# Unconditional Lower Bounds on the Time-Approximation Tradeoffs for the Distributed Minimum Spanning Tree Problem

Michael Elkin *†

**Abstract**

The design of distributed *approximation* protocols is a relatively new rapidly developing area of research. However, so far little progress was done in the study of the *hardness of distributed approximation*.

In this paper we initiate the systematic study of this subject, and show strong *unconditional lower bounds on the time-approximation tradeoff* of the distributed *minimum spanning tree* problem, and some of its variants.

*Department of Computer Science, Yale University, New Haven, CT, USA, 06520-8285, `elkin@cs.yale.edu`.

†Part of this work was done in School of Mathematics, Institute for Advanced Study, Princeton, NJ, USA, 08540. `elkin@ias.edu`.

# 1 Introduction

## 1.1 Distributed Computing

Consider a *synchronous* network of processors with *unbounded computational power*, modeled by an $n$-vertex graph. The initial knowledge of the processors (henceforth, vertices) is very limited. Specifically, each of them has its own *local* perspective of the network (henceforth, graph), which is confined to its immediate neighborhood. The vertices, however, have to compute some *global* function of the graph, such as its *minimum spanning tree* (henceforth, *MST*).

For this end distributed algorithms (henceforth, *protocols*) are designed. There are several measures of *efficiency* of protocols, but in this paper we restrict our attention to one of them, called the *running time*, which is defined as the number of rounds of distributed communication. On each round of communication at most $B$ bits can be sent through each edge, and $B$ is a parameter of the model. The running time efficiency measure of protocols naturally gives rise to a complexity measure of problems, called *time complexity.*

The design of efficient protocols for this model, as well as proving lower bounds on their efficiency, is a vivid area of study known as *locality-sensitive distributed computing* (henceforth, *distributed computing*) (see [Pel00] and the references therein).

## 1.2 Distributed Approximation and Hardness of Approximation

While traditionally the research in the area of distributed computing concentrated on designing protocols that solve *exactly* the problem at hand, some of the more recent research focuses on providing *approximate* solutions for various distributed problems. Most notably, several approximation protocols were recently devised for the minimum dominating set problem [JRS01, DMPRS03, KW03], and for the minimum edge-coloring problem [PS97, DP95, GP97, DP01, CHK01]. Also, an approximation protocol for the maximum matching problem was recently devised in [CHS03].

However, the situation with lower bounds on approximability of distributed problems is by far less satisfactory. Specifically, the existing results on hardness of distributed approximation can be divided to two categories.

First, there are inapproximability results that are based on lower bounds on the time required for *exact* solution of certain problems, and on integrality of the objective functions of these problems. For example, there is a classical result due to Linial [Lin92] saying that 3-coloring an $n$-vertex ring requires $\Omega(\log^* n)$ time. In particular, it implies that any 3/2-approximation protocol for vertex-coloring problem requires $\Omega(\log^* n)$ time.

Second, there are inapproximability results that assume that the vertices are computationally limited, e.g., are allowed to perform at most polynomial in $n$ number of operations. Obviously, under this assumption any NP-hardness inapproximability result immediately gives rise to an analogous result in the distributed model.

Note, however, that neither of these inapproximability results sheds a new light on our understanding of the limitations of distributed computing. Specifically, the results of this sort are just somewhat different *semantic interpretations* of already known lower bounds. Additionally, we believe that imposing restrictions on the computational power of the vertices is as unnatural as limiting the computational power of the parties in the two-party communication complexity model (see [KN96]). In both cases the abstraction of computationally unbounded vertices or parties is necessary to make possible the study of the role that *communication* plays in computation (see also [Lin92, Pel00]).

To summarize, while sophisticated distributed approximation protocols were developed for various problems, so far no real progress was made in the study of the *hardness of distributed approximation.* In this paper we initiate the systematic study of this subject. Specifically, we study the inapproximability of the distributed *MST* problem, and show *strong unconditional lower bounds on the time-approximation tradeoffs* for this problem and some of its variants.

### 1.3 Distributed *MST* Problem

The (distributed) *MST* problem is one of the most important problems in the area of distributed computing, and was subject of extensive research [GHS83, CT85, Gaf85, Awe87, GKP98, KP98, PR99, LPP01, Elk04].

The most time-efficient protocol known for this problem is due to Elkin [Elk04], and its running time is $O(\mu(G,\omega) \cdot \log^3 n + \sqrt{\frac{n \log^* n}{B}} \log n)$, where $\mu(G,\omega)$ stands for the *MST-radius* of the weighted graph $(G,\omega)$. The definition of the *MST*-radius $\mu(G,\omega)$ is given in Appendix C.1, but for the rest of this discussion it is sufficient to keep in mind that for any graph $(G,\omega)$, $\mu(G,\omega) \leq \Lambda(G) \leq n$, where $\Lambda(G)$ stands for the *unweighted diameter* of the graph $G$.

On the negative side, Peleg and Rubinovich [PR99] have shown a lower bound of $\Omega(\frac{\sqrt{n}}{B})$ on the time complexity of the *MST* problem restricted to graphs of small diameter (at most $O(n^\delta)$ for arbitrarily small positive $\delta > 0$).

In this paper we show that *approximating*[1] the *MST* problem within a ratio $H$ on graphs of small diameter requires $T = \Omega(\sqrt{\frac{n}{H \cdot B}})$ time. In other words, we derive an *unconditional lower bound on the time-approximation tradeoff* for the *MST* problem, specifically, $T^2 \cdot H = \Omega(\frac{n}{B})$. Substituting $H = O(1)$ into this formula shows that approximating the *MST* problem within *any constant factor* requires $\Omega(\sqrt{\frac{n}{B}})$ time, improving the lower bound of [PR99] by a factor of $\sqrt{B}$ (recall that the lower bound of [PR99] applies only for the *exact* solution of the *MST* problem).

Moreover, our lower bound implies that for any $0 < \epsilon < 1$, approximating the *MST* problem within a factor of $\left(\frac{n}{B}\right)^{1-\epsilon}$ requires $\Omega\left(\left(\frac{n}{B}\right)^{\epsilon/2}\right)$ time. The latter means, in particular, that the $\left(\frac{n}{B}\right)^{1-\epsilon}$-approximate *MST* problem is *not a local* problem, i.e., cannot be solved in time polylogarithmic in $n$. This lower bound, like all the other lower bounds that we prove in this paper, applies even to *randomized*[1] protocols.

We remark that lower bounds on time-approximation tradeoffs are rather rare even outside the area of distributed computing. In the standard Turing machine model of computation, a huge body of research on hardness of approximation is being conducted (see, e.g., [AL97] and the references therein). However, it seems that the only lower bounds on time-approximation tradeoffs in this model are implicit. One example is the maximum clique problem, where strengthening the complexity-theoretic assumptions leads to stronger lower bounds [EH00]. Implicit lower bounds on time-approximation tradeoffs were also shown by Chakrabarti et al. [CCGL99] in the *cell-probe* model of computation for the nearest-neighbor problem. Finally, in the context of *property testing* such lower bounds are explicit, but it should be noticed that in that context a different notion of approximation is employed.

### 1.4 Additional Results

One direction of recent research on the distributed *MST* problem was to refine the lower bound of Peleg and Rubinovich [PR99] that applies to the *exact* computation of the *MST* on graphs $G$ with diameter $\Lambda(G) = O(n^\delta)$, $0 < \delta < 1/2$, and to prove similar lower bounds for the *MST* problem restricted to graphs of even smaller diameter. Specifically, Peleg and Rubinovich themselves [PR99] have shown a lower bound of $\Omega(\frac{\sqrt{n}}{B \cdot \log n})$ for the *MST* problem restricted to graphs $G$ of diameter $\Lambda(G) = O(\log n)$, and Lotker et al. [LPP01] have shown lower bounds of $\Omega(\frac{n^{1/3}}{B})$ (respectively, $\Omega(\frac{n^{1/4}}{B})$) for the *MST* problem restricted

[1]See Section 1.5 for the formal definitions of the notions of *approximation* and *randomization* in this context.

| $\Lambda$ | Lower bound on the exact computation | Our improved lower bound on the exact computation | Our lower bound on the time-approximation tradeoff |
|---|---|---|---|
| $n^\delta$, $0 < \delta < 1/2$ | $\Omega(\frac{\sqrt{n}}{B})$ [PR99] | $\Omega(\sqrt{\frac{n}{B}})$ | $T^2 \cdot H = \Omega(\frac{n}{B})$ |
| $\Theta(\log n)$ | $\Omega(\frac{\sqrt{n}}{B \cdot \log n})$ [PR99] | $\Omega(\sqrt{\frac{n}{B \cdot \log n}})$ | $T^2 \cdot H = \Omega(\frac{n}{B \cdot \log n})$ |
| Constant (at least 3) | $\Omega(\frac{n^{1/3}}{B})$ [LPP01] | $\Omega((\frac{n}{B})^{\frac{1}{2} - \frac{1}{2\Lambda - 2}})$ | $T^{2+\frac{2}{\Lambda-2}} \cdot H = \Omega(\frac{n}{B \cdot \Lambda})$ |
| 4 | $\Omega(\frac{n^{1/3}}{B})$ [LPP01] | $\Omega((\frac{n}{B})^{1/3})$ | $T^3 \cdot H = \Omega(\frac{n}{B})$ |
| 3 | $\Omega(\frac{n^{1/4}}{B})$ [LPP01] | $\Omega((\frac{n}{B})^{1/4})$ | $T^4 \cdot H = \Omega(\frac{n}{B})$ |

Table 1: The summary of previously known and new lower bounds on the $MST$ problem restricted to graphs of diameter at most $\Lambda$.

to graphs of diameter $\Lambda(G) \leq 4$ (resp., $\Lambda(G) \leq 3$). Recall that all these lower bounds apply only to the *exact* $MST$ problem.

In addition to the lower bound on the time-approximation tradeoff for the general variant of the $MST$ problem (see Section 1.3), we also show a lower bound on the time-approximation tradeoff for the $MST$ problem restricted to graphs of diameter $\Lambda(G) \leq \Lambda$, for $\Lambda = 3$ and all even $\Lambda$ in the range $4 \leq \Lambda = O(\log n)$. Specifically, denoting the running time of an approximation protocol by $T$, and its approximation ratio by $H$, we show that $T^{2+\frac{2}{\Lambda-2}} \cdot H = \Omega(\frac{n}{\Lambda \cdot B})$.

Note that this result improves all the previous lower bounds for the *exact* computation of the $MST$. Specifically, it improves the result of [PR99] for $\Lambda = O(\log n)$ by a factor of $\sqrt{B \cdot \log n}$, and the results of [LPP01] for $\Lambda = 4$ (resp., $\Lambda = 3$) by a factor of $B^{2/3}$ (resp., $B^{3/4}$). Moreover, our result gives rise to a lower bound of $\Omega((\frac{n}{B})^{1/2-\epsilon})$ for the exact computation (or even approximation within *any constant factor*) of the $MST$ on graphs of constant diameter $O(1/\epsilon)$, significantly improving the previously best-known lower bound of $\Omega(\frac{n^{1/3}}{B})$. Table 1 summarizes the previously known lower bounds on the time complexity of the $MST$ problem restricted to graphs of diameter at most $\Lambda$, parameterized by $\Lambda$, along with our improved lower bounds on this problem.

On the positive side, we devise an $H$-approximation protocol for the $MST$ problem with running time $O(\Lambda(G) + \frac{\omega_{max}}{H-1} \cdot \log^* n)$, where $\omega_{max}$ is the ratio between the maximal and the minimal weight of an edge in the input graph $(G, \omega)$. It follows that the approximate $MST$ problem becomes easy when $\omega_{max}$ is small (our lower bounds on the $H$-approximate $MST$ problem apply for $\omega_{max} = \Omega(\sqrt{n} \cdot H^{3/2})$).

Finally, we also derive some upper and lower bounds on the time-approximation tradeoff of the *shortest-path tree* problem.

**Structure of the paper:** Our main result (the lower bound of $T^2 \cdot H = \Omega(\frac{n}{B})$ on the time-approximation tradeoff for the general variant of the $MST$ problem) is proved in Section 2. In Appendix A we present the extensions of this result to the $MST$ problem restricted to graphs of even smaller (than $n^\delta$, $0 < \delta < 1/2$) diameter. In Appendix B we generalize our lower bounds even further, and show that they apply for $\omega_{max} = \Omega(\sqrt{n} \cdot H^{3/2})$. Appendix C is devoted to our approximation protocols for the $MST$ and $SPT$ problems, and to the lower bound on the time-approximation tradeoff for the $SPT$ problem. Finally, proofs of some lemmata from Section 2 are deferred to Appendix D.

### 1.5 Preliminaries

For a graph $G = (V, E)$, a *spanning tree* is an acyclic connected subgraph $\tau = (V, E')$, $E' \subseteq E$. For a weighted graph $(G = (V, E), \omega)$ with a non-negative weight function $\omega : E \to R$, a *minimum spanning tree (MST)* is a spanning tree $\tau = (V, E')$ with minimum weight $\omega(\tau) = \sum_{e \in E'} \omega(e)$.

An *H-approximate MST* $\tau$ for a graph $(G, \omega)$ is a spanning tree of weight that is at most $H$ times greater than the weight of the $MST$ of the graph $(G, \omega)$. A protocol $\Pi$ is said to be an *H-approximation* for the $MST$ problem if for every input graph $(G, \omega)$ it outputs an $H$-approximate $MST$ $\tau$.

Our lower bounds apply to *randomized* protocols with bounded worst-case running time. In other words, these protocols necessarily terminate within specified time bounds, but they are allowed to err with some constant probability $0 < q < 1/2$. Two possible types of error are allowed. First, a protocol may produce a subgraph of the input graph $(G, \omega)$ that is not an $H$-approximate $MST$ of $(G, \omega)$. This subgraph may contain cycles or multiple connectivity components. Secondly, the protocol may return $\perp$, indicating that it failed to compute the correct answer.

# 2 Lower Bounds on Time-Approximation Tradeoffs

## 2.1 CorruptedMail Problem

In this section we show a lower bound on the tradeoff between the possible approximation ratio for the $MST$ problem and the running time of a distributed protocol that may achieve this approximation ratio.

We start with describing the family of graphs that will be used in the proof of our lower bounds.

For a sufficiently large positive integer $n$, let $\Gamma$, $m$ and $p$ be positive integer parameters that satisfy $p \leq \log n$ and $(m+1)\Gamma + \frac{(m+1)^{1+1/p}-1}{(m+1)^{1/p}-1} = n$. Let $d = (m+1)^{1/p}$ (assume that $d$ is integer; non-integrality issues affect only lower-order terms of our results, and are, therefore, ignored).

