TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 10/11 pp230–238
DOI: 10.26599/TST.2019.9010061
Volume 26, Number 2, April 2021

Reliable Data Storage in Heterogeneous Wireless Sensor Networks by Jointly Optimizing Routing and Storage Node Deployment

Huan Yang, Feng Li*, Dongxiao Yu*, Yifei Zou, and Jiguo Yu

Abstract: In the era of big data, sensor networks have been pervasively deployed, producing a large amount of data for various applications. However, because sensor networks are usually placed in hostile environments, managing the huge volume of data is a very challenging issue. In this study, we mainly focus on the data storage reliability problem in heterogeneous wireless sensor networks where robust storage nodes are deployed in sensor networks and data redundancy is utilized through coding techniques. To minimize data delivery and data storage costs, we design an algorithm to jointly optimize data routing and storage node deployment. The problem can be formulated as a binary nonlinear combinatorial optimization problem, and due to its NP-hardness, designing approximation algorithms is highly nontrivial. By leveraging the Markov approximation framework, we elaborately design an efficient algorithm driven by a continuous-time Markov chain to schedule the deployment of the storage node and corresponding routing strategy. We also perform extensive simulations to verify the efficacy of our algorithm.

Key words: reliable data storage; routing; node deployment; heterogeneous sensor networks

1 Introduction

In the past decades, we have witnessed the rapid development of Wireless Sensor Networks (WSNs) in various applications[1]. In the era of big data, a diversity of sensors has been pervasively deployed and networked for monitoring and surveying purpose[2–5]. Nevertheless, due to the proliferation of WSNs, which results in big sensory data[6–11], data storage has become a very challenging issue. The sensor nodes usually cannot provide qualified data storage services because they have only limited resources (e.g., for computing, data processing and storage, etc.). To enhance the capability of data storage in a WSN, one option is to employ designated storage nodes, forming a heterogeneous wireless sensor network, such that the data sensed by the regular sensor nodes are delivered to the storage node through multi-hop transmissions.

Nevertheless, as the sensor nodes may be deployed in hostile regions (e.g., battle fields and earthquake scenarios), fault tolerance should be one of the main concerns. Specifically, when a storage node is destroyed, all data contained in the storage node are corrupted. To address this issue, one popular solution is to introduce data redundancy. For instance, the reliability of the data storage can be considerably enhanced by relying on erasure codes[12, 13]. However, the induced data redundancy results in increased data traffic, which may be unaffordable to energy-limited sensor networks[14]. Furthermore, managing the increasing volume of data results in a considerable overhead in the storage nodes. The storage cost should also be taken into account when

• Huan Yang is with College of Computer Science and Technology, Qingdao University, Qingdao 266071, China. E-mail: cathy_huanyang@hotmail.com.
• Feng Li and Dongxiao Yu are with School of Computer Science and Technology, Shandong University, Qingdao 266237, China. E-mail: {fli, dxyu}@sdu.edu.cn.
• Yifei Zou is with Department of Computer Science, the University of Hong Kong, Hong Kong 999077, China. E-mail: yfzou@cs.hku.hk.
• Jiguo Yu is with School of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China and also with Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan 250014, China. E-mail: jiguoyu@sina.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2019-11-26; accepted: 2019-12-06

designing data storage schemes.

In this study, we design an algorithm to minimize the weighted sum of the data delivery and storage costs by jointly optimizing data routing and storage node deployment. In particular, by leveraging the Markov approximation framework[8], we elaborately design a Continuous-Time Markov Chain (CTMC). Through the algorithm, we can adaptively deploy storage nodes and update the data routing scheme, such that the induced cost can be nearly minimized and the reliability of the data storage can be ensured. The efficacy of our algorithm can be verified by theoretical analyses and extensive simulations.

The rest of this paper is organized as follows. We introduce our system model and formulate our optimization problem in Section 2. We present our algorithm and the related analysis in Section 3. We evaluate our algorithm through extensive simulations in Section 4. We survey related literatures in Section 5 and finally conclude our paper in Section 6.

2 System Model and Problem Formulation

We first introduce our system model in Section 2.1 and some preliminaries on storage reliability in Section 2.2. We then formulate our optimization problem in Section 2.3.

2.1 System model

We consider a bipartite graph $\mathcal{G}$ consisting of two disjoint sets of vertices $\mathcal{N}$ and $\mathcal{U}$. Let $\mathcal{N} = \{n_i\}_{i=1,\dots,|\mathcal{N}|}$ represent the set of $|\mathcal{N}|$ sensor nodes and $\mathcal{U} = \{u_k\}_{k=1,\dots,|\mathcal{U}|}$ represent the $|\mathcal{U}|$ sites where we can deploy our storage nodes. For each pair of $n_i \in \mathcal{N}$ and $u_k \in \mathcal{U}$, there exists an edge (n_i, u_k) between them, with which we define a weight $c_{i,k}$ representing the energy cost we have to spend on delivering one unit of data from n_i to the storage node placed at u_k. The data delivery cost assignment can be computed by, for example, any state-of-the-art shortest path algorithms, e.g., Dijkstra's algorithm or Floyd-Warshall algorithm, or other state-of-the-art distributed shortest path algorithms[15, 16].

