, especially in the case of resource hungry.

From the model for the problem definition, we can see that the optimization at a single sensor can guarantee the system level optimization. However, the system level optimization doesn't necessarily require the optimization at single sensor. The consistency requirement at a single sensor is more rigorous than the consistency requirement for the whole sensor network. So, we may loosen the consistency requirements for individual sensor a little, and the consistency requirements for the whole sensor network will still hold at a very high probability.

In our design, we use the last reading to estimate the previous reading which is consistent when the transmission is reliable. However, it only reduces the messages having data within the consistency bound. If other techniques can accurately estimate the value of the data out of the range of the consistency bound, they can further reduce the number of delivered messages, which will be our future work.

6 Performance Evaluation

To evaluate the performance of the proposed adaptive, lazy, energy-efficient protocol, we have implemented the protocol in TinyOS using the PowerTOSSIM [9, 24] 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.

6.1 Simulation Setup

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 a tree-based structure with height of four, i.e., the depth of the tree, in which all the internal nodes have three children as shown in Figure 3. The sensors periodically collect data from its children and report the readings to its parent based on a TDMA schedule. Besides, the sensors may have the ability of aggregation. The data consistency requirements at each sensor are preloaded from the sink by broadcast.

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 characteristics. For example, for the perspective of temporal, the reading of `Temp` changes very fast, and the reading of the `Press` is relatively stable, while the reading of `Humid` may change very fast during the raining time while it is stable otherwise. To simulate the data dynamics in different areas, we intentionally separate the whole area to three sub-areas as the three subtrees shown in Figure 3, where the reading for parameters changes fast in the left subtree, and the reading is relatively stable in the right subtree, while the reading in the middle subtree changes fast at first period then becomes stable.

Three algorithms will be simulated in our experiment. First, the method without considering both lazy and adaptive approaches will be simulated, denoted as *Simple*. Second, the method with lazy approach but without adaption is simulated, denoted as *Lazy*. Finally, the method considering both lazy and adaptive approaches is simulated, denoted as *Alep*.

6.2 Evaluation Metrics

The goal of the *Alep* protocol is to save energy by reducing the number of delivered messages and to examine the tradeoff between the energy efficiency and data consistency. Thus, *Alep* will be examined in three ways: *Does this protocol reduce the number of the messages and extend the lifetime of WSN*? *Does this protocol improve the accuracy of data*? And *what is the tradeoff between the number of delivered messages and the data accuracy*?

To measure the reduction of the number of delivered messages, we count the total number of messages that have been sent at each sensor. Meanwhile, we measure how much energy can be saved and how much the lifetime of WSN can be extended using the mechanism provided by PowerTOSSIM, including the total energy consumed in sampling, transmission, and computing. The definition of the lifetime of WSN is referred from [21], which is the maximum number of messages the WSN can gather before the WSN loses connectivity or coverage.

To answer the question of the effect of reduced messages to the data consistency, we propose a new performance metric called *data inconsistency factor*, 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.

To examine the tradeoff between the energy consumption and the data accuracy, we adjust the value of the temporal consistency bound and the value consistency bound, which are two parameters in the *Alep* protocol. By adjusting these parameters, we can get different simulation results in terms of energy consumption and data accuracy.

6.3 Number of Delivered Messages

Usually collecting more data is a way to improve the data accuracy; 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 the data dynamics is low, the 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 **??** lists the number of delivered messages at each sensor without and with aggregation respectively. The x-axis is the ID of each sensor, and the y-axis denotes the number of delivered messages. Note that the y-axis of Figure **??** (a) and **??** (b) 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* generally delivers the most number of messages and *Lazy* transfers almost the least number of messages in both cases of with and without data 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 **??** (a), 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 **??** (b), 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. To take full advantage of adapting sample rate, we need an intelligent adaptation scheme, which is our future work.

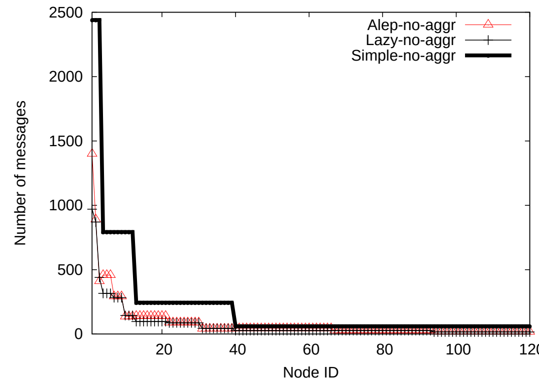

Figure 5. Number of delivered messages without aggregation.