Consider a family $\mathcal{G}$ of graphs that contains one unweighted $n$-vertex graph $G_n = (V_n, E_n)$ for infinitely many positive integers $n$. The vertex set $V_n$ is comprised of $\Gamma$ vertex-disjoint paths $P_1, P_2, \ldots, P_\Gamma$ with $m+1$ vertices each, and a $d$-regular tree $\tau$ of depth $p$ with its own vertex set $V(\tau)$ (that is disjoint from $\bigcup_{i=1}^{\Gamma} V(P_i)$). Observe that $|V(\tau)| = 1 + d + \ldots + d^p = \frac{d^{p+1}-1}{d-1} = \frac{(m+1)^{1+1/p}-1}{(m+1)^{1/p}-1}$. Let $rt$ be the root of $\tau$, i.e., the only vertex that had degree $d$ in $\tau$; all the other vertices have either $d$ children and a parent, or they have only a parent. The latter vertices are called *leaves* of $\tau$. Let $s = z_0, z_1, \ldots, z_m = r$ denote the leaves of $\tau$. The edge set of the graph $G_n$ consists of the edge set $E(\tau)$ of the tree $\tau$, the edge set $\bigcup_{j=1}^{\Gamma} E(P_j)$ of the $\Gamma$ paths $P_1, P_2, \ldots, P_\Gamma$, and $m+1$ stars $S_i$, $i = 0, 1, \ldots, m$. Let $P_j = (v_j^{(0)}, v_j^{(1)}, \ldots, v_j^{(m)})$, $j = 1, 2, \ldots, \Gamma$, denote the vertices of the path $P_j$, $j = 1, 2, \ldots, \Gamma$. Then the edge set of the star $S_i$, for $i = 0, 1, \ldots, m$, is the set $\{(z_i, v_j^{(i)}) \mid j = 1, 2, \ldots, \Gamma\}$ of edges.

Family $\mathcal{G}^\omega$ of *weighted* graphs contains $2^\Gamma$ $n$-vertex graphs for each $n$ such that $G_n \in \mathcal{G}$. Each of these $2^\Gamma$ graphs has the vertex set $V_n$ and the edge set $E_n$, but the weights of edges are different for at least one edge in any two distinct $n$-vertex graphs of $\mathcal{G}^\omega$. The edges of the paths $P_1, P_2, \ldots, P_\Gamma$, as well as the edges of the tree $\tau$, are all of weight zero in all $2^\Gamma$ graphs. The edges of the stars $S_i$ for $i = 1, 2, \ldots, m-1$ are all of weight infinity in all $2^\Gamma$ graphs. All the edges of the star $S_m$ have unit weight in all $2^\Gamma$ graphs. Each of the $\Gamma$ edges of the star $S_0$ may have weight zero or infinity.

The family $\mathcal{G}$ of unweighted graphs generalizes the family (that we refer to as $\mathcal{G}''$) of unweighted graphs that was used in [LPP01] to prove a lower bound of $\Omega(n^{1/3}/B)$ on the number of rounds that are required for a distributed protocol to compute *exact MST* on graphs of diameter 4. However, the choice of weights above is inherently different from the one used in [LPP01], and this difference will be discussed below.

Consider the $MST$ problem restricted to the family $\mathcal{G}^\omega$ of graphs. We will show that for any $1 \leq$

$H = o(n)$, *any* distributed protocol that given a graph $G \in \mathcal{G}^\omega$ constructs a $H$-approximate *MST* for $G$ requires $\Omega((\frac{n}{p \cdot B \cdot H})^{\frac{1}{2} - \frac{1}{2(2p+1)}})$ rounds.

This proof is done by a reduction from a problem of distributed delivery of information throughout the graphs of family $\mathcal{G}$. The problem, referred by us as CorruptedMail problem, generalizes the mailing problem of [PR99, LPP01] in two senses. First, the family $\mathcal{G}$ of graphs is somewhat more general than the corresponding families $\mathcal{G}'$ and $\mathcal{G}''$ in [PR99, LPP01], and this enables us to get lower bounds that are parameterized on the diameter. Secondly, the mailing problem of [PR99, LPP01] requires *exact* delivery of all the input bits, whereas our CorruptedMail problem allows the protocol to make a restricted number of *one-sided errors* (i.e., some zero input bits might be delivered as ones, but not vice versa). This modification is geared to capture the situation when the (oracle) protocol for *MST* (its existence is assumed by the reduction) does not compute the exact *MST*, as it is assumed in [PR99, LPP01], but rather provides to the reduction some (possibly quite loose) approximation of it.

Let $\alpha$ and $\beta$, $0 < \alpha < \beta < 1$, be two additional parameters of the construction that will be fixed later. For a bit string $\chi \in \{0,1\}^\Gamma$ and an index $j = 1, 2, \ldots, \Gamma$, let $\chi_j$ denote the $j$th bit of $\chi$. The *Hamming weight* of $\chi$, denoted $hwt(\chi)$, is the number of indices $j = 1, 2, \ldots, \Gamma$ such that $\chi_j = 1$. For two bit strings $\chi, \chi' \in \{0,1\}^\Gamma$, the string $\chi'$ is said to *dominate* $\chi$ if for each $j = 1, 2, \ldots, \Gamma$, $\chi_j = 1$ implies $\chi'_j = 1$. Consider some mapping $\phi : \{0,1\}^\Gamma \to \{0,1\}^\Gamma$, and suppose $\phi(\chi) = \chi'$. The mapping $\phi$ is said to make an error of the the *first* (respectively, *second*) *type* in the $j$'th position, if $\chi_j = 0$ and $\chi'_j = 1$ (resp., $\chi_j = 1$ and $\chi'_j = 0$).

CorruptedMail$(\alpha, \beta)$ problem is defined on unweighted graphs $G_n \in \mathcal{G}$. Recall that for each graph $G \in \mathcal{G}$, there are two designated vertices $s$ and $r$, $s, r \in V_n$. The input to this problem is a bit string $\chi \in \{0,1\}^\Gamma$ of length $\Gamma$ with Hamming weight $hwt(\chi) = \alpha\Gamma$. The input is provided to the vertex $s$ only. The output, returned by the vertex $r$, is a string $\chi' \in \{0,1\}^\Gamma$ of Hamming weight at most $\beta\Gamma$, and it is required that the output string $\chi'$ will dominate the input string $\chi$. As the parameters $\alpha$ and $\beta$ are fixed throughout this section, we will refer to CorruptedMail$(\alpha, \beta)$ problem as CorruptedMail problem.

CorruptedMail problem is a generalization of Mail problem. Mail problem was introduced in [PR99], and used later on in [LPP01]. It was defined on graphs of families $\mathcal{G}'$ and $\mathcal{G}''$ (in [PR99] and [LPP01], respectively), and each graph $G \in \mathcal{G}' \cup \mathcal{G}''$ has a pair of designated vertices, $s$ and $r$ (like each graph in the family $\mathcal{G}$ has). The input to this problem is also a bit string $\chi \in \{0,1\}^\Gamma$, and the input is provided to the vertex $s$ only. However, in Mail problem $\chi$ may be an arbitrary bit string of length $\Gamma$, and it is not required to have a Hamming weight of $\alpha\Gamma$. More importantly, the output of Mail problem, returned by the vertex $r$ (as in CorruptedMail problem), should be the bit string $\chi$ (in other words, the output bit string $\chi'$ is required to be equal to the input bit string $\chi$). This condition is far more restrictive then the condition that we introduced in CorruptedMail problem, specifically, that $\chi'$ should dominate $\chi$ and have Hamming weight at most $\beta\Gamma$.

Observe, however, that the restriction that $\chi'$ should dominate $\chi$ guarantees that the errors that are done throughout the delivery of the bit string $\chi$ are *one-sided* (i.e., all the errors are of the first type). This is in contrast to the usual setting of error-correcting codes, where two-sided error is allowed. One of the crucial properties required for a reduction from CorruptedMail problem to approximate *MST* problem to work is that approximating the *MST* with sufficiently small factor causes only one-sided errors in the delivery of the bit string $\chi$. In our reduction, the latter is ensured by an appropriate choice of the weights of the edges in $\mathcal{G}^\omega$. This is the essential difference between our choice of weights of edges for $\mathcal{G}^\omega$, and choices of weights of edges in analogous families of graphs in [PR99] and [LPP01].

We remark that enabling a symmetric two-sided error error in CorruptedMail problem, and using error-correcting codes (with possible list-decoding) directly would also lead to a proof of hardness of distributed approximation of the *MST* problem within some small constant factor (smaller than 2, to the best of our knowledge). The bottleneck in this case is the fact that corrupting a bit string $\chi$ that has Hamming weight $\alpha\Gamma$ in at least $\alpha\Gamma$ arbitrary positions in an arbitrary way may make the string almost indistinguishable from a string that is drawn out of the uniform distribution over all the strings of Hamming weight at most $2\alpha\Gamma$, even from information-theoretic point of view. However, this is not the case when only one-sided error is allowed. Then, as we will show, one can allow $(\beta-\alpha)\Gamma$ corrupted positions for $1 > \beta \gg \alpha > 0$, and still the corrupted string $\chi'$ will carry on a significant amount of information (roughly, $\Gamma \cdot \alpha \cdot \log(1/\beta)$ bits for sufficiently small positive $\alpha > 0$; note that the amount of information carried by $\chi$ is $\Gamma \cdot Entropy(\alpha) = \Gamma \cdot (\alpha \log(1/\alpha) + (1-\alpha)\log(1/(1-\alpha)))$). This way, we are able to prove a hardness of distributed approximation of the *MST* problem within a factor of, roughly, $1/\alpha$, for arbitrarily small $\alpha > 0$.

As we mentioned, to ensure one-sided error of the reduction, an appropriate setting of the weights of edges of graphs in $\mathcal{G}^\omega$ is required. The setting that we described has, however, the drawback of huge ratio, denoted $\omega_{max}$, between the biggest weight of an edge and the smallest one. Specifically, it is $\frac{\infty}{0}$ (this is, by the way, also the case in the reductions of [PR99] and [LPP01] that show lower bounds on the distributed complexity of *exact MST* problem). It is easy to see, however, that this ratio can be reduced to $O(n^2)$. However, if one wants to get an even smaller $\omega_{max}$, it is no longer possible to guarantee a one-sided error in the delivery of the bit string $\chi$ in the reduction. Nevertheless, as we will show in Section B, controlling the ratio between $n^2$ and $\omega_{max}$ controls also the number of errors of the second type. In this case we get to a situation when the delivery may suffer an *asymmetric two-sided error*, with a huge possible number of errors of the first type (specifically, $(\beta-\alpha)\Gamma$ for some $1 > \beta \gg \alpha > 0$), and a reasonably small number of errors of the second type (specifically, some constant fraction of $\alpha\Gamma$). We will show that this way one can achieve the same lower bound on time-approximation tradeoff as in the case that $\omega_{max} = \frac{\infty}{0}$ (that is, $T^2 \cdot H = \Omega(n/B)$), but with $\omega_{max} = \sqrt{n} \cdot H^{3/2}$, which is significantly smaller than $n^2$ for most values of $H$.

**Lemma 2.1** *For any deterministic protocol* $\Pi$ *for the* CorruptedMail$(\alpha,\beta)$ *problem, its set of all possible outputs* $\{\Pi(\chi) \mid \chi \in \{0,1\}^\Gamma, hwt(\chi) = \alpha\Gamma\}$ *contains at least* $\Omega(2^{l(\alpha,\beta)\Gamma})$ *elements.*

**Proof:** Let $\mathcal{A} = \{\chi \in \{0,1\}^\Gamma \mid hwt(\chi) = \alpha \cdot \Gamma\}$ be the set of all possible bit strings that may serve as input for the vertex $s$. Let $\Upsilon = \{\chi' \in \{0,1\}^\Gamma \mid hwt(\chi') \le \beta \cdot \Gamma\}$ be the set of all possible bit strings that may be returned by the vertex $r$. Consider the bipartite graph $(\mathcal{A}, \Upsilon, E(\mathcal{A},\Upsilon))$ with $\mathcal{A}$ serving as the set of left-hand vertices, $\Upsilon$ serving as the set of right-hand vertices, and $E(\mathcal{A},\Upsilon) = \{(\chi,\chi') \mid \chi \in \mathcal{A}, \chi' \in \Upsilon \text{ s.t. } \chi' \text{ dominates } \chi\}$. Observe that $|\mathcal{A}| = \binom{\Gamma}{\alpha\Gamma} = \frac{\Gamma!}{(\alpha\Gamma)!((1-\alpha)\Gamma)!}$. Consider some bit string $\chi' \in \Upsilon$. The number of the bit strings $\chi \in \mathcal{A}$ that are dominated by $\chi'$ is at most $D = \binom{\beta\Gamma}{\alpha\Gamma} = \frac{(\beta\Gamma)!}{((\beta-\alpha)\Gamma)!(\alpha\Gamma)!}$.

A subset $\Upsilon' \subseteq \Upsilon$ is said to *dominate* the set $\mathcal{A}$, if for each bit string $\chi \in \mathcal{A}$ there exists a bit string $\chi' \in \Upsilon'$ that dominates $\chi$. As each bit string $\chi' \in \Upsilon$ may dominate at most $D$ bit strings of $\mathcal{A}$, it follows that any subset $\Upsilon' \subseteq \Upsilon$ that dominates $\mathcal{A}$ has cardinality at least

$$|\Upsilon'| \ge \frac{|\mathcal{A}|}{D} = \frac{\Gamma!((\beta-\alpha)\Gamma)!}{((1-\alpha)\Gamma)!(\beta\Gamma)!} .$$

Using Stirling formula to approximate the factorials, we get

$$|\Upsilon'| \;\ge\; \left(\frac{(\beta-\alpha)^{\beta-\alpha}}{(1-\alpha)^{1-\alpha}\beta^\beta}\right)^\Gamma \cdot \sqrt{\frac{\beta-\alpha}{(1-\alpha)\beta}} \cdot (1-o(1)) .$$

For $\beta$ and $\alpha$ such that $\beta - \alpha > 0$ is at least some constant it follows that $|\Upsilon'| = \Omega\left(\left(\frac{(\beta-\alpha)^{\beta-\alpha}}{(1-\alpha)^{1-\alpha}\beta^\beta}\right)^\Gamma\right)$.

Let $h(\alpha,\beta) = \frac{(\beta-\alpha)^{\beta-\alpha}}{(1-\alpha)^{1-\alpha}\beta^\beta}$, and let

$$l(\alpha,\beta) = \log h(\alpha,\beta) = (\beta-\alpha)\log(\beta-\alpha) - (1-\alpha)\log(1-\alpha) - \beta\log\beta . \tag{1}$$

It follows that $|\Upsilon'| = \Omega(2^{l(\alpha,\beta)\Gamma})$, for any subset $\Upsilon' \subseteq \Upsilon$ of bit strings that dominate $\mathcal{A}$. ∎

We next show that for any protocol $\Pi$ with worst-case running time at most $t$ for $t$ in certain range, its set of possible outputs (over all possible inputs) is small, namely, at most exponential in $t$. (To argue this we use determinism of the protocol, and that the vertex $r$ that returns the output could not get too much information about the input.) It will follow that the running time $t$ of a protocol $\Pi$ cannot be too small, unless the protocol $\Pi$ is incorrect. To prove an upper bound on the size of the set of possible outputs of any correct protocol, we show that the set of all possible *configurations* of the vertex $r$ is small as function of $t$.

