Data Compression for Energy Efficient Communication on Ubiquitous Sensor Networks

Hsing-Shao Liu¹, Chi-Cheng Chuang², Chih-Chung Lin², Ray-I Chang²*, Chia-Hui Wang³ and Ching-Chia Hsieh¹

¹Department of Information Management, National Taiwan University, Taipei, Taiwan 10617, R.O.C.
²Department of Engineering Science, National Taiwan University, Taipei, Taiwan 10617, R.O.C.
³Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan, Taiwan 33348, R.O.C.

Abstract

Ubiquitous sensor networks (USNs) implemented by wireless sensor networks (WSNs) play a critical role for the next generation ubiquitous networks. Nowadays, the evolving “Internet of Thing”, which has been deemed as a mainstream of future Internet, raises even more the already high hopes of WSN and has attracted the interest of the research community worldwide. Since the sensor nodes are often equipped with the battery that usually cannot be replaced or recharged, the way to extend their lifetime becomes a critical issue. Because radio communication is the main cause of power consumption for sensor nodes, several approaches have been proposed to save power by compressing the volume of sensing data during communication. In this paper, we propose an efficient compression mechanism for WSN by treating sensing data as the raw data of an image for compression. We also introduce the user-acceptable data error which can be defined by a user to enhance the compression efficiency. Experimental results show that our mechanism can reach a better compression ratio compared with other approaches in either higher or lower correlated data scenario. More power can be saved during the radio communication due to lower volume of data. Furthermore, since our mechanism only involves addition and subtraction operations, the extra calculating burden resulted from compression can be minimized and thus, the extra power used to such calculation is negligible compared with the power used for transmitting a single bit data.

Key Words: Ubiquitous Sensor Network, Wireless Sensor Network, Internet of Thing, Data Compression, JPEG-LS

1. Introduction

Ubiquitous sensor network (USN), often implemented by wireless sensor network (WSN), plays a critical role for the next generation ubiquitous networks. It renders a wide range of application potential including environment and weather monitoring, object tracking, security surveillance, traffic control and so on [1–3].

With the help of the transparent interconnection between WSN and IP networks such as the work of [4], users are able to access the data delivered by WSN anytime and anywhere, realizing the meaning implied by the word “ubiquitous”. Nowadays, the evolving “Internet of Thing”, which has been deemed as a mainstream of future Internet, raises even more the already high hopes of WSN and has attracted the interest of the research community worldwide [5]. A general application scenario of WSN is shown in Figure 1. Each group of distributed sensors
nodes will transmit the sensed data to the node (we call this node as a super node) which is often nearest to the sink device. The sink device is usually connected with a computer to relay all the data it received from the super node to the Internet. Then, users can access the data measured and collected by WSN via Internet.

However, since these sensors are usually deployed in unattended or dangerous areas doing long-time measurements, in which recharging or replacing the battery is inconvenient, one of the challenges to WSN development is the power issue. Especially for super nodes, the heavy traffic will exhaust their energy more quickly than the other sensor nodes [6]. If the super nodes run out of energy, all the data cannot be delivered to the sink and then the lifetime of this WSN ends. Therefore, several power-saving techniques to maximize the lifetime of WSN have been proposed. These techniques can be classified as follows.

**Synchronization mechanism** [7]: Since the power consumption in the idle state of a wireless communication module is almost equal to transmission or reception, it uses the sleep/wakeup scheduling algorithm to make the sensor nodes work or sleep at the appropriate time in synchronization.

**The shortest path** [7]: Since the WSN is a multi-hop network, and the power consumption will be proportional to the hops sensing data passed by. This manner will make a power saving by finding the shortest path for the sensor data to the sink node.

**In-network processing** [8,9,10–13]: Using the data compression or data aggregation techniques to remove the redundancy by exploiting the spatial or temporal correlation in the sensing data.

Along the course of “in-network processing”, it is observed that the utilization of data processing on microcontroller and wireless communication operation are the main causes of power consumption and the power used to transmit a single bit is equal to the consumption amount of processing thousands of commands [8]. Therefore, decreasing the transmission volume or frequency is able to help reduce power consumption. Following above idea, in this paper, we propose one more efficient data compression mechanism for power saving by modified a standard image compression technique called JPEG-Lossless (JPEG-LS). We believe that if the compression ratio can be effectively enhanced, more power can be saved due to the reduction of transmission volume. The advantages of our proposed approach will be demonstrated in the later simulation experiment. This paper is organized as follows. We introduce the related researches on the in-network processing in section 2. Then the proposed algorithm is described in section 3. The simulation results are illustrated in section 4. The future work and conclusion are made in section 5.

