

Maximum Lifetime Data Regeneration for Persistent Storage in Wireless Sensor Networks

Soji Omiwade
Department of Computer Science
University of Houston
Houston, TX 77204
oomiwade@uh.edu

Rong Zheng
Department of Computer Science
University of Houston
Houston, TX 77204
rzheng@uh.edu

Abstract—Erasure codes have been employed to achieve persistent storage in distributed storage networks. Recent work has shown that, in addition to reduction in storage space requirements, the communication bandwidth in the data regeneration process can be further reduced by using Regenerating Codes. In this paper, we consider the issue of energy-efficient data regeneration in wireless sensor networks with the objective of minimizing energy expenditure and thereby maximizing network lifetime. We formally prove the NP-hardness of finding the optimal set of source nodes and corresponding routes for data regeneration in general networks, and devise an optimal polynomial algorithm, TROY, for acyclic networks; the cardinality of the set is pre-defined. Building upon TROY, we devise a heuristic algorithm for general networks and show, through extensive simulation studies, that this heuristic is near-optimal.

I. INTRODUCTION

Any data sensing node in a wireless sensor network (WSN) may store its own data, distributedly store it within the WSN or upload the data immediately to a central location. These three methods of WSN data storage are respectively called local, in-network and external storage. Local storage is undesirable because if a node fails—becomes unreachable to the rest of the network either because of software faults in the node itself or destruction of the node due to environmental conditions—the data it stores is lost. In-network data storage avoids data loss by storing any data item robustly either via replication or erasure coding.

Erasure codes have been established as a means to improve data robustness without incurring much storage overhead compared to replication, the obvious alternative for fault tolerance [1]. Two types of operations may occur in a network of storage nodes when erasure codes are employed. The first is *reconstruction* of the original data, which happens when an external entity needs to retrieve the stored information. The second operation is *regeneration* of lost/damaged coded pieces, and occurs when some storage node fails and its stored information needs to be recovered onto a *newcomer* node to maintain the level of redundancy of the system prior to the node's failure.

Using Regenerating Coding (RC) codes—a generalization of (n, k) -MDS codes, recent advancements have been shown to significantly reduce the data regeneration bandwidth [1]. To the best of our knowledge, little work has been done in

designing power-aware source selection and routing schemes for RC codes. A common metric in power-aware protocols in WSNs is network lifetime, defined as the time until the first node is depleted of its energy resource during the data regeneration process [2], [3]. This definition, is motivated by the fact that once a node is depleted of its energy, it becomes inactive and thus cannot relay traffic or generate sensing data.

Our goal then in this paper is to provide efficient algorithms that maximize the network lifetime by determining (i) *which* set of nodes and (ii) *which* routes to use in the data regeneration of the newcomer. Our contribution is multi-folded, as we (i) prove that maximizing the network lifetime for data regeneration is an NP-hard problem, (ii) propose an optimal maximum lifetime algorithm, TROY, for acyclic networks, and (iii) develop a heuristic for general networks that is empirically shown to be near optimal.

The rest of the paper is organized as follows. A background on RC codes is provided in Section II. In Section III, we formally define the maximum network lifetime problem for data regeneration and prove its hardness. In Section IV, we provide and characterize a novel optimal network lifetime algorithm for data regeneration in acyclic networks. In Section V, a heuristic solution for general networks is proposed, while extensive evaluation results are provided in Section VI. Related work is briefly reviewed in Section VII followed by a conclusion in Section VIII.

II. BACKGROUND

MDS codes: For (n, k) -MDS erasure codes, a data item of size B is partitioned into k equal sized pieces of size B/k , encoded and stored on n storage nodes, such that each node stores an encoded piece of size B/k . The original data can then be reconstructed by having a given data collector download k pieces from k nodes [1]. Therefore, if a storage node fails, a newcomer node can replace the failed node's data by simply reconstructing the original data, and then re-encoding to obtain the lost piece.

