# AN UNCONDITIONAL LOWER BOUND ON THE TIME-APPROXIMATION TRADEOFF FOR THE DISTRIBUTED MINIMUM SPANNING TREE PROBLEM [1]

MICHAEL ELKIN *

**Abstract.**
The design of distributed approximation protocols is a relatively new rapidly developing area of research. However, so far little progress was made 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.

**Key words.** Minimum Spanning Tree, Distributed Algorithms, Hardness of Approximation..

**AMS subject classifications.** 05C05, 05C85,90C27

## 1. Introduction.

### 1.1. Distributed Computing.

Consider a synchronous network of processors with unbounded computational power, modeled by an undirected $n$-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*).

To 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 the running time, defined as the worst-case number of rounds of distributed communication. On each round of communication at most $B$ bits can be sent through each edge, where $B$ is a parameter of the model. The running time efficiency measure of protocols naturally gives rise to a complexity measure of problems, called the time complexity.

The design of efficient protocols for this model, as well as proving lower bounds on their efficiency, is a lively area of study known as locality-sensitive distributed computing (henceforth, *distributed computing*) (see [31] and 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 the problem at hand *exactly*, 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 [22, 9, 25, 35], for the minimum edge-coloring problem [33, 20, 5], for the maximum matching problem [15, 21, 7], and for distance estimation problem [10, 14]. Another relevant important direction is the study of distributed algorithms for finding approximate solutions for positive linear programs initiated by

*Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel, `elkinm@cs.bgu.ac.il`. This work was done in the Department of Computer Science, Yale University, New Haven, CT, USA, 06520-8285, and in School of Mathematics, Institute for Advanced Study, Princeton, NJ, USA, 08540.

[1]Preliminary version of this paper was published in STOC'04 [12].

Papadimitriou and Yannakakis in [34], and continued with the works of Bartal et al. [4], and Kuhn et al. [26]. Recently Moscibroda and Wattenhofer [30], and Grandoni et al. [19] used techniques based on linear programming for developing distributed approximation algorithms for several natural problems.

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 fundamental result due to Linial [27] 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 results of this sort are just somewhat different semantic interpretations of already known lower bounds, and as such, they provide no new insights that were not provided by the original lower bounds. Moreover, we believe that the study of the role of *locality* in distributed computing should be conducted in the cleanest possible model, and, consequently, restricting the computational power of the vertices is obstructive to the goals of this study (see [27, 31]).

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 has been the subject of extensive research [16, 8, 18, 1, 17, 23, 32, 28, 11].

The most time-efficient protocol known for this problem was devised in [11], 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 $(G=(V,E),\omega)$, $\omega : E \to R^+$. The definition of the *MST*-radius $\mu(G,\omega)$ is somewhat involved (see [11]), but for the rest of this discussion it is sufficient to keep in mind that for any graph $(G,\omega)$, $\mu(G,\omega)\le\Lambda(G)\le n$, where $\Lambda(G)$ stands for the *unweighted* diameter of the graph $G$.

On the negative side, Peleg and Rubinovich [32] 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 an arbitrarily small positive $\delta > 0$).

In this paper we show that *approximating* the *MST* problem within a ratio $H$ on graphs of small diameter requires $T = \Omega(\sqrt{\frac{n}{H\cdot B}})$ time. (The result applies to graphs of diameter $\Omega(n^\delta)$ for an arbitrarily small constant $\delta > 0$, exactly like the result of [32]; see Section 2 for the formal definitions of the notions of *approximation* and *randomization* in this context.) 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 [32] by a factor of $\sqrt{B}$ (recall that the lower bound of [32] 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* protocols.[2] We also remark that our lower bounds, as well as the lower bounds of [32, 28], apply to the scenario when the vertices have their own distinct identity numbers at the beginning of computation.

**1.4. Additional Results.** One direction of recent research on the distributed $MST$ problem was to refine the lower bound of Peleg and Rubinovich [32] 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 [32] 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. [28] have shown lower bounds of $\Omega(\frac{n^{1/3}}{B})$ (respectively, $\Omega(\frac{n^{1/4}}{\sqrt{B}})$) for the $MST$ problem restricted to graphs of diameter $\Lambda(G) \le 4$ (resp., $\Lambda(G) \le 3$). For graphs of diameter $\Lambda(G) \le 2$ (respectively, $\Lambda(G) = 1$) upper bounds of $O(\log n)$ (resp., $O(\log \log n)$) were shown in [28] (resp., [29]). Recall that all the aforementioned 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, in this paper we also show a lower bound on the time-approximation tradeoff for the $MST$ problem restricted to graphs of diameter $\Lambda(G) \le \Lambda$, for $\Lambda = 3$ and all even $\Lambda$ in the range $4 \le \Lambda = O(\log n)$. 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 [32] for $\Lambda = O(\log n)$ by a factor of $\sqrt{B \cdot \log n}$, and the results of [28] for $\Lambda = 4$ (resp., $\Lambda = 3$) by a factor of $B^{2/3}$ (resp., $B^{1/4}$). 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 1.1 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, our techniques enable to derive some similar upper and lower bounds on the time-approximation tradeoff of the *shortest-path tree* problem, but they are omitted for the sake of brevity.

| Diameter $\Lambda$ | Previous lower bound on the exact computation | New lower bound on the exact computation | Lower bound on the time-approximation tradeoff |
|---|---|---|---|
| $n^\delta$, $0 < \delta < 1/2$ | $\Omega(\frac{\sqrt{n}}{B})$ [32] | $\Omega(\sqrt{\frac{n}{B}})$ | $T^2 \cdot H = \Omega(\frac{n}{B})$ |
| $\Theta(\log n)$ | $\Omega(\frac{\sqrt{n}}{B \cdot \log n})$ [32] | $\Omega(\sqrt{\frac{n}{B \cdot \log n}})$ | $T^2 \cdot H = \Omega(\frac{n}{B \cdot \log n})$ |
| Constant (at least 3) | $\Omega(\frac{n^{1/3}}{B})$ [28] | $\Omega((\frac{n}{B})^{\frac{1}{2} - \frac{1}{2\Lambda - 2}})$ | $T^{2 + \frac{2}{\Lambda - 2}} \cdot H = \Omega(\frac{n}{B \cdot \Lambda})$ |
| 4 | $\Omega(\frac{n^{1/3}}{B})$ [28] | $\Omega((\frac{n}{B})^{1/3})$ | $T^3 \cdot H = \Omega(\frac{n}{B})$ |
| 3 | $\Omega(\frac{n^{1/4}}{\sqrt{B}})$ [28] | $\Omega((\frac{n}{B})^{1/4})$ | $T^4 \cdot H = \Omega(\frac{n}{B})$ |

TABLE 1.1

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

**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 3. In Section 4 we present the extensions of this result to the *MST* problem restricted to graphs of diameter smaller than $n^\delta$. In Section 5 we generalize our lower bounds even further, and show that they apply for $\omega_{max} = \Omega(\sqrt{n} \cdot H^{3/2})$. In Section 6 we describe our approximation protocol for the *MST* problem.
**Related work:** Our line of research was continued in a recent work of Kuhn et al. [24], who proved remarkable lower bounds on time-approximation tradeoffs for several problems, including the *maximum dominating set* and *maximum matching*. See also the recent survey [13], and references therein.