### 2. Related Works

In this section, we will briefly review the typical techniques used to address the power issue of WSN by data compression in the course of in-network processing. Their distinctive features and limitation will be highlighted and described in the following.

**Pipelined In-Network Compression** [10]: In this approach, the data compression capability comes from sharing the common parts in the data format. The time of sensing data staying in the aggregation node is extended to combine sensing data into one packet and remove the redundant parts. Before compression, the format of sensing data in the WSN is <sensing data, node ID, timestamp>. After compression, the format will be transformed as <shared prefix, suffix list, node ID list, timestamp list>. Shared prefix is the common part of the sensing data while the suffix list is the remaining part.

![Figure 1. Application scenario of USN over Internet.](image-url)
Node ID list is the conjunction of sensor ID and timestamp is the conjunction of the time when the sensing data has been measured. Figure 2 illustrates such data compression approach. Through shared the common parts (i.e. 100), the total bits can be reduced from 33 bits to 27 bits. The more common parts the sensing data has, the higher compression ratio this approach can acquire. Specifically, this approach can work efficiently when the data have high correlation. But, when the high correlation no longer exists in the sensing data, an inefficient compression ratio is expected due to few shared common part which can be identified.

**Static Huffman Coding (SHC)** [11]: This SHC approach proposed a simple compression algorithm to convert analog signals to digital signals which is installed on a sensor node to collect data. For data compression, this method used the DC coefficients (brightness component) codebook in JPEG baseline algorithm with the assumption that the sensing data are continuous and high correlated. Then it calculated the difference $d_i$ between the former and the latter sensing data. Since the codebook of DC coefficients is known before coding with the probabilities of the sensing data, it can compress $d_i$ with high compression ratio. Although this method is simple and easy to implement, it can only be applied when the statistics of the sensing data are known beforehand to assign the probabilities for Huffman coding in the DC coefficients’ codebook.

**Modified Adaptive Huffman Coding (MAH)** [12]: This approach proposed MAH which improved the SHC [11] and did not require the knowledge about statistics of the sensing data for assigning the probabilities in advance. The original adaptive Huffman coding (AHC) [13] will construct the binary tree during the coding process. In the beginning, there is only one node named Not Yet Transmitted (NYT) from which the binary tree is constructed. Finally, the code will consist of two parts. Prefix indicates the code by traversing the tree which includes data. Suffix indicates the binary representation of the sensing data.

In MAH [12], the sensor node just transmits the code by combining the prefix and suffix parts. This brings a problem where the initial data will have few probabilities to get a shorter code since the initial probability can only be assigned after the binary tree is first constructed. MAH combined both the advantages of the SHC and AHC. The leaves of the tree are assigned with a set of symbols rather than one single symbol in [11]. The code is changed and finally made up of two parts. Prefix indicates the code by traversing the tree which includes the data. Suffix indicates the array index of the data appearing in the set of the symbol. The simulation result showed that MAH had a better compression ratio in the medially correlated data. But, it still had a poor compression ratio in the high correlated data in comparison with SHC [11].

### 3. Proposed Method

Since the sensor nodes can be deployed under various application domains, the correlation of sensing data is also varied. Our goal is to reach a good compression ratio in the data either with a high correlation or with a low correlation. Since the application of sensor nodes is often used to report the status of interested area to users [14,15], to represent the status in an image is intuitive. Users can learn about the measurements or data distribution from the image. At this observation, we propose a WSN architecture using improved JPEG-LS for data compression. JPEG-LS is based on the LOCO-I [16] and is one of the JPEG operation modes [17]. The sensing data are treated as raw data of a pixel. These raw data will be collected to form an image and then compressed through the improved JPEG-LS. It is expected that the compression ratio will be improved after a period of collection. Figure 3 illustrates the details of the modeller in JPEG-LS. The images sampled by the modeller will be calculated $g_1 = d - a$, $g_2 = b - c$ and $g_3 = c - a$ through the gradient module, and these differences represent the local gradient. The samples are determined as in the float region while $g_1 = g_2 = g_3 = 0$. Then the samples will be sent into Run mode. Otherwise, the samples

![Figure 2. Pipelined in-network compression.](image-url)
will be sent into Regular mode. The fixed predictor is depicted in (1). An edge will be detected by \( a, b \) and \( c \). When \( c \geq \max(a, b) \), a horizontal edge is detected by \( \max(a, b) = a \) and a vertical edge by \( \max(a, b) = b \). Similarly, when \( c \leq \min(a, b) \), a horizontal edge is detected by \( \min(a, b) = a \) and a vertical edge by \( \min(a, b) = b \).

