Abstract — The goal of Wireless Sensor Networks is to extract useful global information from individual sensor readings, which are typically collected and aggregated over a spanning tree. However, the spanning tree structure is not robust against communication errors; a low-quality (i.e., high-error-rate) wireless link close to the tree root may result in a high rate of global information loss. Therefore, many schemes have been proposed to achieve fault-tolerant aggregation. Intuitively, using timely link-quality information, which is gathered by continuous monitoring and error-rate measurement of network links, improves the performance of fault-tolerant aggregation schemes.

In this paper, we show that this intuition is not always true. In particular, we show that using link-quality information in an intuitive but wrong way results in degraded performance in some schemes, and therefore, care should be taken in using link-quality information. We also show that some schemes make better usage of link-quality information than others, and some schemes are more robust to errors in link-quality estimation than others. We support our findings by an extensive simulation study, and we focus on the (more general) class of fault-tolerant duplicate-sensitive aggregation schemes.

I. INTRODUCTION

In large-scale Wireless Sensor Networks (WSNs), sensor measurements are often aggregated within the network to filter redundancy and reduce overhead [5], [13]. However, communication errors are frequent in links of WSNs [15], and when a spanning-tree is used for aggregation (e.g., [10]), a single packet loss over a low-quality (i.e., high-error-rate) network link causes the loss of the measurements of a whole sub-tree. Schemes proposed to achieve reliable data delivery in WSNs make use of redundancy in network paths with special care to prevent each sensor reading from contributing to the final result more than once (e.g., [3], [4], [10]–[12]).

Intuitively, fault-tolerant aggregation schemes would benefit from the availability of timely link-quality information. For instance, paths with the best link qualities can be selected to minimize errors in delivered final result. However, three issues may limit the benefit of link-quality information. First, the way that link-quality information get incorporated in a fault-tolerant aggregation scheme has an impact on how far the scheme can benefit from the information. For example, in the RideSharing [4] scheme, the error in the delivered result depends not only on the quality of the links between children and parents (in the data-aggregation tree) but also on the quality of the links between parents themselves and on the transmission schedule of the network nodes, in which case finding the optimal selection of links is not trivial. Second, in some fault-tolerant aggregation schemes (e.g., the hash-based schemes [3], [11], [12]) there are other sources of error in the final result besides network errors, enforcing an upper bound on the reduction of error rate attainable by using link-quality information. Third, the techniques used to measure link quality, such as Received Signal Strength Indicator (RSSI) values [1], [8], [14] and probe packets [6], [9], are error-prone. For example, RSSI values are of variable and less than perfect quality, and probe packets record potentially stale data in highly dynamic environments.

In this paper, we investigate the problem of how to incorporate link-quality information in fault-tolerant data aggregation schemes in WSNs. In particular, we address the following two questions: (1) is the accuracy improvement gained by using link-quality information the same across all fault-tolerant aggregation schemes? and (2) are some fault-tolerant aggregation schemes more robust to link-quality measurement errors than the others? These questions are important to the network designer to identify the usability and the applicability scenarios (if any) of the different aggregation schemes.

To answer the above questions, we conducted a simulation study to compare the error rate in three state-of-the-art fault-tolerant aggregation schemes, namely partial parents [10], Hash-based [3], [11], [12], and RideSharing [4]. Our findings are as follows. Our first result is counter-intuitive; in Section II we show that although the partial parents aggregation scheme has better accuracy than the non-fault-tolerant tree aggregation when links have the same quality [10], an intuitive generalization of the partial parents scheme has worse accuracy than tree aggregation. Second, in Section III we show that the benefit of using link-quality information in the Hash-Based [3], [11], [12] aggregation schemes is limited by the errors induced by the hashing mechanism employed in these schemes. Third, in Section IV we show that the problem of finding the optimal way of using link-quality information in the RideSharing scheme [4] is NP-hard. In Section V we compare the performance of the three schemes together and identify the deployment scenarios in which each should be used. Finally, in Section V we show that the Hash-based and RideSharing

\[1\text{We assume that (possibly inaccurate) link quality information is readily available, and we do not confine ourselves to particular measurement techniques.}\]
schemes are more robust to link-quality estimation errors than the partial parents scheme.

