Noname manuscript No.
(will be inserted by the editor)

Improved Structures for Data Collection in Static and Mobile Wireless Sensor Networks

Jon Crowcroft · **Michael Segal** · **Liron Levin**

the date of receipt and acceptance should be inserted later

Abstract In this paper we consider the problem of efficient data gathering in sensor networks for arbitrary sensor node deployments. The efficiency of the solution is measured by a number of criteria: total energy consumption, total transport capacity, latency and quality of the transmissions. We present a number of different constructions with various tradeoffs between aforementioned parameters. We provide theoretical performance analysis for our approaches, present their distributed implementation and discuss the different aspects of using each. We show that in many cases our output-sensitive approximation solution performs better than the currently known best results for sensor networks. We also consider our problem under the mobile sensor nodes environment, when the sensors have no information about each other. The only information a single sensor holds is its current location and future mobility plan. Our simulation results validate the theoretical findings.

Keywords wireless sensor networks · optimization algorithms · approximation guarantees · data gathering

Preliminary version of this paper has been accepted to IEEE INFOCOM 2014 conference.

J. Crowcroft
Computer Laboratory, Cambridge University, Cambridge, United Kingdom, E-mail: Jon.Crowcroft@cl.cam.ac.uk.

M. Segal
Department of Communication Systems Engineering, Ben-Gurion University of the Negev, E-mail: segal@cse.bgu.ac.il.

L. Levin
Department of Communication Systems Engineering, Ben-Gurion University of the Negev, E-mail: levinlir@gmail.com.

1 Introduction

A wireless sensor network (WSN) consists of transceivers (nodes) that are located in the plane, communicate by radio and have no fixed communication backbone. The temporary physical topology of the network is determined by the relative disposition of the sensor nodes and the transmission range assignment of each of the nodes. The combination of these two factors produces a directed communication graph where the nodes correspond to the transceivers and the edges correspond to the communication links. The transmission range of each sensor node is determined by the assigned transmission power.

Our **main objective** in this paper is to construct efficient communication backbones for multi-hop data collection with aggregation in WSNs for arbitrary sensor node deployments, while measuring the efficiency based on the next four metrics.

- *Total energy consumption* is probably one of the most important parameters of a WSN as the sensor nodes are often deployed in areas where battery replacement is infeasible [8]. Wireless communication is a major contributer to the energy budget of a node. In this paper we focus on minimizing the total energy consumed by all nodes for communication purposes.
- The *transport capacity* metric represents the sum of rate-distance products over all the active links. It is measured in *bit-meters* and was first introduced by [19]. The idea behind this measure is to capture both the notion of the overall rate and distance that the information travels in a network.
- *Hop-diameter* is another important metric which reflects the depth of the data gathering tree, i.e. the maximum number of hops from any of the sensor nodes to the base station.

– The *stretch* of the paths connecting sensor nodes to the base station. The distance stretch factor has a strong effect on the quality of geographic routing protocols [17]. These protocols use greedy forwarding decisions based on the geographic progress towards the destination, thus having a low distance stretch factor in the underlying topology graph is essential for efficient and successful geographic routing.

We will follow two main approaches in our constructions. The first is based on so-called *balance nodes*, where the main motivation is to build data collection routes based on centrally located nodes in topologies which are already efficient in terms of some of the metrics. In our second approach we examine the addition of shortcut links to the currently constructed topology in order to allow the required tradeoff between studied criteria. Finally, we consider the problem when the nodes are allowed to change their positions and the topology will change with respect to the nodes movement. It makes sense then to indicate the time interval $[t_s, t_f]$, for which the induced topology is stable, where t_s and t_f are the start and finish times. Note that some other links might appear and disappear during the time interval, however the important links, which define the required topology remain unchanged.

In what follows we describe the model, discuss previous work, and describe our contribution. In Section 2 we show our balanced nodes based construction. The shortcut edges design is described in Section 3. Section 4 outlines a possible distributed implementation of our algorithms. Section 5 deals with the mobile version of the problem and possible extensions. We present our simulation results supporting our theoretical analysis in Section 6. Finally, we discuss some possible future research and conclude in Section 7.

1.1 System settings

A wireless sensor network (WSN) consists of n wireless sensor nodes, $S = \{s_1, \ldots, s_n\}$, distributed in some area A. These nodes perform monitoring tasks and periodically report to a base station r which is located somewhere within the area A (we consider different locations throughout the paper). During the report phase, the sensor nodes propagate a message to the base station through a *data collection tree*, $T_S = (S \cup \{r\}, E_S)$, rooted at r. We consider *data collection with aggregation*, where every node $s \in S$ forwards a single unit size *report message* to its parent. The message holds an accumulated information collected from a subtree of T_S rooted at s. An example of this scenario can be found in temperature monitoring systems for fire prevention, intrusion detection, seismic readings, etc.

We assume the use of *frame-based* MAC protocols which divide the time into frames, containing a fixed number of slots. The main difference from the classic TDMA is that instead of having one access point which controls transmission slot assignments, there is a localized distributed protocol mimicking the behavior of TDMA. The advantage of a frame-based (TDMA-like) approach compared to the traditional IEEE 802.11 (CSMA/CA) protocol for a Wireless LAN is that collisions do not occur, and that idle listening and overhearing can be drastically reduced. When scheduling communication links, that is, specifying the sender-receiver pair per slot, nodes only need to listen to those slots in which they are the intended receiver – eliminating all overhearing. When scheduling senders only, nodes must listen in to all occupied slots, but can still avoid most overhearing by shutting down the radio after the MAC (slot) header has been received. In both variants (link and sender-based scheduling) idle listening can be reduced to a simple check if the slot is used or not. Several MAC protocols were developed to adapt classical TDMA solutions which use access points to ad-hoc settings that have no infrastructure; these protocols employ a distributed slot-selection mechanism that self-organizes a multi-hop network into a conflict-free schedule (see [36, 43]).

Let $d(u,v)$ be the Euclidean distance between two sensor nodes $u, v \in S$. It is customary to estimate that the energy required to transmit from u to v is proportional to $d(u,v)^\mu$, where μ is the *path-loss coefficient*. In perfect conditions $\mu = 2$, however in more realistic settings (in presence of obstructions or noisy environment) it can have a value between 2 and 4 (see [35]). In this paper we assume $\mu = 2$ for simplicity. However, it is possible to extend our results for other values of μ which are greater than 2.

Let $E(T_S)$ be the **energy requirement** to execute a single report phase. Note that every sensor performs a single transmission, during which it sends a single message to its parent in T_S. Thus, the energy requirement is proportional to the sum of squares of lengths of the edges E_S. The focus of this paper is to study the asymptotic performance of data collection trees, thus we can express $E(T_S)$ as follows, $E(T_S) = \sum_{(u,v)\in E_S} d(u,v)^2$.

Minimizing the energy requirement is one of the primary optimization objectives when deploying a WSN due to the very low battery reserves at the sensor nodes and the high costs that are associated with replacing these batteries (if at all possible).

The second measure that we are interested in is **transport capacity**, $D(T_S)$, of the data collection tree T_S. As mentioned earlier, the main idea which stands behind this metric is to capture the spatial rate of the network, which is represented by the total rate over some distance. In our scenario, the rate on all links is fixed as all the nodes transmit an aggregated, unit-size message, to the parent in the collection tree and the schedule is conflict-free. Thus, to maximize the transport capacity we need to minimize the total distance traveled by information, which is the sum of lengths of all the links, $D(T_S) = \sum_{(u,v)\in E_S} d(u,v)$.

Another critical aspect in the design of a WSN is the **hop-diameter** of T_S. The data flows from the leafs of the delivery tree to the base station, where each intermediate node waits to receive the report messages from all its children, before sending its own report message to its parent. Therefore, the hop-diameter of T_S, denoted as $H(T_S)$, determines the delay of data collection.

Finally, we also aim to decrease the **stretch**, $\delta(T_S)$, of the paths in the data collection tree T_S connecting sensors with the base station (root) of T_S. Let $d_T(u,r)$ be the length of the unique path $p_T(u,r)$ connecting u with the root r in T_S. Then, the stretch factor of this path $\delta(p_T(u,r)$ is the ratio $\frac{d_T(u,r)}{d(u,r)}$. The stretch factor of the paths in T_S is defined as $\max_{u\in T_s} \delta(p_T(u,r))$.

Unfortunately, it is impossible to achieve optimal performance in all four measures at the same time. For example, minimizing the hop-diameter results in all nodes transmitting to the base station, which is disastrous in terms of transport capacity or energy consumption, whereas the best topology to minimize energy consumption[1] results in a relatively high hop-diameter. While we are interested in arbitrary deployments of sensor nodes, it was shown by Milyeykovsky et al. [33] that single-hop construction having optimal hop-diameter and stretch factor by choosing a centdian as a root node may lead to very bad transport capacity and energy consumption.

