# An Effective and Efficient Method for Handling Transmission Failures in Sensor Networks

Heejung Yang and Chin-Wan Chung

Korea Advanced Institute of Science and Technology, Korea
heejung@islab.kaist.ac.kr, chungcw@kaist.edu

**Abstract.** The suppression scheme is a solution for limited energy constraints in sensor networks. Temporal suppression, spatial suppression and spatio-temporal suppression are proposed to reduce energy consumption by transmitting data only if a certain condition is violated. Among these suppression schemes, spatio-temporal suppression is the most energy efficient than others because it combines the advantages of temporal suppression and spatial suppression. A critical problem of these suppression schemes is the transmission failure because every nonreport is considered as a suppression. This causes the accuracy problem of query results. In this paper, we propose an effective and efficient method for handling transmission failures in the spatio-temporal suppression scheme. In order to detect transmission failures, we devise an energy efficient method using Bloom Filter. We also devise a novel method for recovering failed transmissions which can save energy consumption and recover failed values more accurately. The experimental evaluation shows the effectiveness of our approach. On the average, the energy consumption of our approach is about 39% less than that of a recent approach and the accuracy of the query results of our approach is about 55% more accurate than that of the recent approach in terms of the error reduction.

**Keywords:** Sensor networks, Spatio-temporal suppression, Transmission failures.

## 1 Introduction

Sensor networks give us new opportunities for observing and interacting with the physical world. They are composed of a large number of sensor nodes and each sensor node has capabilities of sensing, processing, and communication. These sensor nodes are deployed in environments where they may be hard to access and provide various useful data. Habitat and environmental monitoring are representative applications of sensor networks. For example, in Great Duck Island project[6], sensor nodes were deployed in the nests of the storm petrels and biologists can collect various scientific data to analyze their lifestyle.

While sensor networks enable continuous data collection on unprecedented scales, there are challenges because of the limited battery resources on each sensor node. Since sensor nodes are usually deployed in an unattended manner, it is

X. Zhou et al. (Eds.): DASFAA 2009, LNCS 5463, pp. 92–106, 2009.

not easy to replace their batteries. Therefore reducing energy consumption is a major concern in sensor networks. Batteries of sensor nodes are depleted by sensing, computation, and communication. Among these tasks, communication is the primary source of energy consumption. Thus several techniques were proposed to resolve the limited energy constraint by reducing the communication.

The suppression is one solution to reduce communication using the temporal/spatial correlation of sensor readings. Each sensor node transmits its sensor reading only if the value of the sensor reading violates a certain condition related to the temporal/spatial correlation. In the temporal suppression scheme, sensor nodes do not transmit their readings, if the current reading is similar to the last transmitted reading. The base station assumes that any nonreport values are unchanged from the previously received ones. In the spatial suppression scheme, there are several groups of sensor nodes having similar sensor readings. Each group has one leader node and this leader node reports for its group. Both of these suppression schemes can lower energy consumption by reducing the number of transmissions. And it is possible to combine these two approaches. For example, sensor nodes are grouped by using the spatial correlation, and the leader and member nodes use the temporal suppression. The leader node makes the representative value for the group based on sensor readings received from its member nodes. This approach can greatly reduce the communication using the temporal and spatial correlation of sensor nodes.

However this suppression has a critical weakness. This problem is caused by the fact that every nonreport is considered as a suppression. Sensor networks are prone to transmission failures due to interference, obstacles, and congestions, etc. In the suppression scheme, these transmission failures create ambiguity. A nonreport may either be a suppression or a failure but there is no way to differentiate between them. [10] proposed a framework, BaySail (*BAYesian analysis of Suppression and fAILures*), to deal with transmission failures in the suppression scheme. Each node adds some redundant information on every report which consists of the last $r$ transmission timestamps and direction bits indicating whether each report is increased or decreased compared to the previously reported sensor reading. Using this information, the base station estimates missing readings using the Bayesian inference. Therefore the missing readings are estimated more accurately. However this approach cannot be applied to the spatio-temporal suppression scheme.

The spatio-temporal suppression scheme is to combine the advantages of both spatial and temporal suppression schemes and can reduce more energy consumption. In this suppression scheme, leader nodes make their representative values based on their member nodes' sensor readings, and transmit these representative values to the base station. Therefore transmission failures can be classified into two categories such as failures within each group and failures from each leader node to the base station. BaySail only considered the latter transmission failures. It is difficult to apply BaySail approach on transmission failures within groups because the Bayesian inference is complex and time-consuming while leader nodes have very limited resource constraints compared to the base station. If we do not resolve transmission failures within groups in using the spatio-temporal suppression scheme, the leader node will treat every nonreport as a suppression and make the representative value based on these inaccurate data and then transmit this to the base station. Thus we need a new method to deal with transmission failures more effectively in the spatio-temporal suppression scheme.