We now get back to the formal argument. For some fixed sufficiently large $n$, consider again the graph $G = (V, E) = G_n \in \mathcal{G}$ that was described in the beginning of the section. Intuitively, we next argue that information can be delivered through $G_n$ from $s$ to $r$ in quite a slow rate. Similar statement was proven in [PR99, LPP01] regarding somewhat different families of graphs $\mathcal{G}'$ and $\mathcal{G}''$.

Our proof has a similar structure to that of [PR99], but as the structure of the family $\mathcal{G}$ is somewhat more complicated than that of $\mathcal{G}'$ or $\mathcal{G}''$, the proof requires more delicate argument. Basicly, all the three proofs (due to [PR99], [LPP01], and ours) construct a sequence of low-capacity cuts, and argue that each bit has to cross all these cuts. As the cuts have low capacities, no cut can be crossed by "many" bits simultaneously, implying a lower bound on the number of rounds of distributed computation. However, while the choice of the cuts and the proof that they have low capacity are rather straightforward in [PR99, LPP01], it is somewhat more involved in our case.

The lower bounds in [LPP01] on *exact* computation of the $MST$ on graphs with constant diameter (specifically, 3 and 4) are no bigger than $\Omega(n^{1/3}/B)$. Our construction enables to get a lower bound of $\Omega((n/B)^{1/2-\epsilon})$ for any $\epsilon > 0$ for graphs with constant diameter (equal to $O(1/\epsilon)$) on *approximate* computation of the $MST$. For the specific case of diameter 3 and 4 our construction improves the lower bounds of [LPP01] by factors $B^{3/4}$ and and $B^{2/3}$, respectively.

We remark that one could use the family of graphs of [PR99] to prove a lower bound on the time-approximation tradeoff of the $MST$ problem restricted to graphs with diameter at most polynomial in $n$, and the family $\mathcal{G}$ in our proof is required to prove such a tradeoff for this problem restricted to graphs of constant diameter. This simpler version of our result (that uses the family $\mathcal{G}'$ of [PR99] and does not introduce the family $\mathcal{G}$) could actually be proved by defining appropriate weights for the edges of the graphs of the family $\mathcal{G}'$, and proving the lower bound on distributed complexity of the CORRUPTEDMAIL problem on $\mathcal{G}'$. (However, the different choice of weights is crucial to ensure the one-sided error, as was already discussed.)

For a rooted tree $(\tau', rt')$, let the *ancestor-descendent* (resp., *parent-child*) relation, denoted $AD(\tau', rt')$ (resp., $PC(\tau', rt')$), be the set of pairs of vertices $(u, w) \in V(\tau')^2$ such that the vertex $u$ is an ancestor (resp., parent) of the vertex $w$ in the tree $(\tau', rt')$. For a vertex $u \in V(\tau')$, let $par_{(\tau',rt')}(u)$ denote the *parent* of $u$ in the rooted tree $(\tau', rt')$.

Recall that the graph $G = G_n$ contains as a subgraph the $d$-regular rooted tree $(\tau, rt)$ of height $p$, with $m+1$ leaves $s = z_0, z_1, \ldots, z_m = r$. For $i = 0, 1, 2, \ldots, m$, let $\tau(i)$ denote the connected subtree of $\tau$ with minimal number of vertices, such that its set of leaves, $Leaves(\tau(i))$, is equal to $\{z_i, z_{i+1}, \ldots, z_m\}$. Let the *root* of $\tau(i)$, denoted $rt(\tau(i))$, be the closest vertex of $\tau(i)$ to the root $rt$ of the tree $\tau$.

Observe that for any pair of vertices $(x, y) \in AD(\tau, rt)$ such that $x, y \in V(\tau(i))$ for some $i = 0, 1, \ldots, m$, $(x, y) \in AD(\tau(i), rt(\tau(i)))$. Also, $AD(\tau(i), rt(\tau(i))) \subseteq AD(\tau, rt)$, for each $i = 0, 1, \ldots, m$. Note that both statements are true for the parent-child relation $PC$ as well.

We define the *tail* sets, $T_0, T_1, \ldots, T_m$ as follows. The tail set $T_0$ contains the entire vertex set $V$ of $G$ except the vertex $s$, i.e., $T_0 = V \setminus \{s\}$. For $i = 1, 2, \ldots, m$, $T_i = \{v_j^{(i')} \mid j = 1, 2, \ldots, \Gamma, i' = i, i+1, \ldots, m\} \cup V(\tau(i))$. Fix also some arbitrary total order $\psi$ on on the vertex set $V$, and let $\vec{T_i} = \psi(T_i)$ denote the ordered sequence of elements of $T_i$, for $i = 0, 1, \ldots, m$.

Consider some protocol $\Pi$ for the CorruptedMail problem, and consider an execution $\varphi_\chi$ of this protocol on some input bit string $\chi$. Following [PR99], let the *state* of the vertex $v$ at the beginning of round $t$ during the execution $\varphi_\chi$ of the protocol $\Pi$, denoted $\sigma(v,t,\chi)$, be the $deg(v)$-tuple of sequences of messages that the vertex $v$ received on its incoming links. For the vertex $s$, the state of $s$ also includes the input string $\chi$. (Other vertices receive no input in the CorruptedMail problem.)

For some subset $U \subseteq V$ of vertices, such that $\vec{U} = (u_1, u_2, \ldots, u_\ell)$, let *configuration* of $U$ in the execution $\varphi_\chi$ at the beginning of round $t$, denoted $C(U,t,\chi)$, be the sequence of states $(\sigma(u_1,t,\chi), \sigma(u_2,t,\chi), \ldots, \sigma(u_\ell,t,\chi))$. Let $\mathcal{C}(U,t)$ denote the collection of all possible configurations of the subset $U$ at the beginning of round $t$ over all possible executions $\varphi_\chi$ (i.e., for all possible legal input bit strings $\chi$; given a bit string, the execution $\varphi_\chi$ is fixed). Let $\rho(U,t)$ denote the cardinality of $\mathcal{C}(U,t)$.

In what follows let us assume that the rounds are indexed starting from 0 and not from 1. Obviously, this may affect the lower bounds by at most an additive term of 1. At the beginning of round $t = 0$, i.e., at the beginning of the execution of the protocol, all the vertices except $s$ are in some fixed initial state, that is independent of the input bit string $\chi$. Hence, $\rho(T_0, 0) = \rho(V \setminus \{s\}, 0) = 1$. We next prove an upper bound on the number of configurations of the tail set $T_t$, after relatively small number of rounds $t \le m-1$.

**Lemma 2.2** *For* $t = 0, 1, \ldots, m-1$, $\rho(T_t, t) \le (2^{B+1} - 1)^{t \cdot p \cdot d}$.

**Proof:** The proof is by induction on $t$. The induction base was argued above. We next prove the induction step. Observe that $T_0 \supseteq T_1 \supseteq \ldots \supseteq T_m$. Suppose we are given a configuration $C \in \mathcal{C}(T_i, i)$. Note that $C$ uniquely determines the messages that are sent at round $i$ by vertices of $T_i$. Therefore, the only messages that may cause multiple configurations of $T_{i+1}$ are those that are sent by the vertices of $V \setminus T_i$ to the vertices of $T_{i+1}$. Denote this set of edges by $E_{i,i+1}$. Observe that the edges of the paths $P_1, P_2, \ldots, P_\Gamma$ do not belong to $\bigcup_{i=0}^{m-1} E_{i,i+1}$. It follows that $E_{i,i+1} \subseteq E(\tau)$, for $i = 0, 1, \ldots, m-1$. For $i = 0$, clearly, $E_{0,1} = \{(z_0, par_{(\tau,rt)}(z_0))\}$. More generally, $E_{i,i+1}$ is a subset of edges of $E(\tau)$ that are incident both to $V(\tau(i+1))$ and to $V \setminus V(\tau(i))$, $i = 0, 1, \ldots, m-1$. See Figure 1 for illustration.

Let $Cut_i$ denote the subset of edges of $E(\tau)$ that are in the cut between $V(\tau(i))$ and the rest of $V(\tau)$, for $i = 1, 2, \ldots, m$. It follows that $E_{i,i+1} \subseteq Cut_{i+1}$, for $i = 0, 1, \ldots, m-1$. We next argue that the size of this cut is never greater than $p \cdot d$.

**Lemma 2.3** *For* $i = 1, 2, \ldots, m$, $|Cut_i| \le p \cdot d$.

**Proof:** Observe that for any edge $(u,w) \in Cut_i$, either $u$ is a parent of $w$ in the tree $\tau$ or vice versa. For a fixed $i = 1, 2, \ldots, m$, let $L_j = \{(u,w) \in Cut_i \cap AD(\tau, rt) \mid d_\tau(rt, u) = j\}$, for $j = 0, 1, \ldots, p-1$, be the *jth layer* of the edge set $Cut_i$. Note that $Cut_i = \bigcup_{j=0}^{p-1} L_j$. It remains to argue that $|L_j| \le d$, for $j = 0, 1, \ldots, p-1$.

To this end, we will show that for any pair of edges $e_1 = (u_1, w_1), e_2 = (u_2, w_2) \in L_j$ (assume, without loss of generality, that $u_1 = par_{(\tau,rt)}(w_1)$ and $u_2 = par_{(\tau,rt)}(w_2)$), $u_1$ is equal to $u_2$. It will follow that all the edges of $L_j$ share a common endpoint, and, furthermore, this endpoint is a parent of all the other endpoints of these edges. As every vertex has at most $d$ children in the tree $\tau$, this would imply $|L_j| \le d$.

Consider a pair of edges $e_1, e_2$ as above. First, suppose $u_1 \notin V(\tau(i))$, $w_1 \in V(\tau(i))$. As the parent-child relation in $\tau(i)$ is a subset of the parent-child relation in $\tau$, it follows that $w_1 = rt(\tau(i))$.

As $d_\tau(rt, w_1) = j+1$, it follows that $w_1$ may be incident to at most one edge of $L_j$, that is, $e_1$. For any other vertex $v \in V(\tau(i)) \setminus \{w_1\}$, $d_\tau(rt, v) > d_\tau(rt, w_1) \ge j+1$. I.e., $d_\tau(rt, v) \ge j+2$. But for any edge of $L_j$, both its endpoints are at distance at most $j+1$ from the root $rt$ of $\tau$. Hence, no edge of $L_j$ is incident to a vertex $v \in V(\tau(i)) \setminus \{w_1\}$, proving that in this case $|L_j| = 1$ (and $e_1 = e_2$).

It remains to consider the case when $u_1, u_2 \in V(\tau(i))$ and $w_1, w_2 \in V(\tau) \setminus V(\tau(i))$. Suppose also that $e_1 \ne e_2$, as otherwise they share a common endpoint $u_1$, and we are done. By definition of $\tau(i)$, as

$u_1, u_2 \in V(\tau(i))$, there exist indices $a, b = i, i+1, \ldots, m$, such that $(u_1, z_a), (u_2, z_b) \in AD(\tau(i), rt(\tau(i)))$. Note that as $\tau(i)$ is acyclic, $a \neq b$. Assume, without loss of generality, that $i \leq a < b$. Consider a descendent $z_c$ of $w_2$. Note that for any two vertices $u_1$ and $u_2$ such that neither $(u_1, u_2)$ nor $(u_2, u_1)$ belong to the ancestor-descendent relation of the rooted tree $(\tau, rt)$, if some leaf-descendent of $u_1$ has smaller index than some leaf-descendent of $u_2$, then *any* leaf-descendent of $u_1$ has smaller index than *any* leaf-descendent of $u_2$. As $d_\tau(rt, u_1) = d_\tau(rt, u_2) = j$, it follows that $(u_1, u_2) \notin AD(\tau, rt)$, $(u_2, u_1) \notin AD(\tau, rt)$. Hence, $a < b$ and $(u_1, z_a), (u_2, z_b), (u_2, z_c) \in AD(\tau, rt)$ imply that $c > a \geq i$. By definition of $\tau(i)$, it follows that $z_c \in V(\tau(i))$. Recall that $u_2 \in V(\tau(i))$ as well. As $\tau(i)$ is a connected subtree of $\tau$, the entire path that connects $u_2$ to $z_c$ in $\tau$ belongs to $\tau(i)$. I.e., in particular, $w_2 \in V(\tau(i))$, contradiction. This proves that $u_1 = u_2$. Hence, $|L_j| \leq d$, implying $|Cut_i| \leq p \cdot d$. ▮

We conclude that $|E_{i,i+1}| \leq p \cdot d$ for $i = 0, 1, \ldots, m-1$. In other words, for $i = 0, 1, \ldots, m-1$, there are at most $p \cdot d$ edges that are incident to the tail set $T_{i+1}$, such that given a configuration $C$ in $\mathcal{C}(T_i, i)$ of the vertices of $T_i$ at the beginning of round $i$, the messages that are sent over these edges at round $i$ are not determined by this configuration.

Recall that at most $B$ bits can be delivered through an edge in each round. Therefore, the number of possible messages that may be sent through an edge in a given direction in one round is at most $\sum_{\ell=0}^{B} 2^\ell = 2^{B+1} - 1$. Observe that there is only one relevant direction of sending messages in our case, that is, towards the vertices of the tail set $T_{i+1}$. Hence, the number of possible messages that may be sent through at most $p \cdot d$ edges in one round is at most $(2^{B+1} - 1)^{p \cdot d}$. It follows that for $i = 0, 1, \ldots, m-1$, $\rho(T_{i+1}, i+1) \leq (2^{B+1} - 1)^{p \cdot d} \cdot \rho(T_i, i)$. This proves Lemma 2.2. ▮

**Lemma 2.4** *The deterministic complexity of* CorruptedMail *problem is* $\Omega(\min\{m, \frac{\Gamma}{p \cdot B \cdot m^{1/p}} \cdot \alpha \cdot \log(1/\beta)\})$.