In theory it is impossible to devise a range assignment that will satisfy the topology requirement for a given period of time without being aware of the future location changes. Each node has its own *mobility plan*, which is composed of direction vectors, velocity, acceleration, and so on. Basch et al. [3, 4] proposed an elegant method to handle topology updates for mobile nodes. They proposed a framework to maintain an invariant of a set of moving objects in a discrete manner, called the *kinetic data structure* (KDS in short). They introduce the idea of keeping certificates as triggers for updates. When an object moves and a certificate fails, the consistency of the kinetic data structure is invalidated and an update is mandatory. Each failure of a certificate incurs a setup of up to a constant number of new certificates. Hence we are allowed to monitor the dynamics of a set of objects discretely and efficiently. The kinetic data structure requires that we know the mobility plan (a specification of the future motion) of all nodes, and that the trajectory of each disk can be described by some low-degree algebraic curve. These structures are extremely efficient for topology maintenance, but do not address the issue of energy efficiency or the construction of initial topology. The approach taken in this paper resembles the spirit of KDS. Additional results for topology control in mobile networks may be found in [18, 22, 23, 28].

1.2 Previous work

To the best of our knowledge, the only work which takes into account three of the aforementioned performance measures simultaneously (except of paths stretch) is by Milyeykovsky et al. [33]. They consider the random uniformly spreaded sensor nodes in unit size square in the plane and three dimensional space and present centroid-based hierarchical construction with hop-diameter of $O(\log n)$ that performs optimally (up to constant factor) in terms of energy and transport for three dimensional space and provides $O(\log n)$ approximation factor for energy consumption and asymptotically optimal transport capacity for planar case.

Below we discuss some other of the related work on data collection, energy efficiency, transport capacity, bounded-hop and bounded paths' stretch communication.

It has been proved in [39, 44] that using the minimum spanning tree for data collection (gathering) with aggregation achieves optimal solution in terms of **total energy consumption**. Elkin et al. [15] proposed the solution for the broadcast tree construction (which is easily deformable into the data collection tree) such that the total energy consumption is of factor ρ from optimal bound (which is proportional to the weight of minimal spanning tree for the set of nodes where the weight of edge is defined as the squared Euclidean distance between the nodes) and the hop-diameter is $n/\rho + \log\rho$, for any chosen integer parameter $\rho, 1 \le \rho \le n$. Their solution [15] is based on Hamiltonian cycle construction of weight proportional to the weight of minimal spanning tree for squared distances with a consequent design of the hierarchical tree using this cycle. For more details regarding energy consumption in data gathering problem, we refer the reader to a recent survey by Ramanan et al. [29], and a paper by Li et al. [31] which cover a diverse set of data gathering algorithms in ad-hoc networks.

The notion of **transport capacity** was introduced by Gupta and Kumar in [19]. They showed that for any layout of n wireless nodes in an area of size A, with each node being able to transmit W bits per second to a fixed range, the overall transport capacity is at most $(W\sqrt{An})$ bit-meters per second under both interference models (protocol and physical). In [25] the authors derive upper bounds on the transport capacity as a function of the geographic location of the nodes. It has also been shown that the scaling of transport capacity depends, among other factors, on channel attenuation and path loss [45, 46, 47].

Some communication backbones with **bounded hop distances** and/or **bounded paths stretch** between participating

[1] The Euclidean minimum spanning tree minimizes the energy consumption, see [39, 44].

nodes have also been studied. For the linear layout of nodes and an upper bound on hop-distance, Kirousis et al. [27] developed an optimal power assignment algorithm for strong connectivity in $O(n^4)$ time. In the Euclidean case, [11] obtains constant ratio algorithms for the bounded-hop vertex connectivity for well spread instances. Beier et al. [5] proposed an optimal algorithm to find a bounded-hop minimum energy path between pairs of nodes. In [7] the authors obtain bicriteria approximation algorithms for connectivity and broadcast while minimizing the hop-diameter and energy consumption. Funke and Laue [16] provide a PTAS for the h-broadcast algorithm in time linear in n. Additional results for bounded range assignments can be found in [10, 12, 40, 30]. Li et al. [32] consider a problem of constructing energy-efficient broadcast tree with bounded stretch paths. However the approximation factor shown in [32] for the total energy consumption can be as worse as $\Omega(n^2\Delta)$, where Δ stands for the degree of the obtained tree. Segal and Shpungin [38, 41] consider several spanner (opposite to data collection tree) constructions under the total energy consumption and hop-diameter, but ignoring the transport capacity measure.

Not too much has been done with respect to the problem in mobile environment. Yun et al. [48] propose an algorithm for maximizing the lifetime of a wireless sensor network when there is (only) a mobile sink and the underlying application can tolerate some amount of delay in delivering the data to the sink.

1.3 Our results

We study the power assignment problem in wireless sensor networks so as to produce data collection tree while optimizing several properties of the construction: energy cost, transport capacity, hop-diameter and stretch of the paths. Our constructions work for arbitrary sensors deployments. Let us denote by OPT_e the minimal possible value that can obtained by $E(T'_S)$ for some T'_S and denote by OPT_c the minimal possible value that can be obtained by $D(T''_S)$ for some T''_S. Let T' be the minimal spanning tree for $S \cup \{r\}$, $w(T')$ be the weight of T', $w(e^*(T'))$ be the weight of the heaviest edge in T'. Then our first construction for T_S guarantees the following bounds for given hop-diameter parameter $h, 1 \le h \le n$: $E(T_S) = O(\frac{n^2h^2 \cdot w^2(e^*(T'))}{w^2(T')} \cdot OPT_e)$, $D(T_S) = O((1+\frac{2}{\alpha-1}) \cdot \frac{n \cdot d(e^*(T'))}{d(T')} \cdot OPT_c)$, $h(T_S) = h$, $\delta(T_S) = \alpha$, for $\alpha > 1$. The second construction for T_S produces the following results: $E(T_S) = O(n^{2-2\varepsilon} \cdot h \cdot OPT_e)$, $D(T_S) = O(n^{2-\varepsilon} \cdot h \cdot OPT_c)$, $h(T_S) = h$, $\delta(T_S) = n^{\varepsilon} \cdot h^2$. Note that all the upper bounds derived in this article are compared with the best possible corresponding lower bound for the optimal solution. Thus, the produced results serve as approximation guarantees for the considered problems. We also show that our bounds are held for mobile setting up to constant factor.

2 Multi-hop collection for any deployment: transport, hop and stretch

In this section we propose hierarchical structure which has guaranteed bounds for transport capacity and hop-diameter in the scenario where the nodes S are placed anywhere in the area A.

We start by describing the hierarchical structure obtained by *balanced tree partitioning* and then show how it can be used to produce an efficient communication backbone.

2.1 Balanced tree partitioning

We begin with some notation. Given a tree $T = \{V,E\}$, denote the set of nodes in the subtree of T, T', by $V(T')$ and the set of the edges in T', by $E(T')$. Denote the induced tree on the set $V' \subseteq V$ of nodes by $T_{V'}$. Next we provide the definition of balanced tree partition followed by a proof that it exists for any tree T.

Definition 21 (Balanced Tree Partition)**.** *Given a tree $T = \{V,E\}$, a partition into two connected subtrees of T, (T_1,T_2), where $T_1 = \{V(T_1), E(T_1)\}$ and $T_2 = \{V(T_2), E(T_2)\}$ is called a* balanced tree partition *iff the following conditions hold:*

- $V(T_1) \cup V(T_2) = V(T)$.
- *There exists $v \in V$ such that $V(T_1) \cap V(T_2) = \{v\}$.*
- $|V_2| \le |V_1| \le 2|V_2|$.

We refer to v in the second condition above as the balance *node.*

Theorem 22 (Balanced Tree Partitioning)**.** *For any tree T there always exists a balanced tree partition.*

Proof. We prove the theorem by contradiction. Suppose that for every node $v \in V$, the partition we obtain does not satisfy the claim of the theorem, that is for every $v \in V$, and for any partition of T into T_1 and T_2 such that $V(T_1) \cup V(T_2) = V(T)$, $V(T_1) \cap V(T_2) = \{v\}$, and $|V(T_1)| > 2|V(T_2)|$. Let v' be the node in V such that the ratio $\frac{|V(T_1)|}{|V(T_2)|}$ is minimized over all possible choices of nodes. Let $k = |V(T_1)|$ and $m = |V(T_2)|$.

Let us consider a set $N_{T_1}(v')$ of the neighboring nodes of v' that belong to T_1. Since we assumed that v' gives us the minimal ratio $\frac{|V(T_1)|}{|V(T_2)|}$ over all possible choices of nodes, it means that a different partition (T'_1, T'_2) which can be obtained from partition (T_1,T_2) by moving some node $w \in N_{T_1}(v')$ into T_2 and possibly some other nodes from T_1 which are connected to w as well, should produce a larger ratio.

Since we have that $|V(T'_1)| < |V(T_1)|$ and $|V(T'_2)| > |V(T_2)|$ the larger ratio can be produced only when $|V(T'_2)| > 2|V(T'_1)|$, because otherwise the ratio $\frac{|V(T'_1)|}{|V(T'_2)|}$ will be less than $\frac{|V(T_1)|}{|V(T_2)|}$ which contradicts our assumption. Moreover, if $|V(T'_2)| > 2|V(T'_1)|$ we can always bound the number of nodes added to T_2 (denoted by a) to be at most $|V(T_1)/2|$. This is because $|N_{T_1}(v')|$ is at least 2 - in this case we can always add to T_2 the smaller subtree of T_1 rooted at some node of $N_{T_1}(v')$. The reason for the fact that $|N_{T_1}(v')| > 1$ comes from our assumption that the ratio $\frac{|V(T_1)|}{|V(T_2)|}$ is minimized over all possible choices of nodes. If $|N_{T_1}(v')| = 1$ we have that $\frac{|V(T_2 \cup N_{T_1}(v'))|}{|V(T_1) \setminus N_{T_1}(v'))|}$ will be less than $\frac{|V(T_1)|}{|V(T_2)|}$.