In this paper, we propose an effective and efficient method to resolve the transmission failures in the spatio-temporal suppression scheme. Our approach can detect transmission failures more energy efficiently and recover the failed values more accurately. The contributions of this paper are as follows:

- We extend the transmission failure handling for the spatio-temporal suppression scheme.
- We devise an energy efficient method to distinguish a suppression and a transmission failure while considering resource constraints of leader nodes. Some additional information related to the previous transmissions has to be added on every report to distinguish suppression and transmission failure although this increases energy consumption. We propose a method to represent this information compactly and to identify transmission failures effectively.
- We devise an effective method to recover the value of a failed transmission. Since sensor networks have very limited energy constraints, we propose an energy efficient method to notify the transmission failure to the sender while recovering the failed value more accurately.
- We experimentally evaluate our approach against other approaches to deal with transmission failures. Experimental results show the effectiveness of our approach in the spatio-temporal suppression scheme. On the average, the energy consumption of our approach is about 39% less than that of BaySail and the accuracy of the query results of our approach is about 55% more accurate than that of BaySail in terms of the error reduction.

The remainder of this paper is organized as follows. Section 2 reviews related works of reducing energy consumption and handling transmission failures in sensor networks. In Section 3, we describe our proposed approach to resolve transmission failures in the spatio-temporal suppression scheme. The experimental results are shown in Section 4. Finally, in Section 5, we conclude our work.

## 2 Related Work

The suppression is proposed to reduce energy consumption in sensor networks. In suppression, data is transmitted only if a certain kind of condition is violated. Suppression schemes can be classified into three categories such as temporal suppression, spatial suppression and spatio-temporal suppression.

The temporal suppression scheme uses the temporal correlation of sensor readings. If the current value is different from the previously transmitted value by more than a certain threshold, this value is transmitted. [7] and [4] use temporal suppression. [7] uses bounded filters to suppress stream data. If the current value lies inside a bounded filter, the data source does not transmit this value. In [4], dual Kalman Filter is used to suppress data as much as possible. The server activates a Kalman Filter $KF_s$ and at the same time, a sensor activates a mirror Kalman Filter $KF_m$. The dual filters $KF_s$ and $KF_m$ predict future data values. Only when the filter at a sensor $KF_m$ fails to predict future data within the precision constraint then the sensor sends updates to $KF_s$.

The spatial correlation between sensor nodes is used in the spatial suppression scheme. In this scheme, a node suppresses its sensor reading if it is similar to those of its neighboring nodes. There is a group of nodes having similar values and one node is selected to represent the group. [5] proposes the spatial suppression scheme using a small set of representative nodes. These representative nodes constitute a network snapshot and are used to provide approximate answers to user queries.

The spatio-temporal suppression scheme combines the advantages of both of the above schemes. [2] uses replicated dynamic probabilistic models for groups. These groups are made using the disjoint-cliques approach and then build models for each of them. Both the base station and sensor nodes maintain a pair of dynamic probabilistic models of how data evolves and these models are kept synchronized. The base station computes the expected values of sensor readings according to the model and uses it as the answer. Data is transmitted only if the predicted value is not within the error bound. [9] suggests a novel technique called CONCH (*Constraint Chaining*) combining both suppression schemes. Based on the minimum spanning forest covering all sensor nodes, CONCH temporally monitors spatial constraints which are differences in values between neighboring nodes. For each edge in the minimum spanning forest, one node is designated *updater* and the other one is designated *reporter*. *Updater* triggers a report if its value has changed. *Reporter* triggers a report if the difference between its node's value and *reporter*'s value has changed. The set of reports collected at the base station is used to derive all node values. To cope with transmission failures, it uses multiple, different forests over the network. Failure probabilities are integrated into edge weights to get more reliable forests. If transmission failure occurs, reported difference values from each forest are inconsistent. To recover the failed values, the maximum-likelihood approach is used.

All these suppression schemes can reduce energy consumption by suppressing the transmission. But they do not consider transmission failures except CONCH. But the method used in CONCH is only applicable to their approach. In a general suppression scheme, since every nonreport is considered as a suppression, it causes the accuracy problem. Because transmission failures are prone to sensor networks, it is needed to resolve this problem. [10] addresses this problem and proposes a solution in the temporal suppression scheme. It adds some redundant information about the previous transmissions to every report and infers the failed value using the Bayesian inference. However, this approach does not consider the spatio-temporal suppression scheme. In the spatio-temporal suppression scheme, there are two kinds of transmission failures which are failures from a member node to a leader node and failures from a leader node to the base station. Although this approach can be applied to failures from a leader node to the base station, it cannot be used at a leader node due to resource constraints of sensor nodes. Therefore we need a new approach to resolve transmission failures in the spatio-temporal suppression scheme.