\[
\hat{\text{MED}} = \begin{cases} 
\min(a, b) & \text{if } c \geq \max(a, b) \\
\max(a, b) & \text{if } c \leq \min(a, b) \\
 a + b - c & \text{otherwise}
\end{cases}
\]  

(1)

The JPEG-LS also provides the lossy compression called near lossless compression. It uses the Differential Pulse Code Modulation (DPCM) for coding the difference between the sample and predicted value in the modeler. This is tantamount to allowing an error \( \Delta \) between the samples and the original data. The specification [18] treats the lossless mode as a special case of near-lossless compression with \( \Delta = 0 \). Based on the mechanism of JPEG-LS, we improve and build it in our proposed WSN architecture, in which the WSN network can be logically classified into three functional layers as shown in Figure 4.

**Layer 3:** The interested areas can be further clustered into several sub-regions. For example, in Figure 4, four sub-regions are clustered in terms of A, B, C, and D. There are 64 end devices in each sub-region. The end devices are responsible for sensing the data and transmitting it back to the super node in layer 2.

**Layer 2:** The super node is responsible for processing and forwarding the data to sink. It integrates the sensing data from layer 3 to construct an image and then compresses it through the improved JPEG-LS.

**Layer 1:** The action of the sink node is determined by the data it received from layer 2. There are two categories of the received data. If the sink node receives the index, it will display the image with the correspondent index in the image table (the codebook); whereas if the sink node receives the lossless image, the sink node will store it into the image table and then display it.

The WSN operation from layer 1 to layer 3 can be generalized as two phases: data pre-collection and JPEG-LS predictor modification. The purpose of the data pre-collection is for transmitting a shorter symbol after a period of time as illustrated in Figure 5(a). It contains two stages: initial stage and compression stage.

In the initial stage as shown in Figure 5(b), the system will take some time to collect the lossless image \( P_j \). If there is no identical \( P_j \) in the image table, the super node will store it into the image table. The image table

---

**Figure 3.** The modeler in JPEG-LS.

**Figure 4.** The hierarchy of WSN with improved JPEG-LS.
has two columns: index and lossless image. The indexes $x_1, x_2, \ldots, x_i$ represent any integer which is smaller than $k$. After a period of time $t_k$, the system will start the compression stage. The period of time $t_k$ is determined by the correlation of sensing data. Specifically, it takes a shorter time $t_k$ with the high correlated data while in the low correlated data a longer time $t_k$ is necessary for better compression efficiency.

In the compression stage as shown in Figure 5(c), the user-acceptable error $\Delta$ defined by users is taken into consideration. The error $\Delta$ can be evenly distributed into all the sensing data by adding the error into JPEG-LS. It will be less accurate due to the subtraction of $P_{t_{k+n}}$ and $P_{t_k}$ without adding the error to the JPEG-LS. When the lossless image $P_{t_{k+n}}$ is obtained in layer 2, it will be processed by lossy and lossless compression. The lossless image will be stored into the image table temporary. If we can find an image in the image table within an error $\Delta$ after making the subtraction of samples and lossless images in the image table, the correspondent index will be transmitted. On the contrary, the lossless image will be transmitted back to layer 1 if there are no allowable

![Diagram of data pre-collection stages](image-url)
error degrees within the user-acceptable range.

To improve the compression efficiency, we add the user-acceptable error $\Delta$ and modify the predictor in JPEG-LS. For instance, we can make the compressed data with an error $\Delta$ by using the gradient computation. If $g_1 \leq \Delta$, $g_2 \leq \Delta$, and $g_3 \leq \Delta$, the subtractions between every pixel on the image will be less than or equal to error $\Delta$. The image which satisfies Eq. (2) will be user-acceptable [16].

$$\left| P_{h,a}(x,y) - P_i(x,y) \right| \leq \Delta$$

(2)

The subtractions between $P_{h,a}(x,y)$ and $P_i(x,y)$ will be greater than error $\Delta$ with some sudden drastic changes in the environment which will extensively increase or decrease the sensing data in a short time. In these situations, the lossless image stored temporary in the image table in layer 2 will be transmitted to the image table in layer 1 for updating. Generally, the sensor nodes are often used to measure temperature, humidity or other climate-related figures [19,20]. Therefore, our modification of JPEG-LS predictor is based on the application domain of temperature measurement. To enhance the efficiency of JPEG-LS predictor for the image produced by our topology, we keep the edge detection in the predictor. As shown in Figure 6, given that the formation of the images in layer 2 is grid-patterned, if we detect the cross area of four separate pixels, we return the sample value $x$ directly.