To conclude, we have $k > 2m$ (or in other words $|V(T_1)| > 2|V(T_2)|$) and $2(k-a) < m+a, 1 \leq a \leq k/2$. These inequalities have no solution and thus we have reached a contradiction. □

2.2 Obtaining a tree partition

Below we describe the algorithm that chooses the balanced tree partition of minimum spanning tree T with n nodes in $O(n)$ time. We also emphasize that the algorithm uses only local information and, therefore can be easily implemented in a distributed way. First, for each edge $e = (u,v) \in T$, we compute how many nodes are located in the subtree T_u of T that includes u but not v and in the subtree T_v of tree T that includes v but not u. Notice that $T_u \cap T_v = \emptyset$ and $T_u \cup T_v \cup \{e\} = T$. This can be done by a simple scanning of the tree T, starting from the leafs, and converging towards the internal nodes of the tree while counting the number of nodes on the way. For the distributed version, we first establish the connectivity and run the algorithm of Awerbuch [2] that builds a minimum spanning tree. In order to find the above-mentioned values for each edge, we use the convergecast process. Next, we use a well-known fact that the maximal degree of any node in T is at most 5 [34]. It means that the number of partitions that can be possibly made (per each node) is constant. Using the information computed in the previous step, we can find each such partition in $O(1)$ time. Thus, we can find the best partition for each particular node in $O(1)$ time (and in $O(n)$ time for all the nodes) and the best balanced tree partition in $O(n)$ time.

2.3 Data collection through tree partitioning

Using the above, we can define our hierarchical construction in the following recursive fashion. We find the balance node v (which we assume is the location of the base station as well) of the minimum spanning tree T and split the tree into two subtrees T_1 and T_2 sharing the same node v. Next we connect v with recursively computed balance nodes of T_1 and T_2, respectively, and continue in the same way. Clearly, the hop-diameter of the obtained hierarchy H (in fact, it is a binary tree) will be $O(\log n)$ and the total sum of edges, or in other words, transport capacity $D(H)$ will be $O(\log n \cdot w(MST))$, where $w(MST)$ is the weight of minimal spanning tree for our set of nodes, when the edge weight is defined as the Euclidean distance between two nodes. We point out that such a partition can be obtained in $O(n \log n)$ time since we will spend $O(n)$ for each level of the hierarchy. For distributed version, we observe that the total runtime will be bounded by $O(n)$ since the balance nodes in each level of the hierarchy can be computed in parallel and the time needed for computations in each level of the hierarchy is decreasing proportionally to fraction $\frac{2}{3}$ of n (the size of the largest component).

We can provide some tradeoff mechanism between the hop-diameter of the hierarchy and the total distance of the obtained edges. Using the above-mentioned partitioning procedure we can build an m-ary tree hierarchy (for any integer parameter m, $1 < m \leq n$). Instead of performing a (binary) partition of the tree T and connecting every parent node with two recursively computed children, we can partition the tree into m components by using a balanced partition, choosing the largest component from all components in current level and continue partitioning until we have m components. Next we connect the parent node with the m children which are recursively computed in the same manner. The obtained hierarchy H' has hop-diameter of $\log_m n$; however the total transport capacity $D(H')$ deteriorates to $O(m \log_m n \cdot w(MST))$. We may note here that Hassin and Peleg [21] suggested another construction based on separators of the tree; however the obtained degree of the resulted tree (and, therefore, the hop-diameter) can not be chosen (opposite to our strategy) and can be twice as large as the degree of the initial tree.

2.4 Augmenting shortest paths

At this stage we have a collection data tree T_S having hop-diameter of $O(\log n)$ and having transport capacity of weight $O(w(MST) \cdot \log n)$. Clearly, the weight of minimal spanning tree is equal to the optimal transport capacity OPT_c that can ever exist (each node, except of the root, needs to connect to its parent; thus the total transport capacity equals the sum of the edges' weights in the tree). Thus, our collection data tree has transport capacity of weight $O(OPT_c \cdot \log n)$. It may happen that the given root node r does not coincide with the current root of T_S. In that case we simply redirect the corresponding edges towards r. This procedure does not change asymptotically any bound derived for T_S.

Next we proceed following the construction suggested by Khuller et al. [26]. A LAST is a combination of a minimum spanning tree and a shortest path tree. Given a graph G, edges' weight function w and a source node r, Khuller et al. [26] presented a linear time algorithm, which computes a spanning tree of T, so that its weight is at most β times the weight of a minimum spanning tree of G, and for every node v, $p_T(v,r) \leq \alpha \cdot d(v,r)$, where $\alpha > 1$ and $\beta \geq 1 + \frac{2}{\alpha - 1}$. A spanning tree that complies with the bounds is called an (α,β)-LAST. Basically, the algorithm for computing (α,β)-LAST works as follows. First, we compute the minimal spanning tree T of initial graph G. In the following step, a preorder scan is performed over the vertices of T and the comparison is made between the weight of the existing path $p_T(v,r)$ of currently scanned node v and the weight of the shortest path existing in G between multiplied by α. In case that the weight of currently existing path is larger, we add the edges of the shortest path in G to T. Finally, the unnecessary edges are removed from T by running shortest path tree algorithm on T from r.

We are going to incorporate our currently constructed T_S into the Khuller et al. [26] algorithm above. In particular, instead of computing minimum spanning tree, we supply T_S as the first step of the algorithm and the rest remains the same. We also note that the weight of the shortest path between any node v in G and r is simply $d(v,r)$, i.e. we might need to add only one directed edge between r and v to T_S in case that the length of current path violates the given requirement. We also note that this procedure can only decrease the current hop-diameter of T_S. The same proof for the obtained weight of spanning tree holds to our case as well, i.e. at the end of the algorithm we obtain a tree T_S having hop-diameter $O(\log n)$, transport capacity of $O(OPT_c \cdot (1+\frac{2}{\alpha-1})\log n)$ and stretch factor $\delta(T_S) = O(\alpha)$, for $\alpha > 1$.

2.5 Lower bound for sum of squares with bounded hop

One may wonder whether there is hope for designing bounded hop hierarchy for arbitrary points positions that simultaneously provides good bounds for transport capacity and the sum of squared distances, i.e. the energy requirement. Unfortunately, the following example shows that it is not possible (see Fig. 1). Consider the unweighted n-path: any tree T' with hop-diameter Δ, contains an edge with an interval length of at least $(n-1)/\Delta$, and so its weight is at least $(n-1)^2/\Delta^2$. Observe that for the minimum spanning tree T, $E(T) = n-1$. However, for tree T', we have $E(T') \geq n-2+(n-1)^2/\Delta^2$. It means that the approximation ratio we can have while aiming for the hop-diameter of Δ is at least $\frac{n-1}{\Delta^2}$. For example, it follows that for $\Delta = \log n$ we can not build any hierarchy having energy requirement less than $\frac{n-1}{\log^2 n}$ times the optimal one.

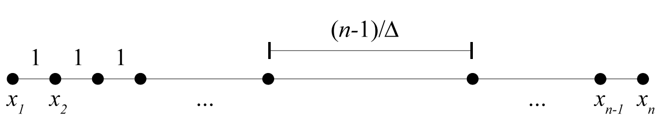


Fig. 1 Demonstrating a lower bound for sum of squares of distances.

3 Multi-hop collection for any deployment: altogether with energy

In this section we are going to build a structure that can perform well under all of the proposed criteria: energy consumption, transport capacity, hop-diameter and stretch of the paths to root node. As we already mentioned, Elkin et al. [15] proposed the solution for the data collection tree problem such that the total energy consumption is of factor ρ from optimal bound OPT_e and the hop-diameter is $n/\rho + \log\rho$, for any chosen integer parameter $\rho, 1 \leq \rho \leq n$. In some sense, this is almost best (up to logarithmic factor) we can do in a view of lower bound example above. Nevertheless, below we present a novel construction that in many cases outperforms the construction of Elkin et al. [15].

Let us denote by T' the minimal spanning tree for the set of nodes where the weight of edge is defined as the squared Euclidean distance between the nodes, and $w(T')$ is the total weight of the edges in this tree. As can be easily seen, $w(T')$ is the lower bound for the energy consumption for data collection tree (each node, except the root, transmits to its parent in the tree). The main weakness in the approach of Elkin et al. [15] is that they completely ignore the current hop-diameter of T'. Given a desired bound h for hop-diameter and $h(T') > h$, they immediately transform T' into corresponding Hamiltonian cycle with a consequent hierarchy construction. However, in cases when $h(T')$ not exceeds by much at h (as we show, it can be as much as $O(h^2)$), we can do better. Below we present and analyze our construction.