RC Codes: To regenerate the coded piece from a failed node, d nodes upload a *fraction* of their respective coded piece to the newcomer incurring less total data regeneration traffic compared to reconstruction [1]. For these codes, we denote

the upload bandwidth per each of the d uploading nodes as β . RC codes are able to achieve bandwidth reduction by tuning α and d , parameters that are fixed in classical (n, k) -MDS codes. Specifically, given the d uploading nodes each provides a fragment of size β , we have the following relation:

$$\alpha(d, \beta) = \begin{cases} \frac{B}{k}, & \beta \in [f(0), +\infty) \\ \frac{2B - g(i)\beta}{2(k-i)}, & \beta \in [f(i), f(i-1)) \end{cases}, \quad (1)$$

where $i = 1, \dots, k-1$ and

$$f(i) \triangleq \frac{2B}{2ik - i^2 - i + 2k + 2kd - 2k^2},$$

$$g(i) \triangleq (2d - 2k + i + 1)i.$$

As an example, if $\alpha = B/k$, then we can simplify (1) to obtain (2). Note then that each node stores the minimum possible for any (n, k) -MDS codes. The resultant codes are called *Minimum Storage Regenerating* (MSR) codes. (2) shows that the parameter d can be tuned to reduce the total download bandwidth.

$$(\alpha_{\text{MSR}}, \beta_{\text{MSR}}) = \left(\frac{B}{k}, \frac{B}{(d-k+1)k} \right) \quad (2)$$

$$(\alpha_{\text{MBR}}, \beta_{\text{MBR}}) = \left(\frac{2Bd}{2kd - k^2 + k}, \frac{2B}{2kd - k^2 + k} \right) \quad (3)$$

The minimum regeneration bandwidth can be achieved when $\beta = f(k-1)$ for any given d , as shown in (3). This assignment trades off minimum storage for minimum regeneration bandwidth. The resultant codes are called *Minimum Bandwidth Regenerating* (MBR) codes. As a final note, for any given α and any RC code, the regeneration bandwidth decreases with d .

III. THE d -RC PROBLEM

A. Network model

Consider a network modeled as a graph $G(V, A)$, where $V \triangleq \{1, \dots, n\}$ is the set of all nodes and $A = \{(i, j) : j \in \mathcal{N}(i)\}$ is the set of all links in the network, and $\mathcal{N}(i) : i \in V$ is the set of direct neighbors of node i . Each node has a limited energy, U_i , and each link (i, j) is associated with an energy cost per unit data transmitted from i to j , a_{ij} . Here, we assume the power consumption for packet reception is similar to the idle power consumption, and thus can be treated as a constant independent of whether the nodes are involved in data relay or reception. Without loss of generality, assume node n is the newcomer and q_{ij} is the total amount of regeneration data relayed on link (i, j) toward n . The network lifetime is associated with the time until the first node in N runs out of power.

$$T = \min_{i \in N} \frac{U_i}{\sum_{j \in \mathcal{N}(i)} a_{ij} q_{ij}} \quad (4)$$

The d -RC problem is to choose d nodes to upload their respective data unit, in the data regeneration of the newcomer, as well as d corresponding routes—one route per node—such that the maximum network lifetime is achieved with respect to the data regeneration traffic in the WSN. We characterize this

problem first by modeling it as an integer linear programming (ILP) problem, using a path formulation to model regeneration traffic. Specifically, $f^i(P)$ is an indicator variable denoting whether path P will be used by node i . Let \mathcal{P}^i be the set of all paths from node i to node n . For any node i in N , y_e^i is the total flow (i.e., the total number of bits delivered) on edge e from node i . An indicator variable δ_i is associated with a node i denoting whether node i will *share* its data for the regeneration. Let $T' = 1/T$, then maximizing T is equivalent to minimizing T' under the following constraints:

$$\sum_{i \in \mathcal{N}(0)} \delta_i = d \quad (5)$$

$$\sum_{P \in \mathcal{P}^i} f^i(P) = \delta_i \quad \forall i \in \mathcal{N}(0) \quad (6)$$

$$\sum_{j \in \mathcal{N}(0)} \sum_{P \in \mathcal{P}^j : e \in P} f^j(P) = y_e^e \quad \forall e \in E^+(i), i \in N \quad (7)$$

$$\sum_{e \in E^+(i)} a_e y_e^e \leq \frac{U_i}{\beta} T' \quad \forall i \in V \quad (8)$$

Constraint (5) ensures that exactly d nodes are selected. By $f^i(P) \in \{0, 1\} : i \in N$, constraint (6) ensures that each of d sources uses exactly one route toward the newcomer. Constraint (7) is due to flow conservation: the amount of traffic relayed on edge e is equal to the total traffic from all source nodes whose routes include e . Note that $f^j(P) \in \{0, 1\}$ implies $y_e^e \in \mathbb{Z}$. Finally, constraint (8) relates the network lifetime defined in (4) with the ratio of node i 's initial energy to the energy expended in routing data to the newcomer. We conclude the formulation by noting that a node may relay other node's data in addition to *uploading* its own data.