Let $\mathcal{S} = \{s_j\}_{j=1,\dots,|\mathcal{S}|}$ represent a set of storage nodes, with $|\mathcal{S}|$ representing the size of $\mathcal{S}$. We assume τ_j denotes the cost for storing one unit of data at s_j. Specifically, the data storage cost includes not only the energy consumption on data communication (e.g., receiving the data from the sensor nodes and responding to the remote data queries), but also the one on maintaining and managing the received data. We suppose that the storage nodes have long-range communication modules to exchange messages with each other and handle remote data queries. We also assume that the storage nodes are equipped with mobile mechanisms[17, 18], if re-deployment is demanded for adapting to dynamic network states.

2.2 Reliable data storage

The reliability of a data storage can be achieved via replication or coding techniques. In the former scheme, the original data are replicated for multiple times, and each copy is placed at different storage nodes. In the latter scheme, we encode the original data by Maximum Distance Separable (MDS) codes, and then place each of the encoded data in a distinct storage node. In both schemes, the reliability is ensured by data redundancy. For instance, visiting only some of the storage nodes to partially access the replicated or encoded data is sufficient to recover the original data.

We hereby take coding techniques as an example. We suppose that λ_i is the data produce rate of sensor node n_i. Specifically, in each sensing-storage period (we assume that the data storage is performed periodically in individual nodes. In each period, the data cumulated in the sensing phase are encoded and delivered to the storage nodes.), the amount of data sensed by n_i is λ_i. These original data can be encoded by redundant coding schemes, e.g., MDS codes parameterized by (μ_i, λ_i). Particularly, node n_i encodes the λ_i original data into $(\mu_i - \lambda_i)$ parity data. Both the original and parity ones are then delivered from the source node n_i to μ_i distinct storage nodes. As the original data can be regarded as the ones encoded by an identity matrix, we call both the original and parity data "encoded data" for brevity in the following. The original data can be recovered based on any λ_i of the μ_i encoded data. In another word, we can tolerate the corruptions of any up to λ_i out of the μ_i encoded data (or the storage nodes containing the encoded data). Considering the whole system, at most $\min_i\{\mu_i - \lambda_i\}$ storage node failures are tolerable.

In the data replication scheme, we suppose that each node n_i replicates each of its data for multiple times. As the data are all the same, accessing any one of them is sufficient. Therefore, when these copies are stored in different storage nodes, we can tolerate only $\min_i\{\mu_i - 1\}$ failures of the storage node. Using this method, although the computation overhead in the encoding process can be avoided, we have to spend more energy on data delivery. Specifically, node n_i needs to deliver each of its μ_i copies (for every original data) to different storage nodes.

Our optimization framework is readily compatible to both replication and coding-based schemes by scaling the data delivery and storage costs; nevertheless, we focus only on the latter one in this study.

2.3 Problem formulation

Let $x_{i,k} \in \{0,1\}$ be a binary variable indicating if node $n_i \in \mathcal{N}$ delivers one of its encoded data to the storage node placed at site u_k (if any). We also suppose that $y_{j,k} \in \{0,1\}$ indicates if storage node s_j is deployed at site u_k. Given a coding scheme $\{(\mu_i, \lambda_i)\}_{i=1,\ldots,|\mathcal{N}|}$, we formulate our optimization problem of Joint roUting and Storage node deploymenT (JUST) as follows:

$$\min \sum_{i=1}^{|\mathcal{N}|}\sum_{k=1}^{|\mathcal{U}|} c_{i,k}x_{i,k} + \beta \sum_{i=1}^{|\mathcal{N}|}\sum_{k=1}^{|\mathcal{U}|}\sum_{j=1}^{|\mathcal{S}|} \tau_j x_{i,k} y_{j,k} \quad (1)$$

$$\text{s.t.} \sum_{k=1}^{|\mathcal{U}|} x_{i,k} = \mu_i,\ \forall n_i \in \mathcal{N} \quad (2)$$

$$\sum_{j=1}^{|\mathcal{S}|} y_{j,k} - x_{i,k} \geqslant 0,\ \forall n_i \in \mathcal{N}, u_k \in \mathcal{U} \quad (3)$$

$$\sum_{j=1}^{|\mathcal{S}|} y_{j,k} \leqslant 1,\ \forall u_k \in \mathcal{U} \quad (4)$$

$$\sum_{k=1}^{|\mathcal{U}|} y_{j,k} \leqslant 1,\ \forall s_j \in \mathcal{S} \quad (5)$$

$$x_{i,k} \in \{0,1\},\ \forall n_i \in \mathcal{N}, u_k \in \mathcal{U} \quad (6)$$

$$y_{j,k} \in \{0,1\},\ \forall s_j \in \mathcal{S}, u_k \in \mathcal{U} \quad (7)$$

The quadratic objective Function (1) delivers our aim to minimize the weighted sum of the data delivery and storage costs. We introduce parameter β in Function (1) to make a trade-off between them. The main constraints are explained as follows:

• **Storage reliability Constraint (2)**: Under the encoding scheme $\{(\lambda_i, \mu_i)\}_{i=1,\ldots,|\mathcal{N}|}$, each node n_i encodes its λ_i original data in each sensing-storage period, and the resulting encoded data are delivered to distinct storage nodes.

• **Data flow Constraint (3)**: Each sensor node cannot deliver its data to an "empty" site with no storage node deployed.

• **Deployment Constraints (4) and (5)**: For any site $u_k \in \mathcal{U}$, only one storage node is deployed, and each storage node $n_i \in \mathcal{N}$ is deployed at no more than one site.