## 3 Proposed Approach

The goal of our approach is to resolve transmission failures more energy efficiently and accurately in the spatio-temporal suppression scheme. We assume that sensor nodes are grouped according to the spatial correlation and each group has a leader node. Each leader node makes a representative value for its group and transmits this to the base station using a temporal suppression policy. In other words, the transmission will occur only when the current representative value for a group has been changed more than a user-specified threshold compared to the previously transmitted one. We consider the cases when the user queries are in the form of an aggregation of sensor readings. Therefore, the representative value of a leader is the aggregation of sensor readings of the group to which the leader belongs. For the general cases where the user queries require individual sensor readings, a leader sends all received readings after possible compression. Therefore, the extension of the proposed method to general cases is straightforward. The representative value for each group is made by aggregating over received values and the previous values for the suppressed values. Any aggregation functions such as MIN, MAX, SUM, and AVG can be used for making the representative value. Member nodes are also using the temporal suppression. We assume that the min value and the max value of sensor readings are known previously. In this section, we describe the details of our approach to handle transmission failures in the spatio-temporal suppression scheme.

### 3.1 Overall Approach

In the spatio-temporal suppression scheme, there are two kinds of transmission failures, that is, failures from a member node to a leader node and those from a leader node to the base station. As mentioned, since sensor nodes have very limited resource constraints, we cannot apply a complex method to resolve transmission failures from a member node to a leader node. Therefore, we devise a method which does not require much resource to deal with transmission failures within groups.

First, we have to distinguish a suppression and a transmission failure. We use the compressed history of the previous transmissions to distinguish them. This history information consists of timestamps of the previous transmissions and we compress the history information using Bloom Filter. Based on this compressed history information, we can identify transmission failures using the membership test of Bloom Filter.

**Table 1.** Notation

| Notation | Description |
|---|---|
| $T_{transmitted}$ | the set of timestamps of previous transmissions |
| $T_{succeeded}$ | the set of timestamps of succeeded transmissions |
| $T_{failed}$ | the set of timestamps of failed transmissions |
| $T_{suppressed}$ | the set of timestamps of suppressions |
| $T_{test}$ | the set of timestamps for transmission failure detection |
| $t_{last}$ | the timestamp of the last received data |
| $t_{current}$ | the timestamp of the currently received data |
| $t_{prevS}$ | the earlier timestamp of a pair of adjacent timestamps in $T_{succeeded}$ |
| $t_{nextS}$ | the later timestamp of a pair of adjacent timestamps in $T_{succeeded}$ |
| $t_{duration}$ | the difference between $t_{prevS}$ and $t_{nextS}$, i.e. $t_{duration} = t_{nextS} - t_{prevS}$ |
| $v_{prevS}$ | the sensor reading value at $t_{prevS}$ |
| B | compressed history information (Bloom Filter) |
| m | Bloom Filter size |
| $h_1, h_2, ..., h_k$ | k hash functions used in Bloom Filter |
| B[$h_i$] | the bit position in Bloom Filter by applying $h_i$ |
| $x$ | sampling interval specified in the user query |
| $\delta$ | user-specified error threshold |
| $D_{received}$ | data buffer for successfully received data |
| $D_{sent}$ | data buffer for transmitted data |

After detecting transmission failures, a leader node or the base station requests the retransmission for failures to senders. A retransmission request is constituted of two successfully transmitted timestamps having failed transmissions between them and the quantization is applied to reduce the size of this.

When a member node receives the retransmission request, it can identify two successfully transmitted timestamps having transmission failures between them. Thus the node can calculate differences between failed values and the sensor value of the first timestamp in the retransmission request. The quantization is applied to each difference and this data is transmitted.

Finally, the leader node can recover failed values using the received quantized value. We can recover the range of a failed value based on the sensor value of the first timestamp in the retransmission request. We assign the average value of the range to the corresponding failed value.

In the case of transmission failures from a leader node to the base station, we can apply our approach or the Bayesian inference of BaySail because the base station has no resource constraints. But our approach shows better performance than BaySail according to experimental results.

Table. 1 summarizes the notation used in this paper.

### 3.2 Transmission Failure Detection

In the suppression scheme, if we don't use any specific method to distinguish a suppression and a transmission failure, every nonreport is considered as a suppression. This causes the accuracy problem of query results. Therefore we add timestamps of previous transmissions on every report and use Bloom Filter[1] to compress this history of transmissions. Bloom Filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set.