B. Hardness of d -RC

Next we show that the d -RC problem is NP-hard by considering its decision form: to select d nodes and d corresponding routes such that the network lifetime is at least T for some nonnegative number T . As a decision problem, any instance of the d -RC problem outputs yes iff $R' \geq L'$ for some $L' \geq 0$. The output of any d -RC instance is determined by the assignment of values to $\{q'_{ij}\}$ for each (i, j) in A' .

Lemma 1: The d -RC problem is NP-complete.

Proof: The NP-completeness follows by reducing from the max- L problem [3]. For space constraints, we omit the proof. ■

Lemma 2: The d -RC problem is NP-hard, even when an arbitrary number of relay nodes are included to increase the network connectivity.

Proof: We reduce from the relaxed max- L problem [3], which is known to be NP-complete. As a decision problem, any instance of the relaxed max- L problem seeks $|D| : D \subseteq V - \{n\}$ routes in a sensor network $G(V = \{1, 2, \dots, n\}, A)$, one route per data collecting node in a *given* set D toward sink n , where the network lifetime is at least some nonnegative number, T . Again, U_i , $\mathcal{N}(i)$, and $a_{ij} : j \in \mathcal{N}(i)$ are defined similarly for any node i .

Next, we show that the instance of relaxed max- L achieves network lifetime T from a given set of sensor nodes D if and only if an instance of the d -RC problem chooses a set $D' : |D'| = |D|$ that achieves network lifetime no less than T . To do so, consider the following transformation, where $G(V', E')$ corresponds to our d -RC instance. Let $V' \triangleq V \cup \{R\}$, where R is a set of relays and is given by $\{1', 2', \dots, |N|\}$. For each node i in the relaxed max- L instance, $E' = E \cup \{(i, i')\}$, such that $a_{ii'}$ is 0 if $i \in D$; otherwise $a_{ii'} = \infty$. Finally, if we set d to $|D|$, then for the d -RC instance, node i can forward its data only if i is in D . Since there is no energy loss for each node i in D to forward data to its only neighbor i' , the d -RC problem is equivalent to the relaxed max- L instance. ■

IV. DATA REGENERATION IN ACYCLIC NETWORKS

The hardness of the d -RC problem arises from the exponential number of possible paths from each source node to the newcomer. This motivates us to consider acyclic networks, where we find that the decision problem of finding d optimal sources guaranteeing maximum network lifetime in this case is no longer NP-hard. Specifically, we provide an optimal polynomial algorithm, TROY for the d -RC problem in acyclic networks. The investigation of acyclic networks has practical implications since many sensor network operations utilize a sink tree.

A. Algorithm TROY

To develop the data regeneration algorithm in acyclic networks, we first characterize the set of nodes attaining the optimal lifetime. Since the network is acyclic, each node $i \in N \triangleq V - \{n\}$ has exactly one route to the newcomer n . Let P_i be the route used by node i and let q_i refer to $q_{i\pi(i)}$, where $\pi(i)$ is the next-hop of node i toward the newcomer. Let a_i be the corresponding transmission cost for node i . Suppose that X is the set of all nodes that uploads their respective units to the newcomer. Denote the total data, due to X , on link $(i, \pi(i))$ as $q_i(X)$. We now offer an alternate definition for the network lifetime, in terms of X . For each $v \in N$, let $\pi(v)$ be the parent node of v . Define the *node lifetime* for any node $v \in N$, as shown in (9). For any set of nodes X uploading to the newcomer, we derive the function shown in (10), where $\delta_i(X) = 1$, if $i \in X$. Otherwise, $\delta_i(X) = 0$.

The network lifetime expressed in (4) can now be written as $l(N, X)$. Using (10), we define the *path lifetime*, $l(P_v, X)$, of a node $v \in N$ as the lifetime of the node that depletes its energy first, amongst all nodes in P_v ; relation (11) and (12) follows from this definition. Finally, (13) provides an alternate to our prior definition of the network lifetime in (4).