II. WEIGHTED PARTIAL PARENTS (WPP)

Partial parents is an optimization of the TAG scheme [10] for improving tolerance to packet loss. In this section, we propose and evaluate an intuitive generalization, namely weighted partial parents, to extend partial parents to make use of link-quality information.

In partial parents, each node has two parents and sends half of its value \( c \) to each parent, assuming both links to the two parents have the same quality. In the Weighted Partial Parents (WPP) scheme, we drop the assumption of equiprobable link errors, that is, each link \( i \) connecting a node, \( X \), and its \( i \)th parent, \( P_i \), has its distinct error probability, \( p_i \). The link quality between node \( X \) and parent \( P_i \) is given by \( q_i = 1 - p_i \).

In this case of different link qualities, instead of sending equal shares of node \( X \)'s value to each of its parents as in partial parents [10], it is more intuitive to make each share proportional to the link quality over which it is going to be transmitted. As a result, a good-quality link contributes a larger share into the aggregate value. We transmit \( W_i = \frac{q_i}{\sum_{i=1}^{k} q_i} \) to each parent \( P_i \) where \( W_i \) is given below and \( \sum W_i = 1 \).

\[
W_i = \frac{q_i}{\sum_{i=1}^{k} q_i}
\]

A. Evaluation and Simulation Analysis

\[
\text{Fig. 1. Relative RMS error (parents/node = 3, total nodes = 1000)}
\]

In our first simulation experiment, we compare the WPP scheme to the spanning tree approach while taking into consideration link quality information. The metric of comparison is the relative RMS Error of the aggregate result, that is, the root mean square error normalized to the correct result (other published works call this metric simply RMS error; we call it relative RMS error for accuracy). In the simulation, sensor readings are aggregated to the data sink every epoch (a specified time interval), and we present results for COUNT query; other duplicate-sensitive queries (e.g., SUM and AVG) yielded similar results.

In our simulations, 1,000 sensors were placed randomly with a uniform distribution in a 300 × 300 ft area. The radio range of each node is 30 ft. The data sink is the node nearest to the center. Each simulation run has 1000 epochs, 300 msec each. The radio bandwidth is 38.4 Kbps [2]. The message size is assumed to be 2 bytes. Link errors are assumed to be uniformly distributed in interval \([0, \text{max link error}]\). In our experiments we varied the \( \text{max link error} \) within \([0,0.5]\) and set the maximum number of parents per node to three (“maximum” because the topology of the WSN may not permit those many parents for every node). The error bars in all the figures represent 90% confidence intervals.

Figure 1 shows the relative RMS error versus the average link error rate \( (= \frac{1}{\text{max link error}} \) ). Four schemes were compared: Tree-Random, where a spanning tree is used and a node picks its parent as the first parent it hears from (transmission is according to a randomized, Aloha-like MAC protocol); Tree-BestParent, where the best spanning tree is used (the node picks as its parent the one with the best link quality between the node and the parent); Weights-Random, where a node picks the first \( k \) parents it hears from and assigns weights as given by Equation 1; and Weights-BestK which similarly assigns weights but picks the best \( k \) parents it can hear from. Tree-random does not utilize redundancy and does not take into consideration the communication quality among nodes; consequently, its performance severely degrades with increasing error rate. Its relative RMS error reaches more than 50% for an average link error rate of 25%. Decomposing the value to be aggregated among multiple parents (in this experiment, \( k = 3 \)) slightly improves the accuracy of the delivered result. On the other hand, as shown in the figure, carefully choosing parents with the best links, significantly enhanced the quality of the received value.

Figure 1 highlights a very interesting result. Surprisingly, Tree-BestParent, although with no fault-tolerance, delivers more accurate aggregate results than WPP schemes. The reason for this is that in WPP, although each link error results in losing a smaller value compared to the spanning tree, more links are used and thus more errors occur. This fact is proved below.