4. Simulation Results

We use the Network Simulator 2 (NS2) for simulation experiment. The topology of the WSN network is based on Figure 4. The compression ratio is calculated as Eq. (3).

$$\text{Compression Ratio} = (1 - \frac{\text{Compressed data}}{\text{Original data}})$$

(3)

Our simulation uses the temperature data on MARS acquired from Planetary Atmospheres Node of the Planetary Data System (PDS) in the NASA [21]. We create two sets of data with different features. One set is low correlated data while the other is high correlated. The auto-correlation of the data is illustrated in Figure 7, computed by the PAST software [22]. The simulation parameters are defined in Table 1. Each end device will sense the data once per minute. The user-acceptable error $\Delta$ is determined by the real image which is received in layer 1. When the error is higher than 5°C, there will be a significant error on the image.

In Figure 8(a), the simulation result shows the relations between the time and compression ratio under the varied error $\Delta$ in high correlated data. Significant changes of the compression ratio can be observed in the time in-

![Figure 6. The image obtained by super node in layer 2.](image)

![Figure 7. Auto-correlation of low (a) and high (b) correlated temperature data on MARS.](image)
interval 0 to 125 minutes. In the beginning, transmitting the lossless image at initial stage leads to the compression ratio at 52.4%. After a period of time for data collection, the compression ratio can be improved. Moreover, when the user-acceptable error $\Delta$ becomes larger, the time for reaching the same compression ratio will be reduced gradually. When more data can be collected over time, the probability to transmit an index rather than the lossless image will be higher. After 125 minutes elapsing, the compression ratio will approach about 85%.

In Figure 8(b), the simulation result shows the relations between the time and compression ratio under the varied error $\Delta$ in low correlated data. A longer period of time for data collection is needed to attain better compression efficiency. We can observe that there are significant changes for the compression ratio from the 80th minute to the 125th minute. The test cases with higher user-acceptable errors will improve their compression ratios faster and more significantly than those with lower ones. For example, the compression ratio in the test case with user-acceptable error of 5 starts to be improved at the 83rd minute and reaches 63.2% at the 125th minute. On the contrary, the compression ratio in the test case with user-acceptable error of 1 starts to be improved at 117th minute and reaches only 53.6% at 125th minute. When there is no user-acceptable error (i.e. error $\Delta = 0$), the compression ratio is only 52.4% in both experiments with the two different data correlations.

To further demonstrate our approach, we compare the approaches of [11] and [12] in the same simulation environment. The compression performances of these approaches are examined in both high and low correlated data scenarios at the user-acceptable error 5. The performance result in high correlated data scenario is shown in

<table>
<thead>
<tr>
<th>Table 1. Simulation parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
</tr>
<tr>
<td>Topology</td>
</tr>
<tr>
<td>Protocol</td>
</tr>
<tr>
<td>Nodes</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Simulation time</td>
</tr>
<tr>
<td>Compression Ratio for index transmission</td>
</tr>
<tr>
<td>Compression Ratio for lossless image transmission</td>
</tr>
<tr>
<td>User-acceptable error $\Delta$ (°C)</td>
</tr>
</tbody>
</table>

![Figure 8](image-url)
Figure 9(a). Our approach can reach better compression efficiency than the other approaches. If more sensing data can be collected over time, for example, after a period of 30 minutes, our approach (almost reaches the compression ratio of 85%) will surpass SHC by 13% and MAH by 45%.

The performance result in low correlated data scenario is shown in Figure 9(b). A longer time for collecting data is needed to obtain better compression efficiency. Since the 85th minute our approach can have better compression efficiency than MAH in the low correlated data. If more sensing data can be collected over time, for example, after a period of 100 minutes, our approach (almost reaches the compression ratio of 67%) will surpass SHC by 45.3% and MAH by 25.9%.

In Table 2, we compare the time latency of SHC, MAH and AHC for given compression ratio with different user-acceptable errors and data correlation types. It indicates that the high correlated data can achieve higher compression ratio than low correlated data. As the power consumption of transmitting a single bit is the same as executing thousands of instructions in sensor node, it’s worthy to spend time to achieve good compression ratio as the method we propose in this paper. Furthermore, if user can more tolerate the data error in some acceptable range, the price of time latency can be reduced. We have demonstrated the compression performance of the proposed approach. Our approach can have better compression efficiency than other typical compression approaches used in WSN regardless of the col-

![Image of Figure 9](image_url)

