Yale University
Department of Computer Science

Polylogarithmic Inapproximability of the Radio Broadcast Problem

Michael Elkin [1] [2] Guy Kortsarz[3]

YALEU/DCS/TR-1274
February 2004

[1]Department of Computer Science, Yale University, New Haven, CT, USA, 06520-8285, elkin@cs.yale.edu.
[2]Part of this work was done in School of Mathematics, Institute for Advanced Study, Princeton, NJ, USA, 08540.
[3]Computer Science department, Rutgers University, Camden, NJ, USA. Email: guyk@crab.rutgers.edu

Polylogarithmic Inapproximability of the Radio Broadcast Problem

Michael Elkin [*,†] Guy Kortsarz [‡]

Abstract

We prove that there exists a universal constant $c > 0$ such that the Radio Broadcast problem admits *no additive* $c \cdot \log^2 n$*-approximation*, unless $NP \subseteq BPTIME(n^{O(\log\log n)})$. This is one of the first results that establish a polylogarithmic inapproximability of a natural problem.

On the positive side we show that for every radio network $G = (V, E)$ and source $s \in V$, there exists a radio broadcast schedule for G of length $Rad(G, s) + O(\sqrt{Rad(G, s)} \cdot \log^2 n) = O(Rad(G, s) + \log^4 n)$, where $Rad(G, s)$ is the radius of the radio network G with respect to the source s. This result improves the previously best-known upper bound of $O(Rad(G, s) + \log^5 n)$ due to Gaber and Mansour [GM95].

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

[†]Part of this work was done in School of Mathematics, Institute for Advanced Study, Princeton, NJ, USA, 08540.

[‡]Computer Science department, Rutgers University, Camden, NJ, USA. Email: guyk@crab.rutgers.edu

1 Introduction

1.1 Polylogarithmic Inapproximability

The seminal survey paper [AL96] by Arora and Lund divides NP-hard optimization problems into four broad classes, based on the lower bound on their approximation threshold. These lower bounds are $c > 1$ for some universal constant c, $\log n$, $2^{\log^{1-\epsilon} n}$ for every constant $\epsilon > 0$, and n^δ for some universal constant $\delta > 0$. The corresponding classes are termed by them Class I, II, III and IV. They remark that their classification reflects only the current knowledge at that point, and "future work may reveal other natural classes".

Recently, Halperin and Krauthgamer [HK03], based on the previous work by Halperin et. al. [HK+03], have shown that the Group Steiner problem is $\Omega(\log^{2-\epsilon} n)$-inapproximable for any constant $\epsilon > 0$, and along with the $O(\log^2 n)$-approximation algorithm of Garg et al. [GKR00], this result can be considered as the first indication of the emergence of a new class in Arora-Lund classification, specifically, the class of problems whose approximation threshold is polylogarithmic in n. In this paper we exhibit an example of another problem with polylogarithmic approximation threshold, specifically, the *Radio Broadcast* problem. This example indicates that the behavior of the approximation threshold of the Group Steiner problem is not an accidental anomaly, but rather there is a *new intermediate class* of optimization problems that is located between Classes II and III.

1.2 Radio Broadcast Problem

1.2.1 Definition and Motivation

Consider a synchronous network of processors that communicate by transmitting messages to their neighbors, where a processor receives a message in a given step if and only if precisely one of its neighbors transmit. The instance of the Radio Broadcast problem, called *radio network*, is a pair $(G = (V, E), s)$, $s \in V$, where G is an unweighted undirected graph, and s is a vertex, called *source*. The objective is to deliver one single message that the source s generates to all the vertices of the graph G using the smallest possible number of communication rounds. The prescription that tells each vertex when it should broadcast is called *schedule*; the *length* of the schedule is the number of rounds it uses, and it is called *admissible* if it informs all the vertices of the graph (see Section 2 for a formal definition).

From practical perspective, the interest to radio networks is usually motivated by their military significance, as well as by the growing importance of cellular and wireless communication (see, e.g., [KM98, GM95, BGI86]). The radio broadcast is perhaps the most important communication primitive in radio networks, and, consequently, it has been intensively studied starting from mid-eighties [CR03, KP02, KP03, CGR00, CG+02, CMS03, I02, GM95, KM98, ABLP89, BGI86, CW87, CK85].

From theoretical perspective, the study of the Radio Broadcast problem provided researchers with a particularly convenient playground for the study of such broad and fundamental complexity-theoretic issues as the power and limitations of randomization, and of different models of distributed computation [BGI86, KM98, KP03]. In this paper we study the *approximation threshold* of the *Radio Broadcast* problem; we believe that our results show that this problem is of a particular interest from the stand-point of the theory of Hardness of Approximation as well.

1.2.2 Previous Results

The first known algorithm for the Radio Broadcast problem was devised by Chlamtac and Weinstein in 1987 [CW87]. That algorithm, given an instance (G, s) of the problem, constructs an admissible broadcast schedule of length $O(Rad(G, s) \cdot \log^2 n)$ for this instance, where $Rad(G, s)$ stands for the *radius* of the instance (G, s), that is, the maximum distance $d_G(s, v)$ in the graph G between the source s and some vertex $v \in V$. Their algorithm is *centralized*, i.e., it accepts the entire graph as input.

Soon afterwards Bar-Yehuda et al. [BGI86] devised a distributed randomized algorithm that provides admissible schedules of length $O(Rad(G, s) \cdot \log n + \log^2 n)$. Alon et al. [ABLP89] have shown that the additive term of $\log^2 n$ in the result of [BGI86] is inevitable, and devised a construction of infinitely many instances (G, s) of constant radius that satisfy that any broadcast schedule for them requires $\Omega(\log^2 n)$ rounds. Kushilevitz and Mansour [KM98] have shown that for *distributed* algorithms, the multiplicative logarithmic term in the result of [BGI86] is inevitable as well, and proved that for *any distributed algorithm* for the Radio Broadcast problem there exist (infinitely many) instances (G, s) on which the algorithm constructs a schedule of length $\Omega(Rad(G, s) \cdot \log(n/Rad(G, s)))$. Finally, the gap between the $\log n$ and $\log(n/Rad(G, s))$ was recently closed by Kowalski and Pelc [KP03], and Czumaj and Rytter [CR03].

Gaber and Mansour [GM95] have shown that *centralized* algorithms are capable of constructing much shorter schedules, and devised an algorithm that constructs admissible schedules of length $O(Rad(G, s) + \log^5 n)$. Their result is also the current state-of-the-art in terms of the *existential* upper bounds on the length of broadcast schedules for instances of large radius.

Since, obviously, any schedule for an instance (G,s) requires at least $Rad(G,s)$ rounds, the algorithms for the Radio Broadcast problem [CW87, BGI86, GM95, KP03, CR03] can be interpreted as *approximation algorithms* for the problem. The state-of-the-art in this context are the results by Bar-Yehuda et al. [BGI86] for instances of small radius, and the result of Gaber and Mansour [GM95] for instances of large radius. Note that the latter result provides a solution having both a *constant multiplicative* and a *polylogarithmic additive* approximation guarantees.

Regarding the lower bounds, the NP-hardness of the Radio Broadcast problem was shown by Chlamtac and Kutten [CK85] already in 1985. However, only recently Bar-Yehuda [B01] provided a first evidence that the problem is hard to approximate. Specifically, he devised a reduction from the 3-coloring problem, and has shown that there are universal constants $c > 0$, $0 < c' < 3/14$ such that existence of a polynomial algorithm with (purely) additive approximation guarantee of $c \log n$ would give rise to a polynomial algorithm that colors 3-colorable graphs with $O(n^{c'})$ colors. While the latter task is commonly considered to be hard, no NP-hardness result for it is currently available. Soon afterwards the authors of the current paper have shown that there exists a constant $c > 0$ such that the Radio Broadcast problem cannot be approximated with a (purely) additive approximation ratio of $c \log n$ unless $NP \subseteq BPTIME(n^{O(\log \log n)})$ [EK02].

1.3 Our Results

In this paper we show that there exists a constant $c > 0$ such that the Radio Broadcast problem cannot be approximated with a (purely) additive approximation ratio of $c \log^2 n$, unless $NP \subseteq BPTIME(n^{O(\log \log n)})$. We also show that for any instance (G,s) of the Radio Broadcast problem, there exists an admissible schedule of length $Rad(G,s) + O(\sqrt{Rad(G,s)} \cdot \log^2 n)$, and devise a randomized algorithm that computes this schedule in expected polynomial time. In deterministic polynomial time a variant of our algorithm computes an admissible schedule with slightly larger guarantee on its length, specifically, $Rad(G,s) + O(\sqrt{Rad(G,s)} \cdot \log^{2.5} n)$.

Note that our lower and upper bounds imply that the approximability of the Radio Broadcast problem is best described in terms of (purely) *additive* and *not multiplicative* approximation. (See Section 2.1 for the formal definition of additive and multiplicative approximation thresholds.)

Observe also that our results determine the additive approximation threshold of the Radio

Broadcast problem restricted to instances (G, s) of radius $Rad(G, s) = O(1)$ up to a *constant* factor, and show that the latter is $\Theta(\log^2 n)$; the same is true for the problem when it is restricted to a wider class of instances (G, s) with radius $Rad(G, s) = O(\log n)$ (to see it, note that the upper bound of [BGI86] is better than ours when $\omega(1) = Rad(G, s) = o(\log^2 n)$). When the problem is restricted to a yet wider class of instances (G, s) that have radius $O(\log^c n)$ for some constant $c > 0$, the additive approximation threshold of the problem is between $\Omega(\log^2 n)$ and $O(\log^{2+c/2} n)$, that is, still polylogarithmic.

Together with the result of Halperin and Krauthgamer [HK03] regarding the Group Steiner problem, our results indicate the existence of a new class in Arora-Lund classification. This new class contains problems whose approximation threshold is significantly higher than that of the Set Cover problem, but is significantly lower than that of the Label-Cover problem. Since currently this class provably contains only two problems, specifically, the Group Steiner and the Radio Broadcast problems, it is instructive to compare the available results regarding these two problems.

First, observe that the approximability of the Radio Broadcast problem is best described in terms of the *additive* approximation, while the approximability of the Group Steiner problem is best described in terms of the more traditional *multiplicative* approximation. We remark that results on additive approximability (not necessarily polylogarithmic) are rather scarce; another example the Minimum-Degree Spanning Tree problem for which additive 1-approximation was demonstrated by Furer and Ragavachari [FR94].