Fig. 1 shows the algorithm of compressed history information. We make $m$-bit Bloom Filter to represent the set of timestamps of previous transmissions

```
Algorithm CompressHistoryInfo
Input Previously transmitted timestamp set T_transmitted = {t_1, t_2, ..., t_n}
Output m-bit Bloom Filter B

begin
1.   for each timestamp t in T_transmitted
2.      Compute h_1, h_2, ..., h_k
3.      Set B[h_1(t)] = B[h_2(t)] = ... = B[h_k(t)] = 1
4.   return B
end
```

**Fig. 1.** Algorithm of Compressed History Information

```
Algorithm FailureDetection
Input Received Bloom Filter B, t_last, t_current, D_received
Output The set of failed transmission timestamps T_failed,
       the set of succeeded transmission timestamps T_succeeded

begin
1.   t_test := t_last
2.   while t_test < t_current
3.      t_test += x
4.      Insert t_test into the set of test timestamps T_test
5.   for each timestamp t in T_test
6.      for each hash function h_i
7.         Compute h_i(t)
8.         if all B[h_i(t)] == 1
9.            if data of t is not in the data buffer D_received // Transmission Failure
10.              Insert t into the set of failed timestamps T_failed
11.           else // Transmission Success
12.              Insert t into the set of succeeded timestamps T_succeeded
13.        else // Suppression
14.           Insert t into the set of suppressed timestamps T_suppressed
15.  return T_failed, T_succeeded
end
```

**Fig. 2.** Algorithm of Transmission Failure Detection

$T_{transmitted}$. It has $n$ timestamps. The number of histories (the number of timestamps in $T_{transmitted}$) affects the accuracy of query results. The consecutive transmission failures cause the history information losses. If the number of histories is large, it consumes more energy but lowers the loss rate of the history information. We vary the number of histories in our experiments and show its effect on the energy consumption and the accuracy. For each timestamp in $T_{transmitted}$, we compute $k$ hash functions $h_1, h_2, ..., h_k$ with range $\{0, ..., m-1\}$ and all bit positions B$[h_1(t)]$, B$[h_2(t)]$, ..., B$[h_k(t)]$ are set to 1 in Bloom Filter (Line (2), Line (3)). This Bloom Filter is added to data when data is transmitted.

When receiving such a report at a leader node or at the base station, it applies the membership test of Bloom Filter to distinguish a suppression and a transmission failure. Fig. 2 shows the algorithm of transmission failure detection.

To detect transmission failures, we make the test timestamp set $T_{test}$ from the timestamp of the last received data $t_{last}$ (Line(1) - Line(4)). Since the sampling interval is specified in the user query, we can know the timestamps of possible transmissions. If the sampling interval is $x$ seconds, we know that the data will be transmitted every $x$ seconds. Therefore, to make the $T_{test}$, we start $t_{last}$ and add $x$ to the previously generated test timestamp until it reaches the currently received timestamp $t_{current}$. After that, we check a timestamp $t$ in the $T_{test}$

whether this is in the reported Bloom Filter or not (Line(5) - Line(13)). To check whether $t$ is in the Bloom Filter, we apply $k$ hash functions to $t$. If all $k$ bits of $h_i(t)$ are set in the reported Bloom Filter and the leader node or the base station has data transmitted at $t$, the data is successfully transmitted (Line(11)). If all $k$ bits of $h_i(t)$ are set but the data of the corresponding time is not in the leader node or the base station, we know that the transmission is failed at that time (Line(9)). If any $h_i(t)$ is not set in the reported Bloom Filter, the data is suppressed (Line(13)).

Bloom Filter may yield false positives. To minimize the false positive rate, $k = \ln 2 \times (m/\#\ of\ histories)$ hash functions are used [1]. Note that if $m = 10 \times \#\ of\ histories$, the false positive rate is less than 1%[3]. Therefore, we use $m = 10 \times \#\ of\ histories$ bits for Bloom Filter and find the optimal number of hash functions based on that. In our approach, a false positive means that a suppression is identified as a transmission failure. Specifically, a certain timestamp is considered as transmitted in the reported Bloom Filter but actually it is not. Since the data of this timestamp is not in the leader node or the base station, it is considered as a transmission failure. Although this causes unnecessary retransmission of suppressed data, it does not decrease the accuracy of query results but may slightly increase the accuracy.