First we find T'. We can do this in $O(n\log n)$ time using Delaunay triangulation. Next, we check, whether the resulting tree T' satisfies the requirement of hop-diameter at most h. If yes, we are done. Otherwise, we are going to shorten the tree in the following fashion. We choose the given vertex r to serve as the root and tag every other node using its distances from the root. It can be done using the standard BFS algorithm. Every edge also receives tag being the minimum value between its both endpoints. Next, we make h stages. At stage $j, 0 \leq j \leq h-1$, we remove from the tree T' rooted at v all the edges being tagged $i \cdot h + j$, for every $i, 0 \leq i \leq \frac{|V|-j}{h}$, and connect the nodes tagged $i \cdot h + j + 1$ directly to the root r. We call the resulting tree T_j. After all h stages we choose between h trees T_j the tree having minimal weight. We call this tree T''. We bound the performance of this solution as follows. Since T' has the minimal weight between all h trees, it follows that weight $w(T'') \leq (1/h)\Sigma_{j=0}^{h-1} w(T_j)$. Next, notice that when considering the entire collection $\{T_j\}_{j=0}^{h-1}$ of trees, every edge of T has been replaced no more than once. Every such replacement (we have at most n such replacements) produced a new edge of weight of at most $n^2 \cdot w(e^*(T'))$, where $e^*(T')$ is the largest edge in T'. This is because a weak triangle inequality is satisfied (i.e., for $u,v,w \in V$, $(d(u,w))^2 \leq 2((d(u,v))^2 + (d(v,w))^2)$ and following Cauchy-Schwartz inequality we have that for any $x_1, x_2, \ldots, x_k \in \mathbb{R}$, $(\Sigma_{l=1}^{k} x_l)^2 \leq k \cdot \Sigma_{l=1}^{k} x_l^2$.

SHORTCUT MST

```
Compute minimum spanning tree T' of complete graph defined
on nodes S ∪ {r}.
if hop-diameter of T' is at most h then
  | output T'
Tag nodes in T according to their distance from root r.
Every edge (v,w) receives tag that is minimum between tags of
v and w.
for j = 0 → h−1 do
  | Remove from the tree T' rooted at r all the edges being
  | tagged i·h + j, for every i, 0 ≤ i ≤ (|V|−j)/h
  | Connect the nodes tagged i·h + j + 1 directly to the root r
  | obtaining tree T_j.
Let T'' be the tree of minimal weight from {T_j}_{j=0}^{h−1}.
Output T''.
```

Thus, $\Sigma_{j=0}^{h-1} w(T_j) \leq h \cdot w(T') + n^3 \cdot w(e^*(T'))$. Combining things together we obtain: $w(T'') \leq w(T') + n^3 \cdot w(e^*(T'))/h$ In other words, the weight of T'' provides $1 + \frac{n^3 \cdot w(e^*(T'))}{h \cdot w(T')}$ approximation for optimal solution. The SHORTCUT MST algorithm shows the formal description of aforementioned scheme.

But, in fact, we can do much better. The idea is that every edge cut and its replacement will lead to an additional $O(h^2 \cdot w(e^*(T'))$ increase of energy, instead of current increase of $n^2 \cdot w(e^*(T'))$ per edge. The crux is to cut the edges not in intervals of size h but rather take intervals of size $h, h-1, h-2, \ldots, 1$ and connect the node u (that became disconnected as the result of cut) in the interval of size $t, 1 \leq t \leq h$ to the node (that became disconnected) in the interval of size $t+1$ that lies on the same path in T' as u. The node in interval of size h is connected directly to the root. We perform the same shifting strategy as before. This scheme is presented in IMPROVED SHORTCUT MST algorithm figure and guarantees that every new added edge may need to bypass at most h original edges of T' and, therefore, will lead to $O(h^2 \cdot w(e^*(T'))$ increase in total energy consumption. To conclude, we have a construction that provides $O(\frac{nh \cdot w(e^*(T'))}{w(T')})$ approximation for total energy consumption with hop-diameter of h assuming that the hop-diameter of minimum spanning tree is at most h^2. This result compares well with the best solution to date by Elkin et al. [15] from the following reason. Of course, the value of $w(e^*(T'))$ can be as large as $w(T')$ but in many cases, the weight of the heaviest edge of minimum spanning tree behaves similarly as the length of the average edge in minimum spanning tree or similar to this. For example, for uniformly distributed points in 2-dimensional unit size square, $w(e^*(T')) = O(\log n/n)$ and $w(T')$ is at least $\Omega(1)$, see [37]. It means that for this and similar cases, the approximation factor stands at $h \log n$. If we choose $h = O(\log n)$, then our approximation factor is $O(\log^2 n)$ while the solution in [15] produces approximation of n. Moreover, even if $\frac{w(e^*(T'))}{w(T')} = n^{-\varepsilon}, 0 < \varepsilon < 1$, our approximation is $hn^{1-\varepsilon}$ and for the polylogarithmic values of h our algorithm gives better approximation than in [15]. Speaking of transport capacity, the similar analysis shows that the approximation factor is $1 + \frac{n \cdot d(e^*(T'))}{d(T')}$, where $d(e^*(T'))$ denotes the Euclidean length of the longest edge in T' and $d(T')$ is the total length of the edges in T'. For uniformly distributed points in 2-dimensional unit size square, $d(e^*(T')) = O(\sqrt{\log n/n})$ and $d(T')$ is at least $\Omega(\sqrt{n})$, see [42]. Thus, we obtain sublinear approximation for this and similar cases following similar argument as above.

IMPROVED SHORTCUT MST

```
Compute minimum spanning tree T' of complete graph defined
on nodes S ∪ {r}.
if hop-diameter of T' is at most h then
  | output T'
Tag nodes in T according to their distance from root r.
Every edge (v,w) receives tag that is minimum between tags of
v and w.
for j = 0 → h−1 do
  | Remove from the tree T' rooted at r all the edges being
  | tagged i·h + j − Σ_{p=0}^{i} p, for every i, 0 ≤ i when such edge
  | exists.
  | Connect the nodes tagged i·h + j + 1 − Σ_{p=0}^{i} p directly to
  | the node tagged (i−1)·h + j + 1 − Σ_{p=0}^{i−1} p lying on the
  | same path in T' or to the root r when i = 0 obtaining tree
  | T_j.
Let T'' be the tree of minimal weight from {T_j}_{j=0}^{h−1}.
Output T''.
```

Remark. The IMPROVED SHORTCUT MST scheme can be applied in a bootstrapping fashion for the case when $h \in o(\sqrt{h(T')})$. In particular, we can take T' and cut the edges in intervals of $h(T'), h(T')-1, h(T')-2, \ldots, 1$ by adding shortcut edges. In such a way, we will obtain a new tree T'_1 having hop-diameter $h(T')$. If $h \in o(\sqrt{h(T'_1)})$, we apply the same procedure to T'_1 cutting it in intervals of $h(T'_1), h(T'_1) - 1, h(T'_1) - 2, \ldots, 1$ with addition of shortcut edges. We repeat this process until we obtain a tree of desired diameter. Clearly, the number of bootstrapping steps is at most $O(\log \log h(T'))$ since we shrink the hop-diameter recursively by square root factor.

Now we suggest how to incorporate the stretch of paths characteristics into our solution. We will show that while the stretch of the paths drops to the factor of $h^2(T'') \cdot n^{\varepsilon}, 0 < \varepsilon < 1$, the total energy consumption in new tree increases by factor of at most $n^{1-2\varepsilon}$ from the energy consumption in T''. We start scanning all of the nodes in our constructed tree T''. In general, the technique will work for any data collection tree. If the distance $d(v,r)$ between currently scanned node v to root r is less than $d(e^*(T'))/n^{\varepsilon}, 0<\varepsilon<1$, then we remove the edge from v to its parent node in T'' and put a direct edge between v and r. Notice, that the hop-diameter, $h(T'')$, can only decrease as the result of our procedure.

Let us compute what happens with total energy consumption. In the worst scenario, we replaced all the edges in the tree and, therefore, the total energy consumption is bounded by $\frac{n\cdot w(e^*(T'))}{n^{2\varepsilon}}$ and the approximation factor from the optimal solution is $O(\frac{n^{2-2\varepsilon}\cdot h\cdot w^2(e^*(T'))}{w^2(T')}) \in O(n^{2-2\varepsilon}\cdot h)$. For appropriate values of ε and, for example, for polylogarithmic values of h or even higher, we obtain a sublinear approximation factor. We leave to any interested reader to obtain the exact range of values when the approximation factor for total energy consumption is $o(n)$ (notice that in our analysis we assumed that $w(e^*(T')) = w(T')$ where for many cases as shown above $w(e^*(T'))$ is much smaller than $w(T')$, and, better approximation factor can be derived; the same holds for transport capacity as well).

In order to evaluate the stretch factor of the obtained paths we notice the following. First, if $d(v,r) < d(e^*(T'))/n^{\varepsilon}$, then a shortcut has been added and the stretch of the path between v and r in new tree is 1. Second, if $d(v,r) \geq d(e^*(T'))/n^{\varepsilon}$, then the length of the path between v and r in a new tree is at most $h(T'')\cdot d(e^*(T'')) \leq h^2(T'')\cdot d(e^*(T')) \leq n^{\varepsilon}\cdot h^2(T'')\cdot d(v,r)$. It means that the stretch of the path is at most $n^{\varepsilon}\cdot h^2(T'')$.