Second, while no polylogarithmic *NP-hardness* results are currently available, our lower bound on the approximation threshold of the Radio Broadcast problem is proven under a weaker complexity-theoretic assumption than the lower bound of [HK03] for the Group Steiner problem. Specifically, our lower bound holds unless $NP \subseteq BPTIME(n^{O(\log \log n)})$, and that of [HK03] holds unless $NP \subseteq ZTIME(n^{O(\log n)})$, where $BPTIME(f(n))$ (respectively, $ZTIME(f(n))$) stands for the class of problems that can be solved in *probabilistic* (resp., *expected*) polynomial time.

Third, the Radio Broadcast problem restricted to instances (G, s) of radius $Rad(G, s) = O(\log n)$ is the first example of a problem whose approximation threshold is polylogarithmic and is established up to a *constant* factor. Specifically, we show that this threshold is precisely $\Theta(\log^2 n)$. Note that there is still a gap of $\log^\epsilon n$ for arbitrarily small $\epsilon > 0$ between the state-of-the-art upper and lower bounds on the approximation threshold of the Group Steiner problem (due to [GKR00] and [HK03], respectively).

Finally, our upper bound, that is, the algorithm that constructs schedules of length $Rad(G, s) + O(\sqrt{Rad(G, s)} \cdot \log^2 n)$ should be compared with the algorithm of Gaber and

Mansour [GM95] that constructs schedules of length $O(Rad(G,s) + \log^5 n)$. Note that for instances of small radius this is an improvement by more than a logarithmic multiplicative factor, while for instances of large radius the multiplicative improvement is by a constant factor, specifically 3. For our purposes it was particularly important to establish a *purely additive upper bound*. Such a bound, along with our inapproximability result, proves that the approximability of the Radio Broadcast problem is best described in terms of additive approximation.

We remark that even out of the context of the theory of Hardness of Approximation, our upper bound is of independent interest, as it strictly improves the result of Gaber and Mansour [GM95]. In particular, it provides a partial answer to an open problem that they posed: to determine whether for instances (G, s) of the Radio Broadcast problem with radius $\Omega(\log^2 n) = Rad(G,s) = O(\log^5 n)$, there always exists a schedule of length $O(Rad(G,s))$. Our result, in particular, provides the positive answer to this question for $Rad(G,s) = \Omega(\log^4 n)$.

Remark: Due to space limitations the upper bound and some proofs of the lower bound are deferred to Appendix.

1.4 Proof Techniques

1.4.1 Lower Bound

The proof of our lower bound incorporates the combinatorial construction of Alon et al. [ABLP89] in the reduction of Lund and Yannakakis [LY94]. The latter reduction reduces the Label-Cover problem to the Set Cover problem, and establishes the logarithmic hardness of approximation of the latter. However, the importance of this reduction extends far beyond this specific proof, and the reduction was used as a building block for many results in the area of Hardness of Approximation [ABSS93, EK02, DGKR03, HK03].

In [EK02] the authors of the current paper have shown an additive logarithmic hardness of approximation of the Radio Broadcast problem, essentially by adapting the reduction of [LY94] to the Radio Broadcast problem. However, it seems that the reduction of [LY94] alone cannot provide greater than logarithmic hardness of approximation for the Radio Broadcast problem. The second crucial building block of our lower bound is the result of Alon et al. [ABLP89]. Their construction provides a bipartite graph in which $\Omega(\log^2 n)$ rounds are required to deliver the message from the left-hand side of the graph to its right-hand side.

To explain how we incorporate the construction of [ABLP89] into the reduction of [LY94], we first outline the structure of the reduction of [LY94]. For each edge $\tilde{e}$ of the input instance of the Label-Cover problem, the reduction of [LY94] constructs a bipartite graph $(\mathcal{S}_{\tilde{e}}, M_{\tilde{e}})$, where each vertex $x \in \mathcal{S}_{\tilde{e}}$ is connected to one half of the vertices of the set $M_{\tilde{e}}$. Furthermore, these connections are ingeniously arranged in such a way that the following condition is satisfied: for any $k = o(\log n)$ vertices $x_1, x_2, \ldots, x_k \in \mathcal{S}_{\tilde{e}}$, and for any choice of sets $X_1, X_2, \ldots, X_k$ such that every X_i is either equal to the set N_i of the vertex x_i in $M_{\tilde{e}}$ or to its compliment $M_{\tilde{e}} \setminus N_i$, the set $\bigcup_{i=1}^{k} X_i$ does not cover the vertex set $M_{\tilde{e}}$. Such a bipartite graph is called a *set system*.

The reduction of [LY94] also provides each such *set system* $(\mathcal{S}_{\tilde{e}}, M_{\tilde{e}})$ with a simple *"trapdoor"* that enables to cover the set $M_{\tilde{e}}$ with only two sets, specifically, the set N_i and the set $M_{\tilde{e}} \setminus N_i$. These trapdoors are geared to enable to construct small solutions for the instances of the Set Cover problem whenever the original instance of the Label-Cover problem is a YES-instance, but, however, cannot be used if the original instance is a NO-instance.

The basic idea of our proof of the polylogarithmic lower bound for the Radio Broadcast problem is to replace the set systems of [LY94] by the bipartite graphs of [ABLP89]. To accomplish this, one, however, needs to devise an analogous trapdoor to the construction of [ABLP89], and doing it without ruining the (rather delicate) properties of that construction constitutes a significant technical difficulty.

We remark that the proof of the polylogarithmic lower bound for the Group Steiner problem due to [HK03] also makes an ingenious use of the reduction of [LY94]. However, in that proof the growth of the lower bound beyond the logarithmic barrier should probably be attributed to a composition of many instances that are constructed by the reduction of [LY94].

1.4.2 Upper Bound

Our algorithm for constructing a radio broadcast schedule of length $Rad(G,s) + O(\sqrt{Rad(G,s)} \cdot \log^2 n)$ for every input instance (G,s) is based on the elegant algorithm of Gaber and Mansour [GM95]. The latter algorithm constructs schedules of length $O(Rad(G,s) + \log^5 n)$.

Both algorithms partition the graph into *layers* (or, *super-levels*, cast to the terminology of [GM95]), and construct an $(O(\log n), O(\log n))$-partition of every layer (see Sec. A.3 for the definition) using the algorithm of [LS91] or [AB+92]. One difference between the algorithms is that in our algorithm the layers are *overlapping* and not *disjoint* as in the algorithm of [GM95]. This difference, while complicating the analysis, enables us to reduce the additive term from $O(\log^5 n)$ to $O(\log^4 n)$. Both algorithms use the constructed partitions to control the data flow, and use similar procedures for this purpose. In [GM95] the scheduling of these procedures involves round-robin time sharing, which guarantees that executions of different procedures do not cause collisions. While very convenient for the analysis, this technique carries a price of increasing the length of the constructed schedule by a constant multiplicative factor. Our algorithm is geared to avoid this unnecessary multiplicative factor in order to achieve a *purely additive* approximation of the optimal schedule. To this end, our algorithm carefully combines different building blocks in such a manner that they cause no collision even though they employ *the same rounds*.

2 Preliminaries

2.1 Hardness of Approximation

Consider a minimization problem $\mathcal{P}$. For an instance α of $\mathcal{P}$, let $opt(\alpha)$ denote the value of the optimum solution of the problem $\mathcal{P}$ on the instance α. For a constant $t > 1$, a deterministic algorithm A is said to provide a *multiplicative* (resp., *additive*) *t-approximation* for the problem $\mathcal{P}$ if for every instance α it returns a solution of value at most $opt(\alpha) \cdot t$ (resp., $opt(\alpha) + t$). The same definition applies to randomized algorithms, but in that case the word "returns" should be replaced by "returns with high probability over its coin tosses".

For an algorithm A, the smallest value of the number t that satisfies the above condition is called the *multiplicative* (resp., *additive*) *approximation guarantee* of the algorithm A, and is denoted $t^*(A)$ (resp., $t^+(A)$). Note that this definition readily generalizes to *super-constant approximation guarantees*, i.e., to the case when the approximation guarantee depends on the size of the instances.

Let $\mathcal{A}$ denote the class of all deterministic and randomized approximation algorithms for the problem $\mathcal{P}$ whose running time is at most polynomial in the size of the input. The *multiplicative* (resp., *additive*) *approximation threshold* of the problem $\mathcal{P}$ is defined as $\min\{t^*(A) \mid A \in \mathcal{A}\}$ (resp., $\min\{t^+(A) \mid A \in \mathcal{A}\}$).

2.2 The Radio Broadcast Problem

We start with introducing some definitions and notations.

Definition 2.1 *The set of neighbors of a vertex v in an unweighted undirected graph $G(V,E)$, denoted $\Gamma_G(v)$, is the set $\{u \in V \mid (v,u) \in E\}$. For a subset $X \subseteq V$, the set of neighbors of* *the vertex v in the subset X, denoted $\Gamma_G(v, X)$ (or $\Gamma(v, X)$ when the graph G is clear from the context), is the set $\{u \in X \mid (v, u) \in E\}$.*

Notation 2.2 *1. For a positive integer number n, let $[n]$ denote the set $\{1, 2, \ldots, n\}$.*

2. For an index $i \in \{1, 2\}$, let $\bar{i}$ denote the complimentary index $2 - i$.

We also use the letter ϵ to denote an arbitrarily small positive real universal constant.

Definition 2.3 *Let $G = (V, E)$ be an unweighted undirected graph, and $R \subseteq V$ be a subset of vertices. The set of vertices* informed *by R, denoted $I(R)$, is $I(R) = \{v \mid \exists! x \in R \ \ s.t. \ \ v \in \Gamma_G(x)\}$ (the notation $\exists! x$ stands for "there exists a unique x"). For a singleton set $R = \{x\}$, $I(R) = I(\{x\}) = I(x) = \Gamma_G(x)$.*

A sequence of vertex sets $\Pi = (R_1, R_2, \ldots, R_q)$, $q = 1, 2, \ldots$, is called a *radio broadcast schedule* (henceforth referred as a *schedule*) if $R_{i+1} \subseteq \bigcup_{j=1}^{i} I(R_j)$ for every $i = 1, 2, \ldots, q-1$. Intuitively, this condition means that the vertices that send a message in certain round have to be informed in one of the previous rounds.