Assume, without loss of generality, that \( q_1 > q_2 > ... > q_k \). In WPP, the transmission process can be viewed as \( k \) Bernoulli trials, where trial \( i \) is carrying the value \( W_i \cdot c \) and has \( q_i \) as the probability of success, thus, the mean and variance can be given as:

\[
\text{Mean}_{wpp} = \sum_{i=1}^{k} q_i \cdot W_i \cdot c = \frac{c}{\sum_{i=1}^{k} q_i} \cdot \left( \sum_{i} q_i^2 \right)
\]

\[
< q_1 \cdot c \cdot \frac{q_1 + q_2 + q_3 + ... + q_k}{q_1 + q_2 + q_3 + ... + q_k}
\]

\[
< q_1 \cdot c \cdot \text{Mean}_{\text{TreeBestParent}}
\]

\[
\text{Var}_{wpp} = \sum_{i=1}^{k} q_i^2 \cdot (1 - q_i) \cdot \frac{c^2}{\left( \sum_{i} q_i^2 \right)^2}
\]

Although counter-intuitive, Equation 2 shows (and Figure 1
confirms) that having no redundancy but using the best possible link (Tree-BestParent) yields a better expected value (closer to $c$) than that delivered by the redundant schemes like WPP.$^3$

III. HASH-BASED SCHEMES

Hash-Based Schemes (HBSs) [3], [11], [12] were devised to support duplicate-sensitive aggregation in multipath WSNs. In this section, we study their performance improvement when link quality information is used.

Examples of HBSs are Sketches [3], Synopsis Diffusion [12], and Tributaries and Deltas [11]. In these schemes, sensors broadcast a single message to multiple neighbors simultaneously, and the value of a specific node is included in the final aggregate result as long as there is at least one error-free path from this node to the data sink. A relatively large bit vector (e.g., $O(\log n)$ for COUNT, where $n$ is the total number of sensors) is attached to each data message.

A. Adding Link Qualities to Hash-Based Schemes

Intuitively, in HBSs sending the node’s hashed value to more neighbors/parents improves the scheme’s accuracy, since the value traverses more paths. But, because it also increases the overhead, a node in HBSs is typically restricted to use only a subset of $k$ parents out of the available ones.

When links have different quality ratings, minimizing the aggregate error is simply achieved by selecting the optimal set of parents locally. Each node chooses $k$ (if enough parents are available) parents out of its available parents, such that the links from the node to the selected parents have the highest link qualities. The assigned parents are then notified by the child node that they were picked for its hash vector aggregation. During the data collection phase, the child node broadcasts its hash vector but only the node’s assigned parents will be awake to listen to the channel and to aggregate the node’s hash vector into theirs, while other parents can switch off their transceivers to save power.

B. Evaluation and Simulation Analysis

We investigated the performance of a representative HBS, namely Synopsis Diffusion (SD) [12], with and without link quality information (we obtained similar results with the Sketches [3] scheme). We present results for two SD variations, SD-20 and SD-40; the first uses 20 hash tables per message while the second uses 40 hash tables (increased accuracy with higher overhead). As in Section II-A, we used the relative RMS error to measure the accuracy of the estimated value. In this section we study only the accuracy improvement from using link quality information in HBSs; we compare this improvement with the improvement achieved in other fault-tolerant schemes in Section V.

The simulation setup is the same as in Section II-A except for one difference: the message size is 12 Bytes for SD-20 (20 bit vectors of 2 bytes each with a 30% compression ratio) [3], and 24 Bytes for SD-40. In the following experiments we vary the maximum link error rate. We compare $SD-20RandomK$ and $SD-40RandomK$, in which parents are selected at random, with $SD-20BestK$ and $SD-40BestK$, which both use link quality information.

Figure 2 shows the relative RMS error as a function of the average link error rate. The curves have an almost flat shape, because SD is robust to link losses and the value of a node will only be lost when all the paths connecting this node to the sink have errors. In other words, this saturation in the performance is because that the error is no longer from the network (enough paths exist so that an error-free path always exists from each node to the sink), rather, it is a hashing error. The relative RMS error increases slightly when the average link error is large enough (around 15%) because some nodes have no error-free paths to the sink, and their values are lost. Furthermore, there is always an error in SD aggregation even when the network is error-free. This error is associated with the hash operation of SD (the hash table is a lossy compression method).