The NP-hardness of the above optimization problem holds even in a homogeneous setting where all of the storage nodes have identical storage cost, i.e., $\tau_j = \tau$ for $\forall j = 1,\ldots,|\mathcal{S}|$. In this case, our problem can be transferred into minimizing $\sum_{i,k} c_{i,k}x_{i,k}$ with consideration of Constraints (2) – (7). Specifically, given a coding scheme, our goal is to choose a subset of $\mathcal{U}$ to deploy the storage nodes such that the induced cost of data delivery and storage is minimized while guaranteeing the storage reliability. The above degenerated problem is equivalent to a Fault-Tolerate k-Median Facility Placement (FTkMFP) problem, and hence, our original JUST problem is NP-hard.

3 Algorithm Design and Analysis

In this section, we leverage the notation of Markov approximation[19] to address our JUST problem. In Section 3.1, we first introduce a simplified case where the storage nodes are fixed, in order to inspire the application of the Markov approximation framework to our JUST problem in Section 3.2. We then report our algorithm in detail and give a short discussion in Section 3.3.

3.1 A simplified case with fixed storage nodes

Given a placement of the storage nodes $\{y_{j,k}\}_{j,k}$, our goal is reduced to designing an optimal routing scheme $\{x_{i,k}\}_{i,k}$ such that the resulting data delivery and storage overheads are minimized. In particular, we minimize $\sum_{i,k}(c_{i,k} + \beta\sum_j \tau_j y_{j,k})x_{i,k}$, subjected to $x_{i,k} \leqslant \sum_{j=1}^{|\mathcal{S}|} y_{j,k}$ for $\forall n_i \in \mathcal{N}$ and $u_k \in \mathcal{U}$. The optimal solution to the above problem is to let each sensor node n_i deliver its encoded data to the "nearest" distinct μ_i storage nodes. Specifically, let $\widetilde{\mathcal{S}} \subseteq \mathcal{S}$ be the subset of the sites with storage nodes deployed, such that $y_{i,k} = 1$ for $\forall s_k \in \widetilde{\mathcal{S}}$ and $y_{i,k} = 0$ otherwise. For $\forall s_k \in \widetilde{\mathcal{S}}$ and $\forall n_i \in \mathcal{N}$, we can calculate the total (weighted) cost as $c_{i,k} + \beta\sum_j \tau_j y_{j,k}$. Then, the destination storage nodes for n_i are the top ones with a minimum total cost. The above μ_i-Nearest storage Nodes (μ_i-NN) policy has a polynomial-time complexity. Although introducing a new freedom of storage node deployment has the potential of decreasing the objective cost function, designing the corresponding algorithm becomes much more difficult, such that we have to be contented with nearly optimal solutions, as we have demonstrated in

Section 2.3.

3.2 Markov approximation

Let $\mathcal{F}$ denote a set of all feasible storage node deployment strategies and c_f be the total cost induced by $f \in \mathcal{F}$. As shown above, for $\forall f \in \mathcal{F}$, the data routing strategy and the resulting cost c_f can be computed according to the μ_i-NN policy. Our optimization framework can accommodate various routing subroutines aiming at distinct objectives (e.g., in terms of energy and delay[20–22]). Nevertheless, designing specific routing protocols is beyond the scope of this study.

Our JUST problem (Function (1) and Constraints (2)–(7)) can be rewritten as

$$\min_{f \in \mathcal{F}}\{c_f\} \tag{8}$$

which can be approximated by a log-sum-exp function (which is parameterized by θ):

$$\phi_\theta(\{c_f\}_{f\in\mathcal{F}}) = -\frac{1}{\theta}\log\left(\sum_{f\in\mathcal{F}}\exp(-\theta c_f)\right) \tag{9}$$

where θ is a positive constant. In accordance with Ref. [23], the above approximation can be characterized as follows:

Theorem 1 $\min_{f\in\mathcal{F}}\{c_f\}$ can be approximated by $\phi_\theta(\{c_f\}_{f\in\mathcal{F}})$, owing to the following tight bound:

$$0 \leqslant \phi_\theta(\{c_f\}_{f\in\mathcal{F}}) - \min_{f\in\mathcal{F}}\{c_f\} \leqslant \frac{1}{\theta}\log|\mathcal{F}| \tag{10}$$

The above bound shows that, as ϕ approaches infinity, the approximation gap closes to zero. Specifically, we have

$$\min_{f\in\mathcal{F}}\{c_f\} = -\lim_{\theta\to\infty}\frac{1}{\theta}\log\left(\sum_{f\in\mathcal{F}}\exp(-\theta c_f)\right).$$

Therefore, we usually take a very large value for θ in the numerical computation.

We suppose that each $f \in \mathcal{F}$ is associated with weight $p_f \in [0, 1]$, such that $\sum_{f\in\mathcal{F}} p_f = 1$. The optimization problem (8) has the same optimal value as

$$\min \sum_{f\in\mathcal{F}} c_f p_f \tag{11}$$

$$\text{s.t.} \sum_{f\in\mathcal{F}} p_f = 1 \tag{12}$$

$$p_f \geqslant 0,\ \forall f \in \mathcal{F} \tag{13}$$

where $\{p_f\}_{f\in\mathcal{F}}$ are variables. Weight p_f can be explained as the time fraction of adopting the deployment strategy f in a long run. Clearly, when $p_f = 1$ for the storage node deployment strategy with the minimum cost and $p_f = 0$ for others, Problems (8) and (11) achieve the minimum cost.