The set of vertices *informed by a schedule* Π, denoted $I(\Pi)$, is $I(\Pi) = \bigcup_{R \in \Pi} I(R)$.

Given a graph $G = (V, E)$ and a vertex $s \in V$, a schedule Π is *admissible* with respect to (G, s) if $R_1 = \{s\}$ and $V \subseteq I(\Pi)$.

The *length* of the schedule $\Pi = (R_1, R_2, \ldots, R_q)$ is $|\Pi| = q$.

An instance of the *radio broadcast problem* $\mathcal{G}$ is a pair $(\bar{G} = (\bar{V}, \bar{E}), s)$, where $\bar{G}$ is a graph, and $s \in \bar{V}$ is a vertex. The goal is to compute an admissible schedule Π of minimal length. The *value* of an instance $\mathcal{G}$ of the radio broadcast problem is the length of the shortest admissible schedule Π for this instance.

For any schedule $\Pi = (R_1, R_2, \ldots, R_q)$, the set R_i is called the *ith round* of Π, $i = 1, 2, \ldots, q$. If a vertex v belongs to the set R_i, we say that v is *active* in round i.

2.3 The MAX-REP problem

Definition 2.4 *The MAX-REP problem is defined as follows. The input consists of a bipartite graph $G = (V_1, V_2, E)$. In addition, for $j = 1, 2$, the input contains a partition $\tilde{V}_j$ of V_j into a disjoint union of subsets, $V_1 = \bigcup_{A \in \tilde{V}_1} A$, $V_2 = \bigcup_{B \in \tilde{V}_2} B$. The triple $\mathcal{M} = (G, \tilde{V}_1, \tilde{V}_2)$ is an* instance *of the MAX-REP problem. The size of the instance is $n = |V_1| + |V_2|$. An instance G as above induces a bipartite super-graph $\tilde{G} = (\tilde{V}_1, \tilde{V}_2, \tilde{E})$ in which the sets A and B of the partition serve as the vertices of the super-graph. The edges of the super-graph are*

$\tilde{E}(\mathcal{M}) = \tilde{E} = \{(A,B) \in \tilde{V}_1 \times \tilde{V}_2 \mid a \in A, b \in B, (a,b) \in E\}$. *In other words, there is a (super-)edge between a pair of sets $A \in \tilde{V}_1$, $B \in \tilde{V}_2$ if and only if the graph G contains an edge between a pair of vertices $a \in A$, $b \in B$.*

Denote $\tilde{V} = \tilde{V}_1 \cup \tilde{V}_2$. A pair of vertices $x_1, x_2 \in V_1 \cup V_2$ is called a matching pair with respect to a super-edge $\tilde{e} = (A,B) \in \tilde{E}$ *(henceforth, $\tilde{e}$-*m.p.*) if $(x_1,x_2) \in E$ and either $x_1 \in A$ and $x_2 \in B$ or vice versa.*

A subset $C \subseteq V_1 \cup V_2$ of vertices is said to cover *a super-edge $\tilde{e} = (A,B)$ if it contains an $\tilde{e}$-m.p.. A subset $C \subseteq V_1 \cup V_2$ that satisfies $|C \cap X| = 1$ for every $X \in \tilde{V}$ is called a* MAX-cover*. In other words, a MAX-cover C contains exactly one vertex from each super-vertex.*

An instance $\mathcal{M}$ of the MAX-REP problem is called a YES-instance *if there exists a MAX-cover that covers all the superedges. Such a MAX-cover is called a* perfect MAX-cover.

For a positive real number $t > 1$, an instance $\mathcal{M}$ of the MAX-REP problem is called a t-NO-instance *if any MAX-cover C of the instance $\mathcal{M}$ covers less than $|\tilde{E}|/t$ super-edges of $\mathcal{M}$.*

The MAX-REP problem is equivalent to the maximization variant of the Label-Cover problem (see, e.g., [AL96]).

We also impose several additional (somewhat less standard) restrictions on the set of instances of the MAX-REP problem.

1. All the super-vertices $X \in \tilde{V}$ are of size $|X| = n^{0.4}/2$.
2. The set of super-edges has sufficiently large cardinality, specifically, $|\tilde{E}| = \Omega(n^{0.6} \cdot \log^3 n)$. (Note that by restriction 1, just for the super-graph to be connected, $|\tilde{E}|$ has to be at least $2n^{0.6} - 1$.)
3. **The Star property:** For every super-edge $\tilde{e} = (A,B) \in \tilde{E}$, $A \subseteq V_1$ and $B \subseteq V_2$ and every vertex $b \in B$ there exists exactly one vertex $a \in A$, denoted $\tilde{e}(b)$, such that $(a,b) \in E$. This property is called the *star* property.

Essentially, the star property means that for every super-edge $\tilde{e} = (A,B) \in \tilde{E}$, the graph induced by the subset $A \cup B$ decomposes into a collection of vertex-disjoint stars (a *star* is a graph with all vertices but (maybe) one having degree 1). The vertex with degree larger than 1 is called the *head* of the star; if there are only two vertices in the star, then the head is the vertex that belongs to the super-vertex A. The other vertices of the star are called the *leaves* of the star. See Figure 1 for an example of a MAX-REP instance that obeys the star property.

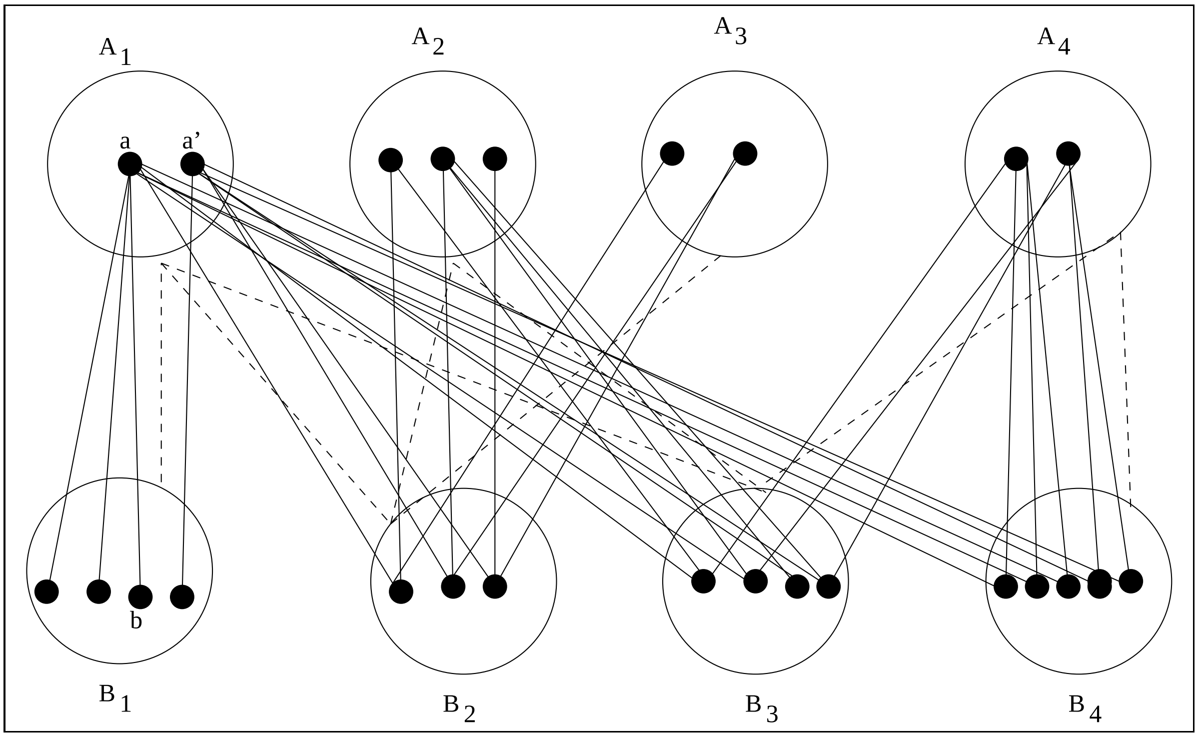

Figure 1: An example of a MAX-REP instance with the star property. Every pair of sets (A_i, B_j) in the partition, induces a collection of disjoint stars with heads in A. The vertices (a, b) form a matching pair that covers the super-edge (A_1, B_1). The vertices a', b do not form a matching pair, and do not cover the super-edge (A_1, B_1).

Theorem 2.5 *[R98] No deterministic polynomial time algorithm may distinguish between the YES-instances and the* $\log^{10} n$*-NO-instances of the MAX-REP problem, unless* $NP \subseteq DTIME(n^{O(\log\log n)})$*, even when the instances of the MAX-REP problem satisfy the conditions (1)-(3).*

The choice of 10 in the exponent of $\log n$ is somewhat arbitrary, and can be replaced by any other constant. For the sake of our proof this constant has to be greater than 8.

2.4 Chernoff Bound

Claim 2.6 *Let* $X_1, X_2, \ldots, X_n$ *be* n *independent identically distributed (henceforth, i.i.d.) boolean random variables, each accepting value 1 with probability* $p = \mu/n$*, for some real number* μ*. Let* $X = \sum_{i=1}^{n} X_i$*. Then for any constant* $0 < \delta < 1$, $\mathbb{P}(X < (1-\delta)\mu) < e^{-\frac{(\delta\mu)^2}{2n}}$.

Proof: Let $Y_i = p - X_i$, and $Y = \sum_{i=1}^{n} Y_i$. Then $\mathbb{E}(Y) = 0$. Now $\mathbb{P}(X < (1-\delta)\mu) = \mathbb{P}(Y > \delta\mu)$, and the latter probability is at most $e^{-\frac{(\delta\mu)^2}{2n}}$, by [AS92], p.240. ∎

3 Reduction

We next describe our randomized reduction from the MAX-REP problem to the radio broadcast problem. This reduction always maps a YES-instance of the MAX-REP problem to an instance of the radio broadcast problem of value $T + O(\log n)$, and with high probability over its own coin tosses the reduction maps a NO-instance of the MAX-REP problem to an instance of the radio broadcast problem of value $T + \Omega(\log^2 n)$, where $T = 1, 2, \ldots$ is a parameter of the reduction.