IV. RIDE SHARING

RideSharing [4] (RS) exploits the inherent redundancy of the wireless medium and depends on the notion of primary and backup parents. When a message between a sensor and its primary parent is lost, possibly one or more other sensors (the backup parents) might have correctly overheared the message. When some of the overhearing sensors have not yet transmitted their own values, they can aggregate the missing value into theirs, and send it with their own data.

RS organizes sensors in a track graph, in which a directed edge from node $X$ to node $Y$ indicates that $Y$ listens to $X$’s communication. As an example, Figure 3 shows a sensor $C$ in track $T_2$ with two parents, primary $P_1$ and backup $P_2$. Assuming no errors, both $P_1$ and $P_2$ receive $C$’s value, but only $P_1$ aggregates it. If a link error occurs and $P_2$ receives $P_1$’s bit vector (see [4] for more details on bit vectors in RS), it detects that $C$’s value is missing and corrects the error by aggregating $C$’s value.

Tracks take turns to transmit in a time-synchronized fashion that starts from the furthest track off the center and proceeds inwards one track at a time. Each round, sensors in the track scheduled for transmission send their values to their parents in the adjacent track closer to the center. The sensors in

![Fig. 2. Relative RMS error (parents/node = 3, total nodes = 1000)](image-url)
A. Adding Link Qualities to RideSharing

The problem of minimizing the error in the aggregate result when the link qualities among the neighboring nodes are known is formulated as follows. First note that the relative RMS errors in RS is affected by (besides the quality of the links between sensors and their parents) the quality of the links between parents, because RS parents listen to each other; and the transmission schedule among the parents, because RS tries to avoid unnecessary transmissions. Assume that parents of each node are given. The objective is then to find a transmission schedule among network nodes to minimize the relative RMS error.

![Fig. 3. Track Topology [4]](image)

![Fig. 4. Minimizing Aggregate Error as a Scheduling Problem](image)

**Fig. 4.** Minimizing Aggregate Error as a Scheduling Problem

An example is depicted in Figure 4, where Nodes 1 and 2 have the same 3 parents (P1, P2, and P3). Six different transmission schedules (orders) are possible by permutations among the parent nodes ([p1, p2, p3], [p2, p1, p3], ..., [p3, p1, p2]). Each schedule assigns a primary parent (node scheduled to transmit first) and backup parents (those scheduled later) for Nodes 1 and 2. The objective is to choose the schedule that minimizes the error in the aggregate result. What follows is a low-overhead, distributed heuristic.

A node C in track i selects as its parents the nodes with the best links (from C to the parents) from track i−1. A transmission schedule is then chosen arbitrarily, and the primary and backup parents are determined accordingly: the primary parent is the first of the selected parents in the schedule followed by the first backup and so on. The main motivation behind this simple heuristic is that the qualities of the links between tracks seem to have more impact on the error rate than those within the same track. If communication links connecting a node C to its parents are error-free then the value of C will be aggregated in the final result irrespective of the quality of the links connecting the parents.

B. NP-Hard Problem Reduction

In this subsection, we prove that finding an optimal parent schedule is NP-hard by reducing the schedule selection problem to the minimum feedback arc set problem [7]. The minimum feedback arc set problem is specified as follows: Given a directed graph $G(V,E)$ it is required to find the minimum set of edges (arcs) $E' \subseteq E$, which, if removed, leave the resultant graph without any cycles (i.e., the result is a DAG). The minimum feedback arc set problem is known to be an NP-hard problem [7].

![Fig. 5. NP-Hard Reduction](image)

**Fig. 5.** NP-Hard Reduction

The reduction algorithm is as follows. Assume that we are given a directed graph $G(V,E)$, for example the graph in Figure 5(a). We construct the following network: the network has a node for each vertex in $G$, and for each arc $i \to j$, node $k$ is inserted as a child for both nodes $i$ and $j$. Link qualities are assigned such that $q_{ki} > q_{kj}$, and $0 < q_{ij} < 1$ where $q_{ij}$ is the quality of the link connecting nodes $i$ and $j$ ($q_{ij} = q_{ji}$). (In Figure 5(b), as an example we applied the transformation for arcs $A \to B$ and $C \to B$. $q_{AX} > q_{XB}$, $0 \leq q_{AB} < 1$, and similarly, $q_{ZC} > q_{CB}$, $0 \leq q_{CB} < 1$.) Consider the constructed network (an example is network ABX in Figure 5(b)). A parent schedule in the constructed network can be modeled as a set of ordered pairs $(i \to j)$, one for each pair of parents $i$ and $j$ such that $i$ is before $j$ in the schedule.