**2. Preliminaries.** For an undirected 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 $\omega : E \to \mathtt{R}^+$, 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* $\tau$.

In a *distributed* exact (respectively, $H$-approximate) *MST* problem, at the beginning of computation each vertex $v$ knows its own identity number, the identity numbers of all its neighbors, and the weights of the edges adjacent to $v$. Let $E_v$ denote the set of these edges. A correct protocol $\Pi$ must guarantee that at the end of computation every vertex $v$ will know which edges of $E_v$ end up to belong to the computed exact (resp., $H$-approximate) *MST*. Formally, the output sets $K_v$ of different vertices $v$ should satisfy (1) that for every $v \in V$, $K_v \subseteq E_v$, (2) $\bigcup_{v \in V} K_v = T$, where $T$ is the exact (resp., $H$-approximate) *MST* tree, and (3) for every edge $e = (v, w) \in E$, either ($e \in K_v$ and $e \in K_w$) or ($e \notin K_v$ and $e \notin K_w$).

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 $0 \leq q < 1/2$. 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.

For a pair of vertices $u, w \in V$, we denote by $dist_G(u, w)$ the *unweighted* distance between $u$ and $w$ in the graph $G = (V, E)$.

**3. A Lower Bound on the Time-Approximation Tradeoff.** 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.

**3.1. Preliminaries, Overview and Discussion.** 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 $p \le \log n$ and $(m+1)\Gamma + \frac{(m+1)^{1+1/p}-1}{(m+1)^{1/p}-1} = n$. Note that $n = \Theta(\Gamma \cdot m)$. 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 $V_n$ comprises 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 $rt$ denote 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 $S_i$, $i \in \{0, 1, \ldots, m\}$. Let $P_j = (v_j^{(0)}, v_j^{(1)}, \ldots, v_j^{(m)})$, $j \in \{1, 2, \ldots, \Gamma\}$, denote the vertices of the path $P_j$, $j \in \{1, 2, \ldots, \Gamma\}$. Then the edge set of the star $S_i$, for $i \in \{0, 1, \ldots, m\}$, is the set $\{(z_i, v_j^{(i)}) \mid j \in \{1, 2, \ldots, \Gamma\}\}$. (See Figure 3.1.)

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 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 the $2^\Gamma$ graphs. The edges of the stars $S_i$ for $i \in \{1, 2, \ldots, m-1\}$ are all of weight infinity in all the $2^\Gamma$ graphs. All the edges of the star $S_m$ have unit weight in all the $2^\Gamma$ graphs. Each of the $\Gamma$ edges of the star $S_0$ may have weight zero or infinity. This induces $2^\Gamma$ different $n$-vertex graphs in $\mathcal{G}^\omega$.

The family $\mathcal{G}$ of unweighted graphs generalizes the family (that we refer to as $\mathcal{G}''$) of unweighted graphs that was used in [28] 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 [28], 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 $1 \le H = o(n/B)$, *any* distributed protocol that given a graph $G \in \mathcal{G}^\omega$ constructs an $H$-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 [32, 28] in two senses.

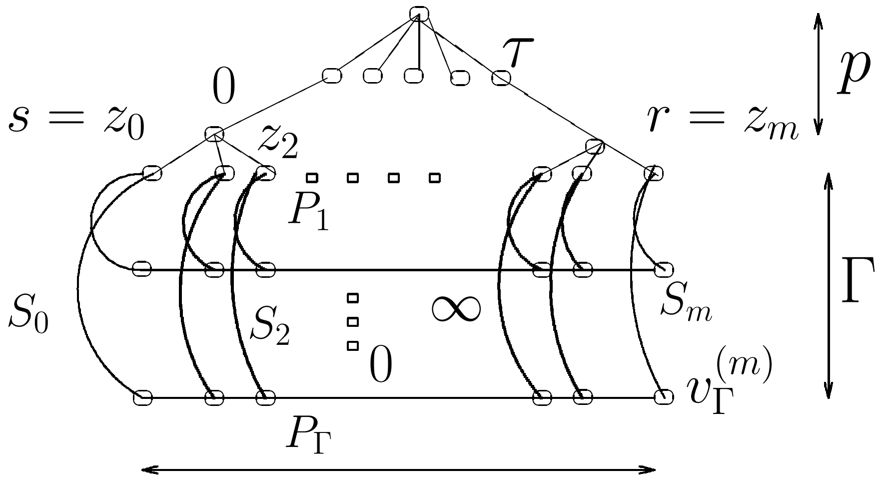


FIG. 3.1. The family $\mathcal{G}$. The edges of the tree $\tau$ and of the paths $P_1, \ldots, P_\Gamma$ have weight zero. The edges of the "internal" stars $S_1, \ldots, S_{m-1}$ have infinite weights. The edges of the right-most star $S_m$ have unit weights, and the edges of the left-most star $S_0$ have weights zero or one, depending on the particular graph.

First, the family $\mathcal{G}$ of graphs is somewhat more general than the corresponding families $\mathcal{G}'$ and $\mathcal{G}''$ in [32, 28], and this enables us to get lower bounds that are parameterized on the diameter. Secondly, the mailing problem of [32, 28] 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 the *MST* problem (its existence is assumed by the reduction) does not compute the exact *MST*, as it is assumed in [32, 28], but rather provides to the reduction some (possibly very loose) approximation of it.

For a bit string $\chi \in \{0,1\}^\Gamma$ and an index $j \in \{1, 2, \ldots, \Gamma\}$, let $\chi_j$ denote the $j$th bit of $\chi$. The *Hamming weight* of $\chi$, denoted $||\chi||$, is the number of indices $j \in \{1, 2, \ldots, \Gamma\}$ 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 $j \in \{1, 2, \ldots, \Gamma\}$, $\chi_j = 1$ 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 $\chi'_j = 0$).

Let $\alpha$ and $\beta$, $0 < \alpha < \beta < 1$, be two additional parameters of the construction that will be fixed later. The $CorruptedMail(\alpha, \beta)$ 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 $||\chi|| = \alpha\Gamma$. 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 and the next sections, we will refer to $CorruptedMail(\alpha, \beta)$ problem as $CorruptedMail$ problem.

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 the choices of weights of edges in analogous families of graphs in [32] and [28].

