RESEARCH ARTICLE

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

Xu Xu1, Weifa Liang1*, Xiaohua Jia2 and Wenzheng Xu1,3

1 Research School of Computer Science, Australian National University, Canberra ACT 0200, Australia
2 Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong
3 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 (CMST) [33] seem to be closely related to the maximizing network throughput with minimal service cost problem. CMST and CMST 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 CMST 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.
In this paper, we consider large-scale remote environmental monitoring (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.

2.2. 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 \( \tau \) 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 \( \tau \). 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 \( e_1, e_2, \ldots, e_L \) 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 \( p_{v,g_i} = \prod_{l=1}^{L} p_{e_l} \), where \( p_{e_l} \) is the reliability of link \( e_l \in E \) with \( 0 < p_{e_l} \leq 1 \).

Let \( L(g_i) \) be the amount of data received by gateway \( g_i \) through tree \( T_i \) within a period of \( \tau \), which is also referred to as the load of gateway \( g_i \), then \( L(g_i) = \sum_{v \in V(T_i)} \left\lfloor p_{v,g_i} \cdot \tau \cdot r \right\rfloor \). 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 \( \tau \) 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 \( \tau \), 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
In this paper, we consider large-scale remote environmental monitoring.

Let \( c_f \) be the fixed cost rate 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 \( \tau \), then

\[
C = K \cdot C_f + \sum_{i=1}^{K} \max\{0, (L(g_i) - Q) \cdot C_p\} \tag{2}
\]

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 \( \tau \), 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 in a sensor network \( G \) 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 \( C = K \cdot C_f + \sum_{i=1}^{K} \max\{0, \sum_{v \in V(T_i)} (p_{v, g_i} \cdot r - Q) \cdot c_p\} \) is minimized.

### 2.4. Approximation ratio

Denote by \( \text{Approx} \) and \( \text{OPT} \) the costs of the approximate solution delivered by an approximation algorithm \( A \) and the optimal solution to an optimization problem, respectively. The approximation ratio of algorithm \( A \) is \( \xi \) 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 \( S = \{a_1, a_2, \ldots, a_n\} \), the subset sum problem is to partition the set into two disjoint subsets \( S_1 \) and \( S_2 \) such that \( \sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i \). The decision version of an instance of the subset sum problem is to determine whether there is a set partition \( S' \) and \( S'' \) such that \( \sum_{i \in S'} a_i = \sum_{i \in S''} a_i = A \), where \( A = \sum_{i \in S} a_i / 2 \). 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 \( v_j \in V \) 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 \( v_j \) and either gateway is \( p_j = a_i / \tau \), 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 \( v_j \in V \) to either gateway during this period is \( p_j \cdot r \cdot \tau \). Clearly, the maximum network throughput of this network is \( \sum_{v_j \in V} (p_j \cdot r \cdot \tau) + \sum_{v_j \in V} (p_j \cdot r \cdot \tau) = \sum_{i \in S} a_i = 2A \). 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 \( V_1, V_2, \ldots, V_h \) such that \( V_l = GW, \cup_{l=2}^{h} V_l = V \) and \( V_l \cap V_j = \emptyset \) if \( i \neq j \) and \( 1 \leq i, j \leq h \). 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 following the previous lemma.

**Lemma 1.** Given an unreliable sensor network \( G(V \cup GW, E) \) with identical link reliability \( p \), \( 0 < p < 1 \), let \( h(v) \) be the number of minimum 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 \( D_{\text{max}} = \sum_{l=2}^{h} \sum_{v \in V_l} p^{l-1}.(\tau \cdot r) = \sum_{l=2}^{h} (|V_l| \cdot p^{l-1} \cdot \tau \cdot r) \).