Without loss of generality, consider a child node (e.g., node X). It will pick $A$ as its primary parent and $B$ as its backup parent if $A$ is scheduled to transmit first, and vice versa. If $A$ transmits first then the expected value of $X$ to be aggregated in the final result is (assuming cascaded RS):

$$\text{Mean}_{A,\text{first}} = V_X \cdot (q_{XA} + (1 - q_{XA})q_{XB}q_{AB})$$

(4)

On the other hand if $B$ was scheduled to transmit first then the expected value is:

$$\text{Mean}_{B,\text{first}} = V_X \cdot (q_{XB} + (1 - q_{XB})q_{AX}q_{BA})$$

(5)

where $V_X$ is the value of Node $X$. Now since by construction $q_{XA} > q_{XB}$ and $0 \leq q_{AB} = q_{BA} < 1$, then $\text{Mean}_{A,\text{first}} > \text{Mean}_{B,\text{first}}$. Hence, the error when $A$ transmits first is smaller.
than that when it transmits later, and thus node $A$ should transmit before node $B$ ($A \rightarrow B$).

Noting that $A \rightarrow B$ is an arc in $G$, consequently, the minimum-error parent schedule has the maximum number (equivalently, violates the minimum number) of the ordered pairs $(i \rightarrow j)$, where $i \rightarrow j$ is an arc in $G$. Next, we construct the graph $G'$ from the optimal schedule as follows. The set of nodes in $G'$ is the set of parents in the network (which is the set of nodes in $G$ by construction of the network). For every ordered pair $i \rightarrow j$ in the minimum-error parent schedule, construct an edge from node $i$ to node $j$ in $G'$ only if $i \rightarrow j$ is an edge in $G$. Since each node is scheduled to transmit once then, $G'$ has no cycles. The graph $G'$ has the maximum number of edges from $G$. Hence, edges in $G$ but not in $G'$ is a solution to the minimum arc feedback set in $G$.

C. Evaluation and Simulation Analysis

To evaluate the performance of RideSharing when taking into consideration link quality information, we used the simulation setup from Sections II-A and III-B, with one difference: the message size depends on the number of children ($2$ bytes + $2$ bits $\times$ number of children). We compare the following schemes: RS with the heuristic described in Subsection IV-A, which uses link-quality information (Cascaded-Best$K$ and Diffused-Best$K$) and plain RS (Cascaded-Random$K$ and Diffused-Random$K$).

Figure 6 shows the relative RMS error as a function of the average link error rate. Note that there is no error in the aggregate result when the network is error free (unlike the 6-12% error in the HBSs in Section III-B). Moreover, although the heuristic proposed for considering the link quality among the nodes is very simple, it is very efficient and significantly improves the performance of the RS schemes (compare Cascaded-Random$3$ and Cascaded-Best$3$). Note that Cascaded $K$ RS achieves better relative RMS error than Diffused $K$ for both Best$3$ and Random$3$ schemes. This is because some link errors are masked by Cascaded $K$, while hurting Diffused $K$ (for instance, an error between the child and the last backup parent to send).

Fig. 6. Relative RMS error (parents/node = 3, total nodes = 1000)

V. PERFORMANCE COMPARISON

In this section, we present simulation results to compare the performance of all the discussed aggregation schemes. The best spanning tree approach, Tree-BestParent, is used as the baseline. We also study the effect of errors in estimating link qualities.

A. Using Link Qualities without Estimation Errors

We first assume no errors in the estimation of link qualities. Figure 7 shows the relative RMS error as a function of average link error when all schemes consider link qualities information. When the link error rate increases, the performance of Tree-BestParent degrades significantly: the relative RMS error reaches 20% for an average link error rate of 25%. WPP (not shown for visual clarity) performed even worse than Tree-BestParent (see Section II-A).

