

RESEARCH ARTICLE

Network throughput maximization in unreliable wireless sensor networks with minimal remote data transfer cost

Xu Xu¹, Weifa Liang^{1*}, Xiaohua Jia² and Wenzheng Xu^{1,3}¹ Research School of Computer Science, Australian National University, Canberra ACT 0200, Australia² Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong³ School of Information Science and Technology, Sun Yat-Sen University, Guangzhou, China

ABSTRACT

In this paper, we consider large-scale remote environmental monitoring (data gathering) through deploying an unreliable wireless sensor network in a remote region. The data monitoring center is geographically located far away from the region of the sensor network, which consists of sensors and gateways. Sensors are responsible for sensing and relaying data, and gateways are equipped with 3G/4G radios and can store the collected data from sensors temporarily and transmit the data to the remote data center through a third-party communication service. A service cost of using this service will be charged, which depends on not only the number of gateways employed but also the volume of data transmitted from each gateway within a given monitoring period. For this large-scale, remote, and unreliable data gathering, we first formulate a problem of maximizing network throughput with minimal service cost with an objective to maximize the amount of data collected by all gateways while minimizing the service cost. We then show that the problem is NP-complete and propose novel approximation algorithms. The key ingredients of the proposed algorithms include building load-balanced routing trees rooted at gateways and dynamically adjusting data load among the gateways. Finally, we conduct experiments by simulations to evaluate the performance of the proposed algorithms. Experimental results demonstrate that the proposed algorithms are very promising, and the obtained solutions are fractional of the optimum in terms of network throughput and the data service cost. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS

unreliable sensor networks; monitoring quality maximization; load-balanced forest; remote data collection; remote data transfer cost; data plan pricing; combinatorial optimization problem

*Correspondence

Weifa Liang, Research School of Computer Science, Australian National University, Canberra ACT 0200, Australia.

E-mail: wliang@cs.anu.edu.au

1. INTRODUCTION

Wireless sensor networks (WSNs) have been widely deployed for habitat monitoring [1], structural health monitoring [2], and environmental sensing [3]. Most existing studies focused on prolonging network lifetime as sensors are usually powered by energy-limited batteries [4]. With the advance of renewable technology, more and more sensors now are powered by ambient energy around their environments such as solar energy and wind energy [5–11]. Sensors in renewable sensor networks can be recharged repeatedly; thus, theoretically, this type of network not only operates environmentally friendly but also runs perpetually, which exhibits great potentials that renewable

sensor networks can be employed for unattended, remote monitoring purposes.

In this paper, we consider a renewable sensor network deployed for monitoring an unattended, remote region of interest, where the monitoring center is geographically far away from the monitoring region [12,13]. One well-known example of such renewable WSNs is the GreenOrbs, which aims to provide all-year ecological surveillance for a forest for supporting forestry observation and research, fire risk evaluation, and succor in the wild [12]. We here assume that the sensor network is a heterogeneous network consisting of common sensors and gateways, where the common sensors are responsible for sensing and relaying sensing data to other common sensors or gateways

through multi-hop relays, while the gateways equipped with larger storages and high-bandwidth radios store the collected sensing data and transmit them to the monitoring center via a third-party communication network. Each gateway adopts a data plan with a fixed cost for a data quota and extra charges applied to the exceeding amount over the quota for a given period. We further assume that wireless communication within the sensor network is unreliable [14–17], which indicates that data loss from a source node to a gateway node is unavoidable. We refer to this sensor network as an unreliable sensor network.

The accumulative volume of sensing data received by all gateways within a given monitoring period is referred to as the network throughput. Clearly, the larger the network throughput, the more the sensing data collected from the sensors, and the better monitoring quality the sensor network will deliver. To maximize the network throughput, sensing data from sensors should be transmitted to the gateways through most reliable routing paths. Given a certain throughput, to minimize the service cost incurred by transferring sensed data via gateways, the amount of data collected by each gateway must be well balanced so that the sum amount of exceeding quota at each gateway is minimized. The challenge to resolve this problem is to maximize the total amount of data collected by all gateways (the network throughput) while keeping the service cost of transferring the collected data to the remote monitoring center minimized. We refer to this optimization problem as the maximizing network throughput with minimal service cost problem.

1.1. Related work

The deployment of WSNs for environmental monitoring has been extensively studied in the past decade [1]. One basic functionality of sensor networks is to collect sensing data from sensors and relay the sensing data to the base station (the data monitoring center) through flows or routing trees, and most existing studies focused on the design of energy-efficient routing protocols to maximize the network lifetime through balancing the energy consumption among sensor nodes, assuming that the base station and the sensors are in the same region and the wireless communication within the network is reliable; thereby, the amount of data received at the base station is equal to the amount of data sent from all source nodes [16,18–31]. In contrast, we assume that the data monitoring center is geographically located far away from the region of the deployed sensors [12,13]. The sensing data need to be forwarded to the center by hiring a third-party service that does incur service costs. We also assume that communication in the sensor network is not reliable, which means that data loss is unavoidable. Thus, the amount of data received at gateways is usually less than the amount of data sent from data sources.

Several well-known optimization problems such as the capacitated minimum spanning tree problem (CMST) [32] and the capacitated minimum Steiner tree problem

(CMSStT) [33] seem to be closely related to the maximizing network throughput with minimal service cost problem. CMST and CMSStT are defined as follows. Given an edge-weighted graph, a root node, a set of source nodes, and a given demand $Q > 2$, the problem is to construct a minimum-cost spanning or Steiner tree rooted at the root node and spanning all source nodes, subject to the sum of demands in the subtree rooted at each child of the root node being no more than Q . CMST and CMSStT have been extensively studied in computer science and operation research over the past 40 years, and there are efficient algorithms for them [33,34]. Liang *et al.* [35] recently studied another capacitated minimum spanning forest problem in WSNs that is to find a forest of routing trees rooted at gateways such that the data load at each gateway is no greater than its capacity. Intuitively, the problem in this paper can be reduced to one of the mentioned problems as follows. The data quota at each gateway is treated as the gateway capacity, and a routing tree rooted at the gateway and spanning some sensor nodes is then built, subject to the gateway capacity constraint. Unfortunately, the problem studied here essentially differs from these mentioned problems in the following aspects: (i) the demand from a source node in the mentioned studies can be routed to a gateway along any routing path without any data loss, whereas data loss in an unreliable sensor network is unavoidable and (ii) existing capacitated tree or forest problems do not allow the gateway capacity to be exceeded, while the data loads of some gateways in our problem are allowed to exceed their data quotas (capacities). Although (data) load balancing among gateways has been studied in the past [35–38], the load balancing studied here is different from the previous ones in that the loads of tree roots are allowed to exceed their capacities, where the load of each tree root is determined not only by the number of its descendants but also by the end-to-end reliability of each routing path between a node and the root.

Another related work [16,17] is to deal with minimizing the service cost while keeping the network throughput above a specified threshold, for which a heuristic was proposed to determine the optimal number of gateways and find routing paths from data sources to the gateways. Unlike the work in [16,17], we here aim to maximize the network throughput while keeping the service cost of achieving the network throughput minimized. We focus on the remote data transfer that incurs the service cost by employing a third-party communication service. We assume that wireless communications in WSNs are unreliable, data loss during its transfer is unavoidable, and the amount of data received at gateways is usually less than the amount of data sent from data sources. Therefore, in this paper, we deal with two coupled challenging problems in an unreliable sensor network. (i) To maximize the amount of sensing data transmitted to the remote monitoring center means that the most reliable routing paths for data routing are preferred. (ii) To minimize the service cost incurred by transferring sensing data to the remote data monitoring center using a third-party telecommunication service

(e.g., 3G), the amount of data transmitted from each gateway must be well balanced; otherwise, heavy-loaded gateways may pay high penalties. To the best of our knowledge, this is the first study of the problem of concern.

1.2. Contributions

The main contributions of this paper are as follows:

- A novel optimization problem of maximizing network throughput with minimal service cost in an unreliable WSN for large-scale remote monitoring is formulated and shown to be NP-complete.
- Novel heuristic and approximation algorithms are proposed, with key ingredients of building load-balanced routing trees and dynamically adjusting the data load among gateways.
- Extensive experiments by simulations are conducted to evaluate the performance of the proposed algorithm and study the impact of different parameters on the algorithm performance. Experimental results demonstrate that the proposed algorithms are very promising, and the obtained solutions are fractional of the optimum.

The remainder of this paper is organized as follows. We introduce the system model, notions, and the problem definition in Section 2. We show that the problem is NP-hard in Section 3. We propose approximation algorithms for the problem with and without uniform link reliability assumptions in Sections 4 and 5, respectively. We conduct extensive experiments by simulations to evaluate the performance of the proposed algorithms in Section 6, and we conclude in Section 7.

2. PRELIMINARIES

In this section, we introduce the system model and notions and define the problem.

2.1. System model

We consider a heterogeneous, unreliable WSN for large-scale remote monitoring, where the monitoring center is geographically far away from the monitoring region. We model the sensor network as an undirected graph $G = (V \cup GW, E)$, where V is the set of sensor nodes, GW is the set of gateways, and E is the set of unreliable links, $n = |V|$ and $K = |GW|$. The data generation rate of sensors is identical, denoted by r . The transmission range of each sensor is fixed and denoted by R . There is a link between two sensor nodes or a sensor node and a gateway node if they are within the transmission range of each other. We assume that both the sensors and the gateways are stationary and their locations are known a priori, where gateways are deployed in strategic locations in the monitoring region. We further assume that gateways have

adequate energy supplies (e.g., solar powered) and large buffer sizes. As gateways are usually costly, the number of gateways K deployed is limited compared with the number of sensors. We thus assume that gateways are not deployed within the transmission ranges of each other. We consider a long-term periodic environmental monitoring application scenario, in which sensors usually have low data generation rates and the generated sensing data are transmitted to the gateways through multi-hop relays. Thus, data burst and bandwidth capacity constraint are not the major issues in such application scenarios. For example, the 802.15.4 link bandwidth of a typical commercial sensor can be up to 250 Kbps [39], while the sampling cycle of each sensor for monitoring ambient conditions, such as humidity and temperature, is more than 30 s [40]. The sensing data from each sensor will be forwarded to one of the gateways, and then the gateways forwarded the collected data to the remote monitoring center via a third-party network. The use of third-party communication service will incur a certain amount of cost, depending on which data plan adopted for the given monitoring period.

2.2. Network throughput and service cost

The accumulative volume of data received by the monitoring center within a specified monitoring period of τ is defined as the network throughput. We assume that the third-party service for remote data transmission is reliable; thus, the network throughput depends on the amount of data collected by the K gateways within a period of τ . Let T_i be the routing tree rooted at a gateway $g_i \in GW$ and $V(T_i)$ be the set of sensors in T_i , $1 \leq i \leq K$. All sensors in the network are included in the K trees, that is, $\bigcup_{1 \leq i \leq K} V(T_i) = V$.