$$l(v, X) \triangleq \frac{U_v}{a_v q_v(X)} \text{ for all } v \in N \quad (9)$$

$$l(W, X) = \min_{u \in W} l(u, X) \text{ for all } W \subseteq N \quad (10)$$

$$l(P_u, X) \leq l(P_v, X) \text{ if } v \in P_u \text{ for all } u, v \in N \quad (11)$$

$$l(P_v, D) = \min(l(P_u \cap P_v, D), l(P_v - P_u, D)) \quad (12)$$

$$l(X) \triangleq \min_{u \in N} l(P_u, X) \quad (13)$$

Require: Condition (14) holds

Ensure: $\forall D' \subseteq N : |D'| = |D| = d, l(D) \geq l(D')$

1: **repeat**

2: $C \leftarrow \{u : \pi(u) \in D \cup \{n\}, \forall u \in N - D\}$

3: $D \leftarrow D \cup \{\arg \max_{u \in C} l(P_u, D \cup \{u\})\}$

4: **until** $|D| = d$

Fig. 1. Starting from an empty set D , TROY iteratively selects d nodes in the network, $G(V, A)$. If $G(V, A)$ is acyclic, then the set D upon TROY's termination maximizes the network lifetime for data regeneration.

Now we are in the position to present our polynomial time data regeneration algorithm, TROY, for achieving maximum network lifetime in acyclic networks. As illustrated in Figure 1, TROY starts from the immediate neighbors of the newcomer and iteratively selects source nodes, until d nodes have been selected. As shown in (14), $d \geq k$ must hold for successful data regeneration, since we use (n, k, d) -RC codes.

$$D = \emptyset, \pi(u) \in \mathcal{N}(u) : u \in N \text{ and } d \geq k > 0 \quad (14)$$

Let D_i be the set of nodes selected up until the end of iteration i . In the $i + 1$ -th iteration, TROY chooses a node u from the immediate neighbors of nodes already in D_i , but not yet included in D_i , such that node u has the maximum node lifetime $l(P_u, D_i)$. To determine the complexity of TROY, consider again the i -th iteration. It takes $O(d)$ to compute the path lifetime associated for any candidate node u in Line 2. There are $O(|D_i|b)$ nodes in C , where b is the average node degree. Moreover, $|D_i| \leq d$ for any iteration i . Hence, the cost of TROY for all d iterations is then $O(d^3b)$. This cost can be made quadratic in d , if in iteration i , we store the path lifetime $l(P_u, D_i - \{v_i\} \cup \{u\})$ of each candidate u , where v_i is the node selected in iteration i . Then in iteration $i + 1$, only the path lifetime of nodes in $\mathcal{N}(v_i) - \{\pi(v_i)\}$, instead of the entire set C , needs to be updated and compared with the stored path lifetime for each candidate $u \neq v_i$ from iteration i . Thus, the time complexity for TROY becomes $O(d^2b^2)$, which is generally better than $O(d^3b)$ as typically $b \ll d$.

B. Optimality of TROY

To prove TROY's optimality, we first establish a few technical lemmas.

Lemma 3: Let P_{v_i} be the path selected in the i -th step (Line 3) of TROY, and D_i be the resultant set. Then $l(D_i)$ equals $l(P_{v_i}, D_i)$.

Proof: We prove by induction. Specifically, we show that the inequality in (15) holds.

$$l(P_{v_i}, D_i) \leq l(P_u, D_i) \text{ for all } u \in D_i \quad (15)$$

Basis: This trivially holds, since the cardinality of D_1 is 1.

Induction step: Suppose the statement is true for iteration i . For each $u \in D_i$, we consider whether or not node u is in $P_{v_{i+1}}$. If $u \in P_{v_{i+1}}$ then (15) holds from (11).

Suppose $u \notin P_{v_{i+1}}$. From (12), note that by subtracting $l(P_u \cap P_{v_{i+1}}, D)$ from both sides of (15), it suffices to show

Require: $D = D'$

Ensure: Condition (22) holds

```

1:  $\forall u \in D, \text{color}[u] \leftarrow \text{white}$ 
2: while  $M(D) \neq \emptyset$  do
3:    $v \leftarrow \arg \max_{u \in D: \text{color}[u] = \text{white}} |P_u|$ 
4:   if  $P_v \cap M(D) \neq \emptyset$  then
5:      $w \leftarrow \arg \min_{u \in P_v \cap M(D)} |P_u|$ 
6:      $D \leftarrow D \cup \{w\} - \{v\}$ 
7:      $\text{color}[w] \leftarrow \text{black}$ 
8:   else
9:      $\text{color}[v] \leftarrow \text{black}$ 
10:  end if
11: end while