SD-20Best$K$ and SD-40Best$K$ are very robust to link errors. If the links are very lossy (large average link error), then SD provides the most accurate aggregate result. As mentioned before, however, there is an error (due to hashing) even for error-free networks. RS schemes (Cascaded-Best$K$ and Diffused-Best$K$) achieve the best performance at low to moderate (< 20%) average link error rates. In particular, Cascaded-Best$3$ outperforms SD-40Best$3$ for an average link error rates up to 20%.

B. Link Qualities with Estimation Errors

As mentioned in Section I, an accurate estimate of the link qualities among the sensor nodes is hard to achieve. We investigate the effect of link-quality estimation inaccuracies on the relative RMS error.

We assume that the error in the estimation of the link quality follows a normal distribution with a zero mean and we changed the standard deviation (relative to the actual link quality value) of the distribution. Consequently, the case of no link quality estimation error (same as Section V-A) is at 0% relative standard deviation, and the estimation error increases as the relative standard deviation of the estimation error increases. Figure 8 shows the relative RMS error of the different aggregation schemes as a function of the relative...
standard deviation of the estimation error. The average link error rate is fixed at 12.5% (average error rate values within [0, 25%] have a similar behavior).

As shown in the figure, the performance of WPP (Weights-Best3) degrades significantly as the estimation error increases. This is because the weight assignment method in WPP depends on the exact value of the link quality (see Equation 1). When the link qualities deviate from the expected value, the shares contributed by each link do not reflect the quality of that link, and a lossy link might be assigned a large share of the value to be aggregated, hence increasing the relative RMS error.

Similar to WPP, the spanning tree (Tree-BestParent) is not robust against link quality estimation errors. In Tree-BestParent each node picks one parent trusting that this link is the best and is supposed to deliver the whole value of the node. As the estimation error increases, the link chosen might not be the best one available and the data values are assigned to lossy links leading to increasing overall relative RMS error.

SD-20Best3, to the contrary, are more robust to estimation errors (we omitted SD-40Best3 for visual clarity but it has a similar behavior). As previously mentioned, the relative RMS error in the aggregate result is mainly due to hashing errors and the value of a node will be delivered as long as there exists at least one error-free path from this node to the sink. The link quality estimation errors have a negligible effect on SD performance.

RS schemes (Cascaded-Best3 and Diffused-Best3), similar to SD, are robust to estimation errors. This robustness is due to the fact that RS does not depend on the exact values of the link qualities but on the order of these qualities. Recall that, in our simple heuristic in Subsection IV-A, a node picks the best parents, and then the role of the parent (primary, first backup, second backup, and so on) is determined based on the transmission schedule. Consider a lossy link \( L \). If there is an error in estimating \( L \)’s quality, this estimation error will only have an effect when the error is large enough to cause \( L \) to replace one of the selected parents. Even in this case, there is still an opportunity (not present in Tree-BestParent) of masking the effect of the estimation error, because other nodes (primary and backup parents) will try delivering the value if it is lost over \( L \). Thus, the effect of the link quality estimation error in the overall aggregate result and on the performance of RS is small.

VI. CONCLUSION

Noting that it is inadequate to assume equiprobable link errors, the use of link quality information intuitively enhances the performance of WSN’s applications. In particular, we showed in this paper how different fault-tolerant, duplicate-sensitive, aggregation schemes for WSNs can take advantage of link quality information. We evaluated the performance of three schemes with respect to relative RMS error: partial-parents, hash-based and RideSharing when they use link quality information.

We found that an intuitive generalization of the partial parents scheme has worse accuracy than tree aggregation. We also found that the benefit of using link-quality information in the Hash-Based [3], [11], [12] aggregation schemes is limited and is less than that gained in the RS scheme. We showed that the problem of using link quality information optimally in RS [4] is NP-hard. Finally, we showed that the Hash-based and RideSharing schemes are more robust to link-quality estimation errors than the partial parents scheme.

ACKNOWLEDGMENTS

This work is partially supported by NSF through grants ANI-0125704, ANI-0325353.

REFERENCES