We remark that allowing a symmetric two-sided error in $CorruptedMail$ problem, and using error-correcting codes (possibly with 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 $\Gamma \cdot \mathit{Entropy}(\alpha) = \Gamma \cdot (\alpha \log(1/\alpha) + (1-\alpha)\log(1/(1-\alpha))))$. This way, we are able to prove a hardness of distributed approximation of the $MST$ problem within a factor of, roughly, $1/\alpha$, for an 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 a very high ratio, denoted $\omega_{max}$, between the largest weight of an edge and the smallest one. Specifically, this ratio is $\Theta(n^2)$. If one wants to prove a similar lower bound for instances of the $MST$ problem with smaller $\omega_{max}$ ratios, 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 5, 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 $\omega_{max} = \Theta(n^2)$ (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 $H(n) = o(n)$. We remark that one cannot expect to achieve such a tradeoff with $\omega_{max} = o(\frac{\sqrt{nH/B}}{\log^* n})$. This is because, as we will show in Section 6, one can approximate the $MST$ on an $n$-vertex graph $G$ within a ratio of $O(\omega_{max}/T')$ in $O(T' \log^* n + \Lambda(G))$ rounds. For $\omega_{max} = o(\frac{\sqrt{nH/B}}{\log^* n})$, the approximation ratio is $H = O(\omega_{max}/T') = o(\frac{\sqrt{nH/B}}{T' \log^* n})$. The number of rounds is $T = T' \log^* n = o(\sqrt{nH/B})$, i.e., $T^2 H = o(\sqrt{n/B})$, contradicting our lower bound.

**3.2. A Lower Bound for the $CorruptedMail$ Problem.** In this section we prove a lower bound on the time complexity of the $CorruptedMail$ problem.

For $0 < \alpha < \beta < 1$, let

$$l(\alpha,\beta) = (\beta-\alpha)\log(\beta-\alpha) - (1-\alpha)\log(1-\alpha) - \beta\log\beta \; . \tag{3.1}$$

Lemma 3.1. *For any $\alpha, \beta$, $0 < \alpha < \beta < 1$,*

$$l(\alpha, \beta) \geq \alpha \cdot \log(1/\beta) \ , \tag{3.2}$$

*Proof.*
Let $f(\alpha, \beta) = l(\alpha, \beta) + \alpha \cdot \log \beta$. It is easy to verify that for a fixed $\beta$, $0 < \beta < 1$, it holds that $f(0, \beta) = 0$. Also,

$$\frac{\partial f}{\partial \alpha}(\alpha, \beta) \ = \ \frac{\partial l}{\partial \alpha}(\alpha, \beta) + \log \beta \ = \ \log\left(\frac{\beta - \beta\alpha}{\beta - \alpha}\right) \ > \ 0$$

for any $\alpha, \beta$ in the range $0 < \alpha < \beta < 1$, and $\frac{\partial f}{\partial \alpha}(0, \beta) = 0$ for any $\beta$, $0 < \beta < 1$. Hence

$$f(\alpha, \beta) = f(0, \beta) + \int_0^\alpha \frac{\partial f}{\partial \alpha}(x, \beta) dx \ \geq \ 0$$

for any $\alpha, \beta$ in the same range. ∎

In the following lemma the $\Omega$-notation hides a universal constant.

Lemma 3.2. *For any deterministic protocol $\Pi$ for the $CorruptedMail(\alpha, \beta)$ problem, its set of all possible outputs $\{\Pi(\chi) \mid \chi \in \{0,1\}^\Gamma, ||\chi|| = \alpha\Gamma\}$ contains at least $\Omega(2^{l(\alpha,\beta)\Gamma})$ elements.*

*Proof.* Let $\mathcal{A} = \{\chi \in \{0,1\}^\Gamma \mid ||\chi|| = \alpha \cdot \Gamma\}$ be the set of all possible bit strings that may serve as input for the vertex $s$. Let $\Upsilon = \{\chi' \in \{0,1\}^\Gamma \mid ||\chi'|| \leq \beta \cdot \Gamma\}$ 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 the left-hand vertices, $\Upsilon$ serving as the set of the right-hand vertices, and $E(\mathcal{A}, \Upsilon) = \{(\chi, \chi') \mid \chi \in \mathcal{A}, \chi' \in \Upsilon$ s.t. $\chi'$ 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

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

Using Stirling formula to approximate the factorials, we get

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

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

Let $h(\alpha, \beta) = \frac{(\beta-\alpha)^{\beta-\alpha}}{(1-\alpha)^{1-\alpha}\beta^\beta}$, and note that $l(\alpha, \beta) = \log h(\alpha, \beta)$. 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 a certain range, its set of possible outputs (over all possible inputs) is at most exponential in $t$. To argue this we use the determinism of the protocol, and the fact 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 relatively small as function of $t$.

For some fixed sufficiently large $n$, consider again a graph $G = (V, E) = G_n \in \mathcal{G}$ that was described in Section 3.1. Intuitively, we next argue that information can be delivered through $G_n$ from $s$ to $r$ in a quite slow rate. Similar statement was proven in [32, 28] regarding somewhat different families of graphs $\mathcal{G}'$ and $\mathcal{G}''$.

Our proof has a similar structure to that of [32], but as the structure of the family $\mathcal{G}$ is somewhat more complicated than that of $\mathcal{G}'$ or $\mathcal{G}''$, the proof requires a more delicate argument. Basically, all the three proofs (due to [32], [28], 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 [32, 28], it is somewhat more involved in our case.

We first need a few definitions. 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 distinct vertices $(u, w) \in V(\tau') \times V(\tau')$ 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 $i \in \{0, 1, 2, \ldots, m\}$, 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 a pair of vertices $x, y \in V(\tau(i))$ such that $x$ is an ancestor of $y$ in the rooted tree $(\tau, rt)$, $x$ is an ancestor of $y$ in the rooted tree $(\tau(i), rt(\tau(i)))$ as well. I.e., $(x, y) \in AD(\tau, rt)$ and $x, y \in V(\tau(i))$ imply that $(x, y) \in AD(\tau(i), rt(\tau(i)))$. Also, $AD(\tau(i), rt(\tau(i))) \subseteq AD(\tau, rt)$, for each $i \in \{0, 1, \ldots, m\}$. Note that both statements are true for the parent-child relation $PC$ as well.

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 $i \in \{1, 2, \ldots, m\}$. We next argue that the size of this cut is never greater than $p \cdot d$.

LEMMA 3.3. *For $i \in \{1, 2, \ldots, m\}$, $|Cut_i| \leq p \cdot d$.*

*Proof.* Fix an index $i \in \{1, 2, \ldots, m\}$. Let $L_j = \{(u, w) \in Cut_i \cap AD(\tau, rt) \mid dist_\tau(rt, u) = j\}$, for $j \in \{0, 1, \ldots, p-1\}$, be the $j$*th layer* of the edge set $Cut_i$. Note that $Cut_i = \bigcup_{j=0}^{p-1} L_j$. It remains to argue that $|L_j| \leq d$, for $j \in \{0, 1, \ldots, p-1\}$.