As Problem (8) is approximated by $\phi_\theta(\{c_f\}_{f\in\mathcal{F}})$, we exploit the unique properties of this approximation to design our algorithm. In detail, the log-sum-exp function $\phi_\theta(\{c_f\}_{f\in\mathcal{F}})$ has a convex and closed conjugate function:

$$\phi_\theta^*(\{p_f\}_{f\in\mathcal{F}}) = -\frac{1}{\theta}\sum_{f\in\mathcal{F}} p_f \log p_f,$$

$p_f \geqslant 0$ for $\forall f \in \mathcal{F}$ and $\sum_{f\in\mathcal{F}} p_f = 1$. In other words, $\phi_\theta(\{c_f\}_{f\in\mathcal{F}})$ is the same as the optimal value of the following optimization problem (so-called "JUST-Approx"):

$$\min \sum_{f\in\mathcal{F}} c_f p_f + \frac{1}{\theta}\sum_{f\in\mathcal{F}} p_f \log p_f \tag{14}$$

$$\text{s.t.} \sum_{f\in\mathcal{F}} p_f = 1 \tag{15}$$

$$p_f \geqslant 0,\ \forall f \in \mathcal{F} \tag{16}$$

Based on Karush-Kuhn-Tucker (KKT) conditions, the optimal solution to our JUST-Approx problem can be represented as

$$p_f^* = \frac{\exp(-\theta c_f)}{\sum_{f'\in\mathcal{F}}\exp(-\theta c_{f'})},\ \forall f \in \mathcal{F} \tag{17}$$

According to Theorem 1, the above definition of p_f^* is a nearly optimal solution to our JUST problem with the gap bounded by $\theta^{-1}\log|\mathcal{F}|$. Specifically, we can nearly minimize the cost via time-sharing across different deployment strategies in a long run according to probability distribution $\{p_f^*\}_{f\in\mathcal{F}}$. However, the number of the feasible storage node deployment strategies, i.e., $|\mathcal{F}|$, is exponentially large; so we cannot directly calculate p_f^* according to Eq. (17).

3.3 Algorithm

To resolve the JUST-Approx problem (and thus to address the JUST problem with approximation optimality), we design a CTMC $\{F(t)\}_{t\geqslant 0}$ to drive the adaptive deployment of the storage nodes and the corresponding routing scheme, where t denotes time and $F(t) \in \mathcal{F}$ denotes the storage node deployment adopted at time t. In the CTMC, state transitions (i.e., storage node redeployments) continuously happen at any point in time and are characterized by transition rates. We assume that, any two of the states have a transition link if and only if the two states have only one storage node deployed at different sites. Note that this assumption does not compromise the reachability between the states and thus the irreducibility of the Markov chain.

The CTMC $\{F(t)\}_{t \geqslant 0}$ has a unique stationary distribution, as it has finite states and is of irreducibility. In the Markov approximation framework, we design a time-reversible CTMC $\{F(t)\}_{t \geqslant 0}$, such that the following detailed balance equation holds for the stationary distribution, i.e.,

$$p_f^* q_{f,f^+} = p_{f^+}^* q_{f^+,f} \tag{18}$$

where q_{f,f^+} and $q_{f^+,f}$ denote the transition rates from f to f^+ and the one from f^+ to f, respectively. In this paper, we define the transition rates q_{f,f^+} as

$$q_{f,f^+} = \delta \frac{\exp(-\theta c_{f^+})}{\exp(-\theta c_f) + \exp(-\theta c_{f^+})} \tag{19}$$

where δ is a constant. Recalling the definition of p_f^* (for $\forall f \in \mathcal{F}$) in Eq. (17), the above definition of q_{f,f^+} apparently satisfies the balance in Eq. (18).

The CTMC $\{F(t)\}_{t \geqslant 0}$ can be implemented based on the two propositions in Ref. [24], as shown in the following:

Proposition 1 For each state $f \in \mathcal{F}$, the sojourn time is a random variable obeying an exponential distribution parameterized by

$$v_f = \sum_{f' \in \mathcal{F}(f)} q_{f,f'} = \delta \sum_{f' \in \mathcal{F}(f)} \frac{\exp(-\theta c_{f'})}{\exp(-\theta c_f) + \exp(-\theta c_{f'})} \tag{20}$$

where $\mathcal{F}(f)$ denotes the set of the states adjacent to f.

Proposition 2 The transition probability from $f \in \mathcal{F}$ to $f^+ \in \mathcal{F}(f)$ is defined as

$$p_{f,f^+} = q_{f,f^+} / v_f \tag{21}$$

if considering the discrete-time counterpart of the CTMC.

According to the above propositions, the optimality of the CTMC-driven method can be explained from the following two folds: On one hand, we adopt the storage node deployments that have lower costs for a longer time; on the other hand, it is more likely to move to the lower-cost storage node deployment in each state transition.

Motivated by the two propositions, we can implement a CTMC to drive the adaptive deployment of the storage nodes by repeating the following steps: (1) Supposing f denotes the current storage node deployment, we set up a countdown timer initialized by an exponential random value $\Delta t \sim \text{Exponential}(v_f)$; (2) once the timer expires, we transit to a new storage node deployment $f^+ \in \mathcal{F}(f)$ according to the transition probabilities defined in Eq. (19). Although we can achieve the optimal distribution $\{p_f^*\}_{f \in \mathcal{F}}$ by repeating the above two steps in a long run, this approach entails some central infrastructures.