Consider an instance $\mathcal{M} = (G, \tilde{V}_1, \tilde{V}_2)$, $G = (V_1, V_2, E)$ of the MAX-REP problem with $V_1 = \bigcup_{A \in \tilde{V}_1} A$, $V_2 = \bigcup_{B \in \tilde{V}_2} B$. The reduction constructs an instance $\mathcal{G} = \mathcal{G}(\mathcal{M}) = (\bar{G}, s)$, $\bar{G} = (\bar{V}, \bar{E})$, $s \in \bar{V}$, of the radio broadcast problem in the following way.

Let $N = n^{0.6}$. The vertex set $\bar{V}$ of the graph consists of the source s, and the disjoint vertex sets $\mathcal{V}_1$ and $\mathcal{V}_2$ (i.e., $\bar{V} = \{s\} \cup \mathcal{V}_1 \cup \mathcal{V}_2$ with $s \notin \mathcal{V}_1 \cup \mathcal{V}_2$, and $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$). It is convenient to visualize the source s on the "left-hand" side of the graph, the vertex set $\bar{V}_1$ in the "middle", and the vertex set $\bar{V}_2$ on the "right-hand" side of the graph. See Figure 2.

The vertex set $\mathcal{V}_1$ contains N copies of every vertex $x \in V = V_1 \cup V_2$; these copies are denoted $cp_1(x), cp_2(x), \ldots, cp_N(x)$. The set of all copies of a vertex x, $\{cp_1(x), cp_2(x), \ldots, cp_N(x)\}$, is denoted by χ_x. For a subset $X \subseteq V$, let χ_X denote $\chi_X = \bigcup_{x \in X} \chi_x$. Let $\hat{J}$ denote the set of indices $\{0.4 \log n, 0.4 \log n + 1, \ldots, 0.6 \log n\}$. For an index $j \in \hat{J}$, and a vertex $x \in V$, let $\chi_x(j)$ denote the subset of the set χ_x that contains only the first $J = 2^j$ copies of the vertex x. I.e., $\chi_x(j) = \{cp_1(x), cp_2(x), \ldots, cp_J(x)\}$. These copies are also called the *j-relevant* copies of the vertex x. For a subset $X \subseteq V$, let $\chi_X(j) = \bigcup_{x \in X} \chi_x(j)$.

The vertex set $\mathcal{V}_2$ is of the form $\mathcal{V}_2 = \bigcup_{\tilde{e} \in \tilde{E}} M_{\tilde{e}}$, where the *ground sets* $M_{\tilde{e}}$ are disjoint, and all have equal size. Each ground set $M_{\tilde{e}}$ is a disjoint union of the sets $M_{\tilde{e}}(j, q)$, $j \in \hat{J}$, $q \in [L]$, with $L = 2 \cdot n^{c_0}$, and c_0 is an integer positive universal constant that will be determined later. The sets $M_{\tilde{e}}(j, q)$ are all of equal size $M = n^{c_0}$, for the same constant c_0.

The edge set $\bar{E}$ of the graph $\bar{G}$ contains edges that connect the source s to the vertices of the set $\mathcal{V}_1$, and edges between the vertices of $\mathcal{V}_1$ and $\mathcal{V}_2$. Specifically, the edge set $\bar{E}$ is formed in the following way.

1. Connect the source s to all the vertices of the set $\mathcal{V}_1$.

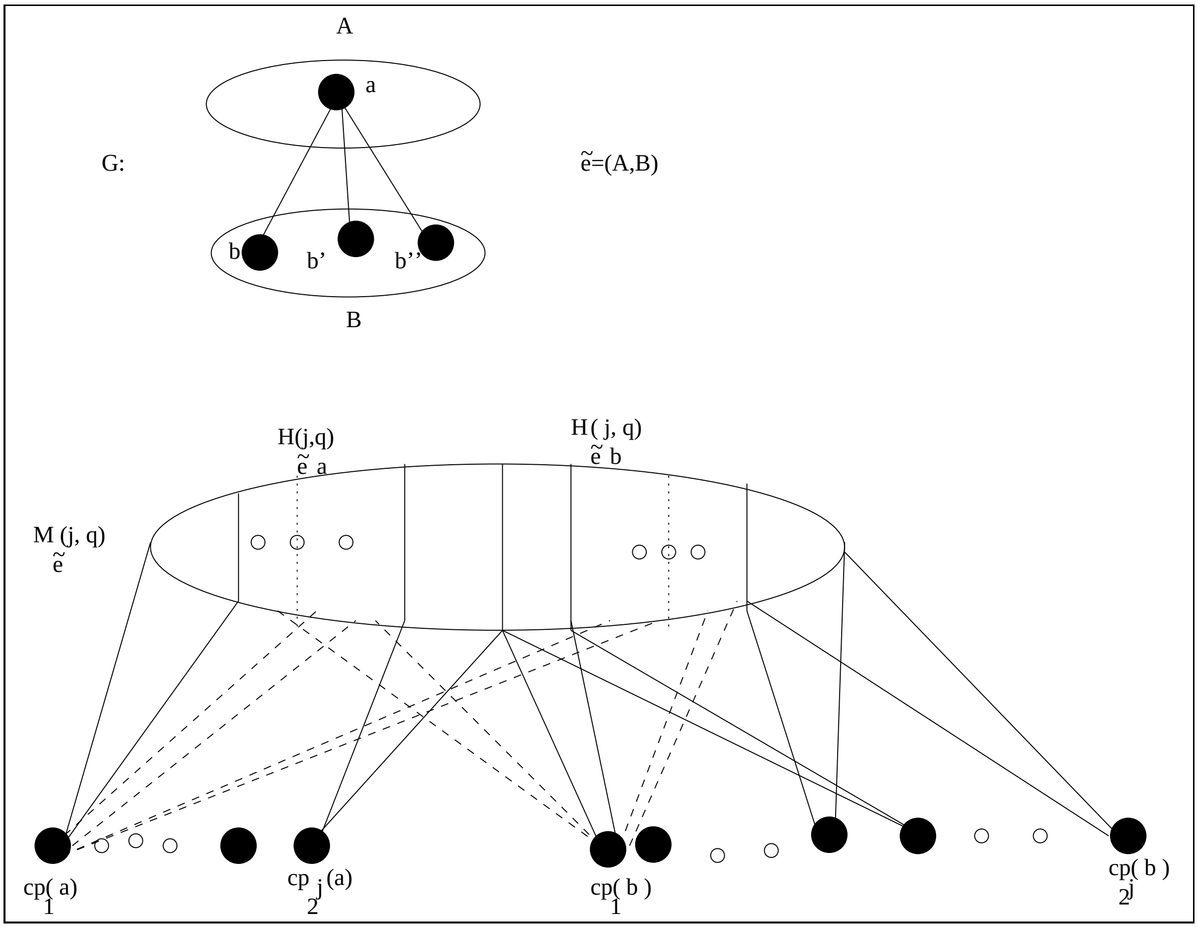

Figure 2: The figure illustrates the permutation step. The copies of the vertex a cover an exact half of the vertices of the set $M_{\tilde{e},a}(j,q)$. The copies of the vertex b cover the complimentary half of the vertices.

2. For every super-edge $\tilde{e} = (A,B) \in \tilde{E}$, for every pair of indices $j \in \hat{J}$, $q \in [L]$ do

   (a) For every vertex $a \in A$

      i. Let $H_{\tilde{e},a}(j,q)$ be an *exact random half* of the set $M_{\tilde{e}}(j,q)$.
      In other words, every subset $H \subseteq M_{\tilde{e}}(j,q)$ that has cardinality $|M_{\tilde{e}}(j,q)|/2$ (for simplicity we ignore all the integrality issues) has the same probability to be chosen to serve as $H_{\tilde{e},a}(j,q)$.

      ii. Let $H^{(1)}_{\tilde{e},a}(j,q)$ be an exact random half of the set $H_{\tilde{e},a}(j,q)$.
      Set $H^{(2)}_{\tilde{e},a}(j,q) = H_{\tilde{e},a}(j,q) \setminus H^{(1)}_{\tilde{e},a}(j,q)$.

   (b) i. For every vertex $b \in B$, set $H_{\tilde{e},b}(j,q) = M_{\tilde{e}}(j,q) \setminus H_{\tilde{e},\tilde{e}(b)}(j,q)$.

      ii. Let $H^{(1)}\tilde{e},b(j,q)$ be an exact random half of the set $H_{\tilde{e},b}(j,q)$.
      Set $H^{(2)}_{\tilde{e},b}(j,q) = H_{\tilde{e},b}(j,q) \setminus H^{(1)}_{\tilde{e},b}(j,q)$.

(The steps 2a-2b will be referred as the *exact partition step.*)

(c) For every vertex $x \in A \cup B$ let $\chi_x^{(1)}(j)$ be a subset of $\chi_x(j)$ that is formed by picking each element of the set $\chi_x(j)$ with probability $1/2$ independently at random.
Set $\chi_x^{(2)}(j) = \chi_x(j) \setminus \chi_x^{(1)}(j)$.
(The step 2c will be referred as the *binomial partition step.*

(d) For every vertex $x \in A \cup B$, and index $i \in \{1, 2\}$ do
Let $\chi_x^{(i)}(j) = \{cp_1^{(i)}(x,j), cp_2^{(i)}(x,j), \ldots, cp_{\bar{\chi}}^{(i)}(x,j)\}$, with $\bar{\chi} = \bar{\chi}(x,i,j) = |\chi_x^{(i)}(j)|$.
Let $S_{\bar{\chi}}$ be the set of all permutations of the set $[\bar{\chi}]$.
Choose a permutation $\sigma \in_R S_{\bar{\chi}}$ uniformly at random from the set $S_{\bar{\chi}}$.
Partition the set $H_{\tilde{e},x}^{(i)}(j,q)$ arbitrarily into $\bar{\chi}$ disjoint subsets $P_{\tilde{e},x}^{(i)}(j,q,\ell)$, $\ell \in [\bar{\chi}]$.
For every index $\ell \in [\bar{\chi}]$, connect the vertex $cp_\ell^{(i)}(x,j)$ to every vertex v in the set $P_{\tilde{e},x}^{(i)}(j,q,\sigma(\ell))$.
(The step 2d will be referred as the *random permutation step.*)

(e) i. For every copy $cp(x) \in \chi_x^{(i)}(j)$, and every vertex $v \in H_{\tilde{e},x}^{(\bar{i})}(j,q)$, with probability $1/J$ insert the edge (x,v) into the edge set $\bar{E}$, independently at random.
ii. Do the same for every copy $cp(x) \in \chi_x^{(i)}(j)$, and every vertex $v \in M_{\tilde{e}}(j,q) \setminus H_{\tilde{e},x}(j,q)$.
iii. Do the same for every copy $cp(x) \in \chi_x \setminus \chi_x(j)$, and every vertex $v \in M_{\tilde{e}}(j,q)$.
(The step 2e will be referred as the *mixing step.*)

The effect of the mixing and permutation steps on the constructed graph is illustrated by Figure 3.