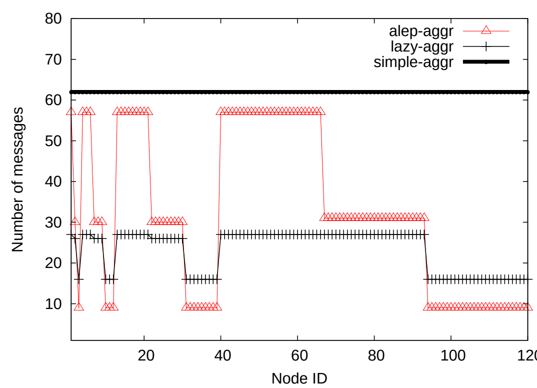

Figure 6. Number of delivered messages with aggregation.

6.4 Lifetime of Sensor Networks

Lifetime of WSN plays an important role in the most passive monitoring applications. Thus, in this section we check the possible lifetime of WSN might be achieved by using *Simple*, *Alep* and *Lazy*. We calculate the lifetime of the sensor network based on the energy consumption of each sensor as defined in [21]. In previous sections, we have seen that the energy is wasted using current implementation. So, we examine the lifetime of sensor network in the case of without unnecessary energy waste.

Figure 7 depicts the possible lifetime of a WSN can achieve by using *Alep*, *Lazy* and *Simple* in both cases of

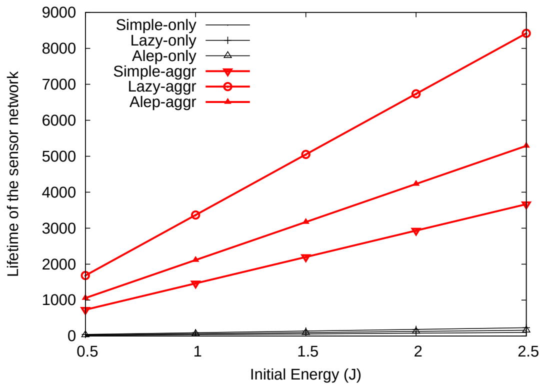

Figure 7. Comparison of lifetime of sensor networks using different approaches.

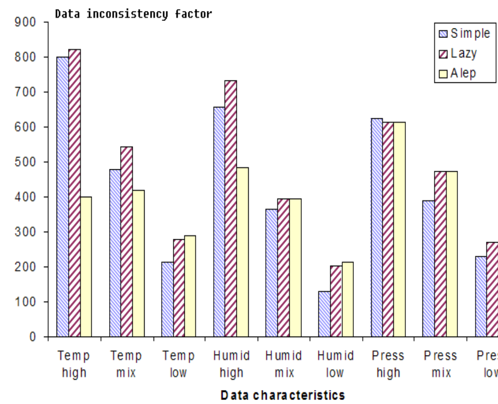

Figure 8. The results of data inconsistency factor.

with and without aggregation. The x-axis denotes the initial energy, and the y-axis shows the possible lifetime of WSN, which is defined as the maximum number of messages the sensor network can handle. In this experiment, three types of data, changing quickly, changing slowly, and changing quickly first then slowly, are collected at all sensors. From the figure, we can see that WSN largely extend the lifetime by using aggregation. The maximum lifetime without aggregation is 234, 6.3% of with aggregation (*Simple*). Moreover both *Alep* and *Lazy* can extend the lifetime by more than 50% and 150% as that using *Simple* because they can reduce the number of messages. Furthermore, *Lazy* usually has longer lifetime than that of *Alep*, because although *Alep* samples less data when the data dynamics is low, it samples more data when the data dynamics is high. The difference between *Alep* and *Lazy* will be further discussed later.

6.5 Data Inconsistency Factor

From above sections, we can see that *Lazy* and *Alep* can largely 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 8 reports the relationship between the data inconsistency factor and different monitoring parameters with variant data dynamics. In the figure, the x-axis is different data types with variant data dynamics and the y-axis represents the calculated data inconsistency factor 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 data inconsistency factor is larger, e.g., the `Temp high` has larger data inconsistency factor than `Temp mix` and `Temp low`, and `Temp high` also has larger data inconsistency factor 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 data inconsistency factor than that of *Simple* and *Lazy* when the data dynamics is high, while it has larger data inconsistency factor than that of *Simple* and has the same data inconsistency factor 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.