Let $e_1, e_2, \dots, e_l$ be the link sequence in the routing path $P(v, g_i)$ in tree T_i from a sensor node $v \in V(T_i)$ to gateway g_i . Denote by p_{v, g_i} the end-to-end reliability of path $P(v, g_i)$, then $p_{v, g_i} = \prod_{i=1}^l p_{e_i}$, where p_{e_i} is the reliability of link $e_i \in E$ with $0 < p_{e_i} \leq 1$.

Let $L(g_i)$ be the amount of data received by gateway g_i through tree T_i within a period of τ , which is also referred to as the load of gateway g_i , then $L(g_i) = \sum_{v \in V(T_i)} (p_{v, g_i} \cdot \tau \cdot r)$. The network throughput thus is

$$D^{(\tau)} = \sum_{i=1}^K L(g_i) = \tau \cdot r \left(\sum_{i=1}^K \sum_{v \in V(T_i)} p_{v, g_i} \right) \quad (1)$$

The cost of transferring the collected data from the gateways to the remote monitoring center for a period of τ is referred to as the remote data transfer cost, or the service cost for short. In this paper, we adopt a popular data plan at each gateway provided by telecommunication companies nowadays, where a fixed cost C_f for a data quota Q is charged within the period of τ , and the extra charge will be applied with a penalty rate c_p for every megabyte of the exceeding data over the data quota during this period. Notice that the penalty rate is usually much higher than the

fixed cost rate c_f for data quota Q , that is, $c_p > c_f = \frac{C_f}{Q}$. Let C_p be the penalty cost of exceeding the quota by an amount of Q , that is, $C_p = c_p \cdot Q$, then $C_p > C_f$. The service cost charged for each gateway depends on not only the volume of its collected data but also its chosen data plan. Denote by C the service cost of the network for the period of τ , then

$$C = K \cdot C_f + \sum_{i=1}^K \max\{0, (L(g_i) - Q) \cdot c_p\} \quad (2)$$

On the right-hand side of Equation (2), the first term is the sum of fixed costs of K gateways, while the second term is the total amount of penalties incurred by the K gateways. The penalty to a gateway g_i is either 0 if the volume of its collected data does not exceed the data quota Q ; or $(L(g_i) - Q) \cdot c_p$, otherwise, for all i with $1 \leq i \leq K$.

2.3. Problem definition

Given a defined unreliable sensor network $G = (V \cup GW, E)$, a specified monitoring period of τ , and a data plan that consists of a fixed cost C_f for a data quota Q and a penalty rate c_p for per megabyte of the exceeding data, the maximizing network throughput with minimal service cost problem in G is to find a forest of routing trees rooted at K gateways such that the accumulative volume of data received by the K gateways is maximized while keeping the service cost of transferring the collected data to the remote monitoring center minimized, where $K = |GW|$. In other words, the problem is to find a forest $\mathcal{F} = \{T_i \mid T_i \text{ that is a routing tree rooted at gateway } g_i \in GW, 1 \leq i \leq K\}$ such that $D^{(\tau)} = \tau \cdot r \left(\sum_{i=1}^K \sum_{v \in V(T_i)} p_{v,g_i} \right)$ is maximized, while keeping the service cost $C = K \cdot C_f + \sum_{i=1}^K \max\{0, \sum_{v \in V(T_i)} (p_{v,g_i} \cdot \tau \cdot r) - Q\} \cdot c_p\}$ is minimized.

2.4. Approximation ratio

Denote by *Approx* and *OPT* the costs of the approximate solution delivered by an approximation algorithm $\mathcal{A}$ and the optimal solution to an optimization problem, respectively. The approximation ratio of algorithm $\mathcal{A}$ is ξ if $\frac{\text{Approx}}{\text{OPT}} \leq \xi$ when the problem is a minimization problem or $\frac{\text{Approx}}{\text{OPT}} \geq \xi$ when the problem is a maximization problem.

3. NP-COMPLETENESS

In this section, we show that the problem is NP-hard by the following theorem.

Theorem 1. *The decision version of the maximizing network throughput with minimal service cost problem in an unreliable sensor network $G = (V \cup GW, E)$ is NP-complete.*

Proof. We show that a special case of the problem is NP-complete through a reduction from an NP-complete problem, namely the subset sum problem [41]. Given a set of positive integers $S = \{a_1, a_2, \dots, a_n\}$, the subset sum problem is to partition the set into two disjoint subsets S_1 and S_2 such that $\sum_{a_i \in S_1} a_i = \sum_{a_j \in S_2} a_j$. The decision version of an instance of the subset sum problem is to determine whether there is a set partition S' and $S'' = S - S'$ such that $\sum_{a_i \in S'} a_i = \sum_{a_j \in S''} a_j = A$, where $A = \sum_{a_i \in S} a_i / 2$.

Given an instance of the subset sum problem, we now construct an instance of a special maximizing network throughput with minimal service cost problem in a sensor network $G(V \cup \{g_1, g_2\}, E)$ with two gateways g_1 and g_2 as follows. V is the set of sensors, and each sensor $v_i \in V$ corresponds to an element in S , and gateways g_1 and g_2 correspond to sets S' and S'' , respectively. There is an edge in E between each sensor node and either gateway or between two sensors if they are within the transmission range of each other. Assume that the link reliability of a link between a sensor v_i and either gateway is $p_i = a_i / \tau$, where $\tau = \max\{a_i \mid 1 \leq i \leq n\} + 1$ is the duration of a monitoring period, while the link reliability between any two sensors is 1. We further assume that the data generation rate is $r = 1$. The amount of data from a sensor $v_i \in V$ to either gateway during this period is $p_i \cdot \tau \cdot r$. Clearly, the maximum network throughput of this network is $\sum_{v_i \in V_1} (p_i \cdot \tau \cdot r) + \sum_{v_j \in V_2} (p_j \cdot \tau \cdot r) = \sum_{a_i \in S} a_i = 2A$. Let the data quota Q be A , that is, $Q = A$, and the fixed cost is C_f . The decision version of this special case of the problem is to determine whether there are two routing trees rooted at the gateways such that the network throughput is maximized while the service cost is minimized, that is, $2C_f$. Note that only when the volume of data received at each root of the two trees is A , the service cost is $2C_f$; otherwise, the cost will be larger than $2C_f$. Obviously, if there is a solution to the problem, there is a corresponding solution to the subset sum problem. Because the subset sum problem is NP-complete and this reduction is in polynomial time, the problem of concern thus is NP-hard. Meanwhile, it is easy to verify whether a given solution has the maximum network throughput $2A$ with service cost $2C_f$ in polynomial time; thus, the problem is in NP-class. As one of its special cases is NP-complete, the problem is NP-complete. $\square$

4. ALGORITHM FOR UNIFORM LINK RELIABILITY

In this section, we deal with a special case of the maximizing network throughput with minimal service cost problem, where each link has an identical reliability p with $0 < p \leq 1$. For this special case, we devise a novel approximation algorithm that achieves the maximum network throughput while keeping the service cost of achieving the throughput bounded. This algorithm will be used as a key component to solve the problem without the uniform link reliability assumption in the next section.

Given the network $G(V \cup GW, E)$, we first construct another network $G' = (V \cup GW \cup \{s_0\}, E')$ as follows. A virtual sink s_0 and edges between s_0 and each gateway node $g_i \in GW$ are included in G' , that is, $E' = E \cup \{(g_i, s_0) \mid g_i \in GW\}$. Let T^{BFS} be a Breadth-first search (BFS) tree in G' rooted at s_0 and h be the depth of T^{BFS} , where the virtual sink s_0 is in layer 0 and all gateways are in layer 1 of T^{BFS} . Let V_l be the set of nodes of T^{BFS} in layer l for all l with $0 \leq l \leq h$. Then, the nodes in $V \cup GW$ are partitioned into h disjoint subsets $V_1, V_2, \dots, V_h$ such that $V_1 = GW$, $\cup_{l=2}^h V_l = V$ and $V_i \cap V_j = \emptyset$ if $i \neq j$ and $j, 1 \leq i, j \leq h$. Note that V_l is the set of nodes whose distance (in terms of the number of hops) to their nearest gateways is at least $l-1$ hops, with $2 \leq l \leq h$.

Having the sensor nodes partitioned based on the BFS tree, we then devise a heuristic for the problem that delivers the maximum network throughput by the following lemma.

Lemma 1. *Given an unreliable sensor network $G(V \cup GW, E)$ with identical link reliability p , $0 < p < 1$, let $h(v) - 1$ be the minimum number of hops from each node $v \in V$ to its nearest gateway in G , then (i) the end-to-end reliability of the most reliable path in G from v to a gateway is $p^{h(v)-1}$ and (ii) the maximum network throughput is $D_{max} = \sum_{l=2}^h \sum_{v \in V_l} p^{l-1} \cdot (\tau \cdot r) = \sum_{l=2}^h (|V_l| \cdot p^{l-1} \cdot \tau \cdot r)$.*

Proof.

- (1) Because the reliability of each link is p , let P_v be the most reliable path from a sensor $v \in V$ to a gateway and $|P_v| = l - 1$, then, the reliability of path P_v is p^{l-1} . Maximizing the end-to-end reliability of path P_v , p^{l-1} is equivalent to minimizing the value of l , while the length of a shortest path from v to any gateway is no less than $h(v) - 1$, following the definition of $h(v)$. Thus, the end-to-end reliability of the most reliable path in G from v to a gateway is $p^{h(v)-1}$.
- (2) For each node $v \in V$ with $h(v) = l$, its contribution to the maximum network throughput for a period of τ is identical, which is $d_l = d_{max}^v = p^{l-1} \cdot (\tau \cdot r)$. Maximizing d_l is equivalent to minimizing l , while $l \geq h(v)$, l is equal to $h(v)$. Thus, $D_{max} = \sum_{v \in V} d_{max}^v = \sum_{l=2}^h (|V_l| \cdot p^{l-1} \cdot \tau \cdot r)$. $\square$

From Lemma 1, it can be seen that the forest consisting of routing trees rooted at the gateways, derived by the removal of the virtual sink s_0 and its adjacent edges from tree T^{BFS} , can deliver the maximum network throughput. However, the corresponding service cost incurred by transferring this network throughput is not optimized. To minimize the service cost, gateways should fully utilize their data quotas to avoid quota exceeding penalties while keeping the maximum throughput unchanged. To this end, we first construct a forest consisting of load-balanced shortest path trees by employing the maximum flow technique, where the length of a “shortest” path is the minimum number of hops from a source node to its gateway. We then refine the load balance among gateways by adjusting their loads through modifying the routing trees to further reduce the service cost as follows.

4.1. Load-balanced forest

We construct a forest $\mathcal{F}$ consisting of K routing trees rooted at the K gateways such that the load among the gateways is well balanced while the maximum network throughput is still maintained. To maximize the network throughput, following Lemma 1, the routing trees in $\mathcal{F}$ are constructed layer by layer. Layer l contains only the nodes in V_l , and the load among the gateways is distributed as balanced as possible, where $1 \leq l \leq h$. The proposed algorithm proceeds iteratively. Within each iteration, a level expansion of each tree is conducted. Let $\mathcal{F}_l$ be the forest spanning the nodes in the first l layers. Initially, $\mathcal{F}_1$ spans all gateways $g_1, g_2, \dots, g_K \in GW$, and the load of each gateway $L(g_i)$ is 0 for all i with $1 \leq i \leq K$. We assume that the nodes in the first l layers have been explored and included in $\mathcal{F}_l$. We now expand the forest $\mathcal{F}_l$ to forest $\mathcal{F}_{l+1}$ by including the nodes in V_{l+1} with the objective to minimize the maximum load among the K gateways in the resultant forest $\mathcal{F}_{l+1}$. To this end, we make use of the maximum flow technique as follows.