We propose an algorithm (see Algorithm 1) to implement the CTMC in a distributed manner. By default, we initially deploy the storage nodes in a randomized fashion (as shown in Line 1). The lifetime of the sensor network can be divided into a sequence of epochs, and a new epoch begins when some storage nodes are redeployed. In each epoch, we first compute a routing strategy according to the current storage node deployment f by the μ_i-NN policy (see Line 3). In every epoch, each storage node s_j generates an exponential random variable $\Delta t_j \sim \text{Exponential}(v_j)$, where

$$v_j = \sum_{f' \in \mathcal{F}_j(f)} q_{f,f'},$$

with $\mathcal{F}_j(f)$ representing the subset of feasible storage node deployments where only s_j is placed at different sites from the one in f (see Line 5). Each storage node s_j then sets up a countdown timer initialized by Δt_j (see Line 6). We assume $j^* = \arg\min_j \{\Delta t_j\}_{j=1,\ldots,|\mathcal{S}|}$, and storage node s_{j^*} is the one whose timer expires first. When the timer expires, s_{j^*} broadcasts a notification message, and the other storage nodes that received the message (before their timers expire) stop counting down. As shown in Line 8, we redeploy s_{j^*} according to the probability distribution:

$$p_{f,f^+} = q_{f,f^+} / v_{j^*}, \quad \forall f^+ \in \mathcal{F}_{j^*}(f) \tag{22}$$

Algorithm 1. Algorithm for jointly optimizing storage node deployment and routing (in each epoch)

```
1  Initialization: randomly initialize the storage node
   deployment f;
2  foreach epoch do
3      Calculate a routing strategy according to f;
4      foreach s_j ∈ S do
5          Generate an exponential random variable
           Δt_j ~ Exponential(v_j);
6          Set up a countdown timer initialized by Δt_j;
7          if no notification message is received then
8              when the timer expires, move to a new site
               according to Eq. (22), and broadcast a
               notification message;
9          end
10         else
11             Stop the timer;
12         end
13     end
14 end
```

The complexity of the above algorithm mainly stems from computing c_{f^+}, i.e., calculating the routing strategy and corresponding cost. The shortest paths from the sensor nodes to the sites and the induced cost can be computed by state-of-the-art polynomial-time algorithms (e.g., Dijkstra's method with a time complexity of $O(|\mathcal{N}|^2)$). As the sensor nodes and the sites where we deploy the storage nodes are fixed, the data delivery paths and their costs can be preloaded in the storage nodes, especially as the storage nodes usually have sufficient memory space. When a storage node is redeployed, it notifies the other ones of its new position by broadcasting $O(\log|\mathcal{U}|)$-length notification messages, such that all the other storage nodes can locally compute c_{f^+}. Moreover, the storage nodes can forward the notifications to the sensor nodes by piggybacking the received new storage node deployment in the acknowledgments to the data deliveries. For each sensor node, once the notification is received from the storage nodes, the sensor node can quickly compute the cost of delivering a data unit in the redeployed storage node and then update the routing strategy according to the μ_i-NN policy.

In the above implementation, each of the storage nodes only needs to set up a local timer. One question is, does such a distributed implementation realize CTMC $\{\mathcal{F}(t)\}_{t\geqslant 0}$ by respecting Propositions 1 and 2? To answer this question, we present the following theorem, which implies that relying on a set of local timers in a distributed manner instead of a centralized one does not compromise the optimality of $\{\mathcal{F}(t)\}_{t\geqslant 0}$ in solving our JUST-Approx problem.

Theorem 2 In Algorithm 1, the sojourn time for each state f is a random variable following an exponential distribution parameterized by rate v_f in Eq. (20), and the transition probability from f to f^+ ($f^+ \in \mathcal{F}(f)$) is $p_{f,f^+} = q_{f,f^+}/v_f$ (see Eq. (21)).

Proof For any storage node deployment $f \in \mathcal{F}$, the sojourn time $\Delta t(f)$ can be defined as

$$\Delta t(f) = \min\{\Delta t_j\}_{j=1,\ldots,|\mathcal{S}|},$$

as shown in Algorithm 1. As $\Delta t_j \sim \text{Exponential}(v_j)$ for $\forall j$, we have $\Delta t(f) \sim \text{Exponential}(\sum_j v_j)$ for $\forall f \in \mathcal{F}$. In another word, $\Delta t(f)$ is an exponentially distributed random variable with the rate defined by $\sum_j v_j = \sum_j \sum_{f' \in \mathcal{F}_j(f)} q_{f,f'}$. Also, since $\bigcup_j \mathcal{F}_j(f) = \mathcal{F}/\{f\}$ and $\bigcap_j \mathcal{F}_j(f) = \varnothing$, we have $\sum_j v_j = \sum_{f' \neq f} q_{f,f'} = v_f$. Therefore, for $\forall f \in \mathcal{F}$,

$$\Delta t(f) \sim \text{Exponential}(v_f).$$

For $\forall j^* \in \{1,\ldots,|\mathcal{S}|\}$, the probability that s_{j^*} is the redeployed storage node with $\Delta t(f) = \Delta t_{j^*}$ can be calculated as $v_{j^*}/\sum_j v_j$. Let $\mathcal{F}_{j^*}(f)$ denote the set of the storage node deployments with s_{j^*} deployed at different sites from the one in f. Then, for $\forall f^+ \in \mathcal{F}_{j^*}(f)$, the probability that we have the transition $f \to f^+$ (given that s_{j^*} has to be redeployed) can be defined by $q_{f,f^+}/v_{j^*}$ (see Eq. (22)). Therefore, $p_{f,f^+} = \dfrac{v_{j^*}}{\sum_j v_j} \cdot \dfrac{q_{f,f^+}}{v_{j^*}} = \dfrac{q_{f,f^+}}{v_f}$ holds for $\forall f^+ \in \mathcal{F}(f)$. ■