Furthermore, the data inconsistency factor 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., the data inconsistency factor exceeds the data consistency requirements of the application. However, *Alep* can keep the data inconsistency factor 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 6.3.

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 6.3 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. It is good to integrate those two approaches to collect accurate data in an energy-efficient way.

6.6 Tradeoff between Energy Efficiency and Data Consistency

We have already seen that *Lazy* and *Alep* can largely reduce the number of delivered messages so that they have potential to save energy and extend the lifetime of WSN, and *Alep* can sufficiently improve the data consistency. Now we are in position to examine the effect of two key factors related with *Alep*: *the temporal consistency bound* and *the value consistency bound*.

First let us consider the effect of the temporal consistency bound to the energy efficiency and data value consistency. If we release the temporal consistency of data, the same set of data will be delivered to the sink regardless of different arrival times. Thus changing the temporal consistency bound will not affect the data inconsistency factor of the collected data. However, releasing the temporal consistency bound does affect the number of delivered messages. Figure **??** (a) displays the relationship between the number of delivered messages and the different temporal consistency bounds ranging from 4 units to 7 units, which is the maximum time to transfer the data to the sink assuming each hop taking one unit time. In the figure, the x-axis is the ID of the sensors and the y-axis is the number of delivered messages.

From the figure, we can see that the increasing of the bound of temporal consistency results in the decreasing of the number of total delivered messages. When the temporal consistency bound is tight as 4, some sensors deliver more than 110 pieces of messages because data combination is not possible. While the temporal consistency bound is raised to 7, sensors deliver only about 50 pieces of messages. Thus, releasing the bound of temporal consistency can reduce the number of delivered messages. However, based on simulation data, the energy consumption almost keeps the same (overlapped in the figure) with the releasing of the bound of the temporal consistency as show in Figure **??** (b). This is because the same reason of idle listening as we mentioned in Section **??**, and we believe that releasing the temporal consistency bound will reduce the energy consumption by reducing the number of delivered messages. In this case a well designed schedule is needed to save energy from idle listening. This problem may be solved automatically in the new version of Motes, such as TelosB [17], which can automatically transfer to the sleeping state. We plan to implement *Alep* in a TelosB testbed next step.

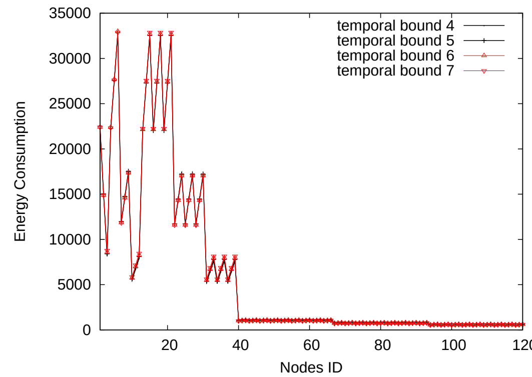

Figure 9. Energy consumption with variant temporal bound using *Alep*.

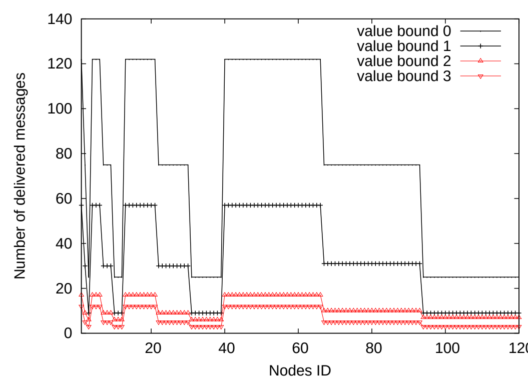

Figure 10. Number of messages with variant value bound using *Alep*.

Having seen that releasing the temporal consistency bound can reduce the number of the delivered messages, next, we examine the effect of the value consistency bound to the number of delivered messages and the data inconsistency factor. Figure **??** (a) shows the number of delivered messages with the relation to the variant value constraints. In the figure, the x-axis is the ID of the sensors and the y-axis shows the number of delivered messages. From the figure, we can see that when the value consistency bound is enlarged, the number of the delivered messages is decreased very fast. Next, we examine the changing of data inconsistency factor with the changing of the value consistency bound.