**Proof.**

(1) Because the reliability of each link is \( p \), let \( P_l \) be the most reliable path from a sensor \( v \in V \) to a gateway and \( |P_l| = l - 1 \), then, the reliability of path \( P_l \) is \( p^{l-1} \). Maximizing the end-to-end reliability of path \( P_l \), \( p^{l-1} \) 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 \( r \) is identical, which is \( d_{l} = d_{\text{max}} = p^{l-1}.(\tau \cdot r) \). Maximizing \( d_{l} \) is equivalent to minimizing \( l \), while \( l \geq h(v) \), \( l \) is equal to \( h(v) \). Thus, \( D_{\text{max}} = \sum_{v \in V} d_{\text{max}} = \sum_{l=2}^{h} (|V_l| \cdot p^{l-1} \cdot \tau \cdot r) \).

From Lemma 1, it can be seen that the forest consisting of routing trees rooted at the gateways, 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 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 load 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 \( T_l \) be the forest spanning the nodes in the first \( l \) layers. Initially, \( T_1 \) spans all gateways \( g_1, g_2, \ldots, g_K \in GW \), 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 \( T_l \). We now expand the forest \( T_l \) to forest \( T_{l+1} \) by including the nodes in \( V_{l+1} \) with the objective of minimizing the maximum load among the \( K \) gateways in the resultant forest \( T_{l+1} \).

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 \( V_f = \{s, t\} \cup \{g_1, g_2, \ldots, g_K\} \cup V_{l+1} \), and \( s \) and \( t \) are the source node and destination node in \( G_f \), and \( c \) is the link capacity function. Let \( V_{l+1} = \{x_1, x_2, \ldots, x_{|V_{l+1}|}\} \). 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 \( B_i = B - \frac{L(g_i)}{d_{i+1}} \), where \( d_{i+1} = p^l \cdot \tau \cdot r \) 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 \( |f| = \sum_{e \in E_f} f(e, v) - \sum_{e \in E_f} f(v, e) \). 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 \( \frac{\max_{1 \leq k \leq K} \{L(g_k)\}}{d_{i+1}} \) and \( \frac{\max_{1 \leq k \leq K} \{L(g_k)\} + |V_{l+1}|/d_{i+1}}{d_{i+1}} \). 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 \( |f| = \sum_{e \in E_f} f(e, v) \) be the total amount of flow entering gateway \( g_i \), then the corresponding nodes in \( V_{l+1} \) will join tree \( T_l \), and the load \( L(g_i) \) of \( T_l \) is updated to \( L(g_i) + |f| \cdot d_{i+1} \) for all \( i \) with \( 1 \leq i \leq K \). Forest \( T_{l+1} \) is then expanded to forest \( T_{l+1} \). The proposed algorithm for constructing a forest, Load Balanced Forest, is described in Algorithm 2.
In this paper, we consider large-scale remote environmental monitoring application with mobile sensor networks. Since the network load among the gateways affects the service quality, no dynamic readjustment of the load is desired. Figure 1 shows the flows and trees constructed by Algorithm 2. The load among the gateways is initially balanced, and the network throughput is maximized. The load adjustment can be achieved through a series of edge swapping between two layers. Specifically, for all layers $i$ with $1 \leq i \leq h-1$, swapping any non-tree edge with a tree edge does not change the network throughput. Otherwise, swapping a non-tree edge with a tree edge changes the network throughput.

**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 $1 \leq i \leq K$, $d_{i+1} = p^i \cdot r$

**Output:** The smallest value of $B$ and the flow $f$

1. $\text{lower_bound} \leftarrow \frac{\max_{g \in G(V_f)} f(g)}{d_{i+1}}$
2. $\text{upper_bound} \leftarrow \frac{\max_{g \in G(V_f)} L(g)}{d_{i+1}} + |V_{i+1}|$
3. while (lower_bound $\neq$ upper_bound) do
   4. $B \leftarrow \lfloor \text{lower_bound + upper_bound} \rfloor$
   5. Assign each edge $(g, t)$ in $G_f$ the capacity of $B_i = B - \frac{L(g)}{d_{i+1}}$
   6. Find a maximum flow $f$ in $G_f$ from $s$ to $t$
   7. if $|f| = |V_{i+1}|$ then
      8. $f$ flow $f$ saturates all outgoing edges from $s$
   9. $\text{upper_bound} \leftarrow B$
   10. else
      11. $\text{lower_bound} \leftarrow B + 1$
   12. end if
13. end while
14. $B \leftarrow \text{upper_bound}$
15. return $B$ and $f$