4 Simulation

In this section, we evaluate the efficacy of our algorithm through extensive simulations. We uniformly deploy 200 sensor nodes in a 100 m × 100 m targeted area. For each sensor node, we choose its data storage reliability requirement uniformly from the range of Refs. [1, 4]. For simplicity, we assume that sensor nodes have identical transmission power and receiving power, such that the data delivery cost associated with each edge can be normalized to 1. We suppose there are 10 storage nodes, and their storage costs are uniformly chosen from Refs. [1,4]. We assume that $\beta = 1$, such that the data delivery and storage costs are of equal importance for our optimization problem.

Figure 1 illustrates the total cost of our algorithm (i.e., the weighted sum of the data delivery and data storage costs) under different numbers of sites where we deploy the storage nodes. We also vary the value of θ to investigate its impact on the total cost of our algorithm. It is shown that, we can decrease total cost by increasing the number of the sites, as we search in a much larger space of the feasible solutions. Figure 1 also shows that taking larger value for θ implies a lower total cost, which is consistent with the fact that we are able to approach the optimum by increasing the value of θ (see Theorem 1).

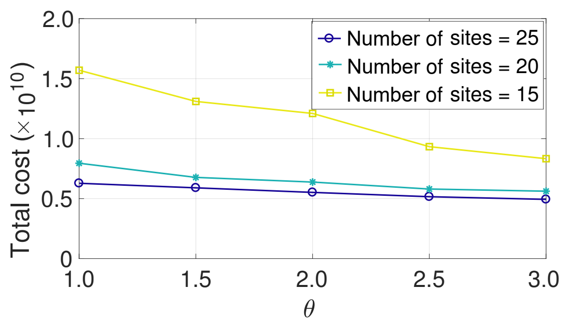


Fig. 1 Total cost of our algorithm under different values of θ.

Moreover, we compare our algorithm with the optimal solution due to the computational intractability of large system settings. We suppose that the storage nodes have identical storage costs to deliberately “shrink” the domain of the feasible solutions. The optimality gaps (defined in Eq. (10)) are shown in Fig. 2. As shown in Fig. 2, increasing θ leads to a reduced optimality gap, which is again consistent with Theorem 1. Another remarkable observation is that, although establishing more sites to place the storage nodes can decrease the total cost, the resulting optimality gap is larger than that in the case where less sites are utilized. This observation can be explained by revisiting Eq. (10) in Theorem 1: The increased number of sites implies a larger feasible region $\mathcal{F}$, and the optimality gap is proportional to the size of the feasible region, i.e., $|\mathcal{F}|$.

5 Related Work

A vast body of proposals have studied the data collection problem by one or multiple sink nodes[25–28]. Although we borrow the idea of empowering the sinks with mobility, these data collection algorithms cannot be directly applied to our problem as neither storage reliability nor the data storage cost is considered. As shown in Section 2.3, integrating both of them into an optimization framework is highly nontrivial.

Some of the exiting proposals have a goal of disseminating the sensed data among the sensor nodes for storage. For example, as demonstrated in Ref. [29], in a tree-structure sensor network, the data delivered from the downstream sensor nodes can be stored in a subset of non-root ones. In Refs. [30, 31], compress sensing and probabilistic broadcasting were integrated to decrease data traffic and energy consumption. In Refs. [32,33], quorum-based data storage strategies were proposed, guaranteeing the storage reliability through data replication.

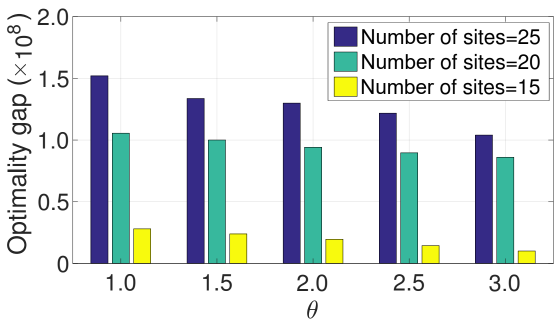


Fig. 2 Optimality gaps of our algorithm under different values of θ.

To deal with the ever-increasing demand on reliable data storage raised by pervasive deployments of sensor networks, robust storage nodes are usually employed in sensor networks. In Ref. [34], a random network coding technique was utilized to encode original data. The encoded data were then randomly delivered to predeployed neighboring storage nodes. In Ref. [17], sensor nodes were divided into a number of clusters with respect to predeployed storage nodes. A mobile sink is then employed to visit the storage nodes to collect the data. In Ref. [35], the problem of energy-efficient routing in the context of data storage was investigated. Specifically, an algorithm was designed to associate the sensor nodes with the storage nodes for energy-efficient data delivery. In Ref. [35], the study relied on the random linear network coding technique to ensure the desired fault-tolerance. In Ref. [36], storage nodes were utilized as “relay” nodes such that the data collected at sensors were first delivered to storage nodes for compression, and the compressed data were then forwarded to a sink node. The resulting energy consumption can be minimized by optimally deploying the storage nodes. However, in Ref. [36], storage reliability was not taken into account.