```

Fig. 2. The relay reduction procedure

that $l(P_u - P_{v_{i+1}}, D)$ is not more than $l(P_{v_{i+1}} - P_u, D)$. To complete the proof, we first show that (16) holds, for the case that v_i is in $P_{v_{i+1}}$ and the case that v_i is not in $P_{v_{i+1}}$.

$$l(P_{v_i}, D_i) \geq l(P_{v_{i+1}}, D_{i+1}) \quad (16)$$

For the former case, (16) follows directly from (11) and the fact that $l(P_{v_i}, D_i)$ is not more than $l(P_{v_i}, D_{i+1})$. For $v_i \notin P_{v_{i+1}}$, (16) follows from the result in (18) as follows: (17) comes from TROY selecting v_i over v_{i+1} in the i -th iteration. Then (18) follows from the fact that adding node v_i to the union of D_i and v_{i+1} cannot decrease the network lifetime.

$$l(P_{v_i}, D_i) \geq l(P_{v_{i+1}}, (D_i - \{v_i\}) \cup \{v_{i+1}\}) \quad (17)$$

$$\geq l(P_{v_{i+1}}, D_i \cup \{v_{i+1}\}) \quad (18)$$

(19) holds because the path lifetime of any node $\theta \in P_u - P_{v_{i+1}}$ is not affected by adding or removing any node in $P_{v_{i+1}}$. (20) follows from the inductive hypothesis and the fact that $P_u - P_{v_{i+1}} \subseteq P_u$ holds. Finally, (21) follows directly from (16).

$$l(P_u - P_{v_{i+1}}, D_{i+1}) = l(P_u - P_{v_{i+1}}, D_i) \quad (19)$$

$$\geq l(P_{v_i}, D_i) \quad (20)$$

$$\geq l(P_{v_{i+1}}, D_{i+1}) \quad (21)$$

Consider now an algorithm A that selects d source nodes for data regeneration. In general, the tree spanned by the d source nodes rooted at the newcomer has more than d nodes. Thus, some nodes serve as relays only. Next, we show that the relay nodes can in fact be removed without reducing the network lifetime.

Let $D' = \{i | i \in N, \delta_i = 1\}$ be the set of source nodes chosen by algorithm A . We define $M(D')$ as the set of nodes that relay traffic for D' in the regeneration. That is, $M(D') = \{i \in P_u | u \in D', \delta_i = 0\}$. Starting from $D' \cup M(D')$, the relay reduction algorithm illustrated in Figure 2, iteratively adjusts D until $M(D)$ is empty. It does this without reducing the network lifetime, as shown in Lemma 4.

Lemma 4: Let D' and D be the source nodes generated by an algorithm A and by applying the procedure in Figure 2 on D' , respectively. Then, $l(D) \geq l(D')$, and (22) holds for D .

$$M(D) = \emptyset, l(D) \geq l(D') \text{ and } |D| = |D'| \quad (22)$$

Proof: As shown in Figure 2, we begin with $D = D'$ and modify D until $M(D) = \emptyset$. Let D_i' and D_i be the set of source nodes before and after the modification in iteration i , respectively. Considering iteration i , Line 3 finds the longest path with some relay node on it. We remove the furthest source node from the newcomer on that path. We then add to D_i the relay node, on that path, that is closest to the newcomer. Clearly, $l(D_i) \geq l(D_i')$. ■

In each iteration of the relay reduction procedure, one node is colored black and is never visited again. Therefore, the algorithm terminates. Moreover, since Line 5 finds the closest node to the newcomer, the algorithm incurs a time complexity of $O(|V|)$. Lemma 4 establishes the fact that the optimal set of d source nodes form a tree together with the newcomer and no relay nodes, that are not source nodes, are needed.

Next, we consider the traffic routed on paths toward the newcomer. For any D and D' , let q_i and q'_i refer to $q_i(D)$ and $q_i(D')$ respectively.

Lemma 5: If there exists a node u such that $q_u > q'_u$, then there exists a node v such that (i) for all x in $P_v - \{v, n\}$, $q_x = q'_x$; (ii) there exists a node w in D , such that $v \in P_w$ and the intersection of $\mathcal{N}(w) - \{\pi(w)\}$ and D is empty. Moreover, for all x in $P_w - P_v$, we have $q_x < q'_x$. (iii) $q_v < q'_v$.

Proof: Given a node u , clearly for each node x in $P_u - \{u, n\}$ one of three cases must hold: (a) for any x in $P_u - \{u, n\}$, $q_x = q'_x$; (b) there exists a node x in $P_u - \{u, n\}$ such that $q_x < q'_x$; or (c) there exists a node x in $P_u - \{u, n\}$ such that $q_x > q'_x$.