We construct a flow network $G_f = (V_f, E_f; c)$, where $V_f = \{s, t\} \cup \{g_1, g_2, \dots, g_K\} \cup V_{l+1}$, s and t are the source node and destination node in G_f , and c is the link capacity function. Let $V_{l+1} = \{x_1, x_2, \dots, x_{|V_{l+1}|}\}$. There is a directed edge in E_f from s to each node $x_j \in V_{l+1}$ with capacity of 1, and there is a directed edge from each gateway g_i to the destination node t with capacity of non-negative integer $B_i = B - \lfloor \frac{L(g_i)}{d_{l+1}} \rfloor$, where $d_{l+1} = p^l \cdot \tau \cdot r$ is the amount of data that can be collected at a gateway from any node in V_{l+1} and B is an integer related to the maximum load among the gateways, which will be defined later. If a node $x_j \in V_{l+1}$ has at least one edge in G connecting a node in the tree rooted at g_i , there is an edge in E_f from node x_j to gateway g_i with capacity of 1. Given a flow f in the flow network $G_f = (V_f, E_f; c)$ from s to t , $f(u, v)$ is the amount of flow in edge $(u, v) \in E_f$ from node u to node v . The value $|f|$ of flow f is defined as $|f| = \sum_{v \in V_f} f(s, v) - \sum_{v \in V_f} f(v, s)$. Figure 1 illustrates the construction of G_f through an example.

Minimizing the maximum load among the gateways is equal to minimizing the value of B for a maximum flow f in G_f from s to t such that $|f| = |V_{l+1}|$ (i.e., each node in V_{l+1} will be included in one of the K routing trees), where the value of B is in the range between $\lfloor \frac{\max_{1 \leq i \leq K} \{L(g_i)\}}{d_{l+1}} \rfloor$ and $\lfloor \frac{\max_{1 \leq i \leq K} \{L(g_i)\} + |V_{l+1}| \cdot d_{l+1}}{d_{l+1}} \rfloor = \lfloor \frac{\max_{1 \leq i \leq K} \{L(g_i)\}}{d_{l+1}} \rfloor + |V_{l+1}|$. The smallest B can be found by algorithm `Smallest_Load`, which is described in Algorithm 1.

With the smallest value of B , a flow f of value $|f| = |V_{l+1}|$ from s to t is obtained. Let $|f_i| = \sum_{x_j \in V_{l+1}} f(x_j, g_i)$ be the total amount of flow entering gateway g_i , then the corresponding nodes in V_{l+1} will join tree T_i , and the load $L(g_i)$ of T_i is updated to $L(g_i) + |f_i| \cdot d_{l+1}$ for all i with $1 \leq i \leq K$. Forest $\mathcal{F}_l$ then is expanded to forest $\mathcal{F}_{l+1}$. The proposed algorithm for constructing a forest, `Load_Balanced_Forest`, is described in Algorithm 2.

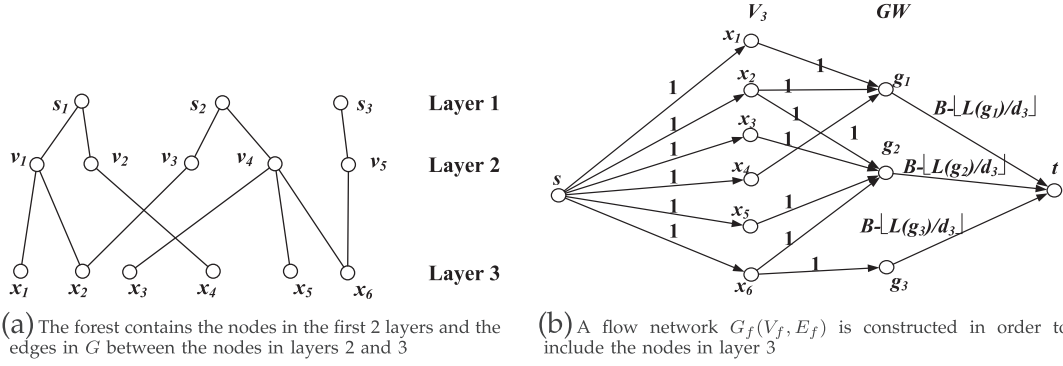


Figure 1. An illustration of constructing flow network G_f . (a) The forest contains the nodes in the first two layers and the edges in G between the nodes in layers 2 and 3, and (b) a flow network $G_f(V_f, E_f)$ is constructed in order to include the nodes in layer 3.

Algorithm 1 Smallest_Load

Input: Flow graph $G_f(V_f, E_f, c)$ and the current load of gateway $L(g_i)$ for all i with $1 \leq i \leq K$, $d_{l+1} = p^l \cdot \tau \cdot r$

Output: The smallest value of B and the flow f

```

1:  $lower\_bound \leftarrow \lfloor \frac{\max_{1 \leq i \leq K} \{L(g_i)\}}{d_{l+1}} \rfloor$ ;
2:  $upper\_bound \leftarrow \lfloor \frac{\max_{1 \leq i \leq K} \{L(g_i)\}}{d_{l+1}} \rfloor + |V_{l+1}|$ ;
3: while ( $lower\_bound \neq upper\_bound$ ) do
4:    $B \leftarrow \lfloor \frac{lower\_bound + upper\_bound}{2} \rfloor$ ;
5:   Assign each edge  $\langle g_i, t \rangle$  in  $G_f$  the capacity of
      $B_i = B - \lfloor \frac{L(g_i)}{d_{l+1}} \rfloor$ ;
6:   Find a maximum flow  $f$  in  $G_f$  from  $s$  to  $t$ ;
7:   if  $|f| = |V_{l+1}|$  then
8:     /* flow  $f$  saturates all outgoing edges from  $s$  */
9:      $upper\_bound \leftarrow B$ ;
10:  else
11:     $lower\_bound \leftarrow B + 1$ ;
12:  end if
13: end while
14:  $B \leftarrow upper\_bound$ ;
15: return  $B$  and  $f$ .
```

Algorithm 2 Load_Balanced_Forest

Input: $G(V \cup GW, E)$, h disjoint subsets of nodes $V_1, V_2, \dots, V_h$, and the link reliability p

Output: The load $L(g_i)$ of gateway node g_i and routing tree T_i rooted at g_i for all i with $1 \leq i \leq K$

```

1:  $L(g_i) \leftarrow 0$  for all  $i$  with  $1 \leq i \leq K$ ; /* The initial load
   of each gateway is 0 */
2:  $\mathcal{F}_1 \leftarrow \{T_i = \{g_i\} \mid 1 \leq i \leq K\}$ ;
3: for  $l \leftarrow 1$  to  $h-1$  do
4:   Calculate the network throughput contribution of
     each node in layer  $l+1$ ,  $d_{l+1} \leftarrow p^l \cdot (\tau \cdot r)$ ;
5:   /* Perform the layer expansion from layer  $l$  to  $l+1$  by
     including all nodes in  $V_{l+1}$  */;
6:   Construct a flow network  $G_f = (V_f, E_f, c)$ ;
7:   Obtain the maximum flow  $f$  from  $s$  to  $t$  in  $G_f$  by
     calling Algorithm 1;
8:   Update the load of gateway  $g_i$ ,
      $L(g_i) \leftarrow L(g_i) + |f_i| \cdot d_{l+1}$  for each  $1 \leq i \leq K$ ,
     where  $|f_i| = \sum_{x_j \in V_{l+1}} |f(x_j, g_i)|$ ;
9:   Expand  $\mathcal{F}_l$  to  $\mathcal{F}_{l+1}$ ;
10: end for
11: return  $\mathcal{F}_h$  and  $L(g_i)$  for all  $i$  with  $1 \leq i \leq K$ .
```

4.2. Dynamic load readjustment

Having the forest of load-balanced routing trees $\mathcal{F}$, the rest is to further balance the load among the gateways through a dynamic readjustment to further reduce the service cost. The load imbalance among the gateways occurs because no information below layer $l+1$ is available when performing node expansions from the current layer l to the next layer $l+1$. Thus, the loads of the gateways in the forest $\mathcal{F}_h$ delivered by algorithm Load_Balanced_Forest may not be balanced. To reduce the service cost by balancing the load among the K gateways, we consider the following cases.

If the load of each gateway is no greater than or no less than its data quota Q , there is nothing to be carried out as the service cost is already the minimum one. Otherwise, the cost can be further reduced by readjusting

the load among the K gateways through a series of edge swapping that replaces a tree edge by a non-tree edge while keeping the obtained network throughput unchanged. To this end, not all but only certain types of non-tree edges can be swapped with the tree edges, which is stated by the following lemma.

Lemma 2. *Given an unreliable sensor network $G(V \cup GW, E)$ with uniform link reliability p and a forest $\mathcal{F}$ consisting of load-balanced routing trees rooted at the gateways, the load adjustment among the gateways can be achieved through a series of edge swapping between tree edges and non-tree edges. Specifically, for all l with $1 \leq l \leq h-1$, swapping any non-tree edge with a tree edge with the endpoints of both edges being in V_l and V_{l+1} does not change the network throughput; otherwise, swapping a*

tree edge with a non-tree edge whose endpoints are not in V_l and V_{l+1} may reduce the obtained network throughput.

Proof. See Appendix A. $\square$

To maintain the maximum network throughput with the minimal service cost, following Lemma 2, we only make use of the non-tree edges between neighboring layers to adjust the load among the gateways. Let $E_{l+1,l} = (V_{l+1} \times V_l) \cap E$ be the edge set between two neighboring layers $l+1$ and l for all l with $1 \leq l \leq h-1$. Let $e_1 = (u, v) \in E_{l+1,l}$ be a tree edge considered with nodes u and v being in layers $l+1$ and l , respectively. Let $L(u)$ be the volume of data collected at node u from all its descendant nodes in the subtree rooted at u , and the volume of data received at gateway g_i from the subtree rooted at node u is $p^l \cdot L(u)$. Let $e_2 = (u, v') \in E_{l+1,l}$ be a non-tree edge. Assume that v and v' are in trees T_i and T_j rooted at gateways g_i and g_j , respectively. We will remove e_1 from T_i and add another non-tree edge e_2 to T_j if this leads to a smaller service cost. We distinguish the edge swapping into the following three cases.

- Case 1: if T_i and T_j are the same tree rooted at g_i , we do nothing.
- Case 2: if the load of tree T_i , $L(g_i)$, is no greater than the data quota Q , we do nothing.
- Case 3: perform edge swapping if (i) T_i and T_j are not the same tree, (ii) $L(g_i) > Q$ while $L(g_j) < Q$, and (iii) $L(g_i) - p^l \cdot L(u) > L(g_j)$. Note that although the routing trees in the forest have been updated, the network throughput delivered by the forest still has not changed by Lemma 2. We thus have the following algorithm for dynamic load readjustment.