### 3.3 Retransmission Request

Using the membership test of Bloom Filter, we can find timestamps of failed transmissions. After detecting transmission failure, we make a retransmission request to the sender. A retransmission request consists of nodeID and two successfully transmitted timestamps, $t_{prevS}$ and $t_{nextS}$, having failed transmissions between them. Fig. 3 shows the algorithm of retransmission request.

```
Algorithm RetransmissionRequest
Input t_current, T_failed, T_succeeded
Output Retransmission request R

begin
    for i = 0; i < length(T_succeeded) - 1; i++
        t_prevS := T_succeeded[i]
        t_nextS := T_succeeded[i+1]
        if t_nextS − t_prevS > x
            for j = 0; j < length(T_failed) - 1; j++
                t_prevF := T_failed[j] // t_prevF, t_nextF: local variables
                t_nextF := T_failed[j+1]
                if t_prevS < t_prevF && t_prevF < t_nextS
                    if t_nextF < t_nextS
                        j++
                    else
                        t_duration := t_nextS - t_prevS
                        n := t_duration / x
                        n_binary := convert n into binary string
                        R := Request(nodeID, t_prevS, n_binary)
    return R
end
```

**Fig. 3.** Algorithm of Retransmission Request

We search $T_{succeeded}$ and $T_{failed}$ to find two timestamps having failed transmissions between them (Line(1) - Line(11)). If we find these two timestamps, we make the retransmission request (Line(12) - Line(15)). To reduce the energy consumption, we transmit the first timestamp of success transmission $t_{prevS}$ and the difference $t_{duration} = t_{nextS} - t_{prevS}$ after encoding it. If the sampling interval is $x$, the range of $t_{duration}$ is $\{0, 1x, 2x, ..., nx\}$ where n is a positive integer. So we represent the $t_{duration}$ using $n$ (Line(12) - Line(14)). We use a small number of bits for representing $n$ more compactly. After making the retransmission request, we transmit this to the corresponding sender.

### 3.4 Failed Value Retransmission

Whenever the sensor node receives the retransmission request, it retransmits the values of failed transmissions. But a naive retransmission of failed values consumes much energy. Therefore, we use the quantization to reduce energy consumption for the failed value retransmission.

Before applying the quantization, we have to decide how many bits will be used in the quantization. We can decide the number of bits for the quantization before starting the query processing. The number of bits is determined by the user-specified error threshold $\delta$ and the range of the input. The difference of the min value $v_{min}$ and the max value $v_{max}$ of sensor readings is the range of the input. By dividing the range of the input by the user-specified error threshold $v_{max}$ - $v_{min}$ / $\delta$, we can get the number of intervals to represent the sensor values. The number of intervals has to satisfy $\#\, of\ intervals \leq 2^{\#\ of\ bits}$. Therefore, we can choose the minimum number of bits satisfying this condition.

```
Algorithm ValueRetransmission
Input Request R(nodeID, t_prevS, n_binary), D_sent
Output Retransmission message retransmissionMSG

begin
    retransmissionMSG := NULL
    Calculate t_nextS from t_prevS and n_binary
    Find transmitted data between t_prevS and t_nextS in D_sent
    num_of_failures := the number of transmitted data between t_prevS and t_nextS in D_sent
    for i = 0; i < num_of_failures; i++
       diff := v_prevS - v_failed // v_failed is the value of the i-th failed transmission
       interval := ⌊|diff|/δ⌋ + 1
       binary := convert interval into binary string
       signIndex := i * num_of_bits // num_of_bits is the number of bits necessary for binary
       if diff < 0
          quantizedValue[signIndex] := 1
          // quantizedValue is an array to store binaries for the transmitted values
       else
          quantizedValue[signIndex] := 0
       for k = 0, nbit = 1; k < length(binary); k++, nbit++ // Quantized Value Setting
          if binary[k] == 1
             quantizedValue[signIndex + nbit] := 1
          else if binary[k] == 0
             quantizedValue[signIndex + nbit] := 0
       retransmissionMSG := (nodeID, t_prevS, quantizedValue)
    return retransmissionMSG
end
```

**Fig. 4.** Algorithm of Failed Value Retransmission

```
Algorithm Recovery
Input Received retransmissionMSG
Output Recovered values for failed transmissions

begin
1.   if retransmissionMSG != NULL
2.     for i = 0; i < num_of_failures; i++
3.       signIndex := i * num_of_bits
4.       if quantizedValue[signIndex] == 1
5.         sign := -1
6.       else
7.         sign := 1
8.       for j = signIndex + 1, nbit = 1; j < signIndex + num_of_bits; j++, nbit++
9.         if quantizedValue[j] == 1
10.            binary[nbit] := 1
11.      interval := convert quantizedValue into decimal number
12.      rangeL := ((interval - 1) * δ) + v_prevS
13.      rangeH := (interval * δ) + v_prevS
14.      recoveredValue := (rangeL + rangeH) / 2
15.      Insert recoveredValue into corresponding failed data value in D_received
end
```

**Fig. 5.** Algorithm of Failed Value Recovery

Fig. 4 shows the algorithm of the failed value retransmission. Based on the received retransmission request, the node identify two timestamps of successful transmissions, $t_{prevS}$ and $t_{nextS}$ (Line(1)). From this, the sensor node can find the number of failed transmissions (Line(3)). We calculate the difference between failed value $v_{failed}$ and the value $v_{prevS}$ for each failed transmission (Line(6)). The difference belongs to a certain interval $(L, H]$ where $L = \lfloor |diff|/\delta \rfloor \times \delta$ and $H = (\lfloor |diff|/\delta \rfloor + 1) \times \delta$. We transform $(diff/\delta) + 1$ into the bit representation using the quantization (Line(7) - Line(18)). For example, let the number of bits for quantization be 3. If the user-specified error threshold $\delta$ is 5 and the difference $diff$ is 12, this belongs to the interval of $(2 \times 5, 3 \times 5]$. Then the quantized value for 3 is 011 and it is transmitted to the leader node or the base station.

