DRE^2: Maximizing Data Resilience in Wireless Sensor Networks in Extreme Environments

Abstract— **Data resilience refers to the ability of long-term viability and availability of data despite insufficiencies of (or disruption to) the physical infrastructure that stores the data. In this paper, we identify, formulate, and solve a new algorithmic problem that addresses data resilience in wireless sensor networks operating in extreme environments. Those environments include remote and inaccessible regions such as underwater or underground, or inhospitable regions under extreme weather. In these extreme environments, it is not feasible to install data-collecting base stations in the field for a long-term. Therefore generated sensory data must be stored inside the network first, and is then collected when uploading opportunities become available. Our goal is to achieve maximum data resilience by preserving the data inside the network for the maximum amount of time, considering that each sensor node has limited storage capacity and unreplenishable battery power. We refer to this problem as *data resiliency in extreme environments (DRE^2)*. We first prove that this problem is generally NP-hard. We design a suite of efficient heuristic algorithms based on different network parameters and metrics.**

Keywords – Data Resilience, Wireless Sensor Networks, Extreme Environment

I. Introduction

Background and Motivation. *Data resilience* refers to the ability of any network to recover quickly and to continue maintaining availability of data despite of disruptions such as equipment failure, power outage, or malicious attack. Due to resource constraints of wireless sensor networks such as unreplenishable battery power and limited storage capacity of sensor nodes [12], link unreliability and scarce bandwidth of wireless medium [18], and the inhospitable and harsh environments in which they are deployed [2], sensor nodes are often prone to failure and vulnerable of data loss. Therefore, how to ensure that data collected at sensor nodes reach the base station reliably despite aforesaid vulnerabilities has been an active research topic since the inception of sensor network research. This line of research is usually named under the umbrella of reliable data transmission [13], data resilience [1], or data persistence [8]. We use data resilience throughout the paper.

In recent years, domain scientists are trying to utilize sensor networks to address some of the most fundamental problems facing human beings, such as disaster warning, climate change, and renewable energy. The emerging sensor networks designed for those scientific applications include seismic sensor networks [17], underground sensor networks [15], underwater or ocean sensor networks [5], wind and solar harvesting [7, 11], and volcano eruption monitoring and glacial melting monitoring [4, 16]. One common characteristic of these sensor networks is that they are all deployed in challenging or extreme environments such as in remote or inhospitable regions, or under extreme weather, to continuously collect large volumes of data for a long period of time. Consequently, it is not practical to deploy data-collecting base stations with power outlets in or near such inaccessible sensor fields. Sensory data generated have to be stored inside the network for some unpredictable period of time and then being collected by periodic visits of robots or data mules [14], or by low rate satellite link [10]. Due to the lack of human intervention and the inadequacy of maintenance in the inhospitable environments, such sensor networks must operate much more resiliently than traditional sensor networks (with base stations and in friendly environments).

Data Resilience Against Storage Overflow Disruption In Extreme Environments. In this paper, we focus on data resilience against sensor storage overflow, wherein storage spaces of some sensor nodes are depleted and therefore it can not store any newly generated data. Storage overflow is a major obstacle existing in above emerging sensor networks, due to the following reasons. First, massive amounts of data in those scientific applications are generated, sensing a wide range of physical properties in real world ranging from solar light to wind flow to seismic activity. Second, storage is still a serious resource constraint of sensor nodes, despite the advances in energy-efficient flash storage [12] with good compression algorithms (data is compressed before stored) and good aging algorithms (fidelity of older data is reduced to make space for newer data). Third, in those extreme environments, it is not possible to replenish or replace the sensors and their storage. As a consequence, the massive sensory data could soon overflow data storage of sensor nodes and causes data loss. From scientific perspective, data are "first class citizens" because every bit of data could potentially be important for scientists to analyze the physical world. Thus how to resiliently maintain sensory data despite storage overflow and prevent data loss in extreme environments becomes a crucial task.

In our network model, there is one sensor node that generates large amounts of sensory data (due to its proximity to the event of interest) and has exhausted its limited storage capacity.[1] We refer to this sensor node with exhausted data storage while still generating new overflow data as *source*

[1]For example, a distributed acoustic monitoring and trace retrieval system, called EnviroMic [9], was designed and deployed to monitor the social behaviors of animals in the wild. In EnviroMic, an acoustic sensor with 1GB flash memory will run out of its storage in just seven hours, when it samples the entire audible spectrum.

node. Other sensor nodes that have available storage are referred to as *storage nodes*. In order to achieve data resilience and prevent data loss, the overflow data generated at the source node needs to be offloaded to the storage nodes, waiting there to be collected when above uploading opportunities become available. Note that sensor node whose generated data has not exceeded its storage capacity is considered as a storage node, since it can still store the overflow data from the source node. The storage nodes that finally store overflow data offloaded from source node are referred to as *destination nodes*.