**Algorithm 2** Load_Balanced_Forest

**Input:** $G(V \cup GW, E)$, $h$ disjoint subsets of nodes $V_1, V_2, \ldots, V_h$, 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 $1 \leq i \leq K$

1. $L(g_i) \leftarrow 0$ for all $i$ with $1 \leq i \leq K$; /* The initial load of each gateway is 0 */
2. $F_i \leftarrow \{T_i = \{g_i\} \mid 1 \leq i \leq K\}$
3. for $i \leftarrow 1$ to $h - 1$ do
4. Calculate the network throughput contribution of each node in layer $l + 1$, $d_{i+1} \leftarrow p^i \cdot (r \cdot r)$;
5. /* Perform the lay expansion from layer $l$ to $l + 1$ by including all nodes in $V_{i+1}$ */
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 $g_i$, $L(g_i) \leftarrow L(g_i) + |f| \cdot d_{i+1}$ for each $1 \leq i \leq K$, where $|f| = \sum_{e \in V_{i+1}} f(e, g_i)$;
9. Expand $F_i$ to $F_{i+1}$;
10. end for
11. return $F_n$ 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 $F$, 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 $F_i$ 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 $F$ 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. \qed

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 $E_{l+1,l} = (V_{l+1} \times V_l) \setminus V$ 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 $F$, 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 $(1 + \frac{C_f}{C_p})$ times of the minimal service cost. The time complexity of the proposed algorithm is $O(|V||E|^2)$, assuming that $|GW| << |V|$, where $C_f$ is the cost of a fixed data quota $Q$ for each gateway during a period of $\tau$, $C_p = \frac{C_f}{p^\tau}$ and $C_p$ is the penalty rate of every exceeding megabyte of data over the quota $Q$.

Proof. See Appendix B. \qed

Notice that the approximation ratio $1 + \frac{C_f}{C_p}$ of the service cost is usually a small constant. For example, one typical data plan in Vodafone is a fixed cost $C_f = $5/GB per month and the penalty rate of...
work throughput via gateway $w$.

If there are multiple edges between nodes not in the trees but are one-hop neighbors of the gateway nodes. We now construct the maximum network throughput, but the service cost is not optimized. We then devise approximation algorithms by incorporating the service cost into consideration.

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 $T_1, T_2, \ldots, T_K$ 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 $\cup_{i \in [K]} V(T_i)$, that is, $V' = \{v \mid (u, v) \in E, u \in \cup_{i \in [K]} V(T_i), v \notin \cup_{i \in [K]} V(T_i)\}$. 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 $g_i$, $p_{uv} = p_{u,v} \cdot p_{v,g_i}$, is maximized. Thus, the contribution of node $v$ to the network throughput via gateway $g_i$ for the period of $\tau$ is $d_{v,g_i}^\tau = p_{uv} \cdot (1 - r)$. 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 $d_{vmax} = \max_{1 \leq i \leq K} (d_{v,g_i}^\tau)$. 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, $d_{vmax}' = \max_{v \in V'} (d_{vmax})$. 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. □

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 $F = \{T_1, T_2, \ldots, T_K\}$ be the forest, which is a feasible solution to the problem. We then perform dynamic load readjustment to the trees in $F$ 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 $\Delta d$ 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 $\Delta d = (p_{u,v} \cdot p_{v,g_i} - p_{u',v'} \cdot p_{v',g_j}) \cdot L(u) > 0$. The swapping increases the network throughput, and accordingly, the service cost for $\Delta d$ will be paid.

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

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) $\Delta d = (p_{u,v} \cdot p_{v,g_i} - p_{u',v'} \cdot p_{v',g_j}) \cdot L(u) > 0$ and (ii) either $L'(g_i) = L(g_i) - p_{u,v} \cdot p_{v,g_i} \cdot L(u) > Q$ or $Q - L(g_j) > Q - L'(g_j)$. 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 $L'(g_j) > Q$, the extra increase on the network throughput will be paid with a less service cost, because only the amount of data $p_{u,v} \cdot p_{v,g_i} \cdot L(u) - (Q - L(g_i))$ incurs an extra charge if
is the maximum number of hops from a sensor to its nearest
no more than
Proof.