Given the load-balanced trees in the forest $\mathcal{F}$, traverse each tree rooted at a gateway g whose load $L(g)$ is greater than the data quota Q by examining its edges layer by layer (from lower layers to higher layers). If swapping a pair of tree and non-tree edges leads to a reduction on the service cost, perform the swapping. This procedure continues until all the tree edges have been examined. The detailed algorithm `Refine_Cost` is described in Algorithm 3.

4.3. Algorithm

The proposed algorithm `Uniform_Link` for the problem with uniform link reliability is described in Algorithm 4.

Theorem 2. Given an unreliable sensor network $G(V \cup GW, E)$ with uniform link reliability p with $0 < p < 1$, there is an approximation algorithm `Uniform_Link` that can achieve the maximum network throughput with at most $(1 + \frac{C_p}{C_f})$ times of the minimal service cost. The time complexity of the proposed algorithm is $O(|V||E|^2)$, assuming that $|GW| \ll |V|$, where C_f is the cost of a fixed data

Algorithm 3 `Refine_Cost`

Input: $G(V \cup GW, E)$, the forest $\mathcal{F}$, $V_1, V_2, \dots, V_h$, and the data quota Q

Output: The refined cost C and the updated forest $\mathcal{F}$.

- 1: Calculate the service cost of forest $\mathcal{F}$: $C = K \cdot C_f + \sum_{i=1}^K \max\{0, (L(g_i) - Q) \cdot c_p\}$;
 - 2: **if** $L(g_i) \leq Q$ or $L(g_i) \geq Q$ for all i /*, the load of each gateway is greater or smaller than the quota Q */ **then**
 - 3: **EXIT**;
 - 4: **else**
 - 5: **for** $l \leftarrow h-1$ down to 1 **do**
 - 6: $E_{l,l+1} \leftarrow (V_l \times V_{l+1}) \cap E$;
 - 7: **while** there are tree edges that have not been examined **do**
 - 8: For any two edges $e_1 = (u, v)$ (a not-examined tree edge) and $e_2 = (u, v')$ (a non-tree edge) in $E_{l,l+1}$, where $v, v' \in V_l$, $v \neq v'$, and $u \in V_{l+1}$, perform the edge swapping if e_1 and e_2 are in two different trees T_i and T_j rooted at gateways g_i and g_j in $\mathcal{F}$, $L(g_i) > Q$ and $L(g_j) < Q$, and $L(g_i) - p^l \cdot L(u) > L(g_j)$;
 - 9: Update the trees rooted at g_i and g_j and their loads;
 - 10: **end while**
 - 11: **end for**
 - 12: Recalculate the service cost C of the resulting forest $\mathcal{F}$ according to Equation (2);
 - 13: **end if**
 - 14: **return** C and $\mathcal{F}$.
-

Algorithm 4 `Uniform_Link`

Input: $G(V \cup GW, E)$, monitoring period of τ , the data plan, and the link reliability p

Output: The network throughput $D^{(\tau)}$ and the cost C

- 1: $\mathcal{F} \leftarrow \emptyset$; /* The forest of routing trees */
 - 2: Partition the nodes in $V \cup GW$ into h disjoint subsets $V_1, V_2, \dots, V_h$, using the information of the BFS tree;
 - 3: Construct a forest $\mathcal{F}$ of load-balanced trees rooted at gateways such that the maximum load among the gateways is minimized, by calling Algorithm 2;
 - 4: Readjust the loads of gateways of routing trees by calling Algorithm 3;
 - 5: **return** $D^{(\tau)}$ and C .
-

quota Q for each gateway during a period of τ , $C_p = c_p \cdot Q$, and c_p is the penalty rate of every exceeding megabyte of data over the quota Q .

Proof. See Appendix B. $\square$

Notice that the approximation ratio $1 + \frac{C_p}{C_f}$ of the service cost is usually a small constant. For example, one typical data plan in Vodafone is a fixed cost $C_f = \$19$ for data quota $Q = 2$ GB per month and the penalty rate of

$c_p = \$0.02/\text{MB}$ [42]. The approximation ratio of the service cost in this realistic data plan is no more than $1 + \frac{C_p}{C_f} = 1 + \frac{0.02 \cdot 1000 \cdot 2}{19} \approx 3$.

5. ALGORITHM FOR NON-UNIFORM LINK RELIABILITY

In this section, we consider the maximizing network throughput with minimal service cost problem under non-uniform link reliability assumption. The proposed algorithm will strive for a non-trivial trade-offs between the network throughput and the incurred service cost. We start with a heuristic algorithm that delivers the maximum network throughput, but the service cost is not optimized. We then devise approximation algorithms by incorporating the service cost into consideration.

5.1. A heuristic for maximizing network throughput

Recall that $T_1, T_2, \dots, T_K$ are the K routing trees rooted at the K gateway nodes. We now construct K routing trees such that the network throughput derived from them is maximized as follows.

Initially, each tree T_i contains only gateway $g_i \in GW$ for all i with $1 \leq i \leq K$. Let $V' \subseteq V$ be the set of nodes not in the trees but are one-hop neighbors of the nodes in $\cup_{i=1}^K V(T_i)$, that is, $V' = \{v \mid (u, v) \in E, u \in \cup_{i=1}^K V(T_i), v \notin \cup_{i=1}^K V(T_i)\}$. The proposed heuristic proceeds iteratively. Within each iteration, only one node in V' is added to one of the K trees. Consider a node $v \in V'$. If there are multiple edges between v and the nodes in T_i , we choose one endpoint $u \in V(T_i)$ such that the reliability of the unique path from v to g_i , $p_{v,g_i} = p_{v,u} \cdot p_{u,g_i}$, is maximized. Thus, the contribution of node v to the network throughput via gateway g_i for the period of τ is $d_{v,g_i}^{(\tau)} = p_{v,g_i} \cdot (\tau \cdot r)$. If there are multiple trees in which v has edges connected to nodes in the trees, the maximum contribution made by v to the network throughput is $d_{\max}^v = \max_{1 \leq i \leq K} \{d_{v,g_i}^{(\tau)}\}$. Having calculated the maximum contribution of each node $v \in V'$, we add a node $v' \in V'$ to a tree T_j in the current iteration if the maximum contribution made by v' via gateway g_j is the maximum one among the nodes in V' , that is, $d_{\max}^{v'} = \max_{v \in V'} \{d_{\max}^v\}$. Node v' is then removed from V' . This procedure continues until $V' = \emptyset$.

The service cost of achieving the maximum network throughput then can be calculated by Equation (2). We refer to this iterative algorithm as algorithm `Max_Throughput`.

Theorem 3. *Given an unreliable sensor network $G(V \cup GW, E)$ with link reliability p_e for each link $e \in E$ with $0 < p_e \leq 1$, algorithm `Max_Throughput` delivers the maximum network throughput without optimizing the service cost of the obtained network throughput.*

Proof. See Appendix C. $\square$

5.2. Approximation algorithm

In this subsection, we propose approximation algorithms for the problem by incorporating the service cost into consideration. For convenience, we assume that the reliability p_e of each link e in $G(V \cup GW, E)$ is within the range of $[p, (1 + \delta)p]$, where $p > 0$ and $(1 + \delta)p \leq 1$.

We first construct a forest in the original network G , consisting of load-balanced trees rooted at the gateways, using algorithm `Load_Balanced_Forest` assuming that each link in G has an identical reliability p . Let $\mathcal{F} = \{T_1, T_2, \dots, T_K\}$ be the forest, which is a feasible solution to the problem. We then perform dynamic load readjustment to the trees in $\mathcal{F}$ to further improve the network throughput while optimizing the service cost as well. Note that the dynamic load readjustment algorithm for the problem with uniform link reliability in the previous section is not applicable to this general setting because performing edge swapping between neighboring layers may reduce the network throughput as links have different reliabilities. We thus need to modify algorithm `Refine_Cost` accordingly. Similar to the discussions in the previous section, we only consider the edge swapping between tree edges and non-tree edges in each layer; otherwise, it will make the problem intractable.

Consider swapping a pair of edges: a tree edge $e_1 = (u, v)$ and a non-tree edge $e_2 = (u, v')$, where node u is in layer $l + 1$ while nodes v and v' are in layer l of trees T_i and T_j rooted at gateways g_i and g_j , respectively. Let Δd be the net network throughput after performing a pair of edge swapping, and let $L(g)$ and $L'(g)$ be the loads of gateway g before and after the edge swapping. The edge swapping will proceed only if $\Delta d > 0$, and there are five cases to be considered.

Case 1: $T_i = T_j$ and $L(g_i) > Q$. Perform the swapping only if $\Delta d = (p_{u,v'} \cdot p_{v',g_i} - p_{u,v} \cdot p_{v,g_i}) \cdot L(u) > 0$. The swapping increases the network throughput, and accordingly, the service cost for Δd will be paid.

Case 2: $T_i = T_j$ and $L(g_i) < Q$. Perform the swapping only if $\Delta d = (p_{u,v'} \cdot p_{v',g_i} - p_{u,v} \cdot p_{v,g_i}) \cdot L(u) > 0$. Although the net increase on the network throughput is Δd , we only pay the extra charge for the amount of data $\Delta d - (Q - L(g_i))$ if $\Delta d - (Q - L(g_i)) > 0$.

Case 3: $T_i \neq T_j$, $L(g_i) > Q$ and $L(g_j) < Q$. Perform the swapping only if the following two conditions are met: (i) $\Delta d = (p_{u,v'} \cdot p_{v',g_j} - p_{u,v} \cdot p_{v,g_i}) \cdot L(u) > 0$ and (ii) either $L'(g_i) = L(g_i) - p_{u,v} \cdot p_{v,g_i} \cdot L(u) \geq Q$ or $Q - L(g_j) > Q - L'(g_i)$. Condition (i) is obvious, while the rationale of condition (ii) is as follows. Assume that the swapping has been performed, the loads of gateway g_i and g_j then become $L'(g_i) = L(g_i) - p_{u,v} \cdot p_{v,g_i} \cdot L(u)$ and $L'(g_j) = L(g_j) + p_{u,v'} \cdot p_{v',g_j} \cdot L(u)$, respectively. Now, if $L'(g_i) \geq Q$, the extra increase on the network throughput will be paid with a less service cost, because only the amount of data $p_{u,v'} \cdot p_{v',g_j} \cdot L(u) - (Q - L(g_j))$ incurs an extra charge if

$p_{u,v'} \cdot p_{v',g_j} \cdot L(u) - (Q - L(g_j)) > 0$, as $L(g_j) < Q$ by our assumption. Otherwise, $(L'(g_i) < Q)$, and the amount of data quota of gateway g_i , $Q - L'(g_i)$, will not be used (wasting part of the paid fixed cost), while the unused data quota of gateway g_j is $Q - L(g_j)$. In that case, only if $Q - L(g_j) > Q - L'(g_i)$, the swapping proceeds, and the result is economical.