**Proof:** Consider a protocol $\Pi$ for the CorruptedMail problem, and let $t$ denote its worst-case running time on inputs of fixed size $\Gamma$. By Lemma 2.2, it follows that if $t \leq m-1$ then $\rho(T_t, t) \leq (2^{B+1}-1)^{t \cdot p \cdot d}$. On the other hand, observe that the number of possible configurations of the vertex $r$ upon the termination of the protocol $\Pi$ is at least the cardinality of some subset $\Upsilon' \subseteq \Upsilon$ of a subset of bit strings that dominates $\mathcal{A}$. By Lemma 2.1, $|\Upsilon'| = \Omega(2^{l(\alpha,\beta)\Gamma})$. It follows that the (worst-case) running time $t$ of the protocol $\Pi$ for the CorruptedMail problem is either at least $t \geq m$ rounds, or it satisfies the the following inequality

$$\Omega(2^{l(\alpha,\beta)\Gamma}) = |\Upsilon'| \leq \rho(\{r\}, t) \leq \rho(T_t, t) \leq (2^{B+1} - 1)^{t \cdot p \cdot d} . \tag{2}$$

(The inequality $\rho(\{r\}, t) \leq \rho(T_t, t)$ follows from the fact that $r \in T_t$ for $t = 0, 1, \ldots, m$.) Hence, $\Omega(l(\alpha,\beta)\Gamma) - O(1) \leq t \cdot p \cdot d \cdot B$. I.e., $t = \Omega(l(\alpha,\beta) \cdot \Gamma/(p \cdot d \cdot B))$. In other words, in both cases $t = \Omega(\min\{m, (l(\alpha,\beta) \cdot \Gamma)/(p \cdot d \cdot B)\})$. Recall that $n = O(\Gamma m)$, and $d = m^{1/p}$. Hence,

$$t = \Omega(\min\{m, l(\alpha,\beta) \frac{n}{p \cdot m^{1+1/p} \cdot B}\}) . \tag{3}$$

Recall that $l(\alpha,\beta)$ is given by (1). For small $\alpha > 0$,

$$l(\alpha,\beta) = \alpha \cdot \log(1/\beta) + o(\alpha) . \tag{4}$$

Set $\beta$ to be a constant between 0 and 1. For $\alpha$ either constant or tending to 0 when $n$ grows, we get $t = \Omega(\min\{m, \frac{n \cdot \alpha}{p \cdot m^{1+1/p} \cdot B}\})$. ▮

Set $m = \left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}$. It follows that any deterministic protocol $\Pi$ that solves the CorruptedMail problem on *every* input bit string $\chi$ requires $\Omega\left(\left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}\right)$ rounds in the worst-case.

Henceforth, the term "randomized protocol" will be used as a shortcut for "randomized protocol that succeeds with at least some constant probability on every input".

**Lemma 2.5** *Any randomized protocol* $\Pi$ *for the* CorruptedMail *problem requires* $\Omega\left(\left(\frac{n\cdot\alpha}{p\cdot B}\right)^{1/2-\frac{1}{2(2p+1)}}\right)$ *rounds.*

The proof of Lemma 2.5 can be found in Appendix D.

## 2.2 Reduction to the Approximate *MST* Problem

In this section we describe the reduction from the CorruptedMail$(\alpha,\beta)$ problem on the family $\mathcal{G}$ of unweighted graphs to the $\frac{\beta}{\alpha}$-approximate *MST* problem on the family $\mathcal{G}^\omega$ of weighted graphs.

The protocol $\Pi_{Corr}$ for the CorruptedMail$(\alpha,\beta)$ problem proceeds in the following way. Given an instance $(G,\chi)$, $G\in\mathcal{G}$, $\chi\in\mathcal{A}$, of the CorruptedMail problem, the vertex $s$ computes the weights of edges $(s,v_j^{(0)})$, $j=1,2,\ldots,\Gamma$, in the following way. If $\chi_j=0$ then the weight of the edge $(s,v_j^{(0)})$ is set to zero, otherwise it is to infinity. All the other weights of edges are set by their endpoints to the values that are determined by the definition of the family $\mathcal{G}^\omega$ of weighted graphs. This setting of weights is performed locally by every vertex, and requires no distributed computation.

Next, the vertices invoke $\frac{\beta}{\alpha}$-approximation protocol $\Pi$ for the obtained instance $G(\chi)$ of the *MST* problem. Upon the termination of the protocol, each vertex $v$ knows which edges among the edges that are incident to $v$ belong to the approximate *MST* tree $\tau_0$ for $G(\chi)$, that was constructed by the protocol. The vertex $r$ calculates the output bit string $\chi'\in\{0,1\}^\Gamma$ in the following way. For each index $j=1,2,\ldots,\Gamma$, if the edge $(r,v_j^{(m)})$ belongs to the tree $\tau_0$, the vertex $r$ sets $\chi'_j=1$. Otherwise, it sets $\chi'_j=0$. Finally, the vertex $r$ returns the bit string $\chi'$.

Observe that whenever the construction of the approximate *MST* tree $\tau_0$ is completed, the computation of the bit string $\chi'$ is performed locally by the vertex $r$, and requires no distributed computation. It follows that the running time of the obtained protocol $\Pi_{Corr}$ for the CorruptedMail$(\alpha,\beta)$ problem is precisely equal to the running time of the $\frac{\beta}{\alpha}$-approximation protocol $\Pi$ for the *MST* problem.

**Lemma 2.6** *For each* $\chi\in\mathcal{A}$*, if* $\tau_0$ *is a* $\frac{\beta}{\alpha}$*-approximate MST for* $G(\chi)$*, then the bit string* $\chi'$*, that is returned by the protocol* $\Pi_{Corr}$*, has Hamming weight at most* $\beta\cdot\Gamma$*, and, in addition, the bit string* $\chi'$ *dominates the input bit string* $\chi$*.*

The proof of Lemma 2.6 can be found in Appendix D.

Therefore, if $\Pi$ is a $\frac{\beta}{\alpha}$-approximation protocol for *MST* on the family $\mathcal{G}^\omega$ of weighted graphs, then $\Pi_{Corr}$ is a protocol for the CorruptedMail$(\alpha,\beta)$ problem on the family $\mathcal{G}$ of unweighted graphs, with the same running time. Recall that any (deterministic or randomized) protocol for the CorruptedMail$(\alpha,\beta)$ problem on the family $\mathcal{G}$ of graphs requires $t=\Omega\left(\left(\frac{n\cdot\alpha}{p\cdot B}\right)^{1/2-\frac{1}{2(2p+1)}}\right)$ rounds. Observe that all the graphs in the family $\mathcal{G}^\omega$ have the same unweighted diameter $2p+2$. The next theorem follows.

**Theorem 2.7** *Any randomized* $H$*-approximation protocol for the MST problem on graphs of diameter at most* $\Lambda$ *for* $\Lambda=4,6,8,\ldots$ *requires* $T=\Omega\left(\left(\frac{n}{H\cdot\Lambda\cdot B}\right)^{1/2-\frac{1}{2(\Lambda-1)}}\right)$ *rounds of distributed computation. I.e.,* $T^{2+\frac{2}{\Lambda-2}}\cdot H=\Omega\left(\frac{n}{\Lambda\cdot B}\right)$ .

In particular, for any $\epsilon>0$, approximation of *MST* on graphs with constant diameter $\Lambda\geq 4$ within a factor of $\left(\frac{n}{B}\right)^{1-\epsilon}$ requires at least $\Omega(\left(\frac{n}{B}\right)^{\epsilon(1/2-\frac{1}{2\Lambda-2})})$ rounds of distributed computation.

**Acknowledgements:** The author is grateful to Oded Regev, Michael Langberg, Avi Wigderson, Alexander Razborov, David Peleg, Zvika Lotker and Vitaly Rubinovich for helpful discussions.

## References


[Awe87] B. Awerbuch, Optimal distributed algorithms for minimum weight spanning tree, counting, leader election and related problems. In *Proc. 19th ACM Symp. on Theory of Computing*, pp. 230-240, May 1987.

[AGLP89] B. Awerbuch, A. Goldberg, M. Luby and S. Plotkin, Network decomposition and locality in distributed computation. in *Proc. 30th IEEE Symp. on Foundations of Computer Science*, pp. 364-369, 1989.

[AL97] S. Arora, C. Lund. Hardness of Approximation, in *Approximation Algorithms for NP-hard Problems*, D. Hochbaum Ed., pp. 399-440, (PWS Publishing Company), 1997.

[CHK01] A. Czygrinow, M. Hanckowiak, and M. Karonski, Distributed O(Delta logn)-edge-coloring algorithm, in *Proc. 9th Annual European Symp. on Algorithms*, LNCS 2161, pp. 345-355, 2001.

[BG92] D. Bertsekas and R.G. Gallager, *Data Networks.* 2nd Edition. Prentice-Hall International, London, 1992.

[CCGL99] A. Chakrabarti, B. Chazelle, B. Gum and A. Lvov, A lower bound on the complexity of approximate nearest-neighbor searching on the Hamming cube, *Proc. 31st Ann. ACM Symp. Theory Comput.*, pp. 305-311, 1999.

[CHS03] A. Czygrinow, M. Hanckowiak, E. Szymanska, Distributed Algorithm for Approximating the Maximum Matching, manuscript.

[CT85] F. Chin and H.F. Ting, An almost linear time and $O(n \log n + e)$ messages distributed algorithm for minimum-weight spanning trees, in *Proc. 26th IEEE Symp. on Foundations of Computer Science*, Los Alamitos, CA, 1985, pp.257-266.

[DMPRS03] D. Dubhasi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivisan, Fast Distributed Algorithm for (Weakly) Connected Dominating Sets and Linear-Size Skeletons. In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms*, pp. 717-724, 2003.

[DP95] G. De Marco and A. Pelc, Near-optimal distributed edge coloring, *Proc. 3rd Annual European Symp. on Algorithms*, pp. 448-459, 1995.

[DP01] G. De Marco and A. Pelc, Fast Distributed Graph Coloring with $O(\Delta)$ colors, In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms*, 2001.

[Elk04] M. L. Elkin, A Faster Distributed Protocol for Constructing Minimum Spanning Tree, In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms*, pp. 352-361, New Orleans, LA, USA, 2004.

[EH00] L. Engebrethsen and J. Holmerin, Towards optimal lower bounds for clique and chromatic number, in *Proc. to 27th Inter. Colloq. on Automata, Languages and Programming*, pp. 2-12, 2000.

[GHS83] R.G. Gallager, P.A. Humblet and P.M. Spira, A distributed algorithm for minimum-weight spanning trees, *ACM Trans. on Programming Lang. and Syst.*, 5:66-77, 1983.

[GKP98] J.A. Garay, S. Kutten and D. Peleg, A sublinear time distributed algorithm for minimum-weight spanning trees, *SIAM J. Comput.*, 27:302-316, 1998.

[GGR96] O. Goldreich, S. Goldwasser, and D. Ron, Property testing and its connection to learning and approximation, In *Proc. of 37th IEEE Symp. on Foundations of Computer Science*, 1996, pp. 339-348.

[Gaf85] E. Gafni, Improvements in the time complexity of two message-optimal election algorithms, in *Proc. 4th Symp. on Principles of Distributed Computing*, ACM, New York, 1985, pp. 175-185.

[GP97] D. Grable and A. Panconesi, Nearly optimal distributed edge colouring in O(log log n) rounds, J. of Algorithms, Vol. 10, No. 3, pp. 385-405, May 1997.

[JRS01] L. Jia, R. Rajaraman, and R. Suel, An Efficient Distributed Algorithm for Constructing Small Dominating Sets, In *Proc. of the 20th ACM Symp. on Principles of Distributed Computing*, pp. 33-42, 2001.

[KN96] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge Press, 1996.

[KP98] S. Kutten and D. Peleg, Fast distributed construction of $k$-dominating sets and applications, *J. Algorithms*, 28:40-66, 1998.

[KW03] F. Kuhn and R. Wattnhofer, Constant-Time Distributed Dominating Set Approximation, to appear in *Proc. of the 22nd ACM Symp. on Principles of Distributed Computing*, Boston, Massachusetts, USA, July 2003.

[Lin92] N. Linial, Locality in distributed graph algorithms. *SIAM J. Comput.*, 21:193-201,1992.

[LPP01] Z. Lotker, B. Patt-Shamir and D. Peleg, Distributed MST for Constant Diameter Graphs, In *Proc. 20th ACM Symp. on Principles of Distributed Computing*, pp. 63-72, Newport, Rhode Island, August, 2001.

[Pel00] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*, SIAM, Philadelphia, PA, 2000.

[PR99] D. Peleg and V. Rubinovich, A near-tight lower bound on the time complexity of distributed MST construction, In *Proc. 40th IEEE Symp. on Foundations of Computer Science*, pp.253-261, 1999.

[PS97] A. Panconesi and A. Srinivasan, Randomized Distributed Edge Coloring via an Extension of the Chernoff-Hoeffding Bounds, SIAM J. on Computing, Vol. 26, No. 2, pp. 350-368, 1997.

[Rub99] V. Rubinovich, Distributed minimum spanning tree construction. M. Sc. Thesis, Bar-Ilan University, Ramat-Gan, Israel, 1999.

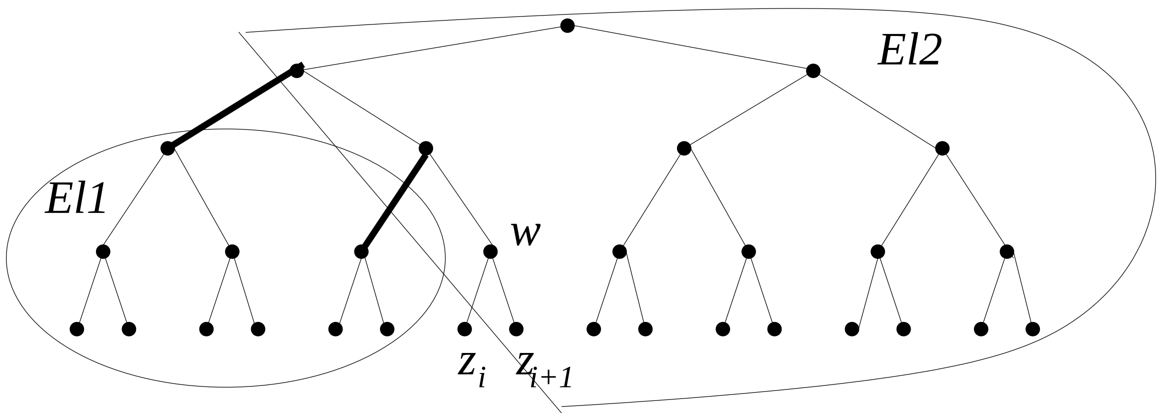


Figure 1: The sets $V(\tau) \setminus V(\tau(i))$ and $V(\tau(i+1))$ are depicted by ellipses $El1$ and $El2$, respectively. Edges of $E_{i,i+1}$ are depicted by thick solid lines. Note that the edge $(z_i, w)$ does not belong to $E_{i,i+1}$, because $z \in V(\tau(i))$.