4: Balance the load among the gateways by readjusting
3: Construct a forest $F$ of load-balanced trees rooted at
gateways such that the maximum load among the gate-
ways is minimized, by calling Algorithm 2, assuming
each link having identical reliability $p$;
1: Let $F ← Φ$; $r^*$ The forest of routing trees $*$!
2: Partition the nodes in $V ∪ GW$ into $h$ disjoint subsets
$V_1, V_2, \ldots, V_h$, using BFS traversal;

Algorithm 5 Appro

Input: $G(V ∪ GW, E)$, monitoring period of $τ$, the data
plan, and link reliability $p^*: E ↦ [p, p(1 + δ)]$
Output: The network throughput $D(τ)$ and the cost $C$

1: $F ← Φ$; $r^*$
2: $P_i(p, τ) = \sum_{j=1}^{p} \log_j^{p}$
3: $\sum_{i=1}^{p} \log_j^{p}$
4: $\sum_{i=1}^{p} \log_j^{p}$
5: $\sum_{i=1}^{p} \log_j^{p}$

Theorem 4. Given an unreliable WSN $G(V ∪ GW, E)$
with link reliability in $[p, (1 + δ)p]$ and $0 \leq δ \leq \frac{1}{p} − 1$, there
is an approximation algorithm Appro for the maximiz-
ing network throughput with minimal service cost problem,
which delivers a solution with no less than $1 + \frac{C_p}{C_f} \rho$ times
of the maximum network throughput while the service cost is
no more than $1 + \frac{C_p}{C_f} \rho$ 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.

Note that the analytical estimation on the approxima-
tion ratio of algorithm Appro is very conservative, which
be confirmed from the later empirical results. The
actual network throughput is no less than 62% of the max-
imum 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 $0 \leq p_{min} \leq p_{max} \leq 1$.
We classify the links in $E$ of $G = (V ∪ GW, E)$ into
$[\log \frac{p_{min}}{p_{max}}]$ different groups by their reliabilities. A sub-
graph $G_i = (V ∪ GW, E_i)$ of $G$ is induced by the links in
group $i$, where $E_i = \{e | e ∈ E$ and $p_e \geq p_i\}$ and
$p_i = \min \{\log i \cdot p_{min} - \log p_{max} | 0 ≤ i ≤ \log \frac{p_{min}}{p_{max}}\}$.
The improved algorithm proceeds as follows. Starting from
$i = \lfloor \log \frac{p_{min}}{p_{max}} \rfloor$, for each $i$, a graph $G_{0,i} = (V ∪ GW ∪$
$\{s_0\}, E_i ∪ \{(s_0, g_j) | g_j ∈ GW\})$ is constructed by adding
a virtual sink $s_0$ and the edges between $s_0$ and every
gateway in GW. If $G_{0,i}$ 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
$G_{0,0}$ is connected, assuming that $G$ is connected. Then, a
solution is found by applying algorithm Appro on graph
$G_{0,0}$. We refer to this improved algorithm as algorithm
Impro_Appro.

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

Proof. See Appendix E.

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the pro-
posed algorithms in terms of network throughput and the ser-
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 $1,000 \times 1,000$ m$^2$ 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 gate-
ways are deployed as follows. The monitoring region is
divided into roughly equal $K$ subregions, and one gate-
way is randomly deployed in each of the subregions. For
simplicity, we assume that the transmission range of gate-
ways is identical to that of sensors. In our experimental
evaluation, the following three different data plans for 1-
month monitoring period (i.e., $τ = 30$ days $× 24$ h $×$
3600 s), provided by Vodafone [42], will be examined: (I)
$Q = 2$ GB and $C_f = 19$; (II) $Q = 4$ GB and $C_f = 29$;
In this paper, we consider large-scale remote environmental monitoring and (III) $Q = 10$ GB and $C_f = $39. Each data plan has the same penalty rate $c_p = $0.02/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 $D^{(t)}$ as $C_f \cdot K + c_p \cdot \max(D^{(t)} - K \cdot Q, 0)$, where $\max(D^{(t)} - K \cdot Q, 0)$ 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 includes the fixed costs of gateways.