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 $|L_j| \leq d$.

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 $w_1 = rt(\tau(i))$. In this case it is easy to that $(u_1, w_1)$ is the only edge in $L_j$, and consequently, $e_1 = e_2$, and $|L_j| = 1$.

It remains to consider the case when $u_1, u_2 \in V(\tau(i))$ and $w_1, w_2 \in V(\tau) \setminus V(\tau(i))$. However, in this case $u_1 \neq u_2$ and $u_1, u_2 \in V(\tau(i))$ imply that either all the descendents of $u_1$ or all the descendents of $u_2$ belong to $V(\tau(i))$, and either of those contradicts $w_1, w_2 \in V(\tau) \setminus V(\tau(i))$. ∎

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 $i \in \{1, 2, \ldots, m\}$, $T_i = \{v_j^{(i')} \mid j \in \{1, 2, \ldots, \Gamma\}, i' \in \{i, i+1, \ldots, m\}\} \cup V(\tau(i))$. It will be more convenient to view the vertex sets $T_i$'s as sequences of vertices. The order of the vertices in these sequences can be arbitrary, and for simplicity we will assume that the vertices are ordered as a monotone increasing sequence of their (distinct) identity numbers.

Consider some protocol $\Pi$ for the $CorruptedMail$ problem, and consider an execution $\varphi_\chi$ of this protocol on some input bit string $\chi$. 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 $U = (u_1, u_2, \ldots, u_\ell) \subseteq V$ of vertices, let the *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 $\mathcal{C}(U, t)$. We remark that our proof does not attempt to analyze the actual sets of configurations, but rather only the cardinalities of those sets.

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 $t \leq m - 1$.

LEMMA 3.4. *For $t \in \{0, 1, \ldots, m-1\}$, $\rho(T_t, t) \leq (2^{B+1} - 1)^{t \cdot p \cdot d}$.*

**Remark:** To provide some intuition for the proof of Lemma 3.4, we illustrate its central idea on a simple example. Consider a path $P = (v_0, v_1, \ldots, v_n)$ with the vertex $v_0$ serving as a sender, and $v_n$ serving as a receiver, that is, $s = v_0$, $r = v_n$. Also, let $T_i = \{v_{i+1}, v_{i+2}, \ldots, v_n\}$ be the tail sets, $i \in \{0, 1, \ldots, n-1\}$.

Suppose we are given a configuration $C$ of the vertices of $T_i$ at the beginning of round $i$. This configuration uniquely determines the messages sent on round $i$ by the vertices of $T_i$, and, in particular, by the vertex $v_{i+1}$. Hence the unique configuration of $T_0$ at the beginning of round 0 (note that $T_0 = V \setminus \{s\}$, and $s$ is the only vertex that receives input) determines the unique configuration of $T_1$ at the beginning of round 1. This is because the only link that connects $T_1$ to $V \setminus T_1$ is the edge $(v_1, v_2)$, but the message sent by the vertex $v_1$ over this link on round 0 is determined by the configuration of $T_0$ at the beginning of round 0, and this configuration is independent of the bit string $\chi$ that the sender $s$ accepts as input.

Applying this consideration inductively we conclude that the configuration of the tail set $T_{n-1} = \{r\} = \{v_n\}$ at the beginning of round $n-1$ is uniquely determined by the initial configuration of the vertices of the set $T_0 = V \setminus \{s\}$, and is independent of the input string $\chi$.

For a more elaborate example consider the graph $H = P \cup S$, where $P$ is the $n$-vertex path that we have just discussed, and $S = \{(v_0, v_i) \mid i \in \{2, 3, \ldots, n\}\}$ is the star rooted at $v_0$. The tail sets are left unchanged.

In this graph the unique configuration of the vertices of $T_0$ at the beginning of round 0 branches into roughly $2^{B(n-2)}$ possible configurations of the vertices of $T_1$ at the beginning of round 1, because the vertices of $T_1$ could receive up to $B(n-2)$ different bits from the vertex $v_0$ on this round, and each distinct combination of these bits may determine a new configuration of the vertices of $T_1$. Repeating this argument we derive that there are roughly $2^{B(n-2)} \cdot 2^{B(n-3)}$ possible configurations of $T_2$ at the beginning of round 2, and, ultimately, at most $2^{O(B \cdot n^2)}$ possible configurations of the tail set $T_{n-1} = \{r\}$ at the beginning of round $n-1$. In other words, intuitively, at most $O(Bn^2)$ bits could have been delivered through this network in $n-1$ rounds.

*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 are not yet determined 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 $i \in \{0, 1, \ldots, m-1\}$. 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 $V \setminus V(\tau(i))$, $i \in \{0, 1, \ldots, m-1\}$. See Figure 3.2 for illustration.

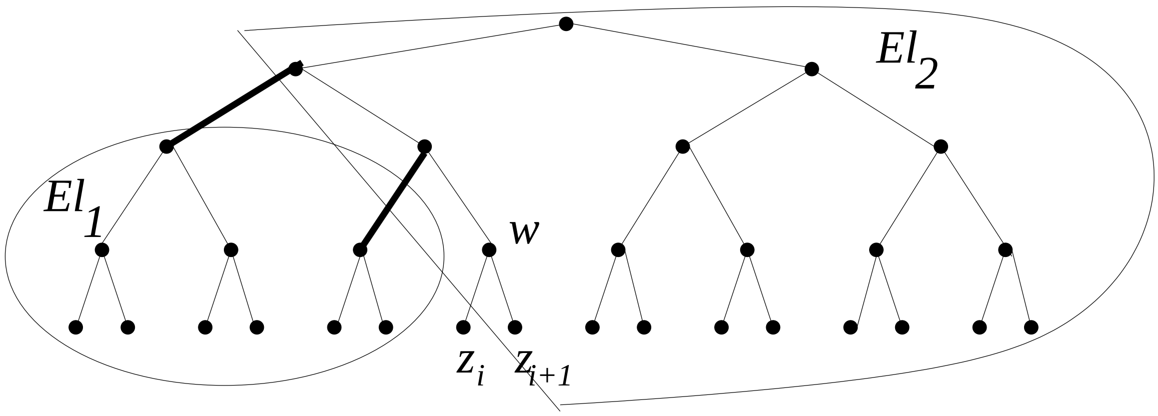


FIG. 3.2. The sets $V(\tau) \setminus V(\tau(i))$ and $V(\tau(i+1))$ are depicted by ellipses $El_1$ and $El_2$, respectively. Edges of $E_{i,i+1}$ are depicted by thick solid lines. Note that, somewhat surprisingly, the edge $(z_i, w)$ does not belong to $E_{i,i+1}$, because $z_i \in V(\tau(i))$.