Case 4: $T_i \neq T_j$, $L(g_i) < Q$, and $L(g_j) < Q$. The discussion is similar to case 3. Perform the swapping only if the following conditions are met: (i) $\Delta d = (p_{u,v'} \cdot p_{v',g_j} - p_{u,v} \cdot p_{v,g_i}) \cdot L(u) > 0$ and (ii) either $L'(g_j) \leq Q$ or $Q - L(g_j) \geq p_{u,v} \cdot p_{v,g_i} \cdot L(u)$.

Case 5: $T_i \neq T_j$, $L(g_i) > Q$, and $L(g_j) > Q$. The discussion is similar to case 3, too. Perform the swapping only if $\Delta d = (p_{u,v'} \cdot p_{v',g_j} - p_{u,v} \cdot p_{v,g_i}) \cdot L(u) > 0$ and $L'(g_i) = L(g_i) - p_{u,v} \cdot p_{v,g_i} \cdot L(u) \geq Q$.

We refer to the proposed approximation algorithm consisting of building a forest of load-balanced trees and dynamic load readjustment as algorithm **Appro**.

Algorithm 5 Appro

Input: $G(V \cup GW, E)$, monitoring period of τ , the data plan, and link reliability $p' : E \mapsto [p, p(1 + \delta)]$

Output: The network throughput $D^{(\tau)}$ and the cost C

- 1: $\mathcal{F} \leftarrow \emptyset$; /* The forest of routing trees */
 - 2: Partition the nodes in $V \cup GW$ into h disjoint subsets $V_1, V_2, \dots, V_h$, using BFS traversal;
 - 3: Construct a forest $\mathcal{F}$ of load-balanced trees rooted at gateways such that the maximum load among the gateways is minimized, by calling Algorithm 2, assuming each link having identical reliability p ;
 - 4: Balance the load among the gateways by readjusting the gateway load through edge swapping, and let $\mathcal{F}'$ be the forest derived from $\mathcal{F}$;
 - 5: Calculate the network throughput $D^{(\tau)}$ and the service cost C through the forest $\mathcal{F}'$.
-

Theorem 4. *Given an unreliable WSN $G(V \cup GW, E)$ with link reliability in $[p, (1 + \delta)p]$ and $0 \leq \delta \leq \frac{1}{p} - 1$, there is an approximation algorithm **Appro** for the maximizing network throughput with minimal service cost problem, which delivers a solution with no less than $\frac{1}{(1 + \delta)^h}$ times of the maximum network throughput while the service cost is no more than $\left(1 + \frac{C_p}{C_f}\right)$ times of the minimum one, where h is the maximum number of hops from a sensor to its nearest gateway and $C_p = c_p \cdot Q$ and C_f are the costs of amount of exceeding data Q and a fixed data quota Q , respectively. The algorithm takes $O(|V| \cdot |E|^2)$ time.*

Proof. See Appendix D. $\square$

Note that the analytical estimation on the approximation ratio of algorithm **Appro** is very conservative, which can be confirmed from the later empirical results. The

actual network throughput is no less than 62% of the maximum network throughput, while the service cost is no more than 107% of the lower bound on the minimum cost (the optimal cost).

5.3. Improved approximation algorithm

Algorithm **Appro** can be further improved in practice if δ is relatively large. Specifically, we assume that each link reliability is in $[p_{min}, p_{max}]$ with $0 < p_{min} \leq p_{max} \leq 1$. We classify the links in E of $G = (V \cup GW, E)$ into $\lceil \log \frac{p_{max}}{p_{min}} \rceil$ different groups by their reliabilities. A subgraph $G_i = (V \cup GW, E_i)$ of G is induced by the links in group i , where $E_i = \{e \mid e \in E \text{ and } p_e \geq p_i\}$ and $p_i = \min \{2^i \cdot p_{min}, p_{max} \mid 0 \leq i \leq \lceil \log \frac{p_{max}}{p_{min}} \rceil\}$. The improved algorithm proceeds as follows. Starting from $i = \lceil \log \frac{p_{max}}{p_{min}} \rceil$, for each i , a graph $G_{s_0,i} = (V \cup GW \cup \{s_0\}, E_i \cup \{(s_0, g_j) \mid g_j \in GW\})$ is constructed by adding a virtual sink s_0 and the edges between s_0 and every gateway in GW . If $G_{s_0,i}$ is disconnected, there is no solution for the problem, which will be shown by Theorem 5. The value of i is decreased by one, and the procedure continues until $G_{s_0,i}$ is connected, assuming that G is connected. Then, a solution is found by applying algorithm **Appro** on graph $G_{s_0,i}$. We refer to this improved algorithm as algorithm **Impro_Appro**.

Theorem 5. *In algorithm **Impro_Appro**, if $G_{s_0,i}$ derived from some of edges of $G = (V \cup GW, E)$ is disconnected, then there is no solution for graph $G_{s_0,j}$, with $0 < i \leq j \leq \lceil \log \frac{p_{max}}{p_{min}} \rceil$. Algorithm **Impro_Appro** can deliver an approximate solution to the problem with the time complexity of $O(|V| \cdot |E|^2 + |E| \cdot \log \frac{p_{max}}{p_{min}})$.*

Proof. See Appendix E. $\square$

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed algorithms in terms of network throughput and the service cost. We also investigate the impact of different parameters on the performance.

We consider a sensor network consisting of 1,000–3,000 sensors randomly deployed in a $1,000 \times 1,000$ m² region. The transmission range of sensors is $R = 60$ m, and the data generation rate of sensors is $r = 10$ bytes/s. The number of gateways K varies from 4 to 10, and the gateways are deployed as follows. The monitoring region is divided into roughly equal K subregions, and one gateway is randomly deployed in each of the subregions. For simplicity, we assume that the transmission range of gateways is identical to that of sensors. In our experimental evaluation, the following three different data plans for 1-month monitoring period (i.e., $\tau = 30 \text{ days} \times 24 \text{ h} \times 3600 \text{ s}$), provided by Vodafone [42], will be examined: (I) $Q = 2$ GB and $C_f = \$19$; (II) $Q = 4$ GB and $C_f = \$29$;

and (III) $Q = 10$ GB and $C_f = \$39$. Each data plan has the same penalty rate $c_p = \$0.02/\text{MB}$.

To show the performance of the proposed algorithms, we also compute the lower bound on the minimum service cost of transferring the amount of the maximum network throughput $D^{(\tau)}$ as $C_f \cdot K + c_p \cdot \max\{D^{(\tau)} - K \cdot Q, 0\}$, where $\max\{D^{(\tau)} - K \cdot Q, 0\}$ is the lower bound on the sum of the amounts of exceeding quota at each gateway in the optimal solution to the problem. Each value in figures is the mean of the results by applying the mentioned algorithm to 30 different network topologies of the same size.

6.1. Performance evaluation with uniform link reliability

We first study the performance of the proposed algorithm `Uniform_Link` for a sensor network in which each link has an identical reliability p with $0.6 \leq p \leq 1$. We evaluate the impact of the number of gateways K on the network performance by varying K from 4 to 10 while keeping p fixed at 0.8 and adopting plan (II). Figure 2(a) indicates that for a given network size, a larger K will result in a higher network throughput. That is because the more the gateways, the less the depth of the routing trees, and the more reliable the routing paths from sensors to the gateways.

Figure 2(b) shows that the service cost is only about 106% of the lower bound cost. When K is fixed, a larger network throughput does not necessarily incur a higher service cost. Also, when n is fixed, the service cost is insensitive to the value of K . For example, when $n = 1,000$, a larger value of K results in a higher service cost because data quota exceeding is very unlikely to occur and the service cost is mainly composed of the fixed costs of K gateways. However, when n is large (e.g., $n = 2,000$ or $3,000$), the service cost decreases with the increase in the number of gateways. The reason behind is that penalties at individual gateways will dominate the service cost when the network throughput becomes much higher.

What followed is to evaluate the impact of link reliability on the network performance by varying p from 0.6 to 1.0 while keeping $K = 6$ and adopting plan (II). Figure 3 shows that the higher the link reliability, the larger the network throughput, and the higher the service cost. The service cost is about 105% of the lower bound cost. It is also worth mentioning that when the link reliability is low, for example, $p = 0.6$, the service cost does not vary much with different network size n because little penalties will be applied, and the service cost mainly included the fixed costs of gateways.

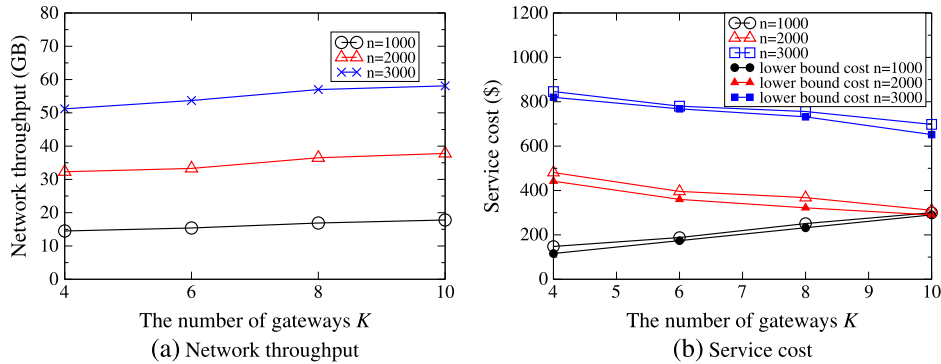


Figure 2. Impact of K on the network performance when $p = 0.8$ and plan (II) is adopted. (a) Network throughput and (b) service cost.

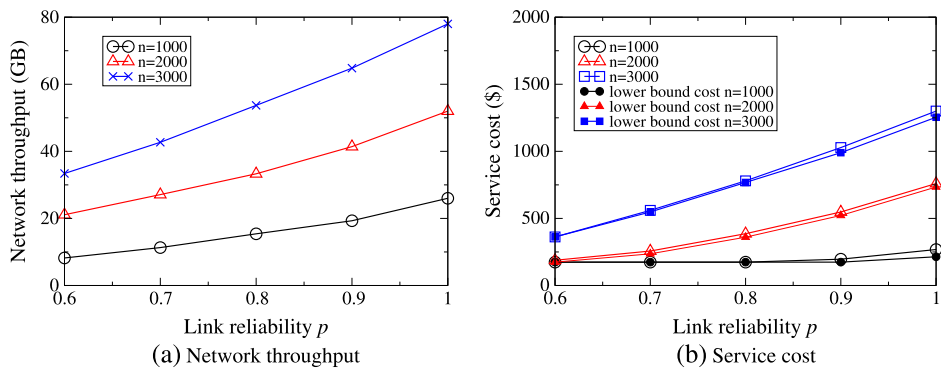


Figure 3. Impact of link reliability p on the network performance when $K = 6$ and plan (II) is adopted. (a) Network throughput and (b) service cost.