# A Dependence on the Diameter

We next argue that a stronger by a factor of $\sqrt{\log n}$ lower bound on the time-approximation tradeoff can be shown, if the requirement that diameter is at most $O(\log n)$ is relaxed. Specifically, if one allows the diameter to depend polynomially on $n$, but still insists that it will be asymptotically negligible with respect to the lower bound on time $T$, then the aforementioned tradeoff can be improved to $T^2 \cdot H = \Omega(\frac{n}{B})$.

For this end, consider the following family $\tilde{\mathcal{G}}$ of unweighted graphs that contains one $n$-vertex graph $\tilde{G}_n$ for infinitely many positive integers $n$. Fix some $n$ such that $\tilde{G}_n \in \tilde{\mathcal{G}}$, and denote $\tilde{G} = \tilde{G}_n$. The vertex set $\tilde{V}$ of the graph $\tilde{G}$ consists of $\Gamma$ paths $P_1, P_2, \ldots, P_\Gamma$, each of length $m^2$ (as in the definition of family $\mathcal{G}$, $\Gamma$ and $m$ are sufficiently large positive integer that will be fixed later on). I.e., for $j = 1, 2, \ldots, \Gamma$, $P_j = (v_j^{(0)}, v_j^{(1)}, \ldots, v_j^{(m^2)})$. In addition, the vertex set $\tilde{V}$ contains a path $P^\tau$ (the path $P^\tau$ plays in the family $\tilde{\mathcal{G}}$ a role that is analogous to the one the tree $\tau$ plays in the family $\mathcal{G}$) of length $m$. I.e., $P^\tau = (s = z_0, z_1, \ldots, z_m = r)$. The edge set $\tilde{E}$ of the graph $\tilde{G}$ contains, in addition to the paths $P_1, P_2, \ldots, P_\Gamma$, and the path $P^\tau$, the stars $S_0, S_1, \ldots, S_m$, where $S_i = \{(z_i, v_j^{(i \cdot m)}) \mid j = 1, 2, \ldots, \Gamma\}$, for $i = 0, 1, \ldots, m$. This completes the description of the family $\tilde{\mathcal{G}}$ of unweighted graphs. This family is a slight generalization of the family $\mathcal{G}'$ of unweighted graphs, that was introduced in [PR99]. Specifically, in [PR99], $\Gamma = m^2$. Observe that the the diameter of the graph $G$ is $\Omega(m)$.

Consider the CorruptedMail$(\alpha, \beta)$ problem on family $\tilde{\mathcal{G}}$ of unweighted graphs. We next argue that it requires $\Omega(\min\{m, \Gamma/B\})$ rounds of distributed computation. The proof proceeds similarly to the proof of an analogous fact for the family $\mathcal{G}$ of unweighted graphs, but it is simpler, because the graphs of the family $\tilde{\mathcal{G}}$ have simpler structure than the graphs of family $\mathcal{G}$. The *tails* sets are $T_0 = \tilde{V} \setminus \{s\}$, $T_i = \{v_j^{(i)}, v_j^{(i+1)}, \ldots, v_j^{(m^2)} \mid j = 1, 2, \ldots, \Gamma\} \cup \{z_{i'} \mid i' = \lfloor i/m \rfloor, \lfloor i/m \rfloor + 1, \ldots, m\}$, $i = 1, 2, \ldots, m$. As in Lemma 2.2, $T_0 \supseteq T_1 \supseteq \ldots \supseteq T_m$, and let $\tilde{E}_{i,i+1}$ be the set of edges that connect one of the vertices of $\tilde{V} \setminus T_i$ with one of the vertices of $T_{i+1}$. It is easy to see that $|E_{i,i+1}| \leq 1$ for $i = 0, 1, \ldots, m^2 - 1$. (Recall that the proof of an analogous upper bound, $|E_{i,i+1}| \leq p \cdot m^{1/p}$, for a graph $G \in \mathcal{G}$ is somewhat more complicated.) Now, a lemma, that is analogous to Lemma 2.2, follows.

**Lemma A.1** *For $t = 0, 1, \ldots, m^2 - 1$, $\rho(T_t, t) \leq (2^{B+1} - 1)^t$.*

Exactly the same argument as one that works for family $\mathcal{G}$ shows that any deterministic protocol for the CorruptedMail$(\alpha, \beta)$ problem on family $\tilde{\mathcal{G}}$ either runs for at least $t \geq m^2 - 1$ rounds, or its running time $t$ satisfies the inequality $\rho(\{r\}, t) \geq |\Upsilon'| = \Omega(2^{l(\alpha,\beta)\Gamma})$ (see inequality (2)). Hence,

$$(2^{B+1} - 1)^t \geq \rho(T_t, t) \geq \rho(\{r\}, t) = \Omega(2^{l(\alpha,\beta)\Gamma}) \ .$$

It follows that $t = \Omega(\min\{m^2, l(\alpha, \beta)\Gamma/\beta\})$. Analogous argument to the one that was presented above for the CorruptedMail$(\alpha, \beta)$ problem on the family $\mathcal{G}$, shows that the same (up to a constant factor) lower bound applies to any *randomized* protocol for the CorruptedMail$(\alpha, \beta)$ problem on the family $\tilde{\mathcal{G}}$ as well.

Finally, for the reduction to the *MST* problem, one needs to introduce the family $\tilde{\mathcal{G}}^\omega$ of weighted graphs. The choice of weights of edges is analogous to the choice of weights for family $\mathcal{G}^\omega$, and is inherently different from the one suggested in [PR99], since it is geared to guarantee a *one-sided error* of the delivery, as was discussed above. Specifically, the edges of the paths $P_1, P_2, \ldots, P_\Gamma$, as well as the edges of the path $P^\tau$, all have weight zero. The edges of the stars $S_1, S_2, \ldots, S_{m-1}$ all have weight $\infty$. The edges of the star $S_m$ all have unit weights, and, finally, each edge of the star $S_0$ may weigh either 0 or $\infty$, so that exactly $\alpha\Gamma$ of them have weights $\infty$.

The reduction from the CorruptedMail$(\alpha, \beta)$ problem on family $\tilde{\mathcal{G}}$ to the $\frac{\beta}{\alpha}$-approximate *MST* problem on family $\tilde{\mathcal{G}}^\omega$ is identical to the reduction between these problems on families of graphs $\mathcal{G}$

and $\mathcal{G}^\omega$, respectively. The analysis of the reduction, described in Section 2.2, applies here as well. It follows that any randomized protocol for the $MST$ problem on the family $\tilde{\mathcal{G}}^\omega$ of weighted graphs requires $t = \Omega(\min\{m^2, l(\alpha,\beta)\Gamma/B\})$ rounds. By (4), $t = \Omega(\min\{m^2, \alpha\cdot\Gamma/B\})$. Substitute $m = \sqrt{\alpha\cdot\Gamma/B}$, set $H = 1/\alpha$, and observe that the number of vertices $n$ in $\tilde{G}$ is $O(\Gamma\cdot m^2)$. We conclude that

**Theorem A.2** *For any constant $\delta > 0$, any randomized $H$-approximation protocol for the MST problem on graphs of diameter at most $O(n^\delta)$ requires $T = \Omega(\sqrt{\frac{n}{H\cdot B}})$ rounds of distributed computation. I.e., $T^2\cdot H = \Omega(\sqrt{\frac{n}{B}})$. In particular, for any $\epsilon > 0$, $\left(\frac{n}{B}\right)^{1-\epsilon}$-approximation requires $\Omega(\left(\frac{n}{B}\right)^{\epsilon/2})$ rounds.*

In particular, this improves the result of [PR99] by a factor of $\sqrt{B}$. Specifically, [PR99] have shown a lower bound of $\Omega(\sqrt{n}/B)$ on the number of rounds required for the *exact* computation of the $MST$, while our result gives a stronger by a factor of $\sqrt{B}$ lower bound on the number of rounds required to approximate the $MST$ within any constant approximation factor.

Similarly, our technique enables to get a lower bound on the time-approximation tradeoff for the $MST$ problem on graphs of diameter 3. This is done by replacing the tree $\tau$ in the family $\mathcal{G}$ of unweighted graphs by a clique of size $m+1$. The clique consists of original "leaves", $s = z_0, z_1, \ldots, z_m = r$. The obtained family generalizes the family $\mathcal{G}''$ due to [LPP01]. The family $\mathcal{G}''$ served in [LPP01] to prove a lower bound of $\Omega(n^{1/4}/B)$ on the number of rounds required for exact computation of the $MST$ on graphs of diameter 3. In their construction, $\Gamma = m^3$.

The tail sets are defined exactly as for the family $\mathcal{G}$, except that $\tau(i) = \{z_i, z_{i+1}, \ldots, z_m\}$, for $i = 1, 2, \ldots, m$. It is easy to see that the edge sets $E_{i,i+1}$ are of cardinality $O(m^2)$ for each index $i$, implying that $\rho(T_t, t) \le (2^{B+1}-1)^{t\cdot m^2}$, for each $t = 0, 1, \ldots, m-1$. The rest of the proof is identical to the described above, and it yields a lower bound of $\Omega(\min\{m, \frac{\Gamma\cdot\alpha}{m^2\cdot B}\})$ on the number of rounds required for any randomized protocol for $\frac{\beta}{\alpha}$-approximate $MST$ on graphs of diameter 3. Substitute $m = (n\cdot\alpha/B)^{1/4}$, $H = 1/\alpha$, and observe that the number of vertices $n$ is $O(\Gamma\cdot m)$. The next theorem follows.

**Theorem A.3** *Any randomized $H$-approximation protocol for the MST problem on graphs of diameter at most 3 requires $T = \Omega(\left(\frac{n}{H\cdot B}\right)^{1/4})$ rounds of distributed computation. I.e., $T^4\cdot H = \Omega(\frac{n}{B})$. In particular, for any $\epsilon > 0$, $\left(\frac{n}{B}\right)^{1-\epsilon}$-approximation requires $\Omega(\left(\frac{n}{B}\right)^{\epsilon/4})$ rounds.*

In particular, this improves the lower bound of [LPP01] for *exact* computation of the $MST$ on graphs of diameter 3 by a factor of $B^{3/4}$.

Observe that our time-approximation tradeoff cannot be generalized further to graphs of diameter at most 2, as there is a protocol for exact computation of the $MST$ in $O(\log n)$ rounds on graphs of diameter 1 and 2, due to [Rub99, LPP01].

# B Dependence on $\omega_{max}$ - Asymmetric Two-sided Error

Observe that in the lower bounds that we described in the previous sections, the ratio $\omega_{max}$ between the maximum weight of an edge in a graph and the minimum weight is $\frac{\infty}{0}$. Actually, the weights $\infty$ and 0 were introduced for the clarity of the presentation, and one can carry on the proof if $\omega_{max} = O(n^2)$. However, allowing $\omega_{max}$ to get even smaller than that breaks the reduction from the CorruptedMail problem to the $MST$ problem. Specifically, it can happen that a protocol for the $MST$ problem will produce a $\frac{\beta}{\alpha}$-approximate $MST$ $\tau_0$, and this tree $\tau_0$ will induce (through the reduction) an output bit string $\chi'$ for the CorruptedMail problem that will not dominate the input bit string $\chi$. This is due to to the fact that now some edges of "infinite" weight can be taken into the approximate $MST$ $\tau_0$, while still keeping relatively small total weight of $\tau_0$. However, fortunately, if $\omega_{max}$ is at least $\frac{\sqrt{n}}{\alpha^{3/2}}$, the number of errors of the second type (i.e., indices $j$ such that $\chi_j = 1$ and $\chi'_j = 0$) is small. This enables to prove a similar time-approximation tradeoff to that of Theorem 2.7 for $\omega_{max}$ that is much smaller than $n^2$ (specifically, $\Omega(\frac{\sqrt{n}}{\alpha^{3/2}})$). As was already mentioned, due to our upper bound on distributed approximation of $MST$ (Section C.1), such a tradeoff becomes impossible for $\omega_{max} = o(\sqrt{n \cdot H/B}/\log^* n)$.

The proof utilizes the following observation that may be interesting on its own right. Consider a bit string $\chi$ of length $\Gamma$ of Hamming weight $\alpha\Gamma$. Suppose that an adversary is allowed to introduce into $\chi$ a *huge* number of corruptions of the first type (i.e., to change from 0 to 1 at most $(\beta - \alpha - \epsilon)\Gamma$ positions for some constant arbitrarily small positive $\alpha$ and $\epsilon$ and constant $\beta < 1$ that is arbitrarily close to 1), and it is allowed to introduce a reasonable number of corruptions of the second type (i.e., to change from 1 to 0 at most a constant fraction of $\varrho \cdot \alpha\Gamma$ of positions for some universal constant $0 < \varrho < 1 - 2\epsilon$). Then the obtained bit string $\chi'$ still carries on a significant fraction of the entropy that was carried on by the original bit string $\chi$!

To capture this intuition, we introduce the AsymMail problem (standing for *Asymmetrically Corrupted Mail*). This problem generalizes the CorruptedMail problem, that was discussed in the previous sections. AsymMail$(\alpha, \beta)$ problem is defined on the same family $\mathcal{G}$ of unweighted graphs as the CorruptedMail$(\alpha, \beta)$ problem. The vertex $s$ accepts as input a bit string $\chi$ of Hamming weight $\alpha\Gamma$, and the vertex $r$ outputs a bit string $\chi'$ of Hamming weight at most $\beta\Gamma$. It is required that for each index $j = 1, 2, \ldots, \Gamma$ such that $\chi_j = 1$, $\chi'_j = 1$ as well, *except for* $\frac{\alpha}{10}\Gamma$ *indices* $j$. (For a pair of bit strings $\chi, \chi' \in \{0,1\}^\Gamma$ that satisfy the above condition, we will say that the bit string $\chi'$ *almost dominates* the bit string $\chi$.)

Consider a bipartite graphs $(\mathcal{A}, \Upsilon, E(\mathcal{A}, \Upsilon))$, with $E(\mathcal{A}, \Upsilon) = \{(\chi, \chi') \mid \chi \in \mathcal{A}, \chi' \in \Upsilon, \chi' \text{ almost dominates } \chi\}$. Fix some bit string $\chi' \in \Upsilon$ with Hamming weight $\lambda\Gamma$, $\frac{9}{10}\alpha \le \lambda \le \beta$. The number of the bit strings $\chi \in \mathcal{A}$ that are almost dominated by $\chi'$ is at most

$$D \;=\; \sum_{y} \binom{\lambda\Gamma}{(\alpha - y)\Gamma} \cdot \binom{(1-\lambda)\Gamma}{y\Gamma}\,, \tag{5}$$

where the index $y$ runs over $y = 0, 1/\Gamma, 2/\Gamma, \ldots, \alpha/10$ (the last index $y$ of this sum is the biggest multiple of $1/\Gamma$ that is smaller or equal than $\alpha/10$).