Since it is not known beforehand when the next uploading opportunities arrive, it is desired that the offloaded data being stored in destination nodes for the maximum amount of time. This is because sensor nodes are powered by battery energy, which eventually will be depleted. When this takes place, its stored data is considered lost and can no longer be collected when uploading opportunities arise. If all the sensor nodes have the same battery draining rate, then nodes with high battery power will deplete their energy later than nodes with low battery power. Thus data stored at destination nodes with higher battery power can last for longer time, getting better chance of survival and of being collected. Therefore it is preferred that data are offloaded to destination nodes with high battery power.

II. Problem Formulation of (DRE^2)

Network Model. The sensor network is represented as an undirected graph $G(V, E)$, where $V = \{1, 2, ..., N\}$ is the set of N nodes, and E is the set of edges. The sensory data is modeled as a sequence of raw data items, each of which has the same size of k bits. Let S be the single source node (with depleted storage space), and $V_s = \{V - \{S\}\}$ be the set of storage nodes. The source node has a overflow data items to be offloaded, denoted as $D = \{D_1, D_2, ..., D_a\}$. Let m_i be the available free storage space at storage node $i \in V_s$, measured in number of data items (that is, storage node i can store m_i more data items). We assume that the total size of the data items to be preserved is less than or equal to the size of the total available storage space in the network, that is, $a \leq \sum_{i \in V_s} m_i$.

Energy Model. Let E_i denote sensor node i's initial energy, which is finite and unreplenishable. For wireless communication energy, we adopt the first order radio model [6] wherein for a k-bit data sent over distance l meters, the *transmission energy* (on the sender side) is $E_t(k, l) = \epsilon_{elec} * k + \epsilon_{amp} * k * l^2$, the *receiving energy* (on the receiver side) is $E_r(k) = \epsilon_{elec} * k$. Here $\epsilon_{elec} = 100nJ/bit$ is the energy consumption per bit on the transmitter circuit and receiver circuit, and $\epsilon_{amp} = 100pJ/bit/m^2$ calculates the energy consumption per bit on the transmit amplifier. Let $w_{u,v}$ denote the total energy consumption when node u sends a k-bit data to its one hop neighbor v over their distance $l_{u,v}$, $w_{u,v} = E_t(k, l_{u,v}) + E_r(k)$. Note that receiving energy is independent of distance between sender and receiver, and that $w_{v,u} = w_{u,v}$. Now for any arbitrary two nodes i and j in the network that are multiple hops away from each other, let $c_{i,j}$ be the minimum energy consumption of sending one data item from i to j along path $P_{i,j}$. Here $P_{i,j}$ is referred to as the *minimum energy consumption path* between i and j. Then $c_{i,j} = \sum_{(u,v) \in SP_{i,j}} w_{u,v}$, wherein both $c_{i,j}$ and $P_{i,j}$ can be easily obtained using Dijsktra's shortest path algorithm by assigning weight $w_{u,v}$ to edge $(u, v) \in E$.

Problem Formulation. We define *offloading function* as $r : D \rightarrow V_s$, indicating that data item $D_j \in D$ is distributed from its source node S to its destination node $r(j) \in V_s$. Let $P_j : S, ..., r(j)$, referred to as the *offloading path* of D_j, be the sequence of distinct sensor nodes along which D_j is distributed from S to $r(j)$ (note that P_j is the same as $P_{S,r(j)}$, we use P_j for simplification of notations). Let $\sigma(i)$ denote node i's successor node in P_j. Let x_{ij} be the energy cost incurred by node i when offloading data item D_j from S to $r(j)$, then

$$x_{ij} = \begin{cases} E_t(k, l_{i,\sigma(i)}) & i = S, \\ E_r(k) & i = r(j), \\ w_{i,\sigma(i)} & i \in P_j - \{S, r(j)\}, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In above, when i is the source node of D_j, it only costs transmission energy $E_t(k, l_{i,\sigma(i)})$; when i is the destination node of D_j, it only costs receiving energy $E_r(k)$. However, when node i is a relaying node of D_j, it costs both receiving and transmission energy, which is $w_{i,\sigma(i)} = E_t(k, l_{i,\sigma(i)}) + E_r(k)$ (note that node i's receiving energy equals to node $\sigma(i)$'s receiving energy). Otherwise, node i is not involved D_j's offloading process thus costs zero amount of energy.