6 Conclusion

To minimize data delivery and storage costs while guaranteeing storage reliability, we have proposed a CTMC-based algorithm in this paper by leveraging the Markov approximation framework. In particular, driven by an elaborately designed CTMC, the deployment of the storage nodes is adaptively scheduled, and the routing scheme is accordingly updated. We evaluate our algorithm through a theoretical analysis and numerical simulations.

We will take dynamic sensor networks into account where sensor nodes may join in or depart from the networks. In this case, we aim to design an on-line algorithm where the storage node deployment and data routing can be adaptively leveraged to the churn of the sensor nodes.

Acknowledgment

This work was partially supported by the Shandong Provincial Natural Science Foundation (No. ZR2017QF005), the National Natural Science Foundation of China (Nos. 61702304, 61971269, 61832012, 61602195, 61672321, 61771289, and 61602269), and the China Postdoctoral Science Foundation (No. 2017M622136).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks: A survey, *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] C. Y. Ai, F. H. Li, and K. J. Zhang, Detecting isolate safe areas in wireless sensor monitoring systems, *Tsinghua Science and Technology*, vol. 22, no. 4, pp. 427–436, 2017.

[3] R. Arcucci, C. Pain, and Y. K. Guo, Effective variational data assimilation in air-pollution prediction, *Big Data Mining and Analytics*, vol. 1, no. 4, pp. 297–307, 2018.

[4] F. Li, J. Luo, S. Q. Xin, and Y. He, Autonomous deployment of wireless sensor networks for optimal coverage with directional sensing model, *Computer Networks*, vol. 108, pp. 120–132, 2016.

[5] F. Li, J. Luo, W. P. Wang, and Y. He, Autonomous deployment for load balancing k-surface coverage in sensor networks, *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 279–293, 2015.

[6] L. Xiang, J. Luo, C. W. Deng, A. V. Vasilakos, and W. S. Lin, DECA: Recovering fields of physical quantities from incomplete sensory data, in *Proc. 9^{th} Annu. IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks*, Seoul, South Korea, 2012, pp. 182–190.

[7] S. Y. Cheng, Z. P. Cai, J. Z. Li, and H. Gao, Extracting kernel dataset from big sensory data in wireless sensor networks, *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 813–827, 2017.

[8] J. Li, S. Y. Cheng, Z. P. Cai, J. G. Yu, C. K. Wang, and D. M. Li, Approximate holistic aggregation in wireless sensor networks, *ACM Transactions on Sensor Networks*, vol. 13, no. 2, pp. 1–24, 2017.

[9] Z. B. He, Z. P. Cai, S. Y. Cheng, and X. M. Wang, Approximate aggregation for tracking quantiles and range countings in wireless sensor networks, *Theoretical Computer Science*, vol. 607, pp. 381–390, 2015.

[10] S. Y. Cheng, Z. P. Cai, and J. Z. Li, Curve query processing in wireless sensor networks, *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5198–5209, 2015.

[11] Q. Y. Meng, K. Wang, X. M. He, and M. Y. Guo, QoE-driven big data management in pervasive edge computing environment, *Big Data Mining and Analytics*, vol. 1, no. 3, pp. 222–233, 2018.

[12] M. Albano and S. Chessa, Distributed erasure coding in data centric storage for wireless sensor networks, in *Proc. 2009 IEEE Symp. Computers and Communications*, Sousse, Tunisia, 2009, pp. 22–27.

[13] A. G. Dimakis, V. Prabhakaran, and K. Ramchandran, Decentralized erasure codes for distributed networked storage, *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2809–2816, 2006.

[14] F. Li, Y. B. Yang, Z. C. Chi, L. Y. Zhao, Y. W. Yang, and J. Luo, Trinity: Enabling self-sustaining WSNs indoors with energy-free sensing and networking, *ACM Transactions on Embedded Computing Systems*, vol. 17, no. 2, pp. 1–27, 2018.

[15] S. Holzer and R. Wattenhofer, Optimal distributed all pairs shortest paths and applications, in *Proc. 2012 ACM Symp. Principles of Distributed Computing*, Madeira, Portugal, 2012, pp. 355–364.

[16] M. Ghaffari and J. Li, Improved distributed algorithms for exact shortest paths, in *Proc. 50^{th} Annu. ACM SIGACT Symp. Theory of Computing*, Los Angeles, CA, USA, 2018, pp. 431–444.

[17] G. Maia, D. L. Guidoni, A. C. Viana, A. L. L. Aquino, R. A. F. Mini, and A. A. F. Loureiro, A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks, *Ad Hoc Networks*, vol. 11, no. 5, pp. 1588–1602, 2013.

[18] Q. Liu, K. Zhang, X. D. Liu, and N. Linge, Grid routing: An energy-efficient routing protocol for WSNs with single mobile sink, *International Journal of Sensor Networks*, vol. 25, no. 2, pp. 93–103, 2017.

[19] M. H. Chen, S. C. Liew, Z. Y. Shao, and C. H. Kai, Markov approximation for combinatorial network optimization, *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6301–6327, 2013.

[20] X. Zheng, Z. P. Cai, J. Z. Li, and H. Gao, A study on application-aware scheduling in wireless networks, *IEEE Transactions on Mobile Computing*, vol. 16, no. 7, pp. 1787–1801, 2017.