Figure **??** (b) shows the relationship between the data inconsistency factor and the value of the data consistency bound. The x-axis is the different value bounds and the y-axis depicts the value of the data inconsistency factor. In

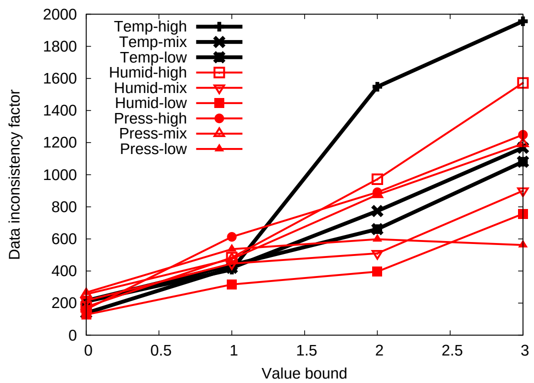

Figure 11. Data inconsistency factor with variant value bound using *Alep*.

the figure, when the data consistency bound is released, the data inconsistency factor increases very quickly, especially when the data dynamics is high. Thus we argue that there is a tradeoff between the data consistency and the energy efficiency. Releasing the data consistency bound results in both energy efficiency and larger data inconsistency factor, so the application should decide the data consistency bound based on its specific data consistency requirements. If the application cares little to the data consistency, it may raise the bound, otherwise, it has to use tightly bound. Data consistency is an application-specific requirements, therefore for future WSN applications, it is better to provide a set of interfaces (API or GUI) to allow application scientists to choose appropriate parameters by themselves, which is our future work.

7 Related Work and Discussions

In this paper, we mainly model the data consistency requirements of data collection in WSN and propose an adaptive, lazy protocol to collect sensing data in an energy efficient way. 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 goals in the design of WSN. Energy efficient protocols have been explored for a long time. Previous work expects to achieve the goal of energy efficiency by designing energy efficient routing protocols such as [20], energy efficient MAC protocols like [28], energy efficient clustering [29], and other energy efficient approaches [19]. However, these approaches mainly focus on finding some energy efficient paths, designing better turn on/off schedules, forming energy efficient clusters, and so on. And none of them has examined the energy efficiency from the view of the data itself, i.e., to adapt the data sampling rate to the data dynamics and keep lazy when data consistency is maintained.

Data consistency is a classical problem in computer architecture, distributed systems, database, and collaborative systems. Interested readers please refer to these textbooks [16, 18, 26]. 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 [22]. In their follow-up work in [5], they model the dynamics of the data items. Their work is similar to ours; however, their work is to collect data from the web, and our work is to collect data in WSN, which is more resource constraint. Moreover, we have different goal in data operations from theirs so that we use a different protocol, and our model for data consistency is more general than theirs. Lu *et al.* propose a spatiotemporal query service in [10], and their goal is to provide a service to enable mobile users to periodically gather information and meet the spatiotemporal performance constraints, but they propose neither data consistency models, nor adaptive protocols. However, their work complements to our effort very well, i.e., we can integrate their approach with our data consistency models by using their service in the scheduling of our protocol. Thus, as far as we know, this is the first model to define the data consistency in WSN.

Adaptive approach is always attractive in system design. Several adaptive protocols including [6] are proposed in literature. However, these protocols are mostly used in the cluster formation, communication patten selection, and duty cycle designing. None of them intends to adapt the data sampling rate according to the data dynamics. A recent paper [12] from Mainland *et al.* uses an adaptive approach to allocate the resource for sensor networks. They model sensors as self-interested agents and use price to tune the behavior of each sensor. They can also adapt the sampling rate to the data dynamics. However, they neither propose a formal model for data dynamics nor consider data consistency in their adaptive approach. Thus their approach is not from the perspective of the data, but from the view of every sensor's profit.

Several work about adaptive sampling rate has been proposed from researchers of database field, sharing the same goal of our Alep protocol. Jain and Chang propose an adaptive sampling for the sensor networks [8]. They employ a Kalman-Filter (KF) based estimation technique and the sensor uses the KF estimation error to adapt the sampling rate. Their approach is different from our approach in that it has to store more data. Moreover Kalman-Filter has matrix operation so that it is much more computing intensive. Marbini and Sacks [13] 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 [11] also adapts the sampling rate based on current network load conditions, but not based on the data dynamics in the data field. Their work complements to our work very well. More work on sampling rate adaption should be done by considering the network load condition and the data feature such as data dynamics and priorities together.