**Figure 9.** Time over compression ratio with different methods.

<table>
<thead>
<tr>
<th>User acceptable error</th>
<th>SHC with 74.3% compression ratio</th>
<th>MAH with 57.9% compression ratio</th>
<th>AHC with 54.6% compression ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>High correlated data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&gt; 200 minutes</td>
<td>136 minutes</td>
<td>124 minutes</td>
</tr>
<tr>
<td>2</td>
<td>161 minutes</td>
<td>67 minutes</td>
<td>60 minutes</td>
</tr>
<tr>
<td>3</td>
<td>67 minutes</td>
<td>26 minutes</td>
<td>23 minutes</td>
</tr>
<tr>
<td>4</td>
<td>34 minutes</td>
<td>13 minutes</td>
<td>11 minutes</td>
</tr>
<tr>
<td>5</td>
<td>9 minutes</td>
<td>4 minutes</td>
<td>3 minutes</td>
</tr>
<tr>
<td>Low correlated data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>&lt; 1 minute</td>
<td>123 minutes</td>
<td>&lt; 1 minute</td>
</tr>
<tr>
<td>2</td>
<td>&lt; 1 minute</td>
<td>116 minutes</td>
<td>&lt; 1 minute</td>
</tr>
<tr>
<td>3</td>
<td>&lt; 1 minute</td>
<td>104 minutes</td>
<td>&lt; 1 minute</td>
</tr>
<tr>
<td>4</td>
<td>&lt; 1 minute</td>
<td>87 minutes</td>
<td>&lt; 1 minute</td>
</tr>
<tr>
<td>5</td>
<td>&lt; 1 minute</td>
<td>85 minutes</td>
<td>&lt; 1 minute</td>
</tr>
</tbody>
</table>
lected data with higher or lower correlation. If more data volume can be compressed, the power used to transmission can be saved. Although additional calculating for compression process is needed in each super node, our proposed approach only involves addition and subtraction operations which are able to minimize the extra calculating burden for super node.

5. Conclusion and Future Works

For the emerging ubiquitous network of next generation, USN which can be implemented by WSN has an undeniable role. It is a beginning step to toward the ubiquitous network, and when lots of local USNs/WSNs connected to the Internet, we can access data which are directly related to our daily lives anytime and anywhere including environment and weather monitoring, object tracking, security surveillance, and traffic control. Since extending the lifetime of each sensor node in WSN is one of the critical issues for its future development, several approaches are proposed to address the power issue recently. Given that the power used to transmit a single bit is equal to the consumption amount of processing thousands of commands [8], in this paper, we propose a WSN with a more efficient compression capability by using improved JPEG-LS. All the sensing data are treated as the raw data of an image. Then we compress the image through the improved JPEG-LS. Better compression efficiency can be obtained by taking the user-acceptable error of an image into consideration. If more data volume can be compressed, it implies more power used to transmission can be saved at the little price of spending additional calculating resource in each super node. Since the proposed approach only involves addition and subtraction operations, the extra burden resulted from such calculating can be minimized. Our proposed approach is demonstrated by the simulation experiments with high and low correlated data. It takes a longer time for data collection in low correlated data while a shorter time in high correlated data. We also make a comparison with other typical compression approach in WSN. The result shows that our approach has better compression efficiency in either high or low correlated data scenario. Moreover, we also present the time needed for reaching a specific compression ratio under the different user-acceptable error $\Delta$. The system can be chosen according to the user-acceptable error $\Delta$ to reach the low or high compression ratio with different time interval.

Our future works are to improve the proposed approach from the dimensions of topology, computation, latency, and predictor. Regarding the network topology, what we take into consideration in this paper is grid-patterned. But in practice, the topology deployments may be random in some specific situations. We will investigate the influence of different topologies on the compression efficiency. Regarding the computation, although super node in layer 2 can save a lot of power consumption in transmitting data at the little price of spending additional calculating resource, the computational burden on super node is still heavy. We will address this issue by using the distributed and cloud computing. Regarding the latency, the size of image table is proportional to the sampling time. Although each sensor node often has an expansion module to insert a memory card for storing the large data, the longer time will be needed for searching a specific item when the volume of data stored in the memory are too large. As a result, how to reduce such searching latency becomes one of our future works. Moreover, we are going to design the training predictor which may be more suitable to cope with the various topologies and environments adaptively.

Acknowledgement

This paper is partially supported by NSC Taiwan under grants NSC 98-2218-E-002-042-.

References


[22] The PAST website [Internet]. Available from: http://folk.uio.no/ohammer/past/.

[23] The IEEE 802.15.4 module for NS2 website [Internet]. Available from: http://www-ee.ccny.cuny.edu/zheng/pub/.

*Manuscript Received: Dec. 23, 2010*  
*Accepted: Dec. 31, 2010*