![Figure 2](image-url)

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](image-url)

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.
In this paper, we consider large-scale remote environmental monitoring.

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, 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 $K = 6$, $n = 2000$, 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...
In this paper, we consider large-scale remote environmental monitoring.

Figure 6. Impact of $K$ on the network performance when $n = 2000$, $p_e \in [0.1, 1]$ 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 $\text{Appro}$ and $\text{Impro\_Appro}$ is close to the maximum one. When the value of $\delta$ increases, the gap of the service cost between $\text{Max\_Throughput}$ 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. □

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 \((1 + \frac{C_{f}}{C_{p}})\) times of the minimal one. Assuming that in the optimal solution, the minimum and maximum loads among gateways are \(L_{\text{min}}\) and \(L_{\text{max}}\), respectively, and let \(C_{\text{min}}\) be the minimal cost. Let \(L'_{\text{min}}\) and \(L'_{\text{max}}\) be the minimum and maximum loads among the gateways in the forest \(F\) delivered by algorithm Uniform_Link, and let \(C\) be the service cost. We analyze the relationship between \(C\) and \(C_{\text{min}}\) by the following three cases.

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

Case 2: \(L_{\text{min}} \geq Q\), \(L_{\text{max}} \geq Q\), and \(C_{\text{min}} \geq K \cdot C_{f}\). If \(L_{\text{min}} \geq Q\), then \(C = C_{\text{min}}\). Otherwise (\(L'_{\text{min}} < Q\)), the loads of some gateways in \(F\) 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 \(C \leq (K - 1) \cdot Q \cdot C_{p} + C_{\text{min}} = K \cdot Q \cdot C_{p} + C_{\text{min}} = C_{\text{min}} \cdot \left(1 + \frac{C_{f}}{C_{p}}\right)\), following the similar discussion in case 1, omitted.

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

Thus, the service cost of the solution is no more than \(1 + \frac{C_{f}}{C_{p}}\) 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 \(O(|E_{l_{i+1}}| \cdot (|V| + |K|)) = O(|E_{l_{i+1}}| \cdot |V|)\) time. Because it takes \(O(|V| + |E|)\) time to find the minimum value of the maximum capacity \(B\) for the flow network, the time spent on layer expansion is \(O(|E_{l_{i+1}}| \cdot |V| \log |V_{l_{i+1}}|)\), and the time for finding the forest is \(\sum_{l=1}^{h-1} O(|E_{l=1}| \cdot |V| \log |V_{l=1}|) = O(|E| \cdot |V| \log |V|)\) 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 \(l, l + 1\) is no more than \(|E_{l_{i+1}}|^{2}\), 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 \(\sum_{l=1}^{h-1} O(|E_{l_{i+1}}| \cdot |V| \log |V_{l_{i+1}}|) + \sum_{l=1}^{h-1} O(|E_{l_{i+1}}|^{2} \cdot |V|) = O(|V| \cdot |E|^{2})\).