IMPROVING THE STRETCH OF PATHS

```
compute T''.
foreach v ∈ V do
    if d(u,r) < d(e*(T'))/n^ε then
        remove edge from v to its parent node in T''.
        add (v,r) to T''.
Output T''.
```

To summarize we have the following theorem.

Theorem 1 *Given an integer value $h, 1\leq h\leq n$ and $\varepsilon, 0<\varepsilon<1$, and assuming that minimal spanning tree of the set of nodes has diameter $O(h^2)$, we can find a data collection tree T_S such that the total energy consumption $E(T_S) = O(n^{2-2\varepsilon}\cdot h\cdot OPT_e)$, $D(T_S) = O(n^{2-\varepsilon}\cdot h\cdot OPT_c)$, $h(T_S) = h$ and $\delta(T_S) = n^{\varepsilon}\cdot h^2$.*

We also may use the LAST construction as described in previous section in order to obtain different tradeoff approximation for our criteria. In particular, applying Khuller et al. [26] produces the following results.

Theorem 2 *Given an integer value $h, 1\leq h\leq n$ and $\varepsilon, 0<\varepsilon<1$, and assuming that minimal spanning tree of the set of nodes has diameter $O(h^2)$, we can find a data collection tree T_S such that the total energy consumption $E(T_S) = O(\frac{n^2h^2\cdot w^2(e^*(T'))}{w^2(T')}\cdot OPT_e)$, $D(T_S) = O((1+\frac{2}{\alpha-1})\cdot\frac{n\cdot d(e^*(T'))}{d(T')}\cdot OPT_c)$, $h(T_S) = h$ and $\delta(T_S) = \alpha$, for $\alpha > 1$.*

4 Distributed implementation

The distributed implementation of our construction heavily depends on the construction on minimal spanning tree (with the consequent manipulations) which is quite straightforward once we established connectivity between the nodes and chose the leader (the root of the tree). For this we can follow two different approaches as described in [33]. The first, described in Dolev et al. [13] forms a temporary underlying topology in $O(n)$ time using $O(n^3)$ message. The second (better) approach is given by Halldórsson and Mitra [20] that shows how to do this in $O(poly(\log\gamma, \log n))$, where γ is the ratio between the longest and shortest distances among nodes. After the topology is established, we can use leader-election algorithm by Awerbuch [2] that shows how to find a leader and minimum spanning tree in a distributed fashion in a network with n nodes in $O(n)$ time using $O(n\log n)$ messages. In our former construction, the leader initiates the process of finding the balance nodes with following hierarchy construction and (α,β)-LAST computation in a distributed fashion as described in [6]. In the latter design, the leader initiates the process of shortcuting edges with the consequent convergecast process towards the leader. Each node (in parallel), computes the edge required to be added to the data collection tree and chooses the largest outgoing edge. The total time and message complexities for each T_i calculation are dominated by the initial minimum spanning tree construction step.

5 Mobile sensors

As the distance between any two nodes $u,v\in S$ may vary in time, we define $d_{u,v}(t)$ to be the Euclidean distance between u and v at time $t\in[t_s,t_f]$. The transmission possibilities resulting from a power assignment vary in time. Let $H_p(t) = (S, E_p(t))$, with $E_p(t) = \{(u,v) : r_u \geq d_{u,v}(t)\}$, be the induced directed communication tree at time $t\in[t_s,t_f]$. Let $G_S = (S,E_S)$ be an undirected complete graph. For any $t\in[t_s,t_f]$, let $w_t(u,v) = (d_{u,v}(t))^2$, for every $(u,v)\in E_S$, a weight function over the edge set E_S. Note that $w_t(u,v)$ matches the amount of energy required to transmit from u to v, at time t. For any weight function w, defined on a weight set E_S, the weight of a graph $H=(S,E_H)$, $E_H\subseteq E_S$, is $w(H) = \sum_{(u,v)\in E_H} w(u,v)$. Consequently, denote the weight of a graph $H(t) = (S,E_H(t))$ as $w(H(t)) = \sum_{(u,v)\in E_{H(t)}} w_t(u,v)$ in time $t\in[t_s,t_f]$. The maximum weight of $H(t)$ through the session $[t_s,t_f]$ is denoted by $w_{max}(H(t)) = \max_{t,t\in[t_s,t_f]} w(H(t))$. Finally, the "critical weight" of $H(t)$ is $w_{cr}(H(t)) = \sum_{(u,v)\in E_{H(t)}} \max_{t,t\in[t_s,t_f]}(d_{u,v}(t))^2$.

For any two nodes $u, v \in S$, in order that an edge (u,v) would exist in every $H_p(t)$, $t \in [t_s, t_f]$, the power assigned to u should be at least the square of the maximum distance between u and v during $[t_s, t_f]$.We define a weight function w' which reflects this amount of energy for any pair of nodes, $w'(u,v) = \max_{t \in [t_s, t_f]} (d_{u,v}(t))^2$, for every $u, v \in S$.

5.1 Basic solution

Consider the problem of finding a power assignment p, which induces a static communication network, $H_p(t) = \{S, E_p(t)\}$, $\forall t \in [t_s, t_f]$, with connectivity topology property, minimizing the total power of the transceivers $c(p)$.

By definition, $(d_{u,v}(t))^2 \geq 0$, $\forall u, v \in S$, $\forall t \in [t_s, t_f]$, and since the nodes move in straight lines, $(d_{u,v}(t))^2$ is also convex and $\max_t (d_{u,v}(t))^2$ is observed in $t = t_s$ or $t = t_f$.

Theorem 3 *$\sum_{(u,v) \in H_p(t)} (d_{u,v}(t))^2$ is maximum in $t = t_s$ or $t = t_f$.*

Proof. For any pair of nodes $u, v \in S$, denote by $y^2_{u,v}(t)$ the linear function, $y^2_{u,v}(t) = a^2_{u,v} t + b^2_{u,v}$, such that $y^2_{u,v}(t_s) = d^2_{u,v}(t_s)$ and $y^2_{u,v}(t_f) = d^2_{u,v}(t_f)$. For any edge $(u,v) \in H_p(t)$, $\sum_{(u,v) \in H_p(t)} y^2_{u,v}(t) = t \cdot \sum_{(u,v) \in H_p(t)} a^2_{u,v} + \sum_{(u,v) \in H_p(t)} b^2_{u,v}$. Clearly, $\sum_{(u,v) \in H_p(t)} y^2_{u,v}(t)$ is maximum in $t = t_s$ or $t = t_f$. Since $d^2_{u,v}(t)$ is convex for $\forall u, v \in S$, $\forall t \in [t_s, t_f]$, then $\sum_{(u,v) \in H_p(t)} (d_{u,v}(t))^2 \leq \sum_{(u,v) \in H_p(t)} y^2_{u,v}(t)$, $\forall t \in [t_s, t_f]$. The theorem follows immediately. □

Consider a connected subgraph $H^\star(t)$ of G_S whose maximum weight $w_{max}(H^\star(t))$ is minimized over all possible choices of connected subgraphs of G_S. Obviously, $H^\star(t)$ is a tree. Let $p^\star$ be the optimal power assignment. Clearly, $c(p^\star) \geq w_{max}(H^\star(t))$.

Denote by $H'(t)$ a connected subgraph of G_S such that the weight $\max\{w(H'(t_s)), w(H'(t_f))\}$ is minimized over all possible choices of connected subgraphs of G_S. Following Theorem 1, we can conclude that $w_{max}(H^\star(t)) \geq \max\{w(H'(t_s)), w(H'(t_f))\}$.

Since the weights of the edges of $H'(t)$, that are not maximal in $t = t_s$ are maximal in $t = t_f$, and vice versa, then we have that $w_{cr}(H'(t)) \leq w(H'(t_s)) + w(H'(t_f)) \leq 2\max\{w(H'(t_s)), w(H'(t_f))\} \leq 2w_{max}(H^\star(t)) \leq 2c(p^\star)$.

Therefore, we can use the following algorithm that computes the power assignment. Consider a complete undirected graph of the nodes in S, G_S. Assign to each edge $\{u,v\} \in E_S$ the weight $\max\{w_{t_s}(u,v), w_{t_f}(u,v)\}$ and find the minimum spanning tree T of the resulting graph. The power assignment p assigns to each node v the power equal to the squared length of the longest edge in the minimum spanning tree attached to node v. The approximation factor follows from the fact that $c(p) \leq 2w(T) \leq 2w_{cr}(H'(t)) \leq 4c(p^\star)$. It is worth to mention here that Dolev et al. [14] have proved that the weight function w' that is defined as: $w'(u,v) = \max\{w_{t_s}(u,v), w_{t_f}(u,v)\}$ satisfies the weak triangle inequality, i.e. $w'(u,v) \leq 2(w'(u,z) + w'(z,v))$, for any $u, v, z \in V$.