The expression $\binom{\lambda\Gamma}{(\alpha-y)\Gamma}\binom{(1-\lambda)\Gamma}{y\Gamma}$ is a monotone increasing function of $y$, whenever $0 \le y \le \alpha(1-\lambda)$. To ensure that $y$ is in this range in (5), it is sufficient to fix $\beta = 3/4$ (or, actually, any other constant smaller than $9/10$). For this choice of $\beta$,

$$D \;\le\; \frac{\alpha}{10}\Gamma \cdot \binom{\lambda\Gamma}{(9\alpha/10)\Gamma}\binom{(1-\lambda)\Gamma}{(\alpha/10)\Gamma}\,. \tag{6}$$

Let $g(\lambda)$ denote the logarithm of base 2 of the right-hand side expression in (6). This function has positive derivative for $\lambda < 9/10$. Hence, setting $\lambda = \beta < 9/10$ maximizes this expression. It follows that

$$D \;\le\; \frac{\alpha}{10}\Gamma \cdot \binom{\beta\Gamma}{(9\alpha/10)\Gamma}\binom{(1-\beta)\Gamma}{(\alpha/10)\Gamma}\,.$$

A subset $\Upsilon' \subseteq \Upsilon$ is said to *almost dominate* the set $\mathcal{A}$, if for each bit string $\chi \in \mathcal{A}$ there exists a bit string $\chi' \in \Upsilon'$ that almost dominates $\chi$.

It follows that any subset $\Upsilon' \subseteq \Upsilon$ that almost dominates $\mathcal{A}$ has cardinality at least

$$|\Upsilon'| \;\ge\; \frac{|\mathcal{A}|}{D} \;\ge\; \frac{\binom{\Gamma}{\alpha\Gamma}}{\frac{\alpha}{10}\Gamma\binom{\beta\Gamma}{(9\alpha/10)\Gamma}\binom{(1-\beta)\Gamma}{(\alpha/10)\Gamma}}\,.$$

The factor $(\alpha/10)\Gamma$ affects only lower-order terms of the lower bound, and, therefore, can be ignored. Using Stirling formula, and analyzing the logarithm $\hat{l}(\alpha,\beta)$ of the right-hand side expression, we get $|\Upsilon'| \geq 2^{\hat{l}(\alpha,\beta)\Gamma}$, with $\hat{l}(\alpha,\beta) = \hat{l}(0,\beta) + \frac{\partial \hat{l}}{\partial \alpha}(\alpha,\beta)\cdot\alpha + o(\alpha) = \alpha(\frac{9}{10}\log(\frac{1}{\beta}\cdot\frac{9}{10}) + \frac{1}{10}\log(\frac{1}{1-\beta}\cdot\frac{1}{10})) + o(\alpha)$.

Consider $f(q) = q\cdot\log(\frac{1}{\beta}\cdot q) + (1-q)\cdot\log\frac{1-q}{1-\beta}$. Note that $f(1) = \lim_{q\to 1} f(q) = \log\frac{1}{\beta}$, and that $f'(q) = c\cdot\log\left(\frac{q}{1-q}\cdot\frac{1-\beta}{\beta}\right)$ for some universal positive constant $c > 0$. Hence, $f'(q) > 0$ whenever $q > \beta$. As we set $q = 9/10$ and $\beta = 3/4$, it follows that $f(9/10) < \log 1/\beta$. Recall that $l(\alpha,\beta) = \alpha\cdot\log 1/\beta + o(\alpha)$, and the fact that $\hat{l}(\alpha,\beta) < l(\alpha,\beta)$ is consistent with the intuition that the entropy of the corrupted bit string when two-sided errors are allowed is smaller than the entropy of the corrupted bit string when only one-sided errors are allowed.

Now, exactly, the same argument that counts the number of possible configurations of the vertices of the graph $G \in \mathcal{G}$ shows that

**Lemma B.1** *Any protocol for* AsymMail$(\alpha,\beta)$ *problem on family $\mathcal{G}$ of graphs requires* $\Omega(\min\{m, \hat{l}(\alpha,\beta)\frac{\Gamma}{p\cdot m^{1/p}\cdot B})$ *rounds.*

Recall that $n = O(\Gamma\cdot m)$, set $m = (\frac{n\cdot\alpha}{p\cdot B})^{1/2-\frac{1}{2(2p+1)}}$, and a lower bound of $\Omega(m) = \Omega((\frac{n\cdot\alpha}{p\cdot B})^{1/2-\frac{1}{2(2p+1)}})$ follows (this lower bound is only by a constant factor smaller than the analogous lower bound for the CorruptedMail problem).

We next describe the reduction from the AsymMail problem to the approximate $MST$ problem with "small" $\omega_{max}$. Consider the family $\mathcal{G}^{\omega m}$ of weighted graphs. This family is constructed out of the family $\mathcal{G}^{\omega}$ of weighted graphs by the following mapping $\psi$. Given a graph $G = (V,E) \in \mathcal{G}^{\omega}$, the mapping $\psi$ constructs a graph in $\mathcal{G}^{\omega m}$ with the same vertex set and the same edge set, but with slightly different edge weights. Let $\omega_1, \omega_2 > 1$ be two positive integer parameters to be fixed later. For an edge $e \in E$, the new weight $\omega'(e)$ is determined by

$$\omega'(e) = \begin{cases} 1, & \omega(e) = 0, \\ \omega_1, & \omega(e) = 1, \\ \omega_1\cdot\omega_2, & \omega(e) = \infty. \end{cases} \tag{7}$$

Consider the same reduction that given an instance of the CorruptedMail problem produces an instance of the $MST$ problem. Consider a composition of this reduction with the mapping $\psi$. Let $\alpha > 0$ be at most a small positive constant, and let $\beta'$ satisfy $\alpha < \beta' < \frac{100}{101}(\beta-\alpha)$. We obtain a reduction from from AsymMail$(\alpha,\beta)$ problem to the $\frac{\beta'}{\alpha}$-approximate $MST$ problem on graphs with $\omega_{max} = \omega_1\cdot\omega_2$.

Consider a weighted graph $G(\chi) \in \mathcal{G}^{\omega m}$ that is formed by the reduction. Recall that $\chi$ is the input bit string of AsymMail problem, and its Hamming weight is $\alpha\cdot\Gamma$. It follows that precisely $\alpha\cdot\Gamma$ edges of the star $S_0$ have weight $\omega_1\cdot\omega_2$ (this is an analogue of $\infty$), and all the rest $(1-\alpha)\cdot\Gamma$ edges of this star have unit weights. It is easy to see that the $MST$ of $G(\chi)$ contains all the edges of unit weight, and, in addition, precisely $\alpha\cdot\Gamma$ edges of weight $\omega_1$ of the star $S_m$.

It follows that the weight of the $MST$, $\omega'(MST) = \alpha\Gamma\cdot\omega_1 + (n-1-\alpha\Gamma)$. Consider the $\frac{\beta'}{\alpha}$-approximate $MST$ $\tau_0$ that is returned by the oracle approximation protocol for $MST$ (its existence is assumed by the reduction). By definition, $\omega'(MST) \leq \omega'(\tau_0) \leq \frac{\beta'}{\alpha}\cdot(\alpha\Gamma\cdot\omega_1 + (n-1-\alpha\Gamma))$. Consider some index $j = 1,2,\ldots,\Gamma$ such that $\chi_j = 1$, and suppose that $\chi'_j = 0$. As $\chi_j = 1$, the weight of the edge $(s, v_j^{(0)})$ is $\omega_1\cdot\omega_2$. As $\chi'_j = 0$, the edge $(r, v_j^{(m)})$ does not belong to the tree $\tau_0$. As $\tau_0$ is a spanning tree of $G(\chi)$, it follows that at least one edge among $HE_j = \{(z_i, v_j^{(i)}) \mid i = 0,1,\ldots,m-1\}$ belongs to the tree $\tau_0$. Observe that for each edge $e \in HE_j$, $\omega'(e) = \omega_1\cdot\omega_2$ ($HE$ stands for "heavy edges"). Note also that for two distinct indices $j \neq j'$, $j,j' = 1,2,\ldots,\Gamma$, $HE_j \cap HE_{j'}$. It follows that if there are $q$ indices $j$ such that $\chi_j = 1$ and $\chi'_j = 0$, then at least $q$ edges of weight $\omega_1 \cdot \omega_2$ belong to approximate $MST$ $\tau_0$. I.e., $q \cdot \omega_1 \omega_2 \le \beta' \Gamma \cdot \omega_1 + \frac{\beta'}{\alpha}(n - 1 - \alpha\Gamma)$. As we allow at most $\alpha\Gamma/10$ errors of the second type, we impose the following condition on $\omega_1$ and $\omega_2$.

$$(\frac{\alpha}{10}\Gamma + 1)\omega_1 \cdot \omega_2 > \beta' \cdot \Gamma \cdot \omega_1 + \frac{\beta'}{\alpha}(n - 1 - \alpha\Gamma) \ . \qquad (8)$$

Consider some index $j = 1, 2, \ldots, \Gamma$ such that $\chi_j = 0$, and suppose that $\chi'_j = 1$. As $\chi_j = 0$, $\omega'(s, v_j^{(0)}) = 1$. As $\chi'_j = 1$, the edge $(r, v_j^{(m)})$ belongs to the tree $\tau_0$. Recall that $\omega'((r, v_j^{(m)})) = \omega_1$. It follows that $\omega'(\tau_0) \ge \omega_1 \cdot (hwt(\chi') - hwt(\chi)) = \omega_1 \cdot (hwt(\chi') - \alpha\Gamma)$. I.e., $\omega_1 \cdot (hwt(\chi') - \alpha\Gamma) \le \omega'(\tau_0) \le \beta'\Gamma \cdot \omega_1 + \frac{\beta'}{\alpha}(n - 1 - \alpha\Gamma)$. We next impose the following condition on $\omega_1$.

$$\Gamma \cdot \omega_1 \ge 100\frac{n}{\alpha} \ . \qquad (9)$$

Then $\beta' \cdot (\Gamma\omega_1 + \frac{n-1-\alpha\Gamma}{\alpha}) \le \frac{101}{100}\beta'\Gamma \cdot \omega_1$. It follows that $hwt(\chi') - \alpha\Gamma \le \frac{101}{100}\beta'\Gamma$. Recall that $\beta' < \frac{100}{101}(\beta - \alpha)$. Hence, $hwt(\chi') \le (\frac{101}{100}\beta' + \alpha)\Gamma < \beta\Gamma$. Hence, under conditions (8) and (9), the output bit string $\chi'$ almost dominates the input bit string $\chi$. It follows that under these conditions, AsymMail$(\alpha, \beta)$ problem reduces to $\frac{\beta'}{\alpha}$-approximate $MST$ problem, for $\beta' < \frac{100}{101}(\beta - \alpha)$.

The conditions (8) and (9) determine how large $\omega_1$ and $\omega_2$ should be for this reduction to work. Inequality (9) means that $\omega_1 \ge 100\frac{n}{\Gamma \cdot \alpha}$, and to satisfy (8), it is enough to ensure that $\frac{\alpha}{10}\Gamma \cdot \omega_1 \cdot \omega_2 > \beta' \frac{101}{100}\Gamma\omega_1$. The latter implies that $\omega_2 > \frac{\beta'}{\alpha} \cdot \frac{101}{10}$. Hence, $\omega_{max} = \omega_1 \cdot \omega_2 > 1010 \cdot \beta' \cdot \frac{n}{\Gamma \cdot \alpha^2}$. As $\beta'$ is a fixed constant ($\beta' < 1$), it follows that $\omega_{max} = \Theta(\frac{n}{\Gamma \cdot \alpha^2})$ is enough. Combining this with the lower bound on AsymMail$(\alpha, \beta)$ problem (Lemma B.1) implies that approximating $MST$ within a ratio of $\Theta(\frac{1}{\alpha})$ on graphs of diameter $2p + 2$ with $\omega_{max} = \Theta(\frac{n}{\Gamma \cdot \alpha^2})$ requires $\Omega(\min\{m, \hat{l}(\alpha, \beta)\frac{\Gamma}{p \cdot m^{1/p} \cdot B})$ rounds for some constant $\beta < 1$.

Recall that $n = O(\Gamma \cdot m)$. Set $H = 1/\alpha$, and $m = \left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}$. We conclude that

**Theorem B.2** *Any randomized $H$-approximation protocol for the MST problem on graphs of diameter at most $\Lambda$ for $\Lambda = 4, 6, \ldots$ with $\omega_{max} = \Theta\left(\left(\frac{n}{\Lambda \cdot B}\right)^{1/2 - \frac{1}{2(\Lambda-1)}} \cdot H^{3/2 + \frac{1}{2(\Lambda-1)}}\right)$ requires at least $T = \Omega\left(\left(\frac{n}{\Lambda \cdot H \cdot B}\right)^{1/2 - \frac{1}{2(\Lambda-1)}}\right)$ rounds. I.e., $T^{2 + \frac{2}{\Lambda - 2}} \cdot H = \Omega(\frac{n}{\Lambda \cdot B})$. In particular, for $\Lambda = \Theta(\log n)$, $\omega_{max} = \Theta(\left(\frac{n}{B \cdot \log n}\right)^{1/2} \cdot H^{3/2})$, and $T^2 \cdot H = \Omega(\frac{n}{B \cdot \log n})$.*

As for the case of unbounded $\omega_{max}$, the factor $\log n$ in the denominator can be eliminated if one allows bigger diameter (specifically, $O(n^{\delta})$ for any constant positive $\delta > 0$). Also, one can get this way a lower bound for approximate $MST$ on graphs with diameter 3 (Theorem B.2 applies as is), by considering AsymMail$(\alpha, \beta)$ problem on family $\mathcal{G}''$ of graphs (see Theorem A.3 and the discussion that precedes it).

We remark that one cannot expect to get $\omega_{max} = o(\sqrt{\frac{n \cdot H}{B}} / \log^* n)$ in Theorem B.2. This is because by Theorem C.4, there exists an $O(\frac{\omega_{max}}{T'})$-approximation protocol for $MST$ problem with running time $O(Diam(G) + T' \log^* n)$. For $\omega_{max}$ as above, the approximation ratio is $H = O(\omega_{max}/T') = o(\sqrt{\frac{n}{B}} \frac{1}{\log^* n} \cdot \frac{\sqrt{H}}{T'})$. I.e., $T' = o(\sqrt{\frac{n}{BH}} \frac{1}{\log^* n})$, and the running time of the protocol is $T = O(Diam(G)) + o(\sqrt{\frac{n}{BH}}) = o(\sqrt{\frac{n}{BH}})$ (assuming that $Diam(G) = o(\sqrt{\frac{n}{BH}})$; indeed, the lower bounds apply for small diameter, and the upper bound apply for arbitrarily large diameter). This protocol provides $H$-approximation for the $MST$ problem. Hence, $T^2 \cdot H = o(\sqrt{n/B})$, contradicting the stronger by a factor $\log n$ version of Theorem B.2 (that applies for $Diam(G) = O(n^{\delta})$ for any constant $\delta > 0$).