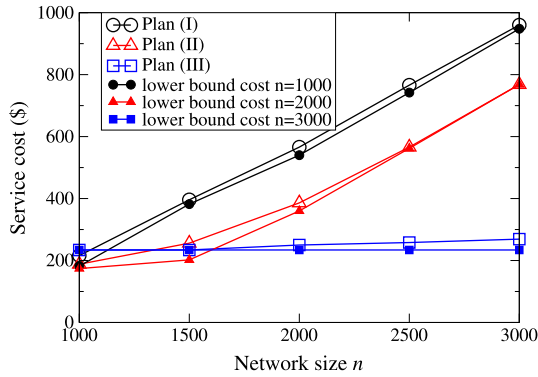


Figure 4. Impact of different data plans on the service cost when $p = 0.8$ and $K = 6$.

We now investigate the impact of different data plans on the service cost while fixing $p = 0.8$ and $K = 6$. From Figure 4, it is observed that when $n = 1000$, the service cost by plan (III) is higher than that of the other two plans because of its higher fixed cost. However, with the growth of n , plan (III) outperforms the other two plans because the data quota $Q = 10$ GB is sufficiently large to transfer the collected data at each gateway, no quota exceeding occurs, and thus, no penalty will be applied, whereas with plans (I) and (II), a larger n results in higher service costs, because of the increasing penalties. Adopting plans (I) and (III) can achieve almost the optimal cost. And with plan (II), the service cost is about 105% of the lower bound cost.

In summary, empirical results demonstrate that the actual approximation ratios of algorithm *Uniform_Link* is much better than that of its analytical counterpart in Theorem 2. The increase on a number of gateways and the reliability of links will lead to the increase on the network throughput, too; however, this does not necessarily incur a higher service cost, as the service cost does depend not only on the network throughput but also on the data plan adopted. Thus, choosing an appropriate data plan for each gateway sometimes can save the service cost substantially.

6.2. Performance evaluation with non-uniform link reliability

We then evaluate the performance of three proposed algorithms *Max_Throughput*, *Appro*, and *Impro_Appro* for the problem in the general setting.

When $K = 6$, $p_e \in [0.1, 1]$ for each link $e \in E$, and plan (II) is adopted. Figure 5 shows that *Max_Throughput* delivers the largest network throughput with the highest service cost among the three mentioned algorithms and *Impro_Appro* leads to a higher network throughput than that of algorithm *Appro* but with a slightly higher service cost. The network throughput delivered by algorithms *Appro* and *Impro_Appro* is no less than 62% and 71% of the maximum one. The service costs by algorithms *Appro* and *Impro_Appro* are no more than 107% and 106% of the lower bounds on their corresponding minimal costs.

We next investigate the impact of the number of gateways K on the network performance when $n = 2,000$, $p_e \in [0.1, 1]$ for each link $e \in E$, and plan (II) is adopted. Figure 6 indicates that a larger K will result in a higher network throughput but does not necessarily incur a higher service cost, because a larger number of gateways can reduce the average length of the routing path from a node to a gateway, which implies more reliable end-to-end data routing and a higher network throughput. Also, the service cost is not necessarily increased even though the amount of the fixed cost goes up, because penalties may be decreased. The service cost approximation ratios of algorithms *Max_Throughput*, *Appro*, and *Impro_Appro* are, respectively, 125%, 113%, and 106%.

We finally study the impact of the range of link reliability on the network performance by varying the ranges of link reliability as follows: $[0.1, 0.2]$, $[0.1, 0.4]$, $[0.1, 0.6]$, $[0.1, 0.8]$, and $[0.1, 1.0]$ while fixing $K = 6$, $n = 2,000$, and adopting plan (II). Accordingly, the values of δ are 1, 3, 5, 7, and 9. Figure 7 shows that with the growth of the link reliability range, the network throughput increases, so does the service cost. It is observed that when the range of link reliability is small, the network throughput delivered

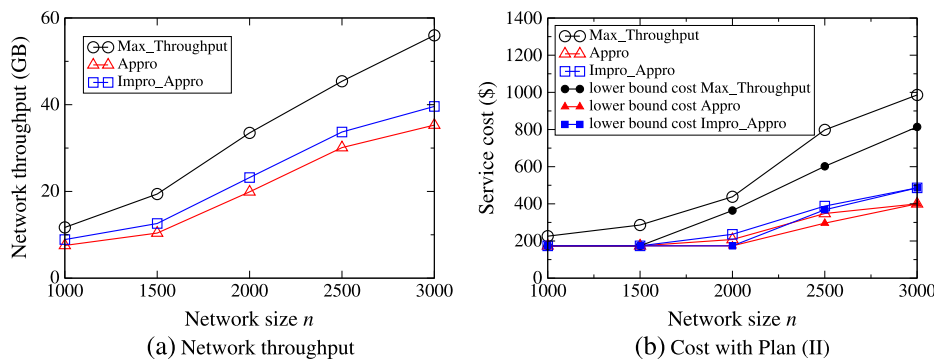


Figure 5. Performance of different algorithms when $K = 6$, $p_e \in [0.1, 1]$ for each link $e \in E$, and plan (II) is adopted. (a) Network throughput and (b) cost with plan (II).

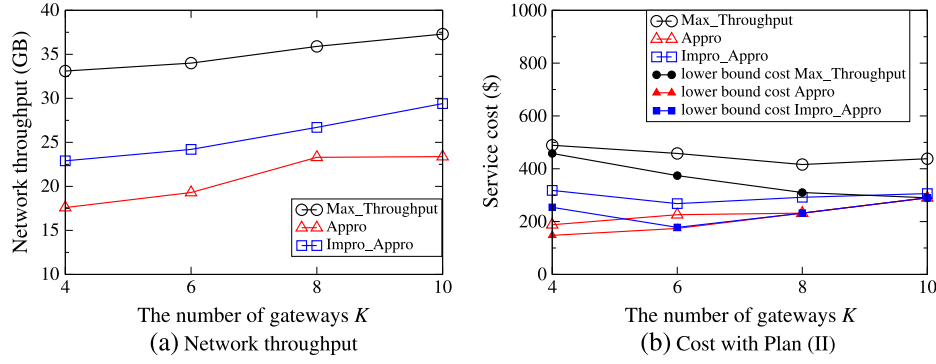


Figure 6. Impact of K on the network performance when $n=2,000$, $p_e \in [0.1, 1]$ for each $e \in E$, and plan (II) is adopted. (a) Network throughput and (b) cost with plan (II).

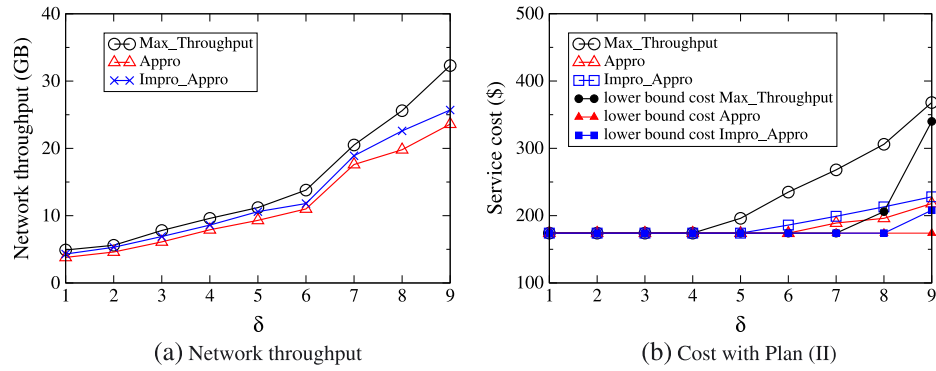


Figure 7. Impact of the range of link reliability on the network performance when $K = 6$, $n = 2000$, and plan (II) is adopted. (a) Network throughput and (b) cost with plan (II).

by algorithms Appro and Impro_Appro is close to the maximum one. When the value of δ increases, the gap of the service cost between Max_Throughput and the other two approximation algorithms becomes larger.

7. CONCLUSION

In this paper, we considered the deployment of an unreliable WSN for large-scale remote monitoring, where the monitoring center is geographically far away from the region of the sensor network. Sensing data must be transferred to the monitoring center through a third-party telecommunication service that incurs the service cost, for which we first formulated a novel maximizing network throughput with minimal service cost problem and showed its NP-completeness. We then proposed approximation algorithms with provable approximation ratios. Finally, we conducted experiments by simulations to evaluate the performance of the proposed algorithms. The experimental results demonstrated that the proposed algorithms are very promising, and the solutions delivered are fractional of the optimum.

APPENDIX A: PROOF OF LEMMA 2

Proof. Let (u, v) be a tree edge with $u \in V_{l+1}$ and $v \in V_l$. Let $L(u)$ be the volume of data collected at node u from all its descendant nodes in the subtree rooted at u , and this volume of data is then forwarded to the gateway that u is attached, and the resulting volume of collected data at the gateway derived from node u is $p^l \cdot L(u)$. Now, if we remove this tree edge (u, v) from the tree and instead add another non-tree edge (u, v') to form a new tree edge where $v' \in V_l$, then the volume of collected data of a gateway derived from the nodes in the subtree rooted at u is still $p^l \cdot L(u)$, which implies that the accumulative volume of collected data at all gateways does not change through this type of edge swapping.

Now, consider a non-neighboring non-tree edge (u, v'') with $v'' \notin V_l$ if it does exist, then the depth $h(v'')$ of node v'' must be equal to or greater than $l+1$ because the length of the shortest path (in hops) between node u and its nearest gateway is $l+1$, that is, $h(v'') \geq l+1$, and the amount of data collected by a reachable gateway from u is $p^{h(v'')} \cdot L(u) \leq p^{l+1} \cdot L(u) \leq p^l \cdot L(u)$ because $p \leq 1$. This implies that the network throughput will be reduced if performing the edge swapping between the tree edge (u, v) and the

non-tree edge (u, v'') , which contradicts the assumption that the network throughput obtained is still unchanged after the edge swapping. The lemma then follows. $\square$

APPENDIX B: PROOF OF THEOREM 2

Proof. Following our analysis, if each sensor sends its data along a shortest path (in terms of the number of hops) to a gateway, it makes the maximum contribution to the network throughput. In algorithm `Uniform_Link`, each sensor routes its data to a gateway along a shortest path; thus, the accumulative volume of data received by all gateways is the maximum one.

We then show that the service cost of the solution is no more than $\left(1 + \frac{C_p}{C_f}\right)$ times of the minimal one. Assuming that in the optimal solution, the minimum and maximum loads among gateways are $L_{\min}$ and $L_{\max}$, respectively, and let $C_{\min}$ be the minimal cost. Let $L'_{\min}$ and $L'_{\max}$ be the minimum and maximum loads among the gateways in the forest $\mathcal{F}$ delivered by algorithm `Uniform_Link`, and let C be the service cost. We analyze the relationship between C and $C_{\min}$ by the following three cases.