Let $E_i^{'}$ denote node i's remaining energy after all the a data items are offloaded. Then,

$$E_i^{'} = E_i - \sum_{j=1}^{a} x_{ij}, \qquad \forall\ i \in V, \quad (2)$$

Definition 1: (**Data Resilience Level.**) Given a sensor network with a data items to be preserved, *data resilience level* is defined as the sum of remaining energy of the destination nodes of all the a data items $\sum_{j=1}^{a} E'_{r(j)}$. It also equals to $\sum_{i \in V_s} \left(E_i^{'} \times \xi(i)\right)$, where $\xi(i)$ is the number of data items node i finally stores. □

Data resilience level indicates the network's best achievable effort to preserve all a data items, since the more energy of the node that stores the data, the longer time of the data it can survive. The objective of DRE^2 is to find a offloading function r and a set of offloading paths $\mathcal{P} = \{P_1, P_2, ..., P_a\}$, to distribute each of the a data items to its destination node, such that the data resilience level of the network is maximized post distribution, i.e.

$$\max_{r, \mathcal{P}} \sum_{1 \leq j \leq a} E'_{r(j)}, \quad (3)$$

under the energy constraint that each node can not spend more energy than its initial energy level:

$$E'_i \geq 0, \qquad \forall\ i \in V, \quad (4)$$

and the storage capacity constraint that the number of data items offloaded to node i is less than or equal to node i's storage capacity:

$$|\{j \mid r(j) = i, 1 \leq j \leq a\}| \leq m_i, \qquad \forall\ i \in V. \quad (5)$$

Rationale of Data Resilience Level. Note that data resilience level is not the sum of the remaining energy of the destination nodes $\sum_{i \in V_s, \xi(i)>0} E_i^{'}$, which, compared to data resilience level defined above, is a less accurate indicator to measure the data resilience performance. For example, consider a scenario wherein one destination node A stores two data items and the other scenario wherein two destination nodes B and C each stores one data item, and $E_A^{'} = E_B^{'} = E_C^{'}$. It is evident that both scenarios achieve the same performance of data resilience, since the two data items in both scenarios will get lost at the same time (when A, B, and C depletes their energy at the same time). This can be best captured by Definition 1, $E_A^{'} \times 2 = E_B^{'} + E_C^{'}$. Using $\sum_{i \in V_s, \xi(i)>0} E_i^{'}$, $E_A^{'} \neq E_B^{'} + E_C^{'}$, which would indicate different data resilience levels.

Theorem 1: The DRE^{2}is NP-hard.

Proof: We show that a special case of this problem is NP-hard by reducing the maximum 3-Dimensional matching (3DM) problem to this special case. 3DM problem has been shown to be NP-hard [3]. The detailed proof is omitted due to space constraint. ∎

III. Heuristic Algorithms for General Graph Topologies

For general graph topologies, since DRE^2 is NP-hard, it is not possible to design time-efficient optimal algorithm. We therefore design a suite of time-efficient heuristic algorithms.

Network-Based Algorithm. In Algorithm 1, it first sorts all the storage nodes in non-ascending order of their initial energy levels. Then it offloads the data items to the sensor node with highest energy level, then the one with the second-highest energy level, so on and so forth, until all the a data items are offloaded.

Algorithm 1: Network-Based Data Resilience Algorithm.
Input: A sensor network,
m_i, E_i, a data items D at source node S
Output: $r : D \rightarrow V_s$;
1. Sort storage nodes in V_s in non-ascending order of their initial energy: $E_{v_1} \geq E_{v_2} \geq ... \geq E_{v_n}$;
2. Find the top $k+1$ highest energy nodes: $v_1, ..., v_k, v_{k+1}$ such that $\sum_{i=1}^{k} m_{v_i} < a \leq \sum_{i=1}^{k+1} m_{v_i}$;
3. **for** $(1 \leq i \leq k)$
4. Offload m_{v_i} data items to node v_i;
Update the energy levels of all the nodes involved;
5. **end for;**
6. Offload $a - \sum_{i=1}^{k} m_{v_i}$ data items to node v_{k+1};
Update the energy levels of all the nodes involved;
7. **RETURN** Data resilience level $\sum_{i=1}^{n} \left(E_i^{'} \times \xi(i)\right)$.