4 Analysis

In this section we analyze the reduction described in the previous section.

We need some additional notation. For a pair of vertices $x \in \mathcal{V}_1$ and $v \in \mathcal{V}_2$, let $(x\ \mathcal{C}\ v)$ denote the event $((x,v) \in \bar{E})$, and let $(x\ \mathcal{N}\ v)$ denote the complimentary event $((x,v) \notin \bar{E})$. For a subset $X \subseteq \mathcal{V}_1$ and a vertex $v \in \mathcal{V}_2$, let $(X\ \mathcal{AC}\ v)$ denote the event $(\forall x \in X,\ (x,v) \in \bar{E})$, and, analogously, let $(X\ \mathcal{AN}\ v)$ denote the event $(\forall x \in X,\ (x,v) \notin \bar{E})$. For a pair of subsets $X \subseteq \mathcal{V}_1$, $U \subseteq \mathcal{V}_2$, let $(X\ \mathcal{AC}\ U)$ denote the event $(\forall x \in X, v \in U,\ (x,v) \in \bar{E})$, and, analogously, let $(X\ \mathcal{AN}\ U)$ denote the event $(\forall x \in X, v \in U,\ (x,v) \notin \bar{E})$.

For a subset $X \subseteq \mathcal{V}_1$, a vertex $v \in \mathcal{V}_2$, and a positive integer $\ell = 1, 2, \ldots$, let $(X\ (\geq \ell\mathcal{C})\ v)$ denote the event $(|\{x \in X \mid (x,v) \in \bar{E}\}| \geq \ell)$. Analogously, let $(X\ (\leq \ell\mathcal{C})\ v)$ denote the

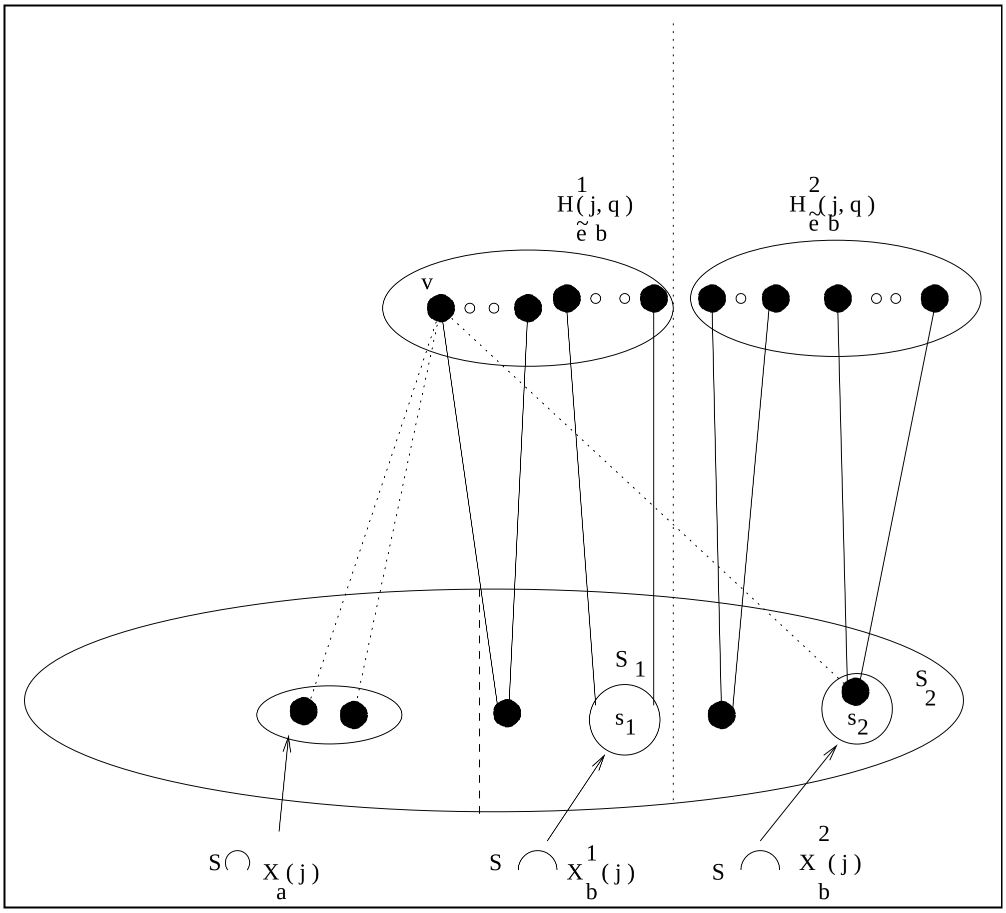

Figure 3: The effect of the mixing and permutation steps on the construction. Only vertices that belong to the set $\mathcal{S}_1$ may be connected to the vertex v by the permutation step. The vertices of the set $\mathcal{S}_2$ are connected to the vertex v independently with probability $1/2^j$. In addition, as $v \in H_{\tilde{e},b}(j,q)$, all the copies of the vertex a are connected to the vertex v with probability $1/2^j$ as well. The figure illustrates the "silent scenario", hence the set $\mathcal{S}_1$ is not connected to the vertex v by the permutation step; the non-edges are depicted by dotted lines.

event $(|\{x \in X \mid (x,v) \in \bar{E}\}| \leq \ell)$, and let $(X\ (\ell\mathcal{C})\ v)$ denote the event $(|\{x \in X \mid (x,v) \in \bar{E}\}| = \ell)$. We also use the notation $(X\ \mathcal{INF}\ v)$ for the event $(X\ (1\mathcal{C})\ v)$; in this case we say that the subset X *informs* the vertex v. For the complimentary event we use the notation $(X\ \mathcal{NINF}\ v)$, and say that the subset X *does not inform* the vertex v. For a subset $U \subseteq \mathcal{V}_2$, we use the notation $(X\ \mathcal{INF}\ U)$ for the event $(\forall v \in U,\ (X\ \mathcal{INF}\ v))$; in this case we say that the subset X informs the subset U. For the complimentary event we use the notation $(X\ \mathcal{NINF}\ U)$, and say that the subset X *does not inform* the subset U. For a collection of subsets $\Pi = (X_1, X_2, \ldots, X_t)$ of $\mathcal{V}_1$ and a vertex $v \in \mathcal{V}_2$ (respectively, a subset $U \subseteq \mathcal{V}_2$), we say that the collection Π *informs the vertex* v (resp., *the subset* U), and denote it by $(\Pi\ \mathcal{INF}\ v)$ (resp., $(\Pi\ \mathcal{INF}\ U)$), if *at least one subset* $X \in \Pi$ informs v (resp., U). For the complimentary event we use the notation $(\Pi\ \mathcal{NINF}\ v)$ (resp., $(\Pi\ \mathcal{NINF}\ U)$ and say that the collection Π *does not inform the vertex* v (resp., *the subset* U).

For a fixed super-edge $\tilde{e} = (A,B) \in \tilde{E}$, and a vertex $x \in A \cup B$, let $star(x)$ denote the subset of vertices of $A \cup B$ that belong to the same star as x with respect to the super-edge $\tilde{e}$. For a subset $S \subseteq A \cup B$, let $cp(S)$ denote the set of all the copies of the vertices of S, i.e., $cp(S) = \{cp_\ell(x) \mid x \in S\}$.

4.1 Basic Properties

Definition 4.1 *For a super-edge $\tilde{e} = (A,B)$, a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$ is called $(\tilde{e}, j)$-*partial *if for every $\tilde{e}$-m.p. (a,b), the set $\mathcal{S}$ contains at most 2^{j-3} j-relevant copies of the vertex a or it contains at most 2^{j-3} j-relevant copies of the vertex b.*

Consider some pair of vertices (a,b) with $a \in A$ and $b \in B$, for some pair of super-vertices $A \in \tilde{V}_1$ and $B \in \tilde{V}_2$. Let $j \in \hat{J}$, $q \in [L]$ be a pair of indices. Consider a subset $\mathcal{S} \subseteq \chi_a(j) \cup \chi_b(j)$, and a vertex $v \in H_{\tilde{e},b}(j,q)$. Recall that the set $\chi_b(j)$ is split into the disjoint union of the subsets $\chi_b^{(1)}(j)$ and $\chi_b^{(2)}(j)$ by a binomial random partition (see Section 3, step 2c). The set $H_{\tilde{e},b}(j,q)$ is split into a disjoint union of the two subsets $H^{(1)}_{\tilde{e},b}(j,q)$ and $H^{(2)}_{\tilde{e},b}(j,q)$, and these two subsets have the same size $M/2$. Let $\mathcal{S}_1 = \mathcal{S} \cap \chi_b^{(1)}(j)$, $\mathcal{S}_2 = \mathcal{S} \cap \chi_b^{(2)}(j)$.