Note that $E_{i,i+1} \subseteq Cut_{i+1}$, for $i \in \{0, 1, \ldots, m-1\}$. Hence, by Lemma 3.3, $|E_{i,i+1}| \leq p \cdot d$ for $i \in \{0, 1, \ldots, m-1\}$. In other words, for $i \in \{0, 1, \ldots, m-1\}$, 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 $\sum_{\ell=0}^{B} 2^\ell = 2^{B+1} - 1$. (In this summation, the case $\ell = 0$ reflects the possibility of transmitting nothing through a certain edge on a certain round.)

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 $i \in \{0, 1, \ldots, m-1\}$, $\rho(T_{i+1}, i+1) \le (2^{B+1} - 1)^{p \cdot d} \cdot \rho(T_i, i)$. Along with $\rho(T_0, 0) = 1$, this implies the statement of the lemma. ∎

Note that the proof argument that counts the number of configurations holds when each vertex knows its unique identity number at the beginning of computation.

In the following lemma and its proof, the asymptotic $\Omega$- and $O$-notations hide universal constants.

LEMMA 3.5. *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 3.4, it follows that if $t \le m-1$ then $\rho(T_t, t) \le (2^{B+1} - 1)^{t \cdot p \cdot d}$. 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 3.2, $|\Upsilon'| = \Omega(2^{l(\alpha,\beta)\Gamma})$. It follows that the (worst-case) running time $t$ of the protocol $\Pi$ for the *CorruptedMail* problem is either at least $t \ge m$ rounds, or it satisfies the following inequality

$$\Omega(2^{l(\alpha,\beta)\Gamma}) = |\Upsilon'| \le \rho(\{r\}, t) \le \rho(T_t, t) \le (2^{B+1} - 1)^{t \cdot p \cdot d} \ . \tag{3.3}$$

(The inequality $\rho(\{r\}, t) \le \rho(T_t, t)$ follows from the fact that $r \in T_t$ for $t \in \{0, 1, \ldots, m\}$.) Hence, $\Omega(l(\alpha,\beta)\Gamma) - O(1) \le t \cdot p \cdot d \cdot B$. 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 $d = (m+1)^{1/p}$. Hence,

$$t = \Omega(\min\{m, l(\alpha,\beta) \frac{n}{p \cdot m^{1+1/p} \cdot B}\}) \ . \tag{3.4}$$

By Lemma 3.1, for $\alpha > 0$ it holds that $l(\alpha,\beta) \ge \alpha \cdot \log(1/\beta)$. Set $\beta$ to be a constant between 0 and 1, and the lemma follows. ∎

Finally, set $m = \left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}$. It follows that

$$t = \Omega\left(\left(\frac{n \cdot \alpha}{p \cdot B}\right)^{1/2 - \frac{1}{2(2p+1)}}\right) \ . \tag{3.5}$$

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 3.6. *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.*

*Proof.* Consider a deterministic protocol $\Pi'$ that accepts as input a string drawn at random from the uniform distribution over the bit strings $\chi$ of Hamming weight $\alpha\Gamma$. For such a protocol to succeed with at least a constant probability $q$ in $t \le m-1$ 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 $\Pi'$, $\rho(T_t, t) \ge \rho(\{r\}, t) \ge q|\mathcal{A}|/D = \Omega(2^{l(\alpha,\beta)\Gamma})$. In other words, up to lower-order terms, the same lower bound (3.5) applies also to a protocol $\Pi'$ as above. By Yao's Minimax theorem [37], 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 (3.5) applies to randomized protocols as well. ∎

**3.3. 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 (see the beginning of Section 3 for the definition of this family).

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 $(s, v_j^{(0)})$, $j \in \{1, 2, \ldots, \Gamma\}$, 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 set 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 $\mathcal{G}^\omega$. (Recall that there is no freedom in setting other weights, and the only difference between two distinct $n$-vertex graphs from $\mathcal{G}^\omega$ is the setting of the weights of the edges of the star $S_0$.) This setting of weights is performed locally by every vertex, and requires no distributed computation.

Next, the vertices invoke a $\frac{\beta}{\alpha}$-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 $j \in \{1, 2, \ldots, \Gamma\}$, 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.

We next argue that the reduction is correct, i.e., that the protocol $\Pi_{Corr}$ solves correctly the $CorruptedMail(\alpha, \beta)$ problem, assuming that the protocol $\Pi$ is a $\frac{\beta}{\alpha}$-approximation protocol for the *MST* problem.

LEMMA 3.7. *For each $\chi \in \mathcal{A}$, if $\tau_0$ is a $\frac{\beta}{\alpha}$-approximate MST for the graph $G(\chi)$, then the bit string $\chi'$, that is returned by the protocol $\Pi_{Corr}$, has Hamming weight at most $\beta \cdot \Gamma$, and, furthermore, the bit string $\chi'$ dominates the input bit string $\chi$.*

*Proof.* Consider a bit string $\chi \in \{0, 1\}^\Gamma$ of Hamming weight $||\chi|| = \alpha\Gamma$. By construction, 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 $j \in \{1, 2, \ldots, \Gamma\}$ 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 $||\chi|| = \alpha\Gamma$, 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 $||\chi'|| \leq \beta\Gamma$. Consider an index $j \in \{1, 2, \ldots, \Gamma\}$ 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 $i \in \{1, 2, \ldots, m-1\}$ 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 $||\chi'|| \leq \beta\Gamma$. ∎

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 3.8. *Any randomized $H$-approximation protocol for the MST problem on graphs of diameter at most $\Lambda$ for $\Lambda \in \{4, 6, 8, \ldots\}$ 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.

We remark that in addition to being the first result on the hardness of distributed approximation, Theorem 3.8 also implies a lower bound of $\Omega((n/B)^{1/2-\epsilon})$ on the time complexity of the exact computation (or even approximation within any constant factor) of the $MST$ problem restricted to graphs with constant diameter $O(1/\epsilon)$. Previously, the best-known lower bound on exact computation of $MST$ on graphs with constant diameter was $\Omega(n^{1/3}/B)$ due to [28]. However, the lower bound of [28] is stronger in the sense that it requires the protocol to work only on graphs of diameter at most 4. Substituting $\Lambda = 4$ in our result yields an improvement by a factor $B^{2/3}$ to the result of [28].