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, but they have not modeled the data consistency and considered adapting sample rate. Sharaf *et al.* study the trade off between the energy efficiency and quality of data aggregation in [23]. They impose a hierarchy of output filters on sensor network to reduce the size of the transmitted data. Data prioritization in TinyDB [11] chooses the most important samples to deliver according to the user-specified prioritization function, which is not as general as our work on data consistency and data dynamics.

8 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 from applications and the feature of data dynamics in the data field. Then, we formally model the data collection problem in the scenario of the passive monitoring application, with the goal of delivering the minimum number of messages under constraints of data consistency, and propose a formal definition for data consistency and data dynamics in WSN. Then, an adaptive, lazy, energy-efficient protocol is proposed to save energy as well as keep the data consistency. A comprehensive experiment has been designed to show the effectiveness and efficiency of the proposed protocol. The results from the simulation show that the proposed protocol indeed reduces the number of messages, saves energy and extends the lifetime of WSN. Finally, the tradeoff between the energy efficiency and data consistency is explored.

Considering the data consistency in the design of WSN is an interesting problem. We plan to extend our first try 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, in this paper, we assume sensors are willing and able to report the correct values; however, this is not true in reality. How to keep data consistency in a non-cooperative and malicious environment is our next step.

References

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

[2] Crossbow technology inc.

[3] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. In *In Proc. of the Fifth Symposium on Operating Systems Design and Implementation(OSDI)*, Dec. 2002.

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

[5] R. Gupta, A. Puri, and K. Ramamritham. Executing incoherency bounded continuous queries at web data aggregators. In *In Proc. of 14th International World Wide Web Conference*, May 2005.

[6] T. He, B. Blum, J. A. Stankovic, and T. F. Abdelzaher. Aida: Adaptive application independent data aggregation in wireless sensor networks. *ACM Transactions on Embeded Computing Systems*, 40(8):102–114, Aug. 2002.

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

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

[9] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: Accurate and scalable simulation of entire tinyos applications. In *Proceedings of the First ACM SenSys'03*, Nov. 2003.

[10] C. Lu et al. A spatiotemporal query service for mobile users in sensor networks. In *ICDCS05*, June 2005.

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

[12] G. Mainland, D. Parkes, and M. Welsh. Decentralized, adaptive resource allocation for sensor networks. In *In Proc. the 2nd Symposium on Networked Systems Design and Implementation*, May 2005.

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

[14] The nyquist-shannon sampling theorem.

[15] C. Olston, J. Jiang, and J. Widom. Adaptive filters for continuous queries over distributed data streams. In *in Proc. of the 2003 ACM SIGMOD international conference on Management of data*, June 2003.

[16] L. L. Peterson and B. Davie. *Computer Networks: A Systems Approach*. Morgan Kaufmann, 2003.

[17] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling ultra-low power wireless research. In *In Proc. of the 14th International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Apr. 2005.

[18] R. Ramakrishnan. *Database Management Systems*. WCB/McGraw-Hill, 1998.

[19] N. Sadagopan and B. Krishnamachari. Maximizing data extraction in energy-limited sensor networks. In *Proceedings of IEEE INFOCOM 2004*, May 2004.

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

[21] K. Sha and W. Shi. Modeling the lifetime of wireless sensor networks. *Sensor Letters*, 3(2):126–135, June 2005.

[22] S. Shah, S. Dharmarajan, and K. Ramamritham. An efficient and resilient approach to filtering and disseminating streaming data. In *In Proc. of 29th International Conference on Very Large Data Bases*, Sept. 2003.

[23] 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.

[24] V. Shnayder et al. Simulating the power consumption of large-scale sensor network applications. In *Proc. of ACM SenSys 2004*, Nov. 2004.

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

[26] A. Tanenbaum and M. Steen. *Distributed Systems: Principles and Paradigms*. Prentice-Hall, 2002.

[27] N. Xu et al. A wireless sensor network for structural monitoring. In *Proc. of ACM SenSys 2004*, Nov. 2004.

[28] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of IEEE Infocom'02*, New York, NY, June 2002.

[29] Q. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proc. of IEEE Conference on Computer Communications (INFOCOM'04)*, Mar. 2004.