Lemma 4.2 *For any positive constant $\epsilon > 0$, for a sufficiently large set $\mathcal{S}$, $\mathbb{P}(|\mathcal{S}_2 \cup (\mathcal{S} \cap \chi_a(j))| \geq |\mathcal{S}|/(2+\epsilon)) \geq 1 - exp(-|\mathcal{S}|)$.*

Proof: Set $\delta = \epsilon/(2+\epsilon)$. Since $\mathbb{E}(|\mathcal{S} \cap \chi_b^{(2)}(j)|) = |\mathcal{S} \cap \chi_b(j)|/2$, by Claim 2.6,

$$\begin{aligned}
\mathbb{P}(|\mathcal{S} \cap \chi_b^{(2)}(j)| \leq |\mathcal{S} \cap \chi_b(j)|/(2+\epsilon)) &= \mathbb{P}(|\mathcal{S} \cap \chi_b^{(2)}(j)| \leq \mathbb{E}(|\mathcal{S} \cap \chi_b^{(2)}(j)|) \cdot (1-\delta)) \\
\leq\ exp(-\frac{(\delta \cdot \mathbb{E}(|\mathcal{S} \cap \chi_b^{(2)}(j)|))^2}{2 \cdot |\mathcal{S} \cap \chi_b(j)|}) &= exp(-\Omega(|\mathcal{S} \cap \chi_b(j)|))\ .
\end{aligned}$$

If $|\mathcal{S} \cap \chi_a(j)| \geq |\mathcal{S}|/(2+\epsilon)$, then the event $(|\mathcal{S}_2 \cup (\mathcal{S} \cap \chi_a(j))| \geq |\mathcal{S}|/(2+\epsilon))$ occurs with probability 1. Otherwise, since $|\mathcal{S}| = |\mathcal{S} \cap \chi_a(j)| + |\mathcal{S} \cap \chi_b(j)|$, it follows that $|\mathcal{S} \cap \chi_b(j)| \geq \frac{1+\epsilon}{2+\epsilon}|\mathcal{S}|$, and so the probability of the event $(|\mathcal{S}_2| \leq |\mathcal{S} \cap \chi_b(j)|/(2+\epsilon))$ is $exp(-\Omega(|\mathcal{S}|))$. Hence $\mathbb{P}(|\mathcal{S}_2| + |\mathcal{S} \cap \chi_a(j)| \leq |\mathcal{S} \cap \chi_b(j)|/(2+\epsilon) + |\mathcal{S} \cap \chi_a(j)|) \leq exp(-\Omega(|\mathcal{S}|))$. Since $|\mathcal{S} \cap \chi_b(j)|/(2+\epsilon) + |\mathcal{S} \cap \chi_a(j)| \geq |\mathcal{S}|/(2+\epsilon)$, it follows that

$$\begin{aligned} &\mathbb{P}(|\mathcal{S}_2| + |\mathcal{S} \cap \chi_a(j)| \leq |\mathcal{S}|/(2+\epsilon)) \\ &\leq \mathbb{P}(|\mathcal{S}_2| + |\mathcal{S} \cap \chi_a(j)| \leq |\mathcal{S} \cap \chi_b(j)|/(2+\epsilon) + |\mathcal{S} \cap \chi_a(j)|) \leq exp(-\Omega(|\mathcal{S}|)) . \qquad \blacksquare \end{aligned}$$

Let $s_1 = |\mathcal{S}_1|$, $s_2 = |\mathcal{S}_2|$, $w = |\chi_b^{(1)}(j)|$ and $u = |\mathcal{S} \cap \chi_a(j)|$. Note that s_1, s_2, and w are random variables, and u is a fixed value that does not depend on random coins.

Lemma 4.3 *For a super-edge $\tilde{e} = (A, B) \in \tilde{E}$, a pair of vertices $a \in A$, $b \in B$, a pair of indices $j \in \hat{J}$, $q \in [L]$, a subset $\mathcal{S} \subseteq \chi_a(j) \cup \chi_b(j)$ and a vertex $v \in M_{\tilde{e}}(j,q)$,*

$$\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v) \ \geq\ (1-1/2^j)^{|\mathcal{S}|} \cdot \sum_{w,s_1} \frac{w-s_1}{w} \cdot \mathbb{P}(|\mathcal{S}_1| = s_1, |\chi_b^{(1)}(j)| = w) \ .$$

Proof: Consider the case $v \in H_{\tilde{e},b}^{(1)}(j,q)$ (the other cases are symmetric). As $v \notin H_{\tilde{e},a}(j,q)$, by the mixing step of the reduction, each copy $cp_i(a)$ of the vertex a is connected to the vertex v with probability $1/2^j$. Hence,

$$\mathbb{P}(((\mathcal{S} \cap \chi_a(j))\ \mathcal{AN}\ v) \mid |\mathcal{S} \cap \chi_a(j)| = u) = (1-1/2^j)^u \ .$$

Consider a vertex $s \in \mathcal{S}_2 = \mathcal{S} \cap \chi_b^{(2)}(j)$. Recall that every vertex $cp(b) \in \chi_b^{(2)}(j)$ is connected to $v \in H_{\tilde{e},b}^{(1)}(j,q)$ with probability $1/2^j$ as well. Note that $u + s_2 \leq |\mathcal{S}|$. Hence,

$$\mathbb{P}(\mathcal{S}_2\ \mathcal{AN}\ v \mid |\mathcal{S} \cap \chi_a(j)| = u, |\mathcal{S}_2| = s_2) = (1-1/2^j)^{s_2} \geq (1-1/2^j)^{|\mathcal{S}|-u} \ .$$

Next, consider the probability of the event $(\mathcal{S}_1\ \mathcal{AN}\ v)$. Note that by the random permutation step, for any fixed vertex $z \in \mathcal{S}_1$, $\mathbb{P}(z\ \mathcal{C}\ v \mid |\chi_b^{(1)}(j)| = w) = 1/w$. Hence, $\mathbb{P}(\exists z \in \mathcal{S}_1 \text{ s.t. } (z\ \mathcal{C}\ v) \mid |\mathcal{S}_1| = s_1, |\chi_b^{(1)}(j)| = w) \leq s_1/w$. It follows that $\mathbb{P}(\mathcal{S}_1\ \mathcal{AN}\ v \mid |\mathcal{S}_1| = s_1, |\chi_b^{(1)}(j)| = w) \geq \frac{w-s_1}{w}$. Hence,

$$\mathbb{P}(\mathcal{S}_1\ \mathcal{AN}\ v) \geq \sum_{w,s_1} \frac{w-s_1}{w} \cdot \mathbb{P}(|\mathcal{S}_1| = s_1, |\chi_b^{(1)}(j)| = w) \ .$$

Since the events $((\mathcal{S} \cap \chi_a(j))\ \mathcal{AN}\ v)$, $(\mathcal{S}_1\ \mathcal{AN}\ v)$ and $(\mathcal{S}_2\ \mathcal{AN}\ v)$ are independent, it follows that

$$\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v) \geq (1-1/2^j)^{|\mathcal{S}|} \cdot \sum_{w,s_1} \frac{w-s_1}{w} \cdot \mathbb{P}(|\mathcal{S}_1| = s_1, |\chi_b^{(1)}(j)| = w) \ . \qquad \blacksquare$$

Consider a subset $\mathcal{S} \subseteq \chi_A(j) \cup \chi_B(j)$. (Notice that in Lemma 4.2, the set $\mathcal{S}$ was a subset of the set $\chi_a(j) \cup \chi_b(j)$ for a specific pair of vertices $a \in A$, $b \in B$.) Let $S = S(\mathcal{S})$ be the subset of vertices of $A \cup B$ that have copies in the set $\mathcal{S}$. Note that the set $\mathcal{S}$ decomposes into a disjoint collection of the subsets $\mathcal{S}(a) = cp(star(a)) \cap \mathcal{S}$, for different vertices $a \in S$. For a yet more general subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$, we will use the notation $S_j = S_j(\mathcal{S})$ for the subset of vertices of $A \cup B$ that have j-relevant copies in the set $\mathcal{S}$.

We will use the notation x to denote an arbitrary vertex $x \in S$, both for $x \in A$ and $x \in B$. Let $PN_x(v)$ denote the subset of copies of the vertex x that have a non-zero probability of being connected to the vertex $v \in M_{\tilde{e}}(j,q)$ *by the random permutation step.* In other words, let $j(v)$ be the index j such that $v \in M_{\tilde{e}}(j,q)$. If $v \in M_{\tilde{e}}(j,q) \setminus H_{\tilde{e},x}(j,q)$, then set $PN_x(v) = \emptyset$. Otherwise, let $i_v(x)$ be the index i such that $v \in H^{(i)}_{\tilde{e},x}(j,q)$. Then set $PN_x(v) = \chi_x^{(i_v(x))}(j)$.

Definition 4.4 *Consider a fixed super-edge $\tilde{e} = (A,B)$, a pair of indices $j \in \hat{J}$, $q \in [L]$, a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$, and a vertex $v \in M_{\tilde{e}}(j,q)$. The exact partitions of the set $M_{\tilde{e}}(j,q)$ into the unions of the subsets $H_{\tilde{e},a}(j,q)$ and $H_{\tilde{e},b}(j,q)$ for different $\tilde{e}$-m.p.s (a,b), the exact partitions of the sets $H_{\tilde{e},x}(j,q)$ into the unions of the subsets $H^{(1)}_{\tilde{e},x}(j,q)$ and $H^{(2)}_{\tilde{e},x}(j,q)$ for different vertices $x \in A \cup B$, and the binomial partitions of the sets $\chi_x(j)$ into the unions of the subsets $\chi_x^{(1)}(j)$ and $\chi_x^{(2)}(j)$ for different vertices $x \in A \cup B$ are called* safe with respect to the pair $(\mathcal{S}, v)$ *if for every vertex $x \in S_j(\mathcal{S})$, $|\mathcal{S} \cap PN_x(v)| \leq 2^{j-3}$.*

In the following lemma the word "probability" and the notation "$\mathbb{P}$" refer to the sample space that is defined by the appropriate exact and binomial partitions.

Lemma 4.5 *Let $\mathcal{S}$ be an $(\tilde{e},j)$-partial subset of $\chi_A \cup \chi_B$ (see Definition 4.1) of size $|\mathcal{S}| \leq 2^j \ln n$, and $v \in M_{\tilde{e}}(j,q)$ be a vertex. The probability that the exact partitions of the set $M_{\tilde{e}}(j,q)$, and the binomial partitions of the sets $\chi_x(j)$ are safe with respect to the pair $(\mathcal{S}, v)$ is at least $1/n^8$.*

Proof: Since $|\mathcal{S}| \leq 2^j \cdot \ln n$, and the stars $cp(star(a))$ with different vertices $a \in A$ are all disjoint, the number of such stars that satisfy $|cp(star(a)) \cap \mathcal{S}| \geq 2^{j-3}$ is at most $8 \cdot \ln n$. Throughout the proof of this lemma, we will call such stars *dangerous.*

Consider a dangerous star $cp(star(a))$. Since the set $\mathcal{S}$ is $(\tilde{e},j)$-partial, *either* it contains at most 2^{j-3} j-relevant copies of the vertex a, *or for all* vertices b with $\tilde{e}(b) = a$, the set $\mathcal{S}$ contains at most 2^{j-3} copies of the vertex b.