We can, in fact, claim that our approximation factor is output-sensitive. To see this, let us enumerate the edges of T in non-decreasing order of their length: $e_1, \ldots e_{n-1}$. Then, there are n_2 edges that contribute twice to power assignment p (from both ends), there are n_1 edges that contribute only once to p (only from one end), and there are some edges that do not contribute at all. In the worst case, the largest length edges is counted twice, and thus their total contribution to p is at most $2\sum_{i=n-n_2}^{n-1} w(e_i)$. The contribution of the rest of the edges is $\sum_{i=n-n_2-n_1}^{n-n_2-1} w(e_i)$. Notice that since $n_1 = n - 2n_2$, the contribution of the rest of edges becomes $\sum_{i=n_2}^{n-n_2-1} w(e_i)$. Thus, to overall contribution to T does not exceed $2\sum_{i=n-n_2}^{n-1} w(e_i) + \sum_{i=n_2}^{n-n_2-1} w(e_i) = w(T) - \sum_{i=1}^{n_2-1} w(e_i) + \sum_{i=n-n_2}^{n-1} w(e_i) = w(T) - \sum_{i=1}^{n_2-1} w(e_i) + w(T) - \sum_{i=1}^{n-n_21} w(e_i) \leq 2w(T) - (n-2)w(e_1)$. On the other hand $c(p) \leq nw(e_{n-1})$. It follows that $c(p) \leq 2w(T) - (n-2)w(e_1)) \leq nw(e_{n-1})$. Consequently, $w(T) \leq \frac{nw(e_{n-1}) + (n-2)w(e_1)}{2}$. Thus, we have that approximation factor $\frac{c(p)}{w(T} \leq 2 - \frac{(n-2)w(e_1))}{w(T)} \leq 2 - \frac{2(n-2)w(e_1))}{nw(e_{n-1}) + (n-2)w(e_1)} = \frac{2w(e_{n-1})n}{w(e_{n-1})n + w(e_1)(n-2)}$. When n is large the bound is close to $\frac{2w(e_{n-1})}{w(e_{n-1}) + w(e_1)}$. The interesting thing is that we know value of $w(e_1)$ (which is equal to the smallest weight in the graph G_S) and can provide an upper bound for $w(e_n)$ (as the largest weight in the graph G_S) even before the execution of our algorithm.

It appears computing the solution tree T_{t_s} at the start of session t_s and T_{t_f} at end of session t_f and finally assigning to each node a maximal power it has from both trees is not enough. It may even not guarantee the connectivity of a given set of sensors as shown in Figure 2. We have 4 ordered nodes, a, b, c, d located on the line, with the distances between them as shown in the Figure. The nodes move with the same speed. The node a moves towards b , b moves towards a, c moves towards d, and d moves towards c, see Figure 2(a). The final position of points is demonstrated at Figure 2(b). The points a and b switched their positions, so the points c and d. As it shown in Figure 2(c), it is not enough to give each point power that enough to cover a distance of $1/3$ although it will guarantee connectivity at t_s and t_f. The distance between a and c is $2/3$ in Figure 2(c) and the network is disconnected.

We can observe that our scheme works well for other topology criteria, e.g. lifetime (the number of rounds) or transport capacity (which is the sum of lengths of all the links, $\sum_{(u,v) \in E_S} d_{u,v}$) of the induced tree. In general, when the batteries' charges of the nodes are the same and the traffic is uniform, the lifetime of the tree is dictated by the length of the longest edge in the tree. Thus, our goal is to find a power assignment $\hat{p}$ such that the length of the longest edge

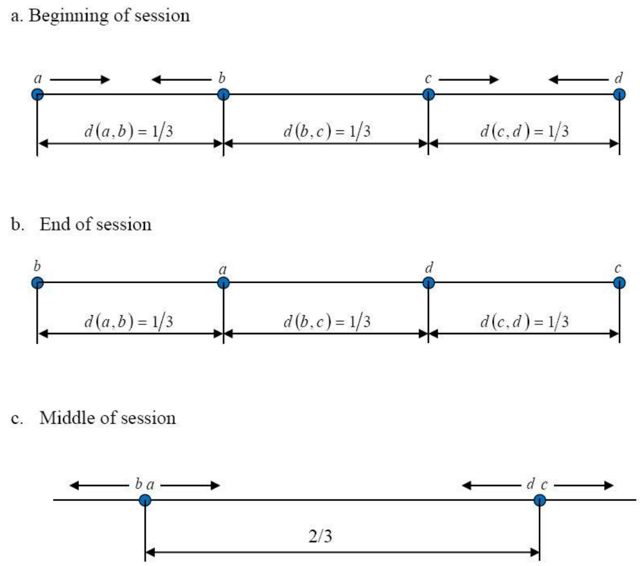


Fig. 2 Counterexample.

in $H_{\hat{p}}(t)$ is minimized. Following the description above, let $\tilde{H}(t)$ be a tree of G_S whose longest edge weight $l_{max}(\tilde{H}(t))$ is minimized over all possible choices of trees of G_S. Let $l_{\hat{p}}$ be the length of the longest edge obtained by the power assignment $\hat{p}$. Clearly, $l_{\hat{p}} \geq l_{max}(\tilde{H}(t))$.

Denote by $H''(t)$ a connected subgraph of G_S such that the longest edge weight $\max\{l_{max}(H''(t_s)), l_{max}(H''(t_f))\}$ is minimized over all possible choices of connected subgraphs of G_S. Following the claim similar to Theorem 1, we can conclude that $l_{max}(\tilde{H}(t)) \geq \max\{l_{max}(H''(t_s)), l_{max}(H''(t_f))\}$.

Since the weights of the edges of $H''(t)$, that are not maximal in $t = t_s$ are maximal in $t = t_f$, and vise versa, then we have that $l_{cr}(H''(t)) \leq l_{max}(H''(t_s)) + l_{max}(H''(t_f)) \leq$ $2\max\{l_{max}(H''(t_s)), l_{max}(H''(t_f))\} \leq$ $2l_{max}(\tilde{H}(t)) \leq 2l(\hat{p})$, where $l_{cr}(H(t)) = \max_{t,t \in [t_s,t_f]}(d_{u,v}(t))^2$.

Thus, the same algorithm will produce the $\frac{4w(e_{n-1})}{w(e_{n-1})+w(e_1)}$ approximation for the lifetime criteria of the tree.

5.2 Fault-tolerance

Andrea and Bandelt [1] give a linear time algorithm for the construction of the Hamiltonian circuit h in T^3, given tree T and a weight function that satisfies a weak triangle inequality. The algorithm is applied to a tree T and an edge $e = (u,v)$ of T. Removing the edge e divides the tree into two subtrees T' and T''. In each subtree the algorithm selects an arbitrary edge $e' = (u,w)$ (for T') and $e'' = (x,v)$ (for T''), and recursively computes a Hamiltonian cycle of T' and T'' that includes the edge e' and e'', respectively. The circuit consists of the cycles in T' and T'' without two edges e' and e''. The two resulting paths are glued together using e and the edge connecting other endpoints of two edges e' and e''. They [1] also show that $w(h) \leq w(T) \cdot (\frac{3}{2}\tau^2 + \frac{1}{2}\tau)$, where τ is the weak triangle inequality parameter that equals 2 in our case. Moreover, it can be shown that the weight of the longest edge in h is at most $O(1)$ times the weight of the longest edge in T. The following theorem applies the above on minimum spanning tree T.

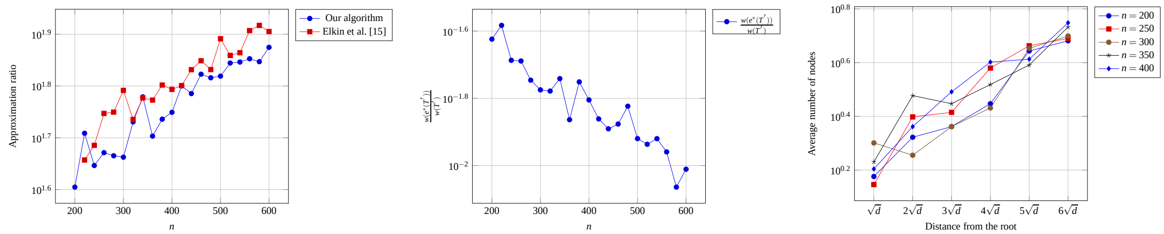
Theorem 4 ([1]) *Let $h = (u_0, u_1, \ldots, u_n = u_0)$, where $u_i \in V$ for $0 \leq i \leq n-1$, be the Hamiltonian circuit as a result of applying the construction in [1] on T. Define $e^*(T)$ and $e^*(h)$ to be the longest edges in T and h, respectively. Then $w(h) = O(w(T))$ and $w(e^*(h)) = O(w(e^*(T)))$.*

Using the aforementioned theorem we can add the fault-tolerance property to the data collection tree we aim to maintain. After computing the required tree, we find the Hamiltonian circuit as described above, and define the power level of each node as the maximum value between the power level in the tree and the power level the node has in the Hamiltonian circuit. Notice that it does not change asymptotically the cost of power assignment but guarantees the existence of two vertex disjoint paths between the root s and any other node. It follows that even if some node will fail, the tree will still function (although its hop-diameter may increase due to the node failure).