### 3.5 Failed Value Recovery

Finally, the leader node or the base station can recover the failed values using received quantized data. Fig. 5 shows the algorithm of failed value recovery.

We can recover the failed values based on $v_{prevS}$ because each quantized value represents the interval to which failed value belongs. Let $q_1, q_2, ..., q_n$ be quantized values for failed values. Because $q_i$ is the bit representation of $diff_i/\delta + 1$, we can get the range $(L, H]$ of the difference using $q_i$ (Line(1) - Line(13)). Then we assign the average value of the range to the corresponding failed value (Line(14)). In the above example used in the failed value retransmission, let $v_{prevS}$ be 33. If we receive 011 at the leader node or the base station, the range of difference value is $(10, 15]$. Thus the original failed value belongs to $(33 + 10, 33 + 15]$ and we assign 45.5 as the failed value.

When a failure occurs during the transmission of a retransmission request or a failed value retransmission, the requestor resends the request after a certain waiting time. This is possible because the requestor is expecting the retransmission. The experimental result shows that this type of failures has little effect on the transmission cost and the accuracy.

## 4 Experimental Evaluation

We perform the experimental analysis to validate our approach using our own simulator. The simulated network consists of one group that is a rectangular grid. We performed experiments for multiple groups, but the pattern of the result for multiple groups was similar to that for a single group. Sensor nodes are placed on grid points and we varied the size of the group as $3 \times 3$, $4 \times 4$, and $5 \times 5$. We assume that the leader node of the group is placed at one hop distance from the base station. The minimum spanning tree is built over all member nodes where each member node can reach to the leader node with the number of hops as small as possible. Sensor readings follow the Gaussian model and are produced at the user-specified sampling rate. Each sensor node generates sensor readings which follow a Gaussian distribution with the mean $\mu$ and the standard deviation $\sigma$. The ranges of $\mu$ and $\sigma$ are 0 to 100 and 0 to 20 respectively. These two parameters for each sensor node are randomly generated between their ranges. The user-specified sampling rate is 10 seconds and the error threshold is 5.

We compare the performance using the energy consumption and the accuracy of query results. The comparison schemes are as follows:

- **ACK**: The acknowledgement is used to detect transmission failures. If a sensor node does not receive an acknowledgement, the corresponding data is retransmitted.
- **BF**: All member nodes and leader nodes are using our proposed approach utilizing Bloom Filter to resolve transmission failures.
- **BF + BaySail**: Our proposed approach BF is used within groups and BaySail is applied to transmissions from the leader node to the base station.
- **Leader BaySail**: A leader node uses BaySail to resolve transmission failures. This assumes that the leader node has no resource constraints.
- **BS BaySail**: Each node transmits its data to the base station using BaySail. This is the original BaySail proposed in [10]. There is no concept of a group in the spatio-temporal suppression scheme.

We change the failure rate from 10% to 50% and vary the number of history information from 1 to 5. Each experiment is run for 5 times and the results are averaged.

### 4.1 Energy Consumption

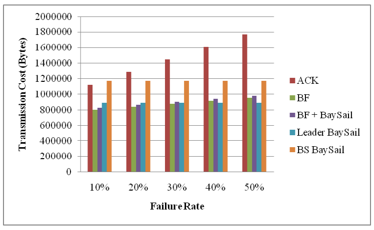
We measure the energy consumption using the amount of transmitted data because the larger the amount of transmission, the more energy is consumed. The basic data sizes used in our experiments are as follows:

| Component | Size (bits) |
|---|---|
| Acknowledgement | 40 (B-MAC protocol [8]) |
| NodeID | 32 |
| Sensor Reading | 32 |
| Timestamp | 32 |