Suppose first that the latter condition holds. In this case, if $v \notin H_{\tilde{e},a}(j,q)$ (and this happens with probability 1/2) then necessarily $PN_a(v) = \emptyset$, and so $|PN_a(v) \cap \mathcal{S}| = 0$. Also, for all the vertices b in this star (i.e., the vertices b with $\tilde{e}(b) = a$), by assumption $|PN_b(v) \cap \mathcal{S}| \leq 2^{j-3}$.

Hence, in this case with probability at least $1/2$, the star $cp(star(a))$ satisfies that for every vertex $x \in star(a)$, $|PN_x(v) \cap \mathcal{S}| \leq 2^{j-3}$.

Similarly, if the set $\mathcal{S}$ contains at most 2^{j-3} j-relevant copies of the vertex a, then with probability $1/2$ the event $(v \in H_{\tilde{e},a}(j,q))$ occurs. Then necessarily, for every vertex b in this star (i.e., such that $\tilde{e}(b) = a$), no copy of b can be made adjacent to the vertex v by the *random permutation step*, and so $PN_b(v) \cap \mathcal{S}| = 0$. Hence, again with probability at least $1/2$, the star $cp(star(a))$ satisfies that for every vertex $x \in star(a)$, $|PN_x(v) \cap \mathcal{S}| \leq 2^{j-3}$.

In other words, for every dangerous star $cp(star(a))$, with probability at least $1/2$, the exact and the binomial partitions are safe with respect to the pair $(cp(star(a)), v)$. Observe that the same event occurs with probability 1 for every star that is not dangerous.

Let us denote by $(Safe\ a)$ the event (the partitions are safe with respect to the pair $(cp(star(a)), v)$). Consider a collection of disjoint stars $\{cp(star(a_1)), cp(star(a_2)), \ldots, cp(star(a_t))\}$. Observe that the above argument actually shows that $\mathbb{P}((Safe\ a_1) \mid \forall \ell \in [t] \setminus \{1\}, (Safe\ a_\ell)) \geq 1/2$. Since there are at most $8 \cdot \ln n$ dangerous stars, the probability that the partitions are safe with respect to all the dangerous stars of the set $\mathcal{S}$ is at least $(1/2)^{8 \ln n} \geq 1/n^8$, as required. ▮

Corollary 4.6 *With probability at least $1/n^8$, for every vertex $x \in S$, $|PN_x(v) \cap \mathcal{S}| \leq 2^{j-3}$.*

4.2 Probability of Silence

In this section we analyze the probability of the event $(\mathcal{S}\ \mathcal{AN}\ v)$ for a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$, and a vertex $v \in M_{\tilde{e}}(j,q)$, $\tilde{e} = (A,B)$.

Let S' be the subset of $S_j(\mathcal{S})$ that contains all the vertices x such that $PN_x(v) \cap \mathcal{S} \neq \emptyset$. For every vertex $x \in S'$, let $z(x) = |\chi_x^{(i_v(x))}(j)|$ and $s(x) = |PN_x(v) \cap \mathcal{S}|$ be random variables. Note that $\sum_x s(x) \leq |\mathcal{S}|$. Under the assumption that $|\mathcal{S}| \leq 2^j \ln n$, it follows that $\sum_x s(x) = \sum_x |PN_x(v) \cap \mathcal{S}| \leq 2^j \ln n$.

Let $W = (\bigcup_{x \in S'} PN_x(v)) \cap \mathcal{S}$.

For a fixed vertex $x \in S'$, consider the probability space Ω_x that is determined by the application of the random permutation step to the vertex x. Specifically, consider the following randomized procedure. First, the set $H_{\tilde{e},x}^{(i_v(x))}(j,q)$ is divided arbitrarily to $|\chi_x^{(i_v(x))}(j)|$ subsets $P_{\tilde{e},x}^{(i_v(x))}(j,q,1), P_{\tilde{e},x}^{(i_v(x))}(j,q,2), \ldots$ of equal size. Then the vertices of the set $\chi_x^{(i_v(x))}(j)$ are permuted by a randomly chosen permutation σ_x, and each copy $cp_\ell(x) \in \chi_x^{(i_v(x))}(j)$ is connected to all the vertices of the set $P_{\tilde{e},x}^{(i_v(x))}(j,q,\sigma_x(\ell))$, where ℓ is the index of the copy $cp_\ell(x)$ according to some arbitrary ordering of the set $\chi_x^{(i_v(x))}(j)$. This randomized procedure determines the probability space Ω_x, and let $\mathbb{P}_x = \mathbb{P}_{\sigma_x}$ denote the probability measure in this space. Also for a pair of values $s(x)$, $z(x)$ of $|PN_x(v) \cap \mathcal{S}|$ and $|PN_x(v)|$ respectively, let $\Omega_{x,s(x),z(x)}$ denote the probability space Ω_x in which necessarily $|PN_x(v) \cap \mathcal{S}| = s(x)$ and $|PN_x(v)| = z(x)$. (Note that the two events $(|PN_x(v) \cap \mathcal{S}| = s(x))$ and $(|PN_x(v)| = z(x))$ do not "live" in the probability space Ω_x, but rather in the probability space that is determined by the binomial partitions.) For a subset $S'' \subseteq S$, let $\mathbb{P}_{S''}$ denote the probability space determined by the choice of the random permutations σ_x for *every* vertex $x \in S''$ (essentially, it is the product space $\Pi_{x \in S''} \Omega_x$). Also, for a fixed vertex $x \in S$, we denote by Ω^*_x the sample space that is defined by

1. the exact partition of the set $M_{\tilde{e}}(j,q)$ to the union of the subsets $H_{\tilde{e},x}(j,q)$ and $M_{\tilde{e}}(j,q) \setminus H_{\tilde{e},x}(j,q)$.
2. the exact partition of the set $H_{\tilde{e},x}(j,q)$ to the union of the subsets $H^{(1)}_{\tilde{e},x}(j,q)$ and $H^{(2)}_{\tilde{e},x}(j,q)$.
3. the binomial partition of the set $\chi_x(j)$.

Let $\mathbb{P}^*_x$ denote the probability measure in the sample space Ω^*_x. We also denote by $\mathbb{P}^{mix}$ (respectively, $\mathbb{P}^{\chi}$) the probability measure in the sample space that is defined by the mixing step (by the binomial partition step).

Consider the probability of the event $((PN_x(v) \cap \mathcal{S})\ \mathcal{AN}\ v)$ in the space Ω_x. For a copy $cp(x) \in PN_x(v)$, $\mathbb{P}_{x,s(x),z(x)}(cp(x)\ \mathcal{C}\ v) = 1/z(x)$. Hence, by union bound,

$$\mathbb{P}_{x,s(x),z(x)}(\exists cp(x) \in (PN_x(v) \cap \mathcal{S}) \ \text{ s.t. } \ (cp(x)\ \mathcal{C}\ v)) \leq s(x)/z(x) \ .$$

Hence,

$$\mathbb{P}_{x,s(x),z(x)}((PN_x(v) \cap \mathcal{S})\ \mathcal{AN}\ v) \ \geq \ \frac{z(x) - s(x)}{z(x)} \ . \tag{1}$$

Hence,

$$\mathbb{P}_x((PN_x(v) \cap \mathcal{S})\ \mathcal{AN}\ v) \ \geq \ \sum_{z(x),s(x)} \frac{z(x) - s(x)}{z(x)} \cdot \mathbb{P}^*_x(|PN_x(v) \cap \mathcal{S}| = s(x), |PN_x(v)| = z(x)) \ . \tag{2}$$

For two different vertices $x, y \in S'$, note that the choices of the random permutations σ_x and σ_y are independent. Also, recall that $PN_x(v) \subseteq \chi_x(j)$, $PN_y(v) \subseteq \chi_y(j)$, and since the sets $\chi_x(j)$ and $\chi_y(j)$ are disjoint, the sets $PN_x(v) \cap \mathcal{S}$ and $PN_y(v) \cap \mathcal{S}$ are disjoint as well. Hence,

$$\mathbb{P}_{\{x,y\}}((((PN_x(v) \cap \mathcal{S})\ \mathcal{AN}\ v), ((PN_y(v) \cap \mathcal{S})\ \mathcal{AN}\ v)) \ = \\ \mathbb{P}_x((PN_x(v) \cap \mathcal{S})\ \mathcal{AN}\ v) \cdot \mathbb{P}_y((PN_y(v) \cap \mathcal{S})\ \mathcal{AN}\ v) \ .$$

These considerations readily generalize to a subset $S' \subseteq S$ of cardinality greater than 2. Note also that for any vertex $x \in S \setminus S'$, $\mathbb{P}_x((PN_x(v) \cap \mathcal{S})\ \mathcal{AN}\ v) = 1$. Hence,

$$\begin{aligned} \mathbb{P}_S(W\ \mathcal{AN}\ v) &= \Pi_{x \in S} \mathbb{P}_x((PN_x(v) \cap \mathcal{S})\ \mathcal{AN}\ v) \qquad (3) \\ &\geq \Pi_{x \in S} \sum_{z(x), s(x)} \frac{z(x) - s(x)}{z(x)} \cdot \mathbb{P}^*_x(|PN_x(v) \cap \mathcal{S}| = s(x), |PN_x(v)| = z(x)) \ . \end{aligned}$$

Lemma 4.7 *For a set $\mathcal{S} \subseteq \chi_A \cup \chi_B$, and a sequence of values $(y(x) \mid x \in S')$ that satisfies that $\sum_{x \in S'} y(x) \leq 2^j \ln n$, and for every $x \in S'$, $y(x) \leq 2^{j-3}$, the following inequality holds:*

$$\mathbb{P}((\mathcal{S}\ \mathcal{AN}\ v) \mid \forall x \in S', s(x) = y(x)) \geq 1/n^8 \ ,$$

where the notation $\mathbb{P}$ stands for the probability measure that is defined by

1. *the random permutation step applied to the vertices $x \in S'$.*
2. *the mixing step applied to the vertices $z \in \mathcal{S} \setminus W$.*

Proof: Recall that for a vertex $x \in S'$, $PN_x(v) = \chi_x^{(i_v(x))}(j)$. By the previous considerations (inequalities (1), (2) and (3)),

$$\begin{aligned} \mathbb{P}_{S'}(W\ \mathcal{AN}\ v \mid \forall x \in S', s(x) = y(x)) &\geq \Pi_{x \in S'} \sum_{z(x)} \left(\frac{z(x) - y(x)}{z(x)} \cdot \mathbb{P}^*_x(|PN_x(v)| = z(x)) \right) \\ &= \Pi_{x \in S'} \sum_{z(x)} \left(\frac{z(x) - y(x)}{z(x)} \cdot \mathbb{P}^*_x(|\chi_x^{(i_v(x))}(j)| = z(x)) \right) \\ &\geq \Pi_{x \in S'} \sum_{z(x)} \left(((1 - \frac{y(x)}{z(x)}) \cdot e^{-1})^{y(x)/z(x)} \cdot \mathbb{P}^*_x(|\chi_x^{(i_v(x))}(j)| = z(x)) \right) \ . \end{aligned}$$

(The last inequality is because for any positive real $u > 0$, $(1 - 1/u)^{u-1} \geq 1/e$, and so $1 - 1/u \geq ((1 - 1/u) \cdot e^{-1})^{1/u}$.)