Case 1: $L_{\min} \leq Q$, $L_{\max} \leq Q$, and $C_{\min} = K \cdot C_f$. If $L'_{\max} \leq Q$, then $C = C_{\min}$. Otherwise ($L'_{\max} > Q$), this implies that the below-quota loads of some gateways in the optimal solution now are reallocated to the other above-quota gateways in $\mathcal{F}$. Because the load of each gateway is less than Q in the optimal solution, the maximum amount of shifted load is no more than $(K-1) \cdot Q$. Thus, $C \leq (K-1) \cdot Q \cdot c_p + C_{\min} < K \cdot Q \cdot c_p + C_{\min} \leq \frac{C_p}{C_f} \cdot Q \cdot c_p + C_{\min} = C_{\min} \cdot \left(1 + \frac{Q \cdot c_p}{C_f}\right) = C_{\min} \cdot \left(1 + \frac{C_p}{C_f}\right)$ because $C_{\min} \geq K \cdot C_f$.

Case 2: $L_{\min} \geq Q$, $L_{\max} \geq Q$, and $C_{\min} \geq K \cdot C_f$. If $L'_{\min} \geq Q$, then $C = C_{\min}$. Otherwise ($L'_{\min} < Q$), the loads of some gateways in $\mathcal{F}$ are less than the data quota Q , while the loads of the rest of gateways are far more than data quotas. For each below-data quota gateways, the amount of its data load is shifted to others that cause a higher penalty cost. In the worst case, there are $(K-1)$ gateways whose loads are below the data quotas, and there is only one gateway whose load is above the data quota; then the service cost is $C \leq (K-1) \cdot Q \cdot c_p + C_{\min} < K \cdot Q \cdot c_p + C_{\min} \leq C_{\min} \cdot \left(1 + \frac{C_p}{C_f}\right)$, following the similar discussion in case 1, omitted.

Case 3: $L_{\min} \leq Q$, $L_{\max} \geq Q$, and $C_{\min} \geq K \cdot C_f$. If $L'_{\min} < L_{\min}$ and $L'_{\max} > Q$, then $C < C_{\min} \cdot \left(1 + \frac{C_p}{C_f}\right)$, following the similar arguments as for cases 1 and 2, omitted. Otherwise, it can be shown that $C < C_{\min} \cdot \left(1 + \frac{C_p}{C_f}\right)$ as well.

Thus, the service cost of the solution is no more than $1 + \frac{C_p}{C_f}$ times of the minimal cost.

The time complexity of algorithm `Uniform_Link` is analyzed as follows. Partitioning the nodes into layers by the BFS transversal to graph G takes $O(|V| + |E|)$ time. In the construction of the forest of load-balanced routing trees, the main algorithm is the layer expansion by invoking a maximum flow algorithm, taking $O(|E_{l,l+1}| \cdot (|V_{l+1}| + K)) = O(|E_{l,l+1}| \cdot |V|)$ time. Because it takes $O(\log |V_{l+1}|)$ time to find the minimum value of the maximum capacity B for the flow network, the time spent on layer expansion is $O(|E_{l,l+1}| \cdot |V| \log |V_{l+1}|)$, and the time for finding the forest is $\sum_{l=1}^{h-1} O(|E_{l,l+1}| \cdot |V| \log |V_{l+1}|) = O(|E| \cdot |V| \log |V|)$. The time complexity of dynamic load readjustment is analyzed as follows. Algorithm `Refine_Cost` proceeds layer by layer. The number of pairs of edge swapping between layers l and $l+1$ is no more than $|E_{l,l+1}|^2$, while each edge swapping takes $O(|V|)$ time by updating at most two routing trees in the forest. Thus, the time spent on dynamic load readjustment is $\sum_{l=1}^{h-1} O(|E_{l,l+1}|^2 \cdot |V|) = O\left(|V| \left(\sum_{l=1}^{h-1} |E_{l,l+1}|^2\right)\right) = O(|V| \cdot |E|^2)$. The time complexity of `Uniform_Link` therefore is $\sum_{l=1}^{h-1} O(|E_{l,l+1}| \cdot |V| \log |V_{l+1}|) + \sum_{l=1}^{h-1} O(|E_{l,l+1}|^2 \cdot |V|) = O(|V| |E|^2)$. $\square$

APPENDIX C: PROOF OF THEOREM 3

Proof. Let $p_{\max}^v$ be the end-to-end reliability of the most reliable routing path from v to one of its reachable gateways, that is, $p_{\max}^v = \max\{p_{v,g_i} \mid g_i \text{ is a reachable gateway from } v \text{ in } G\}$. The maximum network throughput contribution by node v is $d_{\max}^v = p_{\max}^v \cdot \tau \cdot r$, and the maximum network throughput contributed by all nodes during the period of τ is $D_{\max}^{(\tau)} = \sum_{v \in V} (p_{\max}^v \cdot \tau \cdot r)$.

We now show that the accumulative volume of data collected by all gateways in the proposed algorithm is exactly $D_{\max}^{(\tau)}$ by contradiction. Let $v_1, v_2, \dots, v_i$ be the node sequence added to the K routing trees, where node v_j is added to the forest prior to v_{j+1} for all j with $1 \leq j \leq i-1 < |V|$. Assume that v_i is added at iteration i . Let $N(v_i)$ be the set of one-hop neighbors of v_i in the network G and $N_T(v_i)$ be a subset of $N(v_i)$ in which all nodes have been included in the forest. We claim that connecting v_i to one of its neighbors $u \in N_T(v_i)$ will result in the maximum network throughput contribution by v_i . That is, $p_{\max}^{v_i} = \max_{u \in N_T(v_i)} \{p_{v_i,u} \cdot p_{\max}^u\}$. Otherwise, assume that the most reliable routing path from v_i to a gateway is first via its neighbor $u \notin \bigcup_{j=1}^K V(T_j)$ and then via a node $v' \in N_T(v_i)$, which is connected to u , that is, $p_{\max}^{v_i} = \max\{p_{v_i,u} \cdot p_{u,v'} \cdot p_{\max}^{v'} \mid u \in N(v_i) \setminus N_T(v_i), v' \in \bigcup_{j=1}^K V(T_j)\}$. In that case, the maximum network throughput contributions made by nodes u and v_i are $d_{\max}^u = p_{u,v'} \cdot p_{\max}^{v'} \cdot (\tau \cdot r)$ and $d_{\max}^{v_i} = p_{v_i,u} \cdot p_{u,v'} \cdot p_{\max}^{v'} \cdot (\tau \cdot r) = p_{v_i,u} \cdot d_{\max}^u < d_{\max}^u$, respectively. Following the forest construction, u should have already been added to the forest prior to v_i , which contradicts the assumption that

$u \in N(v_i) \setminus N_T(v_i)$ is not yet in the forest when adding v_i to the forest. Thus, every node $v \in V$ is added to the forest via the most reliable path and has the maximum network throughput contribution $p_{\max}^v \cdot \tau \cdot r$. The accumulative volume of data collected by all gateways is $D_{\max}^{(\tau)}$.

APPENDIX D: PROOF OF THEOREM 4

Proof. Let $P(v, g_i)$ be the most reliable path consisting of l links in $G(V \cup GW, E)$ from node $v \in V$ to one of the K gateways, for example, gateway g_i , then the maximum volume of data that can be collected at g_i for a period of τ is $d_{\max}^v = \prod_{e \in P(v, g_i)} p_e \cdot (\tau \cdot r)$, with $l \geq h(v)$ where $h(v)$ is the height of node v in the BFS tree in G' (G' is defined in Section 4). Assume that there is another shortest path P_v from v to another gateway, if each link in P_v is the most reliable one with reliability $(1 + \delta)p$, then the volume of received data at the gateway is $d_U^v = p^{h(v)}(1 + \delta)^{h(v)} \cdot (\tau \cdot r)$, which is an upper bound on the maximum network throughput contribution of v . It is easy to see that $d_U^v \geq d_{\max}^v$ because $h(v) \leq l$. On the other hand, if all links in P_v are the least reliable, then the amount of data received at the gateway is no less than $d_L^v = p^{h(v)} \cdot (\tau \cdot r)$. Let node v be in a tree rooted at gateway g_i and $d_v^{g_i}$ be the volume of data generated from v and received at g_i , following the proposed algorithm, we have $\frac{d_v^{g_i}}{d_{\max}^v} \geq \frac{d_L^v}{d_{\max}^v} \geq \frac{d_L^v}{d_U^v} = \frac{1}{(1 + \delta)^{h(v)}}$. Let $\beta = \frac{1}{1 + \delta}$, then, $d_v^{g_i} \geq d_L^v = \beta^{h(v)} \cdot d_U^v$. Let $d_U^l = \sum_{v \in V_l} d_U^v$ be the upper bound on the accumulative volume of data collected by all gateways from sensors in V_l . Then, if the shortest routing path from each node to gateways is used to route its sensing data to a gateway, the amount of data collected at the gateway is no less than $\beta^{h(v)}$ times the maximum volume of data from node v to any gateway in the network.

Let X_l and Y_l be the sums of data received by the gateways from the nodes in V_l , using the most reliable path and the least reliable shortest path, respectively, then $X_l = \sum_{v \in V_l} \prod_{e \in P(v, g_i)} p_e \cdot (\tau \cdot r)$ and $Y_l = \sum_{v \in V_l} p^l \cdot (\tau \cdot r)$. The approximation ratio of the network throughput *Approx* by the improved algorithm to the optimal one *OPT*, ξ , is

$$\begin{aligned} \xi = \frac{\text{Approx}}{\text{OPT}} &\geq \frac{\sum_{l=2}^h Y_l}{\sum_{l=2}^h X_l} \geq \frac{\sum_{l=2}^h \sum_{v \in V_l} d_L^v}{\sum_{l=2}^h \sum_{v \in V_l} d_{\max}^v} \\ &\geq \frac{\sum_{l=2}^h \sum_{v \in V_l} d_L^v}{\sum_{l=2}^h \sum_{v \in V_l} d_U^v} \geq \frac{\sum_{l=2}^h \beta^l \cdot d_U^l}{\sum_{l=2}^h d_U^l} \geq \frac{\beta^h \cdot \sum_{l=2}^h d_U^l}{\sum_{l=2}^h d_U^l} \\ &= \beta^h = \frac{1}{(1 + \delta)^h} \end{aligned} \quad (\text{D1})$$

The upper bound on the service cost and the analysis of time complexity are almost identical to the ones in Theorem 2, omitted. $\square$

APPENDIX E: PROOF OF THEOREM 5

Proof. As the set of links in $G_{s_0, j}$ is a subset of the set of links in $G_{s_0, i}$ when $j > i$, graph $G_{s_0, j}$ is a subgraph of $G_{s_0, i}$. If $G_{s_0, i}$ is not connected, graph $G_{s_0, j}$ is not connected either, with $0 < i \leq j \leq \lceil \log \frac{p_{\max}}{p_{\min}} \rceil$. For any disconnected graph $G_{s_0, j}$, performing a BFS traversal on it will not obtain a tree because at least one node $v \in V$ is not in the connected component that the virtual sink s_0 and the gateways belong to, which implies that there is not any routing path from node v to any gateway. Thus, there is not a forest in $G_{s_0, j}$ spanning all nodes in V , and no solution can be obtained. There is an i with $0 \leq i \leq \lceil \log \frac{p_{\max}}{p_{\min}} \rceil$ such that $G_{s_0, i}$ is connected, as G is connected. And an approximate solution is obtained whose throughput is with the approximation ratio $\frac{1}{(1 + \delta_i)^h}$, where $\delta_i = \frac{p_{\max}}{2^i \cdot p_{\min}} - 1 \leq \delta = \frac{p_{\max}}{p_{\min}} - 1$.