[21] Z. P. Cai, G. H. Lin, and G. L. Xue, Improved approximation algorithms for the capacitated multicast routing problem, in *Proc. of the 11th COCOON*, Berlin, Germany, 2005, pp. 136–145.

[22] Z. P. Cai, Z. Z. Chen, and G. H. Lin, A 3.4713-approximation algorithm for the capacitated multicast tree routing problem, *Theoretical Computer Science*, vol. 410, no. 52, pp. 5415–5424, 2009.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[24] R. G. Gallager, *Stochastic Processes: Theory for Applications*. Cambridge, UK: Cambridge University Press, 2014.

[25] R. L. Deng, S. B. He, and J. M. Chen, An online algorithm for data collection by multiple sinks in wireless-sensor networks, *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 93–104, 2018.

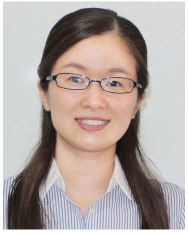
[26] J. Wang, J. Y. Cao, R. S. Sherratt, and J. H. Park, An improved ant colony optimization-based approach with mobile sink for wireless sensor networks, *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6633–6645, 2018.

[27] G. R. Li, S. C. Peng, C. Wang, J. W. Niu, and Y. Yuan, An energy-efficient data collection scheme using denoising autoencoder in wireless sensor networks, *Tsinghua Science and Technology*, vol. 24, no. 1, pp. 86–96, 2019.

[28] F. Li, J. Luo, G. T. Shi, and Y. He, ART: Adaptive frequency-temporal co-existing of ZigBee and WiFi, *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 662–674, 2017.

[29] B. Sheng, Q. Li, and W. Z. Mao, Data storage placement in sensor networks, in *Proc. 7^{th} ACM Int. Symp. Mobile Ad Hoc Networking and Computing*, Florence, Italy, 2006, pp. 344–355.

[30] X. J. Yang, X. F. Tao, E. Dutkiewicz, X. J. Huang, Y. J. Guo, and Q. M. Cui, Energy-efficient distributed data storage for wireless sensor networks based on compressed sensing and network coding, *IEEE Transactions on Wireless Communications*, vol. 12, no. 10, pp. 5087–5099, 2013.
[31] A. Talari and N. Rahnavard, CStorage: Decentralized compressive data storage in wireless sensor networks, *Ad Hoc Networks*, vol. 37, pp. 475–485, 2016.
[32] J. Luo, F. Li, and Y. He, 3DQS: Distributed data access in 3D wireless sensor networks, in *Proc. 2011 IEEE Int. Conf. Communications*, Kyoto, Japan, 2011, pp. 1–5.
[33] C. Zhang, J. Luo, L. Xiang, F. Li, J. C. Lin, and Y. He, Harmonic quorum systems: Data management in 2D/3D wireless sensor networks with holes, in *Proc. 2012 9th Annu. IEEE Communications Society Conf. Sensor, Mesh and Ad Hoc Communications and Networks*, Seoul, South Korea, 2012, pp. 1–9.
[34] R. F. Zeng, Y. X. Jiang, C. Lin, Y. F. Fan, and X. M. Shen, A distributed fault/intrusion-tolerant sensor data storage scheme based on network coding and homomorphic fingerprinting, *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1819–1830, 2012.
[35] J. Tian, T. Yan, and G. L. Wang, A network coding based energy efficient data backup in survivability-heterogeneous sensor networks, *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 1992–2006, 2015.
[36] G. D'Angelo, D. Diodati, A. Navarra, and C. M. Pinotti, The minimum k-storage problem: Complexity, approximation, and experimental analysis, *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1797–1811, 2016.



Huan Yang received the PhD degree in computer science from Nanyang Technological University, Singapore in 2015, the MS degree in computer science from Shandong University, China in 2010, and the BS degree from the Heilongjiang Institute of Technology, China in 2007. Currently, she is an associate professor in the College of Computer Science and Technology, Qingdao University, China. Her research interests include applied optimization, stochastic modeling, and machine learning.

Feng Li received the BS, MS, and PhD degrees in computer science from Shandong Normal University, China in 2007, Shandong University, China in 2010, and Nanyang Technological University, Singapore in 2015, respectively. From 2014 to 2015, he worked as a research fellow in National University of Singapore, Singapore. He jointed the School of Computer Science and Technology, Shandong University, China in 2015, where he is currently an associate professor. His research interests include distributed algorithms and systems, wireless networking, mobile computing, and internet of things.

Dongxiao Yu received the BSc degree from Shandong University in 2006 and the PhD degree from the University of Hong Kong in 2014. He became an associate professor in the School of Computer Science and Technology, Huazhong University of Science and Technology in 2016. He is currently a professor in the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing, and graph algorithms.

Yifei Zou received the BE degree from Wuhan University in 2016. He is currently a PhD student in Department of Computer Science, the University of Hong Kong. His research interests include wireless networks, ad hoc networks, and distributed computing.

Jiguo Yu received the PhD degree from Shandong University in 2004. He became a full professor in the School of Computer Science, Qufu Normal University, Shandong, China in 2007. Currently he is a full professor in Qilu University of Technology (Shandong Academy of Sciences) and Shandong Computer Science Center (National Supercomputer Center in Jinan). His main research interests include privacy-aware computing, wireless networking, distributed algorithms, peer-to-peer computing, and graph theory. Particularly, he is interested in designing and analyzing algorithms for many computationally hard problems in networks. He is a senior member of IEEE, a member of ACM, and a senior member of the China Computer Federation (CCF).