By substituting $\Lambda = \log \frac{n}{H \cdot B}$, we get

Corollary 3.9. *Any randomized $H$-approximation protocol for the MST problem on graphs of diameter $\Lambda = O(\log n)$ requires $T = \Omega(\sqrt{\frac{n}{H \cdot B \cdot \log n}})$ rounds of distributed computation. I.e., $T^2 \cdot H = \Omega(\frac{n}{B \cdot \log n})$.*

A lower bound of $\Omega(\frac{\sqrt{n}}{B \log n})$ on the running time of a protocol that computes the *MST* exactly on graphs of diameter at most $O(\log n)$ was shown in [32]. Our lower bound (Corollary 3.9) shows, in particular, a stronger by a factor of $\sqrt{B \log n}$ lower bound on the running time of a protocol that computes the *MST* exactly or *approximates* it within *any constant factor* on graphs with diameters in the same range.

**4. Dependence on the Diameter.** In this section we show that if the diameter is $\Omega(n^\epsilon)$ for some constant $\epsilon > 0$, then the lower bound of Corollary 3.9 can be strengthened by a factor of $\sqrt{\log n}$.

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 $j \in \{1, 2, \ldots, \Gamma\}$, $P_j = (v_j^{(0)}, v_j^{(1)}, \ldots, v_j^{(m^2)})$. In addition, the vertex set $\tilde{V}$ contains a path $P^\tau$ (the path $P^\tau$ plays in the family $\tilde{\mathcal{G}}$ the role that is analogous to the one that 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 $S_i = \{(z_i, v_j^{(i \cdot m)}) \mid j \in \{1, 2, \ldots, \Gamma\}\}$, for $i \in \{0, 1, \ldots, m\}$. 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 [32]. Specifically, in [32], $\Gamma = m^2$. 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 the analogous fact for the family $\mathcal{G}$ of unweighted graphs, (see Section 3.2), 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 $T_0 = \tilde{V} \setminus \{s\}$, $T_i = \{v_j^{(i)}, v_j^{(i+1)}, \ldots, v_j^{(m^2)} \mid j \in \{1, 2, \ldots, \Gamma\}\} \cup \{z_{i'} \mid i' \in \{\lfloor i/m \rfloor, \lfloor i/m \rfloor + 1, \ldots, m\}\}$, $i \in \{1, 2, \ldots, m\}$. As in Lemma 3.4, $T_0 \supseteq T_1 \supseteq \ldots \supseteq T_m$, 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 $i \in \{0, 1, \ldots, m^2 - 1\}$. (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 3.4, follows.

LEMMA 4.1. *For* $t \in \{0, 1, \ldots, m^2 - 1\}$, $\rho(T_t, t) \leq (2^{B+1} - 1)^t$.

Exactly the same argument as the 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 (3.3)). Hence,

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

It follows that $t = \Omega(\min\{m^2, l(\alpha, \beta)\Gamma/B\})$. 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 different from the one suggested in [32], since it is geared to guarantee that the delivery will suffer from only *one-sided error*, 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 latter reduction was described in Section 3.3. The analysis of the reduction, described in Section 3.3, 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 (3.1), $t = \Omega(\min\{m^2, \alpha\cdot\Gamma/B\})$. Substitute $m = \sqrt{\alpha\cdot\Gamma/B}$, set $H = 1/\alpha$, and observe that the number of vertices $n$ in $\tilde{G}$ is $O(\Gamma\cdot m^2)$. The diameter $\Lambda$ of the graphs from the family $\mathcal{G}$ is $m+2 = O\left(\left(\frac{n}{HB}\right)^{1/4}\right)$. It is easy to see that by replacing the path $P^\tau$ by $k$ paths $P^\tau_1, P^\tau_2, \ldots, P^\tau_k$ of decreasing lengths $m, m^{1/2}, m^{1/4}, \ldots, m^{1/2^k}$ we obtain a similar lower bound for graphs of diameter $\Lambda = O\left(\left(\frac{n}{HB}\right)^{1/2^{k+2}}\right)$. We conclude that

THEOREM 4.2. *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 [32] by a factor of $\sqrt{B}$. Specifically, [32] 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 [28]. The family $\mathcal{G}''$ served in [28] 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 $i \in \{1, 2, \ldots, m\}$. 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 $t \in \{0, 1, \ldots, m-1\}$. 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 4.3. *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 [28] for exact computation of the *MST* on graphs of diameter 3 by a factor of $B^{1/4}$.

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 [36, 28, 29].

**5. 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 infinity. 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 3.8 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 6), such a tradeoff becomes impossible for $\omega_{max} = o(\sqrt{n \cdot H/B}/\log^* n) = o(\sqrt{n/(B\cdot\alpha)}\cdot \log^* n)$.

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 $j \in \{1,2,\ldots,\Gamma\}$ 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

$$D \;=\; \sum_y \binom{\lambda\Gamma}{(\alpha-y)\Gamma}\cdot\binom{(1-\lambda)\Gamma}{y\Gamma}\,, \tag{5.1}$$

where the index $y$ runs over $y\in\{0,1/\Gamma,2/\Gamma,\ldots,\alpha/10\}$ (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 (5.1), it is sufficient to fix $\beta = 3/4$ (or, actually, any other constant smaller than $9/10$). For this choice of $\beta$,

$$D \;\le\; \frac{\alpha}{10}\Gamma \cdot \binom{\lambda\Gamma}{(9\alpha/10)\Gamma}\binom{(1-\lambda)\Gamma}{(\alpha/10)\Gamma}\,. \tag{5.2}$$

Let $g(\lambda)$ denote the logarithm of base 2 of the right-hand side expression in (5.2). 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 $|\Upsilon'| \ge 2^{\hat{l}(\alpha,\beta)\Gamma}$, 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 5.1. *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 (3.5) 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

$$\omega'(e) = \begin{cases} 1, & \omega(e) = 0,\\ \omega_1, & \omega(e) = 1,\\ \omega_1\cdot\omega_2, & \omega(e) = \infty. \end{cases} \tag{5.3}$$

Consider the same reduction that was described in Section 3.3, 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 the $AsymMail(\alpha, \beta)$ 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 the 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 $j \in \{1, 2, \ldots, \Gamma\}$ 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 from the set $HE_j = \{(z_i, v_j^{(i)}) \mid i \in \{0, 1, \ldots, m-1\}\}$ belongs to the tree $\tau_0$ (*HE* stands for "heavy edges"). Observe that for each edge $e \in HE_j$, $\omega'(e) = \omega_1 \cdot \omega_2$. Note also that for two distinct indices $j \neq j'$, $j, j' \in \{1, 2, \ldots, \Gamma\}$, $HE_j \cap HE_{j'} = \emptyset$. 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., $q \cdot \omega_1\omega_2 \leq \beta'\Gamma \cdot \omega_1 + \frac{\beta'}{\alpha}(n - 1 - \alpha\Gamma)$. As we allow at most $\alpha\Gamma/10$ errors of the second type, we impose the following condition on $\omega_1$ and $\omega_2$.