5.3 Interference awareness

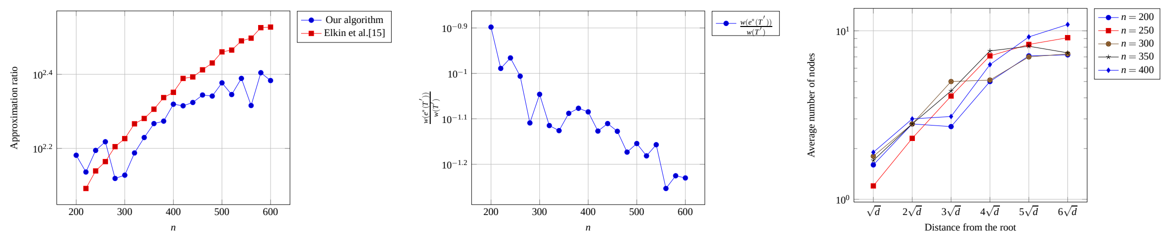
As nodes communicate through radio signals, wireless interference becomes inevitable. Every node receiving simultaneous signals may incorrectly interpret them. High levels of interference decrease the number of transmissions that can happen simultaneously, which has a direct affect on the required number of time slots for the message to propagate from the source to all the other nodes in the network or opposite. We point out here that minimizing the hop-diameter does not necessarily leads to the minimal number of slots for propagating broadcast or convergecast messages. In order to deal with the problem we give separate solutions for convergecast and broadcast scheme. To perform convergecast that starts from the leaves, every node keeps two numbers: the total number of its siblings and its consecutive number (starting from 0) between siblings order. These numbers can be redistributed to every node by its parent in the tree. Then, the node is allowed to send a message to its parent in the tree only if the time slot number equals its consecutive number modulo the total number of its siblings. This prevents from more than one child to send a message simultaneously to its parent. Every node sends a message to its parent only when it received the messages from all of its children.

For propagating the broadcast message, we can use the previous construction of Hamiltonian circuit in order to guarantee the required number of slots for propagating broadcast



(a) Comparison with the best to date algorithm [15].
(b) Ratio between maximum edge weight to MST weight.
(c) Average number of nodes in different levels of MST (of diameter d).

Fig. 3 Random uniform network.



(a) Comparison with the best to date algorithm [15].
(b) Ratio between maximum edge weight to MST weight.
(c) Average number of nodes in different levels of MST (of diameter d).

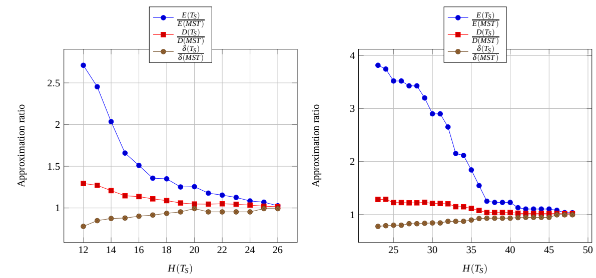
Fig. 4 2D Poisson point process with normalized unit density.

message be equal to some prescribed input parameter k. After we find the Hamilton circuit (call the order of the nodes starting from the source node s as $s, v_1, v_2, \ldots, v_i, \ldots, v_{n-1}, s$, we assign the required power to s to be enough to reach the next n/k nodes in the path, i.e. to reach $v_{\frac{n}{k}}$, and follow the same scheme to assign powers of $v_{\frac{n}{k}}, v_{\frac{2n}{k}}, v_{\frac{3n}{k}}, \ldots$. The suggested scheme assures us that there will be only n/k transmitting nodes (in different time slots) and the total broadcast time will be k. This scheme increases the total energy consumption by factor of n/k.

6 Simulation Results

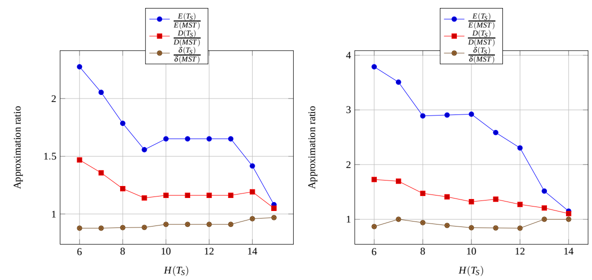
In this section we show some simulation results with respect to the construction in Section 3, where we are mainly interested in comparison of total consumption energy criteria obtained by ours and Elkin et al. [15], which is considered to be the best algorithm in terms of energy consumption with predefined hop-diameter. As we show, the simulation results fully support our theoretical analysis. In our first experiment we have randomly and uniformly distributed n sensor nodes in a square of size 10×10, with the network size n ranging from 200 to 600 in steps of 20, see Fig. 3(a,b,c). We have computed the energy consumption (Fig. 3(a)), the ratio between the weight of the heaviest edge in minimum spanning tree and the weight of minimum spanning tree (Fig. 3(b)) and the number of nodes that located in different levels of minimum spanning tree (Fig. 3(c)). The results are an average of 10 tries for every network size n, where the predefined value that has been taken for required hop-diameter h is the square root of the obtained minimum spanning tree diameter d. This is since for larger values of h our algorithm performs even better as there is no need in doing shortcuts for many nodes. We can, in fact, observe from the Fig. 3(c) that the amount of nodes that need to be shortcut when the value of h close to the diameter of minimum spanning tree is small. As it can be concluded from Fig. 3(a), for the values of n started from 240 our solution always outperforms the one given in [15]. Moreover, Fig. 3(b) also confirms the fact that the ratio between the weight of the heaviest edge in minimum spanning tree and the weight of minimum spanning tree deteriorates as n grows up. As our algorithm depends linearly on such ratio, we deduce that it works really well for real, large-scale deployments. Our second experiment modeled a wireless sensor network by 2D Poisson point process of normalized unit density in an 25×25 region for various values of n, see Fig. 4(a,b,c). This is a standard technique for modeling random wireless network with omni-directional transmission as in [9]. We evaluate the same criteria as in the first experiment. We observe the same tendency for all Fig. 4(a), Fig. 4(b) and Fig. 4(c) as for uniformly placed sensors although the rates are slightly different. We also learn that the ratio in Fig. 4(b) indeed decreases but more slowly than for random uniform network.

Next, for both uniform sensor nodes (Figure 5) and 2D Poisson point process with normalized unit density (Figure 6) we evaluated the total energy consumption, total transport and stretch factor for different values of hop-diameter obtained by Improved Shortcut MST algorithm and compared them with the corresponding values obtained by standard MST solution. We can learn that in terms of energy and transport, our algorithm performs slightly better for the 2D Poisson point process although for both cases the approximation ratio does not exceed the value of 4. For small values of hop-diameter our algorithm produces a tree with better stretch factor than MST. We reproduced the simulation tests under the same criteria, but now (Figure 7 and Figure 8) we compared our parameters when keeping the ratio between our produced tree hop diameter and MST hop-diameter equal to 0.5 (Figure 7) and equal to 0.75 (Figure 8). Our goal was to check whether the increase in the number of nodes has any influence on the produced results. We can conclude from both simulations that in both uniform and Poisson point processes for different ratio of hop-diameters the obtained results for all criteria (energy, transport, stretch factor) remained almost the same with small deviations. Thus, our algorithm works well for small values of the nodes as well as for the large number.



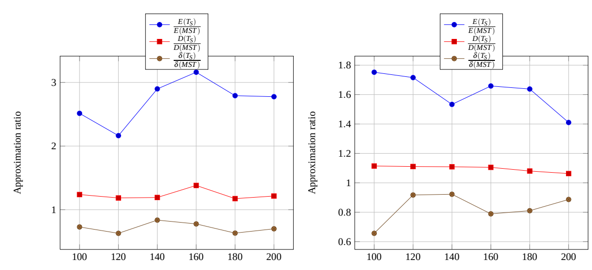
(a) 100 nodes (b) 200 nodes

Fig. 5 Comparison between approximation factor of IMPROVED SHORTCUT MST to MST for 2*D* random uniform network.



(a) 100 nodes (b) 200 nodes

Fig. 6 Comparison between approximation factor of IMPROVED SHORTCUT MST to MST for 2*D* Poisson point process with normalized unit density.



(a) $\frac{H(T_S)}{H(MST)} = \frac{1}{2}$ (b) $\frac{H(T_S)}{H(MST)} = \frac{3}{4}$

Fig. 7 Comparison between approximation factor of IMPROVED SHORTCUT MST to MST for 2*D* random uniform network when diameter ratio is constant.

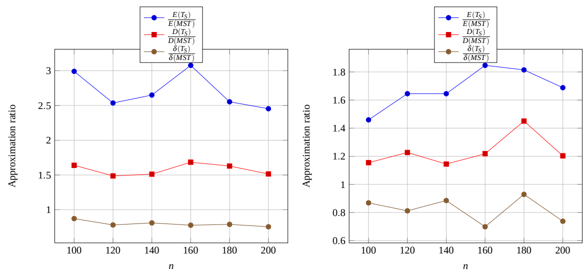
7 Conclusions

In this paper we have presented two constructions for data collection tree with provable performance bounds on total energy consumption, total transport capacity, hop diameter and stretch of the obtained paths from the nodes towards the root of the tree. We have shown that for various sensor nodes deployments our solutions outperforms the previously known schemes. We also considered the problem in mobile setting scenario where the sensors are allowed to move. It would be interesting to investigate how well our structures perform in terms of average hop-diameter (i.e. hop-diameter taken over all paths connecting nodes to the root) which can serve as another potential criteria to optimize for scenarios where sensors send the information towards the root in different time frames and periods of time. It looks like our schemes can be extended to a more general, SINR model, where a transmission is successful if the signal is strong enough compared to the interference (as a result of simultaneous transmissions). This is because we can adopt some of the known techniques for dividing the nodes into interference/transmission regions based on the transmit powers [24].