Time Complexity. Sorting takes $O(n\log n)$. Since it updates the energy level of at most n nodes to offload m_i data items to node v_i $(1 \leq i \leq k)$, offloading to all the $k+1$ highest energy nodes takes $(k+1) \cdot n = O(n^2)$. Therefore time complexity of Algorithm 1 is $O(n^2)$.

Node-Based Algorithm. The difference between Algorithm 2 below and Algorithm 1 is that in Algorithm 1, it finds the storage nodes with top energy levels only once, while in Algorithm 2, in needs to find the storage node with highest energy every time it needs one.

Algorithm 2: Node-Based Data Resilience Algorithm.
Input: A sensor network,
m_i, E_i, a data items D at source node S
Output: $r : D \rightarrow V_s$;
1. **while** (There are still data items to offload)
2. Find the storage node with available storage and
3. highest energy currently, say node i;
4. Offload m_i data items to node i;
5. Update the energy levels of all the nodes involved;
6. **end while;**
7. **RETURN** Data resilience level $\sum_{i=1}^{n} \left(E_i^{'} \times \xi(i)\right)$.

Time Complexity. When offloading to one storage node, it needs to find the highest-energy storage node (which takes $\log n$) and could update the energy level of at most n nodes, taking $O(n)$. Therefore offloading all the data items takes $O(n^2)$.

Data-Based Algorithm. Algorithm 3 below executes in a iterations: in each iteration, one of the a data items is offloaded to the storage node with available space and highest energy in that iteration. Note that Algorithm **??** and Algorithm **??** in linear topology are special cases of Algorithm 3.

Algorithm 3: Data-Based Data Resilience Algorithm.
Input: A sensor network,
m_i, E_i, a data items D at source node S
Output: $r : D \rightarrow V_s$;
1. **for** (each of the a data items)
2. Offload it to the storage node with highest energy
3. and available storage in this iteration;
4. Update the energy levels of all the nodes involved;
4. **end for;**
5. **RETURN** Data resilience level $\sum_{i=1}^{n} \left(E_i^{'} \times \xi(i)\right)$.

Time Complexity. Offloading one data item, which takes $\log n$ to find the node with highest energy and which could update the energy levels of at most n nodes, takes $(\log n + n) = O(n)$. Therefore offloading all the a data items takes $O(a \cdot n)$.

Benefit-Based Algorithm. We first introduce some notation. In Algorithm 4 below, selection of variable B_{ijk} indicates that the k^{th} storage slot of storage node i has been selected to store data item D_j $(1 \leq i \leq N, 1 \leq j \leq a, 1 \leq k \leq m_i)$. Then we formally define $benefit$ of offloading a datum as follows.

Definition 2: (**Benefit of Offloading a Datum.**) Let $\mathcal{B}$ denote the set of variables that have already been selected. Let $t(\mathcal{B})$ denote the data resilience level corresponding to $\mathcal{B}$. The benefit of variable B_{ijk} w.r.t. $\mathcal{B}$, denoted as $\beta(B_{ijk}, \mathcal{B})$, is

the increase of the data resilience level when B_{ijk} is selected given $\mathcal{B}$. That is $\beta(B_{ijk}, \mathcal{B}) = t(\mathcal{B} \cup \{B_{ijk}\}) - t(\mathcal{B})$. □

To maximize the data resilience level, Algorithm 4 endeavors to maximize the $benefit$ of offloading a data item. In particular, in each iteration, it offloads a data item to a storage node with available space such that the benefit of offloading the data item to this storage node is the largest one among offloading to all the storage nodes in that iteration. It terminates when all a data items at source nodes are offloaded.

Algorithm 4: Benefit-based Data Resilience Algorithm.
Input: A sensor network,
Output: $r : D \rightarrow \{n, n-1, ..., 2, 1\}$;
1. $\mathcal{B} = \phi$ (empty set), $j = 1$;
2. **for** $(1 \leq j \leq a)$
3. Let B_{ijk} be the variable with maximum $\beta(B_{ijk}, \mathcal{B})$;
4. $\mathcal{B} = \mathcal{B} \cup \{B_{ijk}\}$;
5. Update the energy levels of all the nodes involved;
6. **end for;**
7. **RETURN** Data resilience level $\sum_{i=1}^{n} \left(E_i^{'} \times \xi(i)\right)$.