It takes $O(|E| + |V|)$ time to check the connectivity of a graph, and the number of connectivity checks is no greater than $\lceil \log \frac{p_{\max}}{p_{\min}} \rceil$. It takes $O(|V| \cdot |E|^2)$ to deliver a solution by calling algorithm *Appro*, referring to Theorem 4. Thus, algorithm *Impro_Appro* takes $O(|V| \cdot |E|^2 + |E| \cdot \lceil \log \frac{p_{\max}}{p_{\min}} \rceil)$ time. It can be seen that although the running time of algorithm *Impro_Appro* is longer than that of algorithm *Appro*, the throughput delivered by it is much higher in comparison with that by algorithm *Appro*. $\square$

REFERENCES

1. Mainwaring A, Polastre J, Szewczyk R, Culler D, Anderson J. Wireless sensor networks for habitat monitoring. In *Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications*, ACM, 2002; 88–97.
2. Xu N, Ranngwala S, Chintalapudi K, Ganesan D, Broad A, Govindan R, Estrin D. A wireless sensor network for structural monitoring. In *Proceedings of SenSys*, ACM, 2004; 13–24.
3. Martinez K, Ong R, Hart J. Glacsweb: a sensor network for hostile environments. In *Proceedings of SECON*, IEEE, 2004; 81–87.
4. Liang W, Xu Y, Shi J, Luo J. Aggregate node placement for maximizing network lifetime in sensor networks. *Journal of the Wireless Communications and Mobile Computing* 2012; **12**: 219–235, Wiley-Blackwell publisher.
5. Han G, Dong Y, Guo H, Shu L, Wu D. Cross-layer optimized routing in wireless sensor networks with duty cycle and energy harvesting. In *Journal of Wireless Communications and Mobile Computing*, Wiley-Blackwell publisher, 2014. DOI: 10.1002/wcm.2468.

6. Ren X, Liang W, Xu W. Data collection maximization in renewable sensor networks via time-slot scheduling. *IEEE Transactions on Computers* 2014.
7. Liang W, Ren X, Jia X, Xu X. Monitoring quality maximization through fair rate allocation in harvesting sensor networks. *IEEE Transactions on Parallel and Distributed System* 2013; **24**: 1827–1840.
8. Dutta P, Hui J, Jeong J, Kim S, Sharp C, Taneja J, Tolle G, Witehouse K, Culler D. Trio: enabling sustainable and scalable outdoor wireless sensor network deployments. In *Proceedings of IPSN*, ACM, 2006; 407–415.
9. Jiang X, Polastre J, Culler D. Perpetual environmentally powered sensor networks. In *Proceedings of IPSN*, ACM, 2005; 463–468.
10. Lin K, Yu J, Hsu J, Zahedi S, Lee D, Friedman J, Kasal A, Raghunathan V, Srivastava M. Heliomote: enabling long-lived sensor networks through solar energy harvesting. In *Proceedings of SenSys*, ACM, 2005; 309.
11. Park C, Chou P. AmbiMax: autonomous energy harvesting platform for multi-supply wireless sensor nodes. In *Proceedings of SECON*, IEEE, 2006; 168–177.
12. GreenOrbs: a long-term kilo-scale wireless sensor network system in the forest. Available from: <http://www.greenorbs.org>.
13. Xu X, Liang W, Wark T, Jeong J. Maximizing network lifetime via 3G gateway assignment in dual-radio sensor networks. In *Proceedings of LCN*, IEEE, 2012; 479–486.
14. Zhu C, Shu L, Hara T, Wang L, Nishio S, Yang L. A survey on communication and data management issues in mobile sensor networks. *Journal of the Wireless Communications and Mobile Computing* 2014; **14**: 19–36, Wiley-Blackwell publisher.
15. Xu X, Liang W, Jia X, Xu W. Maximizing network throughput with minimal remote data transfer cost in unreliable wireless sensor networks. In *Proceedings of Mobihoc*, ACM, 2013; 245–248.
16. Xu X, Liang W, Xu Z. Minimizing remote monitoring cost of wireless sensor networks. In *Proceedings of WCNC*, IEEE, 2013; 1476–1481.
17. Xu X, Liang W, Xu Z. Remote monitoring cost minimization for an unreliable sensor network with guaranteed network throughput. *Information Processing in Agriculture* 2014; **1**: 83–94.
18. Heinzelman W. Application-specific protocol architectures for wireless networks. *Ph.D. Thesis*, Massachusetts Institute of Technology, 2000.
19. Lindsey S, Raghavendra CS. PEGASIS: power-efficient gathering in sensor information systems. In *Proceedings of Aerospace Conference*, IEEE, 2002; 3-1125–3-1130.
20. Intanagonwiwat C, Govindan R, Estrin D, Heidemann J, Silva F. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking* 2003; **11**: 2–16.
21. Kalpakis K, Dasgupta K, Namjoshi P. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Computer Networks* 2003; **42**: 697–716.
22. Goel A, Estrin D. Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk. In *Proceedings of the SODA*, ACM-SIAM, 2003; 499–505.
23. Cristescu R, Beferull-Lonzano B, Vetterli M. On network correlated data gathering. In *Proceedings of the INFOCOM*, IEEE, 2004; 2571–2582.
24. Richenbach P, Wattenhofer R. Gathering correlated data in sensor networks. In *Proceedings of DIALM-POMC*, ACM, 2004; 1–7.
25. Buragohain C, Agrawal D, Suri S. Power aware routing for sensor databases. In *Proceedings of the INFOCOM*, IEEE, 2005; 83–92.
26. Liang W, Liu Y. On-line data gathering for maximizing network lifetime in sensor networks. *IEEE Transactions on Mobile Computing* 2007; **6**: 2–11.
27. Wu Y, Fahmy S, Shroff NB. On the construction of a maximum-lifetime data gathering tree in sensor networks: NP-completeness and approximation algorithm. In *Proceedings of INFOCOM*, IEEE, 2008; 1013–1021.
28. Wu Y, Mao Z, Fahmy S, Shroff NB. Constructing maximum-lifetime data gathering forests in sensor networks. *IEEE/ACM Transactions on Networking* 2010; **18**: 1571–1584.
29. Xu X, Liang W. Monitoring quality optimization in wireless sensor networks with a mobile sink. In *Proceedings of MSWiM*, ACM, 2011; 79–84.
30. Xu Z, Liang W, Xu Y. Network lifetime maximization in delay-tolerant sensor networks with a mobile sink. In *Proceedings of DCOSS*, IEEE, 2012; 9–16.
31. Liang W, Luo J, Xu X. Network lifetime maximization for time-sensitive data gathering in wireless sensor networks with a mobile sink. *Journal of the Wireless Communications and Mobile Computing* 2013; **13**: 1263–1280.
32. Papadimitriou CH. The complexity of the capacitated tree problem. *Networks* 1978; **8**: 217–230.

33. Jothi R, Raghavachari B. Approximation algorithms for the capacitated minimum spanning tree problem and its variants in network design. *ACM Transactions on Algorithms* 2005; **1**: 265–282.
34. Altinkemer K, Gavish B. Heuristics with constant error guarantees for the design of tree networks. *Management Science* 1988; **34**: 331–341.
35. Liang W, Schwitzer P, Xu Z. Approximation algorithms for capacitated minimum spanning forest problems in wireless sensor networks with a mobile sink. *IEEE Transactions on Computer* 2013; **62**: 1932–1944.
36. Marta M, Cardei M. Improved sensor network lifetime with multiple mobile sinks. *Journal of Pervasive and Mobile Computing* 2009; **5**: 542–555.
37. Shan F, Liang W, Luo J, Shen X. Network lifetime maximization for time-sensitive data gathering in wireless sensor networks. *Computer Networks* 2013; **57**: 1063–1077.
38. Yang Y, Fonoage MI, Cardei M. Improving network lifetime with mobile wireless sensor networks. *Journal of Computer Communications* 2010; **33**: 409–419.
39. Texas Instrument. CC2420 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver, 2007. Available from: <http://www.ti.com/lit/ds/symlink/cc2420.pdf>.
40. Intel lab data. Available from: <http://db.csail.mit.edu/labdata/labdata.html>.
41. Cormen TH, Leiserson CE, Rivest RL, Stein C. *Introduction to Algorithms* (3rd edn). MIT Press: Cambridge, Massachusetts, 2009.
42. Vodafone, March 2012. (Available from: <http://www.vodafone.com.au/personal/plans/infinite/index.htm>).

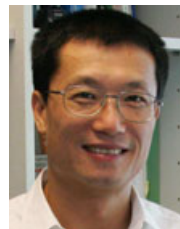
AUTHORS' BIOGRAPHIES



Xu Xu received her ME degree from the Institute of Computing Technology, Chinese Academy of Sciences in 2009 and the BS degree from China Agricultural University in 2006, both in Computer Science. She is currently doing her PhD in the Research School of Computer Science at the Australian National University. Her research interests include sink mobility in wireless sensor networks, routing protocol design for wireless ad hoc and sensor networks, data gathering in wireless sensor networks, and optimization problems.



Weifa Liang (M'99–SM'01) received the PhD degree from the Australian National University in 1998, the ME degree from the University of Science and Technology of China in 1989, and the BSc degree from Wuhan University, China, in 1984, all in computer science. He is currently an Associate Professor in the Research School of Computer Science at the Australian National University. His research interests include design and analysis of energy-efficient routing protocols for wireless ad hoc and sensor networks, cloud computing, graph databases, design and analysis of parallel and distributed algorithms, approximation algorithms, and graph theory. He is a senior member of the IEEE.



Xiaohua Jia received his BSc (1984) and MEng (1987) from the University of Science and Technology of China and DSc (1991) in Information Science from the University of Tokyo. He is currently the Chair Professor with the Department of Computer Science at City University of Hong Kong. His research interests include cloud computing and distributed systems, computer networks, wireless sensor networks, and mobile wireless networks. Prof. Jia is an editor of *IEEE Transactions on Parallel and Distributed Systems* (2006–2009), *Wireless Networks*, *Journal of World Wide Web*, *Journal of Combinatorial Optimization*, and so on. He is the General Chair of ACM MobiHoc 2008, TPC Co-Chair of IEEE MASS 2009, Area-Chair of IEEE INFOCOM 2010, TPC Co-Chair of IEEE GlobeCom 2010—Ad Hoc and Sensor Networking Symp—and Panel Co-Chair of IEEE INFOCOM 2011.



Wenzheng Xu received the BSc and ME degrees, both in computer science, from Sun Yat-sen University, Guangzhou, China, in 2008 and 2010, respectively. He is currently a PhD candidate at Sun Yat-sen University and a visiting student at the Australian National University. His research interests include routing algorithms and protocols design for wireless ad hoc and sensor networks, approximation algorithms, combinatorial optimization, and graph theory.