Acknowledgement. The authors thank to Engineering and Physical Sciences Research Council (EPSRC), United Kingdom for providing support to the work on this paper and to the reviewers for their helpful comments.

References

1. T. Andreae and H.-J. Bandelt. Performance guarantees for approximation algorithms depending on parametrized triangle inequalities. *SIAM J. Discrete Math.*, 8(1):1–16, 1995.
2. B. Awerbuch. Optimal distributed algorithms for minimum weight spanning tree, counting, leader election, and related problems. In *ACM STOC 1987*, pages 230–240, 1987.
3. J. Basch. Kinetic data structures. In *PhD Dissertation, Stanford University*, 1999.
4. J. Basch, L. J. Guibas, and J. Hershberger. Data structures for mobile data. In *SODA'97*, pages 747–756, 1997.



(a) $\frac{H(T_S)}{H(MST)} = \frac{1}{2}$ (b) $\frac{H(T_S)}{H(MST)} = \frac{3}{4}$

Fig. 8 Comparison between approximation factor of IMPROVED SHORTCUT MST to MST for 2D Poisson point process with normalized unit density when diameter ratio is constant.

5. R. Beier, P. Sanders, and N. Sivadasan. Energy optimal routing in radio networks using geometric data structures. In *ICALP'02*, pages 366–376, 2002.
6. Y. Ben-Shimol, A. Dvir, and M. Segal. Splast: A novel approach for multicasting in mobile wireless ad hoc networks. In *IEEE PIMRC'04*, pages 1011–1015, 2004.
7. G. Calinescu, S. Kapoor, and M. Sarwat. Bounded-hops power assignment in ad hoc wireless networks. *Discrete Applied Mathematics*, 154(9):1358–1371, 2006.
8. A. Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, G. Konduri, J. Kulik, W. Rabiner, and A. Wang. Design considerations for distributed microsensor systems. In *CICC'99*, pages 279–286, 1999.
9. C.-K. Chau, R. J. Gibbens, and D. Towsley. Impact of directional transmission in large-scale multi-hop wireless ad hoc networks. In *INFOCOM*, pages 522–530, 2012.
10. A. E. F. Clementi, A. Ferreira, P. Penna, S. Perennes, and R. Silvestri. The minimum range assignment problem on linear radio networks. In *ESA'00*, pages 143–154, 2000.
11. A. E. F. Clementi, P. Penna, and R. Silvestri. On the power assignment problem in radio networks. *Electronic Colloquium on Computational Complexity*, 7(054), 2000.
12. A. E. F. Clementi, P. Penna, and R. Silvestri. The power range assignment problem in radio networks on the plane. In *STACS'00*, pages 651–660, 2000.
13. S. Dolev, M. Segal, and H. Shpungin. Bounded-hop energy-efficient liveness of flocking swarms. *IEEE Transactions on Mobile Computing*, 12(3):516–528, 2013.
14. S. Dolev, M. Segal, and H. Shpungin. Bounded-hop energy-efficient liveness of flocking swarms. *IEEE Trans. Mob. Comput.*, 12(3):516–528, 2013.
15. M. Elkin, Y. Lando, Z. Nutov, M. Segal, and H. Shpungin. Novel algorithms for the network lifetime problem in wireless settings. *ACM Wireless Networks*, 17(2):397–410, 2011.
16. S. Funke and S. Soren Laue. Bounded-hop energy-efficient broadcast in low-dimensional metrics via coresets. In *STACS'07*, volume 4393, pages 272–283, 2007.
17. J. Gao, L. Guibas, J. Hersheberger, L. Zhang, and A. Zhu. Geometric spanner for routing in mobile networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc01)*, pages 45–55, 2001.
18. J. Gao, L. J. Guibas, and A. Nguyen. Deformable spanners and applications. *Computational Geometry: Theory and Applications*, 35(1):2–19, 2006.
19. P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.
20. M. M. Halldórsson and P. Mitra. Distributed connectivity of wireless networks. In *PODC*, pages 205–214, 2012.
21. Y. Hassin and D. Peleg. Sparse communication networks and efficient routing in the plane. *Distributed Computing*, 14(4):205–215, 2001.
22. H. Huang, A. W. Richa, and M. Segal. Approximation algorithms for the mobile piercing set problem with applications to clustering in ad-hoc networks. *MONET*, 9(2):151–161, 2004.
23. H. Huang, A. W. Richa, and M. Segal. Dynamic coverage in ad-hoc sensor networks. *Mobile Networks and Applications*, 10(1-2):9–17, 2005.
24. S. Ji, R. A. Beyah, and Y. Li. Continuous data collection capacity of wireless sensor networks under physical interference model. In *IEEE MASS'11*, pages 222–231, 2011.
25. A. Jovicic, P. Viswanath, and S. R. Kulkarni. Upper bounds to transport capacity of wireless networks. *IEEE Transactions on Information Theory*, 50(11):2555–2565, 2004.
26. S. Khuller, B. Raghavachari, and N. Young. Balancing minimum spanning and shortest path trees. In *SODA'93*, pages 243–250, 1993.
27. L. M. Kirousis, E. Kranakis, D. Krizanc, and A. Pelc. Power consumption in packet radio networks. *Theoretical Computer Science*, 243(1–2):289–305, 2000.
28. K. Kothapalli, C. Scheideler, M. Onus, and A. W. Richa. Constant density spanners for wireless ad-hoc networks. In *SPAA'05*, pages 116–125, 2005.
29. K.Ramanan and E.Baburaj. Data gathering algorithms for wireless sensor networks: A survey. *IJASUC*, 1, December 2010.
30. X. Li, S. Yan, C. Xu, A. Nayak, and I. Stojmenovic. Localized delay-bounded and energy-efficient data aggregation in wireless sensor and actor networks. *Wireless Communications and Mobile Computing*, 11(12):1603–1617, 2011.

31. X.-Y. Li, Y. Wang, and Y. Wang. Complexity of data collection, aggregation, and selection for wireless sensor networks. *IEEE Trans. Computers*, 60(3):386–399, 2011.
32. Y. Li, M. T. Thai, F. Wang, and D.-Z. Du. On the construction of a strongly connected broadcast arborescence with bounded transmission delay. *IEEE Trans. Mob. Comput.*, 5(10):1460–1470, 2006.
33. V. Milyeykovsky, M. Segal, and H. Shpungin. Location, location, location: Using central nodes for efficient data collection in wsns. In *IEEE WIOPT*, 2013.
34. C. L. Monma and S. Suri. Transitions in geometric minimum spanning trees. *Discrete & Computational Geometry*, 8:265–293, 1992.
35. K. Pahlavan and A. H. Levesque. *Wireless information networks*. Wiley-Interscience, 1995.
36. V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient collision-free medium access control for wireless sensor networks. In *SenSys'03*, pages 181–192, 2003.
37. C. Redmond and J. Yukich. Asymptotics for euclidean functionals with power-weighted edges. *Stochastic Processes and their Applications*, 61(2):289 – 304, 1996.
38. M. Segal and H. Shpungin. Improved multi-criteria spanners for ad-hoc networks under energy and distance metrics. In *IEEE INFOCOM'10*, pages 6–10. Also in ACM Transactions on Sensor Networks 2013, to appear.
39. M. Segal and H. Shpungin. On construction of minimum energy *k*-fault resistant topology. *Ad Hoc Networks*, 7(2):363–373, 2009.
40. H. Shpungin and M. Segal. Low-energy fault-tolerant bounded-hop broadcast in wireless networks. *IEEE/ACM Trans. Netw.*, 17(2):582–590, 2009.
41. H. Shpungin and M. Segal. Near optimal multicriteria spanner constructions in wireless ad-hoc networks. *IEEE/ACM Transactions on Networking*, 18(6):1963–1976, 2010.
42. J. M. Steele. Probability and problems in euclidean combinatorial optimization. *Statistical Science*, 8(1):48 – 56, 1993.
43. L. van Hoesel and P. Havinga. A lightweight medium access protocol (lmac) for wireless sensor networks:reducing preamble transmissions and transceiver state switches. In *INSS'04*, pages 205–208, 2004.
44. P.-J. Wan, G. Calinescu, X. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *INFOCOM'01*, pages 1162–1171, 2001.
45. L.-L. Xie and P. R. Kumar. A network information theory for wireless communication: scaling laws and optimal operation. *IEEE Transactions on Information Theory*, 50(5):748–767, 2004.
46. L.-L. Xie and P. R. Kumar. On the path-loss attenuation regime for positive cost and linear scaling of transport capacity in wireless networks. *IEEE Transactions on Information Theory*, 52(6):2313–2328, 2006.
47. F. Xue, L.-L. Xie, and P. R. Kumar. The transport capacity of wireless networks over fading channels. *IEEE Transactions on Information Theory*, 51(3):834–0847, 2005.
48. Y. Yun, Y. Xia, B. Behdani, and J. C. Smith. Distributed algorithm for lifetime maximization in a delay-tolerant wireless sensor network with a mobile sink. *IEEE Transactions on Mobile Computing*, 12(10):1920–1930, 2013.