# C Upper Bounds on Time-Approximation Tradeoffs

In this section we devise distributed protocols for the approximate $MST$ and $SPT$ problems. Our first protocol runs in $O(Diam(G)+n^{\epsilon}\cdot\log^{*} n)$ rounds and constructs $O(\frac{\omega_{max}}{n^{\epsilon}})$-approximate $MST$. Our second protocol constructs $O(\frac{\omega_{max}\cdot Diam(G)}{n^{\epsilon}})$-approximate $SPT$ in $O(Diam(G)+n^{\epsilon})$ rounds. (In both results $\epsilon$ is any fixed number between zero and one.) Note that the approximation ratio of $O(\omega_{max})$ is trivial for both problems. Hence, the tradeoff for $SPT$ is meaningful for the case $Diam(G)=O(n^{\epsilon})$.

The two protocols in this section assume that $B=\log n$. Hence the protocols can be used whenever $B=\Omega(\log n)$, and their running time will be the same. Also, they obviously can be adapted for the case $B=o(\log n)$ incurring an overhead of $O(\frac{\log n}{B})=O(\log n)$.

Throughout the section we assume that the weights of edges are scaled between 1 and $\omega_{max}$.

## C.1 Approximate Minimum Spanning Tree

We start with the approximation protocol for the $MST$ problem.

First, we define the graph parameter called the *MST-radius* [Elk04]. For a vertex $u\in V$ and an edge $e\in E$, let $dist_G(u,e)=\min\{dist_G(u,v)\mid v\in e\}$. For a vertex $u\in V$ and a cycle $C$, let the *vertex-edge radius* of the cycle $C$ with respect to vertex $u$, denoted $VxEdgRad_u(C)$, be $\max\{dist_G(u,e')\mid e'\in E(C)\}$. For a vertex $u\in V$ and an edge $e\in E$ that is adjacent to $u$, let $VxEdgElimRad(u,e)=\min\{VxEdgRad_u(C)\mid C\in\mathcal{Z}(e)\}$. Intuitively, $VxEdgElimRad(u,e)$ is the number of rounds required to the vertex $u$ to discover that the edge $e$ that is adjacent to $u$ does not belong to the $MST$. The *MST-radius* of a weighted graph $(G=(V,E),\omega)$ with respect to a vertex $u\in V$, denoted $\mu_u(G,\omega)$, is defined by $\mu_u(G,\omega)=\max\{VxEdgElimRad(u,e)\mid e\in E, u\in e\}$. The *MST-radius* of the graph $(G,\omega)$, denoted $\mu(G,\omega)$, is the $\ell_\infty$-norm of the vector $\vec{\mu}(G,\omega)$, i.e., $\mu(G,\omega)=\max\{\mu_u(G,\omega)\mid u\in V\}$.

Next, consider an $MST$ $\tau_0$ of the graph $G$. A connected subtree of $\tau_0$ is called a *fragment* of $\tau_0$. A *k-MST forest* $\mathcal{F}$ of a graph $G=(V,E)$ is a collection of vertex-disjoint trees that satisfy

1. $\bigcup_{\tau\in\mathcal{F}}V(\tau)=V$, $\bigcup_{\tau\in\mathcal{F}}E(\tau)\subseteq E$.
2. $|V(\tau)|=\Omega(k)$, $Diam(\tau)=O(k)$.
3. There exists an $MST$ $\tau_0$ for the graph $G$, such that each tree $\tau\in\mathcal{F}$ is its fragment.

The notion of *k-MST* forest is related to the notion of $(\sigma,\rho)$ spanning forest of [KP98]. Trees $\tau\in\mathcal{F}$ of $(\sigma,\rho)$ spanning forest $\mathcal{F}$ have to satisfy properties (1) and (2) with $|V(\tau)|\geq\sigma$ and $Diam(\tau)\leq\rho$, but may not satisfy property (3). It was demonstrated in [KP98] that *k-MST* forest of an $n$-vertex graph $G$ can be constructed in $O(k\cdot\log^{*} n)$ rounds of distributed computation.

The first step of our $n^{\epsilon}$-approximation protocol for $MST$ is to construct $n^{\epsilon}$-*MST* forest. This requires $O(n^{\epsilon}\cdot\log^{*} n)$ rounds. The second step is to construct a BFS tree $\tau$ of the entire graph, rooted at some arbitrary vertex $rt=rt(\tau)$. This requires $O(Diam(G))$ rounds. After the construction of the tree $\tau$, each vertex $v$ in the graph knows its unweighted distance to the root $rt$, $d_\tau(rt,v)$. At the third step of the protocol, convergecasts are conducted in parallel over the spanning trees of the fragments $T\in\mathcal{F}$ of the $n^{\epsilon}$-*MST* forest $\mathcal{F}$. Throughout the convergecast over a fragment $T$, the root of the fragment $T$ learns the identity of the vertex $v\in V(T)$ that is closest in $\tau$ to the root $rt$ of $\tau$. The draws are broken arbitrarily. The fourth step involves broadcasts over the spanning trees of the fragments of the identities of these chosen vertices. Both convergecasts and broadcasts are done in parallel (recall that the fragments are vertex-disjoint), and require $O(\max\{Diam(\tau)\mid\tau\in\mathcal{F}\})$ rounds. As $\mathcal{F}$ is an $n^{\epsilon}$-*MST* forest, it follows that $Diam(\tau)=O(n^{\epsilon})$ for every tree $\tau\in\mathcal{F}$. Hence, these steps require $O(n^{\epsilon})$ rounds. After the broadcasts are over, the fifth step occurs. On the fifth step in each fragment $T\in\mathcal{F}$, the chosen vertex $v\in V(T)$ inserts the edge $e_v=(par_{(\tau,rt(\tau))}(v),v)$ into the tree $\tau_0$, that the protocol constructs. It also informs his parent in $\tau$, $par_{(\tau,rt(\tau))}(v)$, that it was chosen, and the parent inserts the edge $e_v$ into $\tau_0$ as well. In addition, each vertex $w$ in the graph inserts into the tree $\tau_0$ the edges of the $k$-$MST$ forest $\mathcal{F}$ that are incident to $w$. Observe that the fifth step requires only one round of distributed computation (for sending the messages by the chosen vertices to their parents in $\tau$).

This completes the description of the protocol. It follows from our discussion that its running time is $O(Diam(G) + n^{\epsilon} \cdot \log^* n)$. We next argue that it is indeed a $\frac{\omega_{max}}{n^{\epsilon}}$-approximation protocol for $MST$.

**Lemma C.1** *The subgraph $\tau_0$, that is constructed by the protocol, is acyclic.*

**Proof:** Suppose for contradiction that there is a cycle $((u_0,w_0),P_0,(u_1,w_1),P_1,\ldots,(u_{t-1},w_{t-1}),P_{t-1}))$, where $P_i$ is a path in some fragment $F_i$ between $w_i$ and $u_{((i+1) \pmod t)}$, and $(u_i,w_i) \in E(\tau)$, $i = 0,1,\ldots,t-1$. Fragments may appear more than once. (Observe that the cycle cannot be contained entirely in one fragment, because for each fragment $T \in \mathcal{F}$, the edges of $\tau$ with both endpoints in $T$ all belong to the spanning tree of $T$.) It follows that $|d_\tau(rt,u_i) - d_\tau(rt,w_i)| = 1$ for $i = 0,1,\ldots,t-1$.

Assume, without loss of generality, that $d_\tau(rt,u_0) - d_\tau(rt,w_0) = 1$. I.e., $u_0 = par_{(\tau,rt)}(w_0)$. Then $u_1 = par_{(\tau,rt)}(w_1)$, because otherwise both edges $(u_0,w_0),(u_1,w_1)$ connect vertices of the same fragment (the vertices $w_0$ and $u_1$) to their parents in the BFS tree $\tau$. However, at most one such an edge is inserted into the tree $\tau_0$ for each fragment in the approximation protocol. It follows that $u_i = par_{(\tau,rt)}(w_i)$ for $i = 0,1,\ldots,t-1$. Hence,

$$d_\tau(rt,u_0) > d_\tau(rt,w_0) \geq d_\tau(rt,u_1) > \ldots \geq \ldots > d_\tau(rt,w_{t-1}) \geq d_\tau(rt,u_0) \ .$$

This is a contradiction, implying that $\tau_0$ is acyclic. ∎

**Lemma C.2** *The subgraph $\tau_0$ is connected, and it is spanning all the vertices of the graph $G$.*

**Proof:** The second assertion follows directly from the observation that the edge set of the tree $\tau_0$ contains the edge set of the $n^{\epsilon}$-$MST$ forest $\mathcal{F}$, and from the definition of a $k$-$MST$ forest.

For the first assertion, consider some pair of vertices $u$ and $w$ in $V$. Let $T_u$ and $T_w$ be the pair of fragments of the $n^{\epsilon}$-$MST$ forest $\mathcal{F}$ such that $u \in T_u$, $w \in T_w$. If $T_u = T_w$ then $u$ and $w$ are connected in $\tau_0$, because the subgraph $\tau_0$ contains a spanning tree of each fragment $T$ of the $n^{\epsilon}$-$MST$ forest $\mathcal{F}$, and, in particular, of $T_u = T_w$. Otherwise, let $u_0 \in T_u$ (resp., $w_0 \in T_w$) be the closest vertex in $T_u$ (resp., $T_w$) to $rt = rt(\tau)$ (in terms of unweighted distance). To prove that there is a path between $u$ and $w$ in $\tau_0$, it suffices to prove that there is a path between $u_0$ and $w_0$. We prove this by induction on $d_\tau(rt,u_0) + d_\tau(rt,w_0)$.

The induction base is the case when the sum is 0, i.e., $u_0 = w_0 = rt$, and then the assertion is obvious.

For the induction step, consider the parent $v = par_{(\tau,rt)}(u_0)$ of the vertex $u_0$ in the BFS tree $\tau$. Observe that $d_\tau(rt,v) = d_\tau(rt,u_0) - 1 < d_\tau(rt,u_0)$, and that the edge $e = (u_0,v)$ belongs to $\tau_0$. Thus, it suffices to prove that there is a path between $v$ and $w_0$ in $\tau_0$. Let $T_v$ be the fragment of $\mathcal{F}$ that contains the vertex $v$, and let $v_0$ be the vertex that was chosen by the convergecast on this fragment. It follows that $d_\tau(rt,v_0) \leq d_\tau(rt,v) < d_\tau(rt,u_0)$, and, thus, the induction hypothesis is applicable to the pair of vertices, $v_0$ and $w_0$. As the fragment $T_v$ is connected, the lemma follows. ∎

It follows from Lemmas C.1 and C.2 that $\tau_0$ is a spanning tree of the graph $G$.

**Lemma C.3** *The tree $\tau_0$ is $O(\frac{\omega_{max}}{n^{\epsilon}})$-approximate $MST$ of the graph $G$.*

**Proof:** Observe that the weight of $n^{\epsilon}$-$MST$ forest $\mathcal{F}$ is at most $\omega(MST)$, and that the convergecast and braodcast procedures insert into $\tau_0$ at most $n^{1-\epsilon}$ additional edges. Hence, $\omega(\tau_0) = O(n^{1-\epsilon} \cdot \omega_{max}) + \omega(MST)$. Thus,

$$\frac{\omega(\tau_0)}{\omega(MST)} = 1 + O\left(\frac{n^{1-\epsilon} \cdot \omega_{max}}{\omega(MST)}\right) \ .$$

Recall that by our assumption, all the weights are scaled between 1 and $\omega_{max}$. Hence, $\omega(MST) \geq n - 1$. It follows that $\omega(\tau_0)/\omega(MST) = O(\omega_{max}/n^\epsilon)$. ∎

To conclude,

**Theorem C.4** *For any $0 < \epsilon < 1$, there exists a protocol that constructs $H$-approximate MST for an $n$-vertex graph $G$ in $O(Diam(G) + n^\epsilon \cdot \log^* n)$ rounds with $H = 1 + O(\frac{\omega_{max}}{n^\epsilon})$. Let $T = n^\epsilon \cdot \log^* n$. Then $T \cdot H = O(\omega_{max} \cdot \log^* n)$.*

Note that for graphs with small $\omega_{max}$ (e.g., constant, or polylogarithmic in $n$) this protocol provides an approximation ratio that is arbitrarily close to 1, and the running time of the protocol is arbitrarily close to $O(Diam(G))$.

## C.2 Approximate Shortest Path Tree

In this section we devise an approximation protocol for the $SPT$ problem.

The protocol is given as input the graph $G = (V, E, \omega)$, and a designated vertex $s \in V$. A spanning tree $\tau_0$ of $G$ is called a $\rho$-approximate $SPT$ with respect to the source $s$, if the weighted distance between $s$ and $v$ in $\tau$, $d^\omega_\tau(s, v)$, is at most $\rho$ times the weighted distance between them in $G$. Protocol $\Pi$ is said to be $\rho$-approximation for $SPT$ problem, if given $G$ and $s$, it constructs a $\rho$-approximate $SPT$ with respect to $s$.

If $Diam(G) \geq n^\epsilon$ then the BFS tree rooted in $s$ serves as $O(\omega_{max}) = O(\omega_{max} \cdot \frac{Diam(G)}{n^\epsilon})$-approximation for $SPT$ with respect to $s$. This tree can be constructed in $O(Diam(G))$ rounds, that is, within the desired time bounds.

In the case that $Diam(G) < n^\epsilon$, the protocol is to run Bellman-Ford protocol for $n^\epsilon$ rounds. In Bellman-Ford protocol each vertex $v$ maintains a distance estimate $d_v$ of the distance between $v$ and $s$. It also keeps an identity (initially *NULL*) of its parent in the tree. Initially, $d_v = \infty$ for each vertex $v \neq s$, and $d_s = 0$. On each round, each vertex sends a message with its distance estimate to all its neighbors. Upon getting the messages from all its neighbors, each vertex $v$ computes $\min\{d_u + \omega((v, u)) \mid (v, u) \in E\}$, and assigns the value of this minimum to its distance estimate $d_v$. It also writes down the identity of the neighbor $u$ such that $d_u + \omega((u, v)) = d_v$, as the identity of its parent. Normally, the protocol is run for $n - 1$ rounds, and this ensures the construction of *exact SPT* with respect to $s$. For more elaborated discussion on Bellman-Ford protocol, its different variants and their analysis, c.f. [BG92]. For the sake of approximation protocol, we run the Bellman-Ford protocol for $n^\epsilon$ rounds.