$$(5.4) \qquad (\frac{\alpha}{10}\Gamma + 1)\omega_1 \cdot \omega_2 > \beta' \cdot \Gamma \cdot \omega_1 + \frac{\beta'}{\alpha}(n - 1 - \alpha\Gamma) \ .$$

Consider some index $j \in \{1, 2, \ldots, \Gamma\}$ 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 $\omega'(\tau_0) \geq \omega_1 \cdot (||\chi'|| - ||\chi||) = \omega_1 \cdot (||\chi'|| - \alpha\Gamma)$. I.e., $\omega_1 \cdot (||\chi'|| - \alpha\Gamma) \leq \omega'(\tau_0) \leq \beta'\Gamma \cdot \omega_1 + \frac{\beta'}{\alpha}(n - 1 - \alpha\Gamma)$. We next impose the following condition on $\omega_1$.

$$(5.5) \qquad \Gamma \cdot \omega_1 \geq 100\frac{n}{\alpha} \ .$$

Then $\beta' \cdot (\Gamma\omega_1 + \frac{n-1-\alpha\Gamma}{\alpha}) \leq \beta' \cdot (\Gamma\omega_1 + \frac{n}{\alpha}) \leq \frac{101}{100}\beta'\Gamma \cdot \omega_1$. It follows that $||\chi'|| - \alpha\Gamma \leq \frac{101}{100}\beta'\Gamma$. Recall that $\beta' < \frac{100}{101}(\beta - \alpha)$. Hence, $||\chi'|| \leq (\frac{101}{100}\beta' + \alpha)\Gamma < \beta\Gamma$. Hence, under conditions (5.4) and (5.5), 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 any $\beta' < \frac{100}{101}(\beta - \alpha)$.

The conditions (5.4) and (5.5) determine how large $\omega_1$ and $\omega_2$ should be for this reduction to work. Inequality (5.5) means that $\omega_1 \geq 100\frac{n}{\Gamma \cdot \alpha}$, and to satisfy (5.4), 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 5.1) 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 5.2. *Any randomized $H$-approximation protocol for the MST problem on graphs of diameter at most $\Lambda$ for $\Lambda \in \{4, 6, \ldots\}$ 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* $T^2\cdot H = \Omega(\frac{n}{B\cdot\log n})$.

By setting $m$ to be smaller than $\left(\frac{n}{HpB}\right)^{1/2-\frac{1}{2(2p+1)}}$, one can get a smaller lower bound on $T$ but with $\omega_{max} = \Theta(mH^2)$. Also, 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 5.2 applies as is), by considering $AsymMail(\alpha,\beta)$ problem on family $\mathcal{G}''$ of graphs (see Theorem 4.3 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 5.2. This is because by Theorem 6.4, there exists an $O(\frac{\omega_{max}}{T'})$-approximation protocol for *MST* problem with running time $O(\Lambda(G) + T'\log^* n)$. 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 $T = O(\Lambda(G)) + o(\sqrt{\frac{n}{BH}}) = o(\sqrt{\frac{n}{BH}})$ (assuming that $\Lambda(G) = o(\sqrt{\frac{n}{BH}})$; 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 5.2 (that applies for $\Lambda(G) = O(n^\delta)$ for any constant $\delta > 0$).

**6. An Upper Bound.** In this section we devise a distributed protocol for the approximate *MST* problem. Our protocol runs in $O(\Lambda(G) + n^\epsilon\cdot\log^* n)$ rounds and constructs $O(\frac{\omega_{max}}{n^\epsilon})$-approximate *MST*, where $\omega_{max}$ is the ratio between the weights of the heaviest and the lightest edges in the graph $G$, and $\epsilon$ is any fixed number between zero and one. Note that the approximation ratio of $O(\omega_{max})$ is trivial.

Throughout this section we assume that $B = \log n$. Hence the protocol can be used whenever $B = \Omega(\log n)$, and its running time will be the same. Also, obviously it can be adapted to the case $B = o(\log n)$ incurring an overhead of $O(\frac{\log n}{B}) = O(\log n)$. Also, we assume that the weights of edges are scaled between 1 and $\omega_{max}$.

Consider an *MST* $\tau_0$ of the graph $G$. A connected subtree of $\tau_0$ is called a *fragment* of $\tau_0$. When the *MST* $\tau_0$ can be understood from the context, such a fragment will be called an *MST fragment.* A *k-MST forest* $\mathcal{F}$ of a graph $G = (V, E)$ is a collection of vertex-disjoint trees that satisfy

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

The notion of *k-MST* forest is related to the notion of $(\sigma, \rho)$ spanning forest of

[23]. Trees $T \in \mathcal{F}$ of a $(\sigma, \rho)$ spanning forest $\mathcal{F}$ have to satisfy properties (1) and (2) with $|V(T)| \geq \sigma$ and $\Lambda(T) \leq \rho$, but may not satisfy property (3). It was demonstrated in [23] that a $k$-$MST$ 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 the $MST$ problem is to construct an $n^\epsilon$-$MST$ forest $\mathcal{F}$. This requires $O(n^\epsilon \cdot \log^* n)$ rounds. The second step is to construct a BFS spanning tree $\tau$ of the entire graph, rooted at some arbitrary vertex $rt = rt(\tau)$. This requires $O(\Lambda(G))$ rounds. After the construction of the tree $\tau$, each vertex $v$ in the graph knows its unweighted distance to the root $rt$, $dist_\tau(rt, v)$. 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 $O(\max\{\Lambda(T) \mid T \in \mathcal{F}\})$ rounds. As $\mathcal{F}$ is an $n^\epsilon$-$MST$ forest, it follows that $\Lambda(T) = O(n^\epsilon)$ for every tree $T \in \mathcal{F}$. 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 $O(\Lambda(G) + n^\epsilon \cdot \log^* n)$. We next argue that it is indeed an $O(\frac{\omega_{max}}{n^\epsilon})$-approximation protocol for the $MST$ problem.

Lemma 6.1. *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 $T_i$ between $w_i$ and $u_{((i+1) \ mod \ t)}$, and $(u_i, w_i) \in E(\tau)$, $i \in \{0, 1, \ldots, t-1\}$ (where $\tau$ is the BFS spanning tree of the graph). 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 $\tau_0$ with both endpoints in $T$ all belong to the spanning tree of $T$.) It follows that $|dist_\tau(rt, u_i) - dist_\tau(rt, w_i)| = 1$ for $i \in \{0, 1, \ldots, t-1\}$.

Assume, without loss of generality, that $dist_\tau(rt, w_0) - dist_\tau(rt, u_0) = 1$. 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 $i \in \{0, 1, \ldots, t-1\}$. Hence,

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

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

Lemma 6.2. *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 the 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 $dist_\tau(rt, u_0) + dist_\tau(rt, w_0)$.