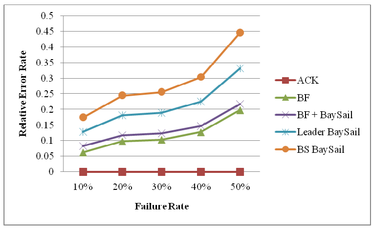
The energy consumption for $5\times5$ grid is shown in (a), (c), (e), (g), (i) of Fig. 6. H is the number of histories used in BF and BaySail. We do not show the results for $3\times3$ and $4\times4$ due to the space limitation. But they also have similar results to those of $5\times5$. BF consumes less energy than the other schemes in almost all cases. ACK's consumption steeply increases when the failure rate increases. Since it has to retransmit the original data until it is successfully transmitted. Energy consumption of BF also increases when the failure rate increases because the number of retransmission requests and value retransmissions increase in accordance with the increased failure rate. Although the number of retransmission requests and value retransmissions increase when the failure rate is high, BF does not require too much energy due to Bloom Filter and the quantization technique. BaySail has the constant energy consumption when failure rates are varied because it transmits a fixed size of data having history information only once and failed values are inferred at the base station. The number of histories in BF and BaySail increases the size of transmitted data. But the transmitted data size of BF is much less than that of BaySail because BF compresses the history information using Bloom Filter. Specifically, we set the size of Bloom Filter to 10 times larger than the number of histories to reduce the false positive rate less than 1%. But this is very small compared to the history size used in BaySail. In the case that the number of histories is 3, the history size is 96 bits in BaySail while 30 bits in BF. Therefore BF does not increase the size of transmitted data severely when the number of histories increases. Consequently, BF is more energy efficient than other schemes. We compare the energy consumption between BF and the original BaySail (BS BaySail) by calculating the reduction of the transmission cost of BS BaySail by using BF. Let the average of the transmission costs for all numbers of histories and all failure rates for BS BaySail be $T$(BS BaySail) and that for BF be $T$(BF). Then ($T$(BS BaySail)$-T$(BF))/$T$(BS BaySail) is about 0.39.

### 4.2 Accuracy

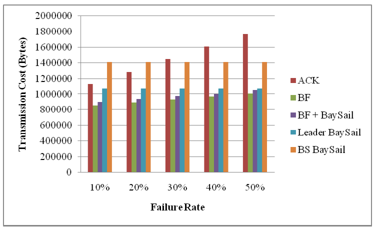
We evaluate the accuracy of query results using the relative error rate. Relative error rate is calculated by $|(recovered\ value - original\ value)/original\ value|$, where *recovered value* is the estimated value after applying our approach to handle transmission failures. The query result is an aggregated value of the group and we use AVG as the aggregation function. We assume that ACK can successfully retransmit failed values in the end. (b), (d), (f), (h), (j) of Fig. 6 show the result of the accuracy for each scheme. BF shows better accuracy than



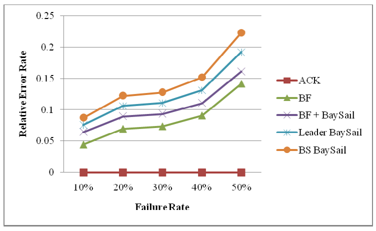

(a) Energy Consumption (H = 1)



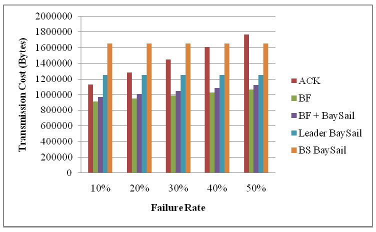

(b) Accuracy (H = 1)



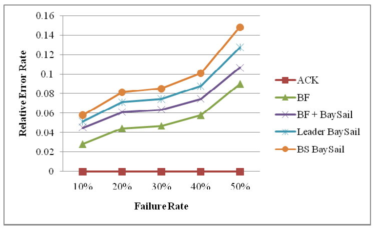

(c) Energy Consumption (H = 2)



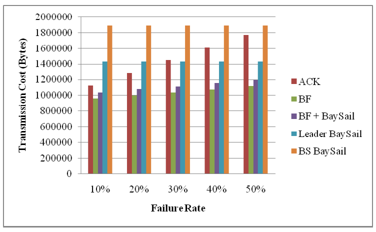

(d) Accuracy (H = 2)



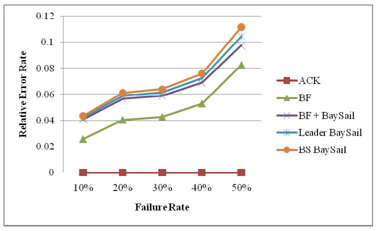

(e) Energy Consumption (H = 3)



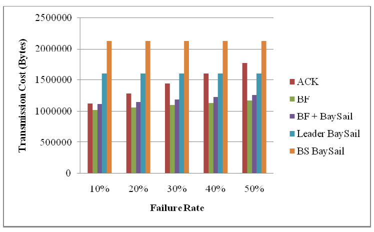

(f) Accuracy (H = 3)



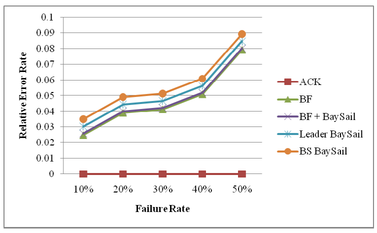

(g) Energy Consumption (H = 4)




(h) Accuracy (H = 4)




(i) Energy Consumption (H = 5)




(j) Accuracy (H = 5)

**Fig. 6.** Results for $5 \times 5$ grid

all other schemes. As for the energy consumption, we compare the accuracy between BF and the BS BaySail by calculating the reduction of the relative error rate of BS BaySail by using BF. Similarly as above, the reduction by using BF is 55%.