In case (a), since $q_u > q'_u$ and $q_x = q'_x$ for all x in $P_u - \{u, n\}$, then necessarily, there exists a node v such that $\pi(v) = \pi(u)$ and $q_v < q'_v$. Thus, the condition (i) and (iii) are satisfied. To see that (ii) holds, consider the subtree rooted at v . Since q_v is less than q'_v , then there exists x in $\mathcal{N}(v) - \{\pi(v)\}$ such that q_x is less than q'_x . By applying this argument recursively, then (ii) also holds. In case (b), we identify the first such node x in $P_u - \{u, n\}$ starting from node n . Setting v to this x and using an arguments similar to those of case (a), we can show that conditions (i) – (iii) hold. In case (c), we again identify the first such node x in $P_u - \{u, n\}$ from node n . Since $q_{\pi(x)} = q'_{\pi(x)}$ and $q_x > q'_x$, then we can find a node $v : \pi(v) = \pi(w)$ and $q_v < q'_v$. Using case (b), conditions (i) – (iii) hold for v . ■

We are now in a position to prove TROY's optimality.

Theorem 1: Algorithm TROY illustrated in Figure 1 attains maximum lifetime in regenerating a newcomer's data.

Proof: Let θ be the last node TROY selects, and D be the resulting set. Also, by Lemma 4, we can suppose $M(D') = M(D) = \emptyset$. Suppose, by contradiction, the premise in (23) holds.

$$l(D') > l(D) \text{ where } D' \neq D, |D| = |D'| \quad (23)$$

Let $\phi \in D$ be such that $l(D) = l(\phi, D)$. Note then that since $l(D') > l(D)$, we have $l(\phi, D) < l(\phi, D')$, and consequently $q_\phi > q'_\phi$. Then by Lemma 5, we can find a node v satisfying conditions (i) – (iii) for that Lemma. Therefore, there exists a node $w \in D'$, but $w \notin D$, such that $\pi(w) \in D \cup \{n\}$ and $v \in P_w$.

We proceed to characterize any D' . Relation (25) holds from Lemma 5, and (26) comes from the fact that removing nodes from D can only increase the network lifetime. Relation (27) holds because node θ is the last node added by TROY in Line 3 in Figure 1. Finally, (28) follows from Lemma 3.

$$l(D') \leq l(P_w, D') \quad (24)$$

$$\leq l(P_w, D \cup \{w\}) \quad (25)$$

$$\leq l(P_w, D \cup \{w\} - \{\theta\}) \quad (26)$$

$$\leq l(P_\theta, D) \quad (27)$$

$$= l(D) \quad (28)$$

But (28) contradicts the premise in (23). Therefore, Theorem 1 holds, and TROY achieves the optimal network lifetime in selecting d nodes to regenerate data at the newcomer. ■

V. GENERAL NETWORKS

Even though TROY is optimal in acyclic networks, no tree construction can always achieve the optimal in generic networks, as illustrated in Figure 3. Motivated by this observation, we devise a data regeneration algorithm, CRETE, illustrated in Figure 4 for general networks. CRETE allows a node to distribute regeneration units by considering passively learned paths toward the newcomer. For the set of distinct nodes D selected by CRETE, let Φ be the corresponding set of routes, such that D and Φ have the same cardinality, and the source node of each route in Φ is in D . We can derive the node lifetime function in (29), where $q_{uv}(\Phi)$ is the amount of data relayed on link (u, v) using Φ . The network lifetime, $l(\Phi)$ is given by $\min_{v \in N} l(v, \Phi)$.

$$l(v, \Phi) = \frac{U_v}{\sum_{v' \in N(v)} a_{vv'} q_{vv'}(\Phi)} \text{ for all } v \in N \quad (29)$$

Using Algorithm CRETE in Figure 4, each node passively learns routes to the newcomer based on routes that have already been selected to regenerate the lost data. Let $P(v)$ be the set of paths that node v has learned to reach the newcomer and D_i be the set of source nodes CRETE has selected until the start of iteration $i + 1$. For iteration $i + 1$, only neighbors of nodes in $D_i \cup \{n\}$ are considered (Line 2); the paths of any considered node u are the paths associated with its neighbors in D_i concatenated with node u (Line 5). Amongst all considered nodes with their respective routes, the node having maximum path lifetime for iteration i is added to D_i to yield D_{i+1} . Moreover, its path will be used to regenerate the newcomer (Line 9).