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 $dist_\tau(rt, v) = dist_\tau(rt, u_0) - 1 < dist_\tau(rt, u_0)$, 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 $dist_\tau(rt, v_0) \le dist_\tau(rt, v) < dist_\tau(rt, u_0)$, 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 6.1 and 6.2 that $\tau_0$ is a spanning tree of the graph $G$.

LEMMA 6.3. *The tree $\tau_0$ is a $(1 + O(\frac{\omega_{max}}{n^\epsilon}))$-approximate $MST$ of the graph $G$.*

*Proof.* Let $\omega : E \to R^+$ denote the weight function that is associated with the graph $G$, and let $\omega(MST)$ denote the weight of the $MST$. Observe that the weight of the $n^\epsilon$-$MST$ forest $\mathcal{F}$ is at most $\omega(MST)$, and that the convergecast and broadcast procedures insert into $\tau_0$ at most $O(n^{1-\epsilon})$ 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, $\omega(MST) \ge n - 1$. It follows that $\omega(\tau_0)/\omega(MST) = O(\omega_{max}/n^\epsilon)$. ▮

To conclude,

THEOREM 6.4. *For any $0 < \epsilon < 1$, there exists a protocol that constructs an $H$-approximate $MST$ for an $n$-vertex weighted graph $(G, \omega)$ in $O(\Lambda(G) + n^\epsilon \cdot \log^* n)$ 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 $O(\Lambda(G))$.

**7. Discussion.** In this paper we presented a lower bound on the time-approximation tradeoff of the distributed $MST$ problem. While this is one of the most fundamental problems in the area of distributed computing, there are many other important distributed problems whose approximation behavior is yet to be explored. In our opinion, it would be of particular interest to establish lower bounds on time-approximation tradeoffs of such problems as maximum dominating set and maximum matching. Interrelating the complexities of these problems via distributed reductions appears to be an even greater challenge.

**Acknowledgements.** The author is grateful to Michael Langberg, Zvika Lotker, Alessandro Panconesi, David Peleg, Alexander Razborov, Oded Regev, Vitaly Rubinovich and Avi Wigderson for helpful discussions. The author also wishes to thank anonymous referees for their suggestions and remarks that helped to improve the presentation in this paper.

REFERENCES

[1] 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.

[2] B. Awerbuch, B. Berger, L. Cowen and D. Peleg, Near-linear cost sequential and distributed constructions of sparse neighborhood covers. In *Proc. Proc. 34th IEEE Symp. on Foundations of Computer Science*, pp. 638-647, 1993.

[3] B. Awerbuch and G. Gallager, A new distributed algorithm to find breadth first search trees. *IEEE Trans. on Information Theory*, IT-33(3):315-322, 1987.

[4] Y. Bartal, J. W. Byers, and D. Raz, Global optimization using local information with applications to flow control. In *Proc. of the 38th IEEE Symp. on the Foundations of Computer Science*, pages 303–312, 1997.

[5] 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.

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

[7] A. Czygrinow, M. Hanckowiak, E. Szymanska, Distributed Algorithm for Approximating the Maximum Matching, manuscript, 2003.

[8] 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.

[9] D. Dubhashi, 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.

[10] M. Elkin, Computing almost shortest paths. In *Proc. 20th ACM Symp. on Principles of Distributed Computing*, pages 53–62, 2001.

[11] M. 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.

[12] M. Elkin, Unconditional Lower Bounds on the Time-Approximation Tradeoffs for the Distributed Minimum Spanning Tree Problem, in *Proc. of the 35th Ann. ACM Symp. on Theory of Computing*, STOC'04, pp. 331-340, June 13-15, 2004, Chicago, Illinois, USA.

[13] M. Elkin, An Overview of Distributed Approximation. In *ACM SIGACT News Distributed Computing Column*, Volume 35, Number 4 (Whole number 132), Dec. 2004, pp. 40-57.

[14] M. Elkin and J. Zhang, Efficient algorithms for constructing $(1+\epsilon, \beta)$-spanners in the distributed and streaming models. In *Proc. 23rd ACM Symp. on Principles of Distributed Computing* , pp. 160-168, 2004.

[15] T. Fischer, A. Goldberg, D. J. Haglin, and S. Plotkin, Approximating matchings in parallel, in *Information Processing Letters archive*, pp. 115-118, Vol. 46, Issue 3, June 1993.

[16] 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.

[17] 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.

[18] 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.

[19] F. Grandoni, K. Konemann, A. Panconesi and M. Sozio. Primal-Dual Based Distributed Algorithms for Vertex Cover with Semi-Hard Capacities. In *Proc. of 24th ACM Symp. on Principles of Distributed Computing*, pp. 118-125, Las-Vegas, Nevada, July, 2005.

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

[21] M. Hanckowiak, M. Karonski, and A. Panconessi, On The distributed complexity of computing

maximal matchings, SIAM J. Discrete Mathematics, Vol 15, No. 1, pp.41-57, 2001.

[22] 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.

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

[24] F. Kuhn, T. Moscibroda, and R. Wattenhofer, What Cannot Be Computed Locally! in *Proc. of 23rd Symp. on the Principles of Distributed Computing (PODC)*, pp. 300-309, St. John's, Newfoundland, Canada, July 2004.

[25] F. Kuhn and R. Wattenhofer, Constant-Time Distributed Dominating Set Approximation, In *Proc. of the 22nd ACM Symp. on Principles of Distributed Computing*, Boston, Massachusetts, USA, July 2003.

[26] F. Kuhn and R. Wattenhofer. Distributed combinatorial optimization. *Technical Report 426, Dept. of Computer Science, ETH, Zurich*, 2004.

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

[28] 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.

[29] Z. Lotker, E. Pavlov, B. Patt-Shamir and D. Peleg, MST Construction in $O(\log\log n)$ communication rounds, in *Proc. Symp. on Parallel Algorithms and Architectures, SPAA'03*, pp. 94-100, San-Diego, CA, USA.

[30] T. Moscibroda and R. Wattenhofer, Facility Location: Distributed Approximation. In *Proc. of 24th ACM Symp. on Principles of Distributed Computing*, pp. 108-117, Las-Vegas, Nevada, July, 2005.

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

[32] 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.

[33] 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.

[34] C. Papadimitriou and M. Yannakakis. Linear programming without matrix. In *Proc. of the 25th ACM Symp. on Theory of Computing*, pages 121–129, 1993.

[35] S. Rajagopalan and V.V. Vazirani, Primal-Dual RNC Approximation Algorithms for (multi)Set (multi)Cover and Covering Integer Programs. SIAM J. of Computing 28, 2 (1998) 525-540.

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

[37] A. Yao, Probabilistic computations: Towards a unified measure of complexity, in *Proc. 17th IEEE Symp. on Foundations of Computer Science*, Providence, Rhode Island, 31 October-2 November 1977, pp. 222-227.