In BF, each failed value is represented as the difference from the previously successfully transmitted data value, and this difference is retransmitted using the quantization. The quantization is used to reduce energy consumption while guaranteeing that the recovered values are not different from the original values more than the user-specified error threshold. Therefore we can recover the failed value more accurately while using less energy than the other schemes. When the transmission failure rate increases, the relative error rate of each scheme also increases. The number of histories also affects the relative error rate. If the transmission failure rate is high, the probability of consecutive transmission failures is also high. Thus the history information could be lost. For example, let the number of histories is 1. If data is transmitted at $t_n, t_{n+1}, t_{n+2}$ but data is successfully transmitted only at $t_{n+2}$, then the history information about $t_n$ is lost and it is considered as a suppression. Therefore the relative error rate is higher than those for larger numbers of histories.

We set the minimum number of bits for the value quantization to satisfy $\#\ of\ intervals \leq 2^{\#\ of\ bits}$. The number of intervals is calculated by $(v_{max} - v_{min})/\delta$ where $v_{max}$, $v_{min}$ and $\delta$ are the max value, min value for sensor readings, and the user-specified error threshold respectively. If we use a value smaller than $\delta$ to calculate the number of intervals, the interval becomes narrowed and the number of bits for the quantization increases. Using this quantization bits and interval, we can get a tighter range for a failed value. This can increase the accuracy while consuming more energy.

## 5 Conclusion

Sensor networks usually have very limited energy resources. To reduce energy consumption, suppression schemes are proposed. Among these suppression schemes, spatio-temporal suppression can dramatically reduce the energy consumption. But the critical weakness of suppression is the transmission failure because this is considered as a suppression. This causes the accuracy problem in the query result.

We propose an effective and efficient method for handling transmission failures in the spatio-temporal suppression scheme. In the spatio-temporal suppression, transmission failures can occur from the member node to the leader node of a group and from the leader node to the base station. In resolving transmission failures, we have to consider the resource constraints of each sensor node. Therefore, we devise an energy efficient method to distinguish a suppression and a failure using Bloom Filter. History information of previous transmissions is inserted into Bloom Filter and we can effectively identify failures using the membership test of Bloom Filter. After detecting transmission failures, the receiver notifies the transmission failures to the sender, which retransmits these failed values using quantization. This quantization can reduce the size of transmitted data and recover the failed values more accurately. The experimental results show that our approach resolves the transmission failures energy efficiently and accurately.

**Acknowledgments.** This work was partially supported by Defense Acquisition Program Administration and Agency for Defense Development under the contract.

## References

1. Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Communications of the ACM 13, 422–426 (1970)
2. Chu, D., Deshpande, A., Hellerstein, J.M., Hong, W.: Approximate data collection in sensor networks using probabilistic models. In: ICDE 2006, p. 48. IEEE Computer Society, Los Alamitos (2006)
3. Fan, L., Cao, P., Almeida, J., Broder, A.Z.: Summary cache: A scalable wide-area web cache sharing protocol. In: IEEE/ACM Transactions on Networking, pp. 254–265 (1998)
4. Jain, A., Chang, E.Y., Wang, Y.-F.: Adaptive stream resource management using kalman filters. In: SIGMOD 2004, pp. 11–22. ACM, New York (2004)
5. Kotidis, Y.: Snapshot queries: Towards data-centric sensor networks. In: ICDE 2005, pp. 131–142. IEEE Computer Society, Los Alamitos (2005)
6. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D.: Wireless sensor networks for habitat monitoring (2002)
7. Olston, C., Jiang, J., Widom, J.: Adaptive filters for continuous queries over distributed data streams. In: SIGMOD 2003, pp. 563–574. ACM, New York (2003)
8. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: SenSys 2004, pp. 95–107. ACM, New York (2004)
9. Silberstein, A., Braynard, R., Yang, J.: Constraint chaining: on energy-efficient continuous monitoring in sensor networks. In: SIGMOD 2006, pp. 157–168. ACM, New York (2006)
10. Silberstein, A., Puggioni, G., Gelfand, A., Munagala, K., Yang, J.: Suppression and failures in sensor networks: a bayesian approach. In: VLDB 2007, pp. 842–853. VLDB Endowment (2007)