Obviously, CRETE is an approximation algorithm via Lemma 1, where we showed that it is NP-hard to achieve

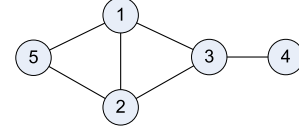


Fig. 3. Consider any $(5, k, 4)$ -RC code where $B = U_i = a_{ij} = 1$, for each $i \in N, j \in \mathcal{N}(i)$. If node 5 is the newcomer, then any tree construction must satisfy $q_{3-i} = 2$, for some $i \in \{1, 2\}$. However, the optimal lifetime is achieved only with the assignment $q_{3-i} = 1$ for all $i \in \{1, 2\}$.

Require: $\Phi = \emptyset; P(n) = \{n\}; P(u) = \emptyset, \forall u \neq n$

```

1: repeat
2:    $C \leftarrow \{u : \mathcal{N}(u) \cap (D \cup \{n\}) \neq \emptyset, \forall u \in N - D\}$ 
3:   for all  $u \in C$  do
4:     for all  $v \in \mathcal{N}(u) \cap (D \cup \{n\})$  do
5:        $P(u) \leftarrow P(u) \cup \{u\} - P(v)$ 
6:     end for
7:      $\xi(u) \leftarrow \max_{P \in P(u)} l(P, \Phi \cup P)$ 
8:   end for
9:    $\Phi \leftarrow \Phi \cup \{\arg \max_{P(u): u \in C} \xi(u)\}$ 
10: until  $|\Phi| = d$ 

```

Fig. 4. CRETE for generic networks

maximum network lifetime. Consider the parameters from the caption in Figure 3 as an illustration, with the exception that $d = 3$, and $a_{2-5} = 2 + \epsilon$, where $0 < \epsilon < 1$. The maximum network lifetime is given by the routes, $\{(1, 5), (2, 5), (3, 1, 5)\}$, and its value is $\frac{1}{2+\epsilon}$. CRETE, however, can output the routes, $(1, 5), (2, 1, 5), (3, 1, 5)$ in that order, with a network lifetime of $\frac{1}{3}$. Hence, even though CRETE always makes the optimal route selection for any iteration i (Line 7), such a selection may be suboptimal for some other iteration $i' > i$.

VI. EVALUATION

We have implemented TROY and CRETE in Python; the results are from simulation runs of 50 randomly generated 25-node random networks. For each network, the newcomer is chosen randomly. The initial energy of all nodes is uniformly distributed in $(0, 30)$. Each node is uniformly positioned in a 25 sq. unit plane, where the transmission cost for link (i, j) is $a_{ij} = 0.001 \cdot d_{ij}^3$, and d_{ij} is the distance between node i and node j ; $j \in \mathcal{N}(i)$ only if $d_{ij} < 3$ units. For each storage scheme, CPLEX is used to select the source nodes and routes optimally [4], using the integer linear programming formulation in Section III-A.

We first study the effects of different fault tolerant storage mechanisms: replication, classical (n, k) -MDS codes, and RC codes. For replication, each node stores a replica of the original data. Hence, $\alpha = \beta = B$. For the erasure codes, we consider d in $\{12, 18\}$. Figure 5 shows the normalized network lifetime achieved by classical MDS and RC codes with respect to replication. We observe that RC codes generally outperform classical erasure codes. From the literature on RC codes, an increase in d results in much lower data regeneration bandwidth [1]; however, Figure 5 shows that an increase in

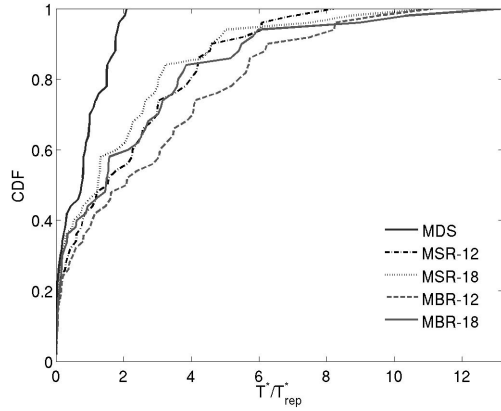


Fig. 5. Network lifetime performance of fault tolerant storage schemes. Results are normalized by the lifetime of replication.