APPENDIX C: PROOF OF THEOREM 3

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

We now show that the accumulative volume of data collected by all gateways in the proposed algorithm is exactly \(D_{\text{max}}^{(\tau)}\) by contradiction. Let \(v_{1}, v_{2}, \ldots, v_{i}\) 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 \(j \leq i \leq 1 < |V|\). Assume that \(v_{j}\) 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_{j}\ to one of its neighbors \(u \in N_{T}(v_{i})\ will result in the maximum network throughput contribution by \(v_{j}\). That is, \(p_{\text{max}}^{u'} = \max \{p_{v_{i}, u} \cdot p_{v_{i}, u}' \cdot p_{\text{max}}^{v} \mid u \in N(v_{i}) \setminus N_{T}(v_{i}), v' \in \mathcal{K}_{v_{i}}(V(T_i))\}. In that case, the maximum network throughput contributions made by nodes \(u\ and \(v_{j}\ are a_{\text{max}}^{v_{j}} = p_{v_{i}, u} \cdot p_{\text{max}}^{v} \cdot (\tau \cdot r)\ and \(d_{\text{max}}^{w} = p_{v_{i}, u} \cdot p_{v_{i}, u}' \cdot p_{\text{max}}^{v} \cdot (\tau \cdot r) = p_{v_{i}, w} \cdot d_{\text{max}}^{w} \cdot \frac{p_{v_{i}, u}}{p_{v_{i}, u}'}, \) respectively. Following the forest construction, \(u\ should have already been added to the forest prior to \(v_{j}\, which contradicts the assumption that
u ∈ N(vj) \ N7(vj) is not yet in the forest when adding vj to the forest. Thus, every node v ∈ V is added to the forest via the most reliable path and has the maximum network throughput contribution pmax · τ · r. The accumulative volume of data collected by all gateways is Dmax.

**APPENDIX D: PROOF OF THEOREM 4**

**Proof.** Let P(v, gj) be the most reliable path consisting of l links in G(V ∪ GW, E) from node v ∈ V to one of the K gateways, for example, gateway gj, then the maximum volume of data that can be collected at gj for a period of τ is dmax = \( p(v) \cdot (\tau \cdot r) \), with \( h(v) \) 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 from v to another gateway, if each link in P is the most reliable one with reliability \( 1 + \delta \), then the volume of received data at the gateway is \( d_U = p(v) \cdot (\tau \cdot r) \). Let node v be in a tree rooted at gateway gj and \( d_U \) be the volume of data generated from v and received at gj, following the proposed algorithm, we have \( \frac{d_U}{d_{max}} \geq \frac{d_U}{d_{max}} \geq \frac{d_U}{d_{max}} = \frac{1}{1 + \delta} \).

Let \( \beta = \frac{1}{1 + \delta} \), then, \( d_U \geq d_{max} \geq d_{max} \cdot \beta \). Let \( \beta = \frac{1}{1 + \delta} \). The approximation ratio of the network throughput Approx by the improved algorithm to the optimal one OPT, \( \xi \), is

\[
\xi = \frac{\text{Approx}}{\text{OPT}} \geq \frac{\sum_{i=2}^{h} Y_i}{\sum_{i=2}^{h} X_i} \geq \frac{\sum_{i=2}^{h} \sum_{v \in V_i} d_U}{\sum_{i=2}^{h} \sum_{v \in V_i} d_{max}} \geq \frac{\sum_{i=2}^{h} \sum_{v \in V_i} \beta d_U}{\sum_{i=2}^{h} \sum_{v \in V_i} d_{max}} = \beta \frac{1}{1 + \delta}.
\]

(D1)

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

**APPENDIX E: PROOF OF THEOREM 5**

**Proof.** As the set of links in \( G_{s,j} \) is a subset of the set of links in \( G_{s,i} \) when \( j > i \), graph \( G_{s,j} \) is a subgraph of \( G_{s,i} \). If \( G_{s,j} \) is not connected, graph \( G_{s,j} \) is not connected either, with \( 0 < i < j \leq \frac{\log p_{max}}{p_{min}} \). For any disconnected graph \( G_{s,j} \), 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 sj 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 \( G_{s,j} \) spanning all nodes in V, and no solution can be obtained. There is an \( i \) with \( 0 \leq i \leq \frac{\log p_{max}}{p_{min}} \) such that \( G_{s,i} \) is connected, as \( G \) is connected. And an approximate solution is obtained whose throughput is with the approximation ratio \( \frac{1}{1 + \delta} \), where \( \delta = \frac{p_{max}}{p_{min}} - 1 \leq \delta = \frac{p_{max}}{p_{min}} - 1 \).

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

**REFERENCES**


In this paper, we consider large-scale remote environmental monitoring


In this paper, we consider large-scale remote environmental monitoring.


**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.