Let $\tau_0$ denote the constructed tree. It is easy to see that after $n^\epsilon$ rounds of execution of Bellman-Ford protocol, each vertex that is on unweighted distance at most $n^\epsilon$ from $s$ has a non-*NULL* parent in the constructed tree $\tau_0$. As $Diam(G) \leq n^\epsilon$, it follows that $\tau_0$ is a spanning subgraph of $G$. As all the weights are positive, it is acyclic. The subgraph $\tau_0$ is also connected, as all the vertices are connected to $s$. Hence, $\tau_0$ is a spanning tree for the graph $G$. Furthermore, observe that for each vertex $v$ such that one of the weighted shortest paths between $s$ and $v$ contains $n^\epsilon$ or less edges, the distance estimate $d_v$ of $v$ is correct. It follows that the weighted distance between $s$ and $v$ in the tree $\tau_0$ is equal to the weighted distance between them in the graph $G$. For any other vertex $v$ in $G$, the weighted distance between $s$ and $v$ in $G$, $d^\omega_G(s, v)$, is at least $n^\epsilon$. Also, note that there is a path between $s$ and $v$ in $G$ that contains at most $Diam(G) \leq n^\epsilon$ edges. Therefore, this path is of weighted length at most $Diam(G) \cdot \omega_{max}$. The weighted distance between $s$ and $v$ in the tree $\tau_0$, $d^\omega_{\tau_0}(s, v)$, is the minimum weighted length of some path between $s$ and $v$ in $G$ that contains at most $n^\epsilon$ edges. It follows that $d^\omega_{\tau_0}(s, v) \leq Diam(G) \cdot \omega_{max} \leq d^\omega_G(s, v) \cdot \frac{\omega_{max} \cdot Diam(G)}{n^\epsilon}$. Hence, the tree $\tau_0$ is an $O(\frac{\omega_{max} \cdot Diam(G)}{n^\epsilon})$-approximate $SPT$ of $G$ with respect to the vertex $s$. Note that the running time of the protocol is $O(Diam(G))$ in the case that $Diam(G) \geq n^\epsilon$, and $O(n^\epsilon)$ otherwise. I.e., the running time of the protocol is $O(Diam(G) + n^\epsilon)$.

## C.3 Inapproximability of the Shortest-Paths Tree Problem

In this section we show that the problem of approximate computing of the shortest-paths tree (henceforth, $SPT$) on graphs of diameter $\Lambda = 4, 6, \ldots$ within a factor of $O(\sqrt{\frac{\omega_{max}}{(n/(\Lambda B))^{\epsilon}}})$ requires $\Omega((\frac{n}{\Lambda B})^{\epsilon})$ rounds of distributed computation, for any $0 < \epsilon < 1/2 - \frac{1}{2\Lambda - 2}$.

This is done by a reduction from MAIL problem, that is a variant of CORRUPTEDMAIL problem in which no error is allowed. The restriction that the input bit string $\chi$ should have Hamming weight $\alpha\Gamma$ is abondoned, but it is required that the output bit string $\chi'$ should be equal to $\chi$. MAIL problem on a different family of graphs was introduced in [PR99], and was used there and in [LPP01] to prove lower bounds on the number of rounds that are required for *exact* computation of the $MST$.

Consider MAIL problem on the family $\mathcal{G}$ of unweighted graphs. As was demonstrated in inequality (3), any protocol for the more relaxed CORRUPTEDMAIL problem requires $\Omega(\min\{m, \frac{\Gamma}{p \cdot m^{1/p} \cdot B}\})$ rounds. The same statement follows for MAIL problem.

The family $\mathcal{G}'^{\omega m}$ of weighted graphs is formed in an analogous way to the family $\mathcal{G}^{\omega m}$ (see Section B). The only difference is in the way that the weight function $\omega'(e)$ is defined. Let $\gamma > 1$ be a parameter that determines the approximation ratio of a protocol for the $SPT$ problem. We will show a lower bound on the number of rounds that such a protocol requires, parameterized by $\gamma$ (among other parameters).

The weight function $\omega''$ is defined (see the discussion preceding (7) for the context) by

$$\omega''(e) = \begin{cases} 1, & \omega(e) = 0, \\ (m+1)\gamma - (2p-1), & \omega(e) = 1, \\ (m+1)\gamma^2 + \gamma, & \omega(e) = \infty. \end{cases} \tag{10}$$

Next, we describe the reduction from *Mail* problem to the $\gamma$-approximate $SPT$ problem.

Given a graph $G \in \mathcal{G}$, and an input bit string $\chi$ to *Mail* problem, the instance of $SPT$ problem is constructed by assigning weights to the edges of $\mathcal{G}$ analogously to the way it was done in Section B, but according to the weight function $\omega''$ that is defined above. In other words, all the edges that got weight zero (resp., one; resp., infinity) in the the reduction from CORRUPTEDMAIL problem would get weight one (resp., $(m+1)\gamma - (2p-1)$; resp., $(m+1)\gamma^2 + \gamma$) in this reduction. Then the reduction invokes the $\gamma$-approximation protocol for the $SPT$ problem with respect to the source $s$. The existence of such a protocol is assumed by the reduction. The protocol returns a $\gamma$-approximate $SPT$ $\tau_0$ with respect to the source $s$.

For any two vertices $u, w \in V$, let $d^{\omega}_{G}(u, w)$ and $d^{\omega}_{\tau_0}(u, w)$ denote the *weighted* distances in $G$ and $\tau_0$, respectively (the distances are with respect to the weight function $\omega''$, defined by (10)).

Given the tree $\tau_0$, the reduction constructs the output bit string $\chi'$ exactly as the reduction from CORRUPTEDMAIL problem. I.e., for $j = 1, 2, \ldots, \Gamma$, if the edge $(r, v^{(m)}_j)$ belongs to the tree $\tau_0$, the bit $\chi'_j$ is set to 1, otherwise it is set to 0.

We next argue that the reduction solves *Mail* problem correctly.

**Lemma C.5** *The output bit string $\chi'$ of the reduction is equal to its input bit string $\chi$.*

**Proof:** For $j = 1, 2, \ldots, \Gamma$, suppose $\chi_j = 0$. It follows that $\omega''((s, v^{(0)}_j)) = 1$. Hence, $d^{\omega}_{G}(s, v^{(m)}_j) = m + 1$. Suppose for contradiction that $\chi'_j = 1$. Then the edge $(r, v^{(m)}_j)$ belongs to the tree $\tau_0$. In this case either one of the edges of the path $P_j$ does not belong to the tree $\tau_0$, or the edge $(s, v^{(m)}_0)$ does not belong to it, or one of the edges of the (unique) path between $s$ and $r$ in $\tau$ does not belong to $\tau_0$ (since $\tau_0$ is acyclic). In the first two cases, $d^{\omega}_{\tau_0}(s, v^{(m)}_j) \geq 2p + \omega''((r, v^{(m)}_j)) = (m+1)\gamma + 1 > \gamma \cdot d^{\omega}_{G}(s, v^{(m)}_j)$, contradiction to the assumption that the tree $\tau_0$ is $\gamma$-approximate $SPT$ with respect to the vertex $s$. In the third case, $d^{\omega}_{G}(s, r) = 2p$ and $d^{\omega}_{\tau}(s, r) \geq m + 1 + (m+1)\gamma - (2p-1) >> 2p\gamma$ (as $m >> 2p$, since $p = O(\log n)$ and

$m$ is polynomial in $n$), contradiction to the assumption that $\tau_0$ is a $\gamma$-approximate $SPT$ with respect to the vertex $s$. Hence, $\chi_j = 0$ implies $\chi'_j = 0$.

Suppose $\chi_j = 1$. Then $\omega''((s, v_j^{(0)})) = (m+1)\gamma^2 + \gamma$. Then the shortest path between $s$ and $v_j^{(m)}$ in $G$ contains the path between $s$ and $r$ in the tree $\tau$, and uses the edge $e = (r, v_j^{(m)})$. Recall that $\omega''(e) = (m+1)\gamma - (2p-1)$. In other words, $d_G^{\omega}(s, v_j^{(m)}) = (m+1)\gamma - (2p-1) + 2p = (m+1)\gamma + 1$. Suppose for contradiction that $\chi'_j = 0$. It follows that the edge $e$ does not belong to the tree $\tau_0$. Then any path between $s$ and $v_j^{(m)}$ in the approximate $SPT$ $\tau_0$ has to use one of the edges $E^i = \{(z_i, v_j^{(i)}) \mid i = 0, 1, \ldots, m-1\}$. However, each of them has weight $(m+1)\gamma^2 + \gamma$. I.e., $d_{\tau}^{\omega}(s, v_j^{(m)}) \geq (m+1)\gamma^2 + \gamma + 1 > d_G^{\omega}(s, v_j^{(m)}) \cdot \gamma$. (The "plus 1" in the first inequality is because the path between $s$ and $v_j^{(m)}$ has to use at least one edge in addition to an edge of $E^i$). Again, it is a contradiction to the assumption that $\tau_0$ is a $\gamma$-approximate $SPT$ with respect to the vertex $s$. ∎

It follows that $Mail$ problem reduces to the $\gamma$-approximate $SPT$ problem. Hence, any $\gamma$-approximation protocol for $SPT$ on graphs of diameter $2p+2$ with $\omega_{max} = (m+1)\gamma^2 + \gamma$ requires $\Omega(\min\{m, \frac{\Gamma}{p \cdot m^{1/p} \cdot B}\})$ rounds. It follows that $\gamma = \Omega(\sqrt{\frac{\omega_{max}}{m}})$. Set $h > 0$ be a positive parameter. Set $\Gamma = p \cdot B \cdot m^{1+1/p+h}$. Recall that $n = O(\Gamma \cdot m)$. The number of rounds is $\Omega(m) = \Omega\left(\left(\frac{n}{pB}\right)^{\frac{p}{p(h+2)+1}}\right)$. The approximation ratio is $\Omega\left(\sqrt{\frac{\omega_{max}}{(n/(pB))^{\frac{p}{p(h+2)+1}}}}\right)$. For a fixed diameter (i.e., fixed $p$), for any $\epsilon$, $0 < \epsilon \leq \frac{p}{2p+1}$, set $h = \frac{p-\epsilon}{p\epsilon} - 2$. We get $\epsilon = \frac{p}{p(h+2)+1}$. Hence,

**Theorem C.6** *Any $H$-approximation protocol for the SPT problem on graphs of diameter at most $\Lambda$, for $\Lambda = 4, 6, \ldots$ for $H = \Theta\left(\sqrt{\frac{\omega_{max}}{(\frac{n}{\Lambda \cdot B})^{\epsilon}}}\right)$ for some $\epsilon$, $0 < \epsilon \leq 1/2 - \frac{1}{2(\Lambda-1)}$, requires $T = \Omega\left(\left(\frac{n}{\Lambda \cdot B}\right)^{\epsilon}\right)$ rounds of distributed computation. I.e., $T \cdot H^2 = \Omega(\omega_{max})$.*

Similarly to Theorems 2.7 and B.2, Theorem C.6 also generalizes to graphs of diameter $\Lambda = 3$.

# D Some Proofs

**Proof of Lemma 2.5:**

Consider a deterministic protocol $\Pi'$ that accepts as input the uniform distribution over the input bit strings $\chi$ of Hamming weight $\alpha\Gamma$. For such a protocol to suceed with at least a constant probability $q$ in $t \leq m-1$ rounds, the number of configurations of the vertex $r$ at the end of the protocol has to be at least the size of a subset $\Upsilon'' \subseteq \Upsilon$ that dominates at least a fraction $q$ of the set $\mathcal{A}$. However, by the same considerations as above, such a subset $\Upsilon''$ must have cardinality at least $q|\mathcal{A}|/D$. In other words, for protocol $\Pi'$, $\rho(T_t, t) \geq \rho(\{r\}, t) \geq q|\mathcal{A}|/D = \Omega(2^{l(\alpha,\beta)\Gamma})$. In other words, up to lower-order terms, the same lower bound applies also to a protocol $\Pi'$ as above. By Yao's Minimax theorem, any randomized protocol that succeeds with probability at least $q$ on *every* input of the CorruptedMail problem requires at least as many rounds as requires a deterministic protocol $\Pi'$ that succeeds with the same probability on the uniform distribution of inputs. Hence, the lower bound applies to randomized protocols as well. ∎

**Proof of Lemma 2.6:**

Consider a bit string $\chi \in \{0,1\}^{\Gamma}$ of Hamming weight $hwt(\chi) = \alpha\Gamma$. Hence, precisely $\alpha\Gamma$ edges of the star $S_0$ have weight $\infty$ in $G(\chi)$. Therefore, the exact $MST$ contains all the edges of the paths

$P_1, P_2, \ldots, P_\Gamma$ and of the tree $\tau$, as all of them have weight zero, and it contains $(1-\alpha)\Gamma$ edges of weight zero that belong to the star $S_0$. In addition, for each index $j = 1, 2, \ldots, \Gamma$ such that $\omega((s, v_j^{(0)}) = \infty$, it contains the edge $(r, v_j^{(m)})$. Recall that the latter edge has unit weight. Also, note that $\omega((s, v_j^{(0)}) = \infty$ implies that $\chi_j = 1$. As $hwt(\chi) = \alpha\Gamma$, it follows that exactly $\alpha\Gamma$ edges of the star $S_0$ have weight $\infty$. Hence, exactly $\alpha\Gamma$ edges of the star $S_m$ belong to the $MST$, implying that its weight is $\alpha\Gamma$.

Consider a $\frac{\beta}{\alpha}$-approximate $MST$ $\tau_0$. By definition of approximate $MST$, its weight is at most $\beta\Gamma$. Hence, in particular, it contains at most $\beta\Gamma$ edges of the star $S_m$, implying that $hwt(\chi') \leq \beta\Gamma$. Consider an index $j = 1, 2, \ldots, \Gamma$ such that $\chi_j = 1$. It follows that the weight of the edge $(s, v_j^{(0)})$ in $G(\chi)$ is $\infty$. As no edge with infinite weight may belong to $\tau_0$ (as its weight is at most $\beta\Gamma$), it follows that neither $(s, v_j^{(0)})$ nor $(z_i, v_j^{(i)})$ for some $i = 1, 2, \ldots, m-1$ may belong to the tree $\tau_0$. It follows that the edge $(r, v_j^{(m)})$ belongs to $\tau_0$, as otherwise the vertex $s$ would not be connected in $\tau_0$ to the vertices of the path $P_j$. The latter would imply that $\tau_0$ is not a spanning tree of the graph $G(\chi)$, contradicting the assumption that it is an approximate $MST$ for $G(\chi)$. Hence, the edge $(r, v_j^{(m)})$ belongs to the tree $\tau_0$. Hence, the bit $\chi'_j$ is set to 1 by the reduction. It follows that the output bit string $\chi'$ dominates the input bit string $\chi$, and that $hwt(\chi') \leq \beta\Gamma$. ▮