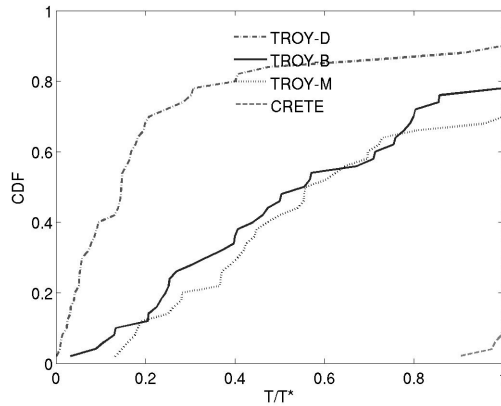


Fig. 6. Network lifetime performance of TROY and CRETE, using MBR codes with $d = 18$. Results are normalized by the optimal solution.

d is undesirable in maximizing the network lifetime for the random topologies utilizing MBR codes.

We now evaluate the performance of TROY and CRETE, using $(25,6)$ -MBR codes. Along with TROY, breadth-first search (TROY-B), depth-first search (TROY-D) and minimum spanning trees (TROY-M) are considered. To account for the residual energy and transmission cost, in constructing the minimum spanning tree, we assign the cost $a_{ij}^{\text{MST}} = \frac{\min(U_i, U_j)}{a_{ij}}$. From Figure 6, we observe that TROY-M performs the best amongst TROY-based schemes since it accounts for the residual energy and transmission costs. Also, CRETE has near optimal performance: in more than 80% of the simulation runs, CRETE is optimal. This is because for any iteration i of CRETE, even if $l(\Phi_i)$ is less than the optimal lifetime $l(\Phi_i^*)$ when d equals i , it is likely that $l(\Phi_{i'})$ equals $l(\Phi_i^*)$ for some iteration $i' > i$.

VII. RELATED WORK

Since RC codes have been shown to have much lower data regeneration bandwidth than replication, much work has

focused on RC-code design and structure [1]. Unlike the work proposed in this paper, little work has considered the impact of RC codes in WSNs. Since the seminal work of Tassiulas et al. to maximize the network lifetime in a WSN, where all nodes route their data to a sink node, several optimal centralized and distributed polynomial-time algorithms have been developed for maximizing network lifetime in the case of single-path and multi-path routing per source node [2]. Kar et al. and Li et al. propose algorithms for maximizing network capacity and network lifetime, respectively [5], [6] for an unknown sequence of data flows in a WSN. None of these works consider power aware routing algorithms from a subset of source nodes in the WSN to maximize the network lifetime. Xiong et al. consider the case where the lifetime is to be maximized only from a *fixed* set of source nodes [3]. Moreover, their algorithms have exponential computation complexity. Our work in this paper, however, proposes polynomial time algorithms that account not only for optimal route selection, but also optimal node selection—a requirement for maximum network lifetime.

VIII. CONCLUSION

In this paper, we have investigated the problem of achieving maximum network lifetime for data regeneration in wireless sensor networks. We proposed two algorithms, TROY and CRETE, and have shown that they have optimal and near optimal performance in acyclic and general networks, respectively.

ACKNOWLEDGMENT

Our sincerest gratitude to Dr. Gabriel Scalosub on his input regarding the hardness of the d -RC problem, and the anonymous reviewers for their valuable comments. Dr. Zheng received the National Science Foundation (NSF) CAREER Award in 2006. This work is partially funded by the NSF under grant NSF CNS 0546391.

REFERENCES

- [1] A. Dimakis, K. Ramchandran, Y. Wu, and C. Suh, "A Survey on Network Codes for Distributed Storage," *Proceedings of the IEEE*, vol. 99, no. 3, 2011.
- [2] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 6–28, 2004.
- [3] S. Xiong, J. Li, and L. Yu, "Maximize the lifetime of a data-gathering wireless sensor network," in *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 2009, pp. 306–314.
- [4] IBM ILOG, "Cplex optimizer," 2010. [Online]. Available: <http://www-01.ibm.com>
- [5] K. Kar, M. Kodialam, T. Lakshman, and L. Tassiulas, "Routing for network capacity maximization in energy-constrained ad-hoc networks," in *INFOCOM 2003. The 22nd Conference on Computer Communications*. IEEE, vol. 1. IEEE, 2003, pp. 673–681.
- [6] Q. Li, J. Aslam, and D. Rus, "Online power-aware routing in wireless ad-hoc networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, p. 107.