Time Complexity. It takes $O(n^2)$ in each iteration, and the total running time of Algorithm 4 is $O(a \times n^2)$. We omit the detailed analysis due to space constraint.

IV. Simulations.

We compare the suite of heuristic algorithms: network-based (referred to as Network), node-based (referred to as Node), data-based (referred to as Data), and benefit-based (referred to as Benefit) in general graph topologies. In our experiments, 100 sensors are uniformly distributed in a region of 1000m × 1000m. Transmission range is 250m. One of the sensor nodes is randomly selected as the source node and the rest 99 nodes are storage nodes. The source node has 1000 overflow data items to be offloaded, each of which has size 512B. The initial energy level of each node is a random number in [100J, 200J]. We vary the storage capacity of each storage node from 11, 13, 15, to 17. That is, each node can store 11, 13, 15 and 17 data items, respectively. **We generate two plots: one shows the performance comparison of the four algorithms in terms of the total data resilience level each achieved, the other shows the performance comparison of the four algorithms in terms of the total energy consumption each incurred.**

References

[1] M. Albano and J. Gao. Resilient data-centric storage in wireless ad-hoc sensor networks. In *Proc. of ALGOSENSOR*, 2010.

[2] H. Chenji and R. Stoleru. Mobile sensor network localization in harsh environments. In *Proc. of DCOSS*, 2010.

[3] Thomas Corman, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009.

[4] S. Edwards, T. Murray, T. O'Farrell, I. C. Rutt, P. Loskot, I. Martin, N. Selmes, R. Aspey, T. James, S. L. Bevan, and T. Baugé. A High-Resolution Sensor Network for Monitoring Glacier Dynamics. *IEEE SENSORS JOURNAL*, 14(11):3926–3931, 2013.

[5] J. Heidemann, M. Stojanovic, and M. Zorzi. Underwater sensor networks: applications, advances and challenges. *Phil. Trans. R. Soc. A*, 370:158 – 175, 2012.

[6] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of HICSS 2000*.

[7] J. Jeong, X. Jiang, and D. Culler. Design and analysis of micro-solar power systems for wireless sensor networks. In *Proc. of INSS*, 2008.

[8] Yunfeng Lin, Ben Liang, and Baochun Li. Data persistence in large-scale sensor networks with decentralized fountain codes. In *Proc. of INFOCOM*, 2007.

[9] L. Luo, Q. Cao, C. Huang, L. Wang, T. Abdelzaher, and J. Stankovic. Design, implementation, and evaluation of enviromic: A storage-centric audio sensor network. *ACM Transactions on Sensor Networks*, 5(3):1–35, 2009.

[10] Ioannis Mathioudakis, Neil M. White, and Nick R. Harris. Wireless sensor networks: Applications utilizing satellite links. In *Proc. of the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007)*, pages 1–5, 2007.

[11] D. Mosse and G. Gadola. Controlling wind harvesting with wireless sensor networks. In *Proc. of IGCC*, 2012.

[12] L. Mottola. Programming storage-centric sensor networks with squirrel. In *Proc. of IPSN*, 2010.

[13] D. Puccinelli and M. Haenggi. Reliable data delivery in large-scale low-power sensor networks. *ACM Trans. Sen. Netw.*, 6(4):28:1–28:41, 2010.

[14] R. Sugihara and R. K. Gupta. Path planning of data mules in sensor networks. *ACM Trans. Sen. Netw.*, 8(1):1:1–1:27, 2011.

[15] Z. Sun and I. F. Akyildiz. On capacity of magnetic induction-based wireless underground sensor networks. In *Proc. of INFOCOM*, 2012.

[16] Rui Tan, Guoliang Xin, Jinzhu Chen, Wen-Zhan Song, and Renjie Huang. Fusion-based volcanic earthquake detection and timing in wireless sensor networks. *ACM Transaction on Sensor Networks (ACM TOSN)*, 9, 2013.

[17] B. Weiss, , H.L. Truong, W. Schott, A. Munari, C. Lombriser, U. Hunkeler, and P. Chevillat. A power-efficient wireless sensor network for continuously monitoring seismic vibrations. In *Proc. of SECON*, 2011.

[18] M. Z. Zamalloa and B. Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Transactions on Sensor Networks*, 3(2):1277–1280, 2007.