Observe that the function $f(z) = ((1 - y/z)e^{-1})^{y/z}$ is monotonically increasing with z for sufficiently large z. Since $2^j/3 \geq n^{0.4}/3$, it follows that for each term for which $z(x) \geq 2^j/3$, $((1 - \frac{y(x)}{z(x)})e^{-1})^{y(x)/z(x)} \geq ((1 - \frac{3 \cdot y(x)}{2^j} e^{-1})^{\frac{3 \cdot y(x)}{2^j}}$. Hence,

$$\begin{aligned} &\sum_{z(x) \geq 2^j/3} \left(((1 - \frac{y(x)}{z(x)}) \cdot e^{-1})^{y(x)z(x)} \cdot \mathbb{P}^*_x(|\chi_x^{(i_v(x))}(j)| = z(x)) \right) \\ &\geq ((1 - \frac{3 \cdot 2^j}{2^j})e^{-1})^{\frac{3 \cdot y(x)}{2^j}} \cdot \sum_{z(x) \geq 2^j/3} \mathbb{P}^*_x(|\chi_x^{(i_v(x))}(j)| = z(x)) \\ &= ((1 - \frac{3 \cdot 2^j}{2^j})e^{-1})^{\frac{3 \cdot y(x)}{2^j}} \cdot \mathbb{P}^*_x(|\chi_x^{(i_v(x))}(j)| \geq 2^j/3) \ . \end{aligned}$$

By Lemma 4.2, $\mathbb{P}^*_x(|\chi_x^{(i_v(x))}(j)| \geq 2^j/3) \geq 1 - \exp(-\Omega(n^{0.4}))$. Hence,

$$\sum_{z(x)} \left(((1 - \frac{y(x)}{z(x)}) \cdot e^{-1})^{y(x)/z(x)} \cdot \mathbb{P}^*_x(|\chi_x^{(i_v(x))}(j)| = z(x)) \right) \geq$$
$$((1 - \frac{3 \cdot y(x)}{2^j})e^{-1})^{\frac{3 \cdot y(x)}{2^j}} \cdot (1 - \exp(-\Omega(n^{0.4}))\ .$$

Recall that for every vertex $x \in S'$, $y(x) \leq 2^{j-3}$. Hence, $\frac{y(x)}{2^j/3} \leq 3/8$, and so $(1 - \frac{3 \cdot y(x)}{2^j}) \cdot e^{-1} \geq \frac{5}{8 \cdot e}$. Hence,

$$\Pi_{x \in S'} \sum_{z(x)} \left(((1 - \frac{y(x)}{z(x)})e^{-1})^{y(x)/z(x)} \cdot \mathbb{P}^*_x(|\chi_x^{(i_v(x))}(j)| = z(x) \right) \geq$$
$$\Pi_{x \in S'}((1 - \frac{3 \cdot y(x)}{2^j})e^{-1})^{\frac{3 \cdot y(x)}{2^j}}(1 - exp(-\Omega(n^{0.4}))) \geq$$
$$(1 - exp(-\Omega(n^{0.4})))\Pi_{x \in S'}(5/8e)^{\frac{3 \cdot y(x)}{2^j}} \geq (1 - exp(-\Omega(n^{0.4})))(5/8e)^{\sum_{x \in S'} y(x)}$$

Recall that $\sum_{x \in S'} y(x) \leq 2^j \cdot \log n$. Hence, a lower bound on this probability is $(1 - exp(-\Omega(n^{0.4})))(5/8e)^{3 \log n} \geq 1/n^6$. In other words, $\mathbb{P}_{S'}(W\ \mathcal{AN}\ v \mid \forall x \in S', s(x) = y(x)) \geq 1/n^6$ for any sequence of values $(y(x) \mid x \in S')$ that are all at most 2^{j-3} and such that $\sum_{x \in S'} y(x) \leq 2^j \log n$.

Recall that for a fixed vertex $x \in S$, $PN_x(v) \cap \mathcal{S}$ is a subset of vertices of the set $\mathcal{S}$ that have a non-zero probability to be connected to v by the permutation step of the reduction. Other vertices of the set $\mathcal{S}$, i.e., the vertices that belong to the set $\mathcal{S} \setminus W$, are connected to v with probability $1/2^j$ independently at random by the mixing step of the reduction. (Note that this is true for all the copies of the vertices $x \in S \setminus S'$ regardless of the index ℓ of the particular copy $cp_\ell(x)$, and, also, for the copies $cp_\ell(x)$ of the vertices $x \in S'$, whenever $\ell > 2^j$.) Hence, $\mathbb{P}^{mix}((\mathcal{S} \setminus W)\ \mathcal{AN}\ v) = (1 - 1/2^j)^{|\mathcal{S} \setminus W|} \geq (1 - 1/2^j)^{|\mathcal{S}|} \geq (1 - 1/2^j)^{2^j \log n} \geq 1/n^2$. Note also that the last calculation applies to the conditional probability $\mathbb{P}^{mix}((\mathcal{S} \setminus W)\ \mathcal{AN}\ v \mid \forall x \in S', s(x) = y(x))$ as well. Finally, the event "no vertex of the set $\mathcal{S} \setminus W$ is connected to the vertex v by the mixing step" is independent of the event "no vertex of the set W is connected to the vertex v by the permutation step". Hence,

$$\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v \mid \forall x \in S', s(x) = y(x)) =$$
$$\mathbb{P}_{S'}(W\ \mathcal{AN}\ v \mid \forall x \in S', s(x) = y(x)) \cdot \mathbb{P}^{mix}((\mathcal{S} \setminus W)\ \mathcal{AN}\ v \mid \forall x \in S', s(x) = y(x)) \geq$$
$$\frac{1}{n^6} \cdot \frac{1}{n^2} = \frac{1}{n^8}\ . \quad \blacksquare$$

Corollary 4.8 *For a pair of indices $j \in \hat{J}$, $q \in [L]$, a super-edge $\tilde{e} = (A, B)$, a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$ of cardinality at most $2^j \log n$, and a vertex $v \in M_{\tilde{e}}(j, q)$,*

$$\mathbb{P}((\mathcal{S}\ \mathcal{AN}\ v) \mid \{\mathcal{S} \cap \chi_x^{(i_v(x))}(j) \mid x \in S'(\mathcal{S})\} \textit{ is a safe partition with respect to } \mathcal{S}) \geq 1/n^8\ ,$$

where the notation $\mathbb{P}$ has the same meaning as in the statement of Lemma 4.7.

Proof: By definition of $s(x)$, $s(x) = |\mathcal{S} \cap \chi_x^{(i_x(v))}(j)|$. By definition of a safe partition, it follows that for every $x \in S'$, $s(x) \leq 2^{j-3}$. Hence, the assertion follows by Lemma 4.7. ∎

Corollary 4.9 *Let j, q, $\tilde{e}$, $\mathcal{S}$ and v be as in Corollary 4.8, and, in addition, suppose that the set $\mathcal{S}$ is $(\tilde{e}, j)$-partial. Then $\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v) \geq 1/n^{16}$, where the notation $\mathbb{P}$ stands for the probability measure that is defined by*

1. *the exact partitions of the set $M_{\tilde{e}}(j,q)$ into the union of the subsets $H_{\tilde{e},x}(j,q)$ and $M_{\tilde{e}}(j,q) \setminus H_{\tilde{e},x}(j,q)$, for every vertex $x \in S$.*
2. *the exact partitions of the sets $H_{\tilde{e},x}(j,q)$ into the union of subsets $H^{(1)}_{\tilde{e},x}(j,q)$ and $H^{(2)}_{\tilde{e},x}(j,q)$, for every vertex $x \in S$.*
3. *the binomial partitions of the sets $\chi_x(j)$, for every vertex $x \in S$.*
4. *the random permutations applied to the vertices $x \in S'$.*
5. *the mixing step applied to the vertices $z \in \mathcal{S} \setminus W$.*

Proof: By Lemma 4.5, with probability $1/n^8$, the random partitions are safe with respect to the set $\mathcal{S}$, i.e., for every vertex $x \in S'$, $s(x) = |\mathcal{S} \cap \chi_x^{(i_v(x))}(j)| \leq 2^{j-3}$. By Corollary 4.8,

$$\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v \mid \mathcal{S} \text{ induces a safe partition}) \ \geq\ 1/n^8 \ .$$

Hence, with probability at least $1/n^{16}$, the set $\mathcal{S}$ induces a safe partition, and the event $(\mathcal{S}\ \mathcal{AN}\ v)$ occurs. Hence, $\mathbb{P}(\mathcal{S}\ \mathcal{AN}\ v) \geq 1/n^{16}$. ∎

Note that the steps of the reduction that are not listed in (1)-(5) (Corollary 4.9) cannot possibly affect the probability of a vertex from the set $\mathcal{S}$ being connected to the vertex v. Therefore, denoting by $\mathbb{P}$ the probability measure in the sample space that is defined by the entire reduction, we conclude that

Corollary 4.10 *Let j, q, $\tilde{e}$, $\mathcal{S}$ and v be as in Corollary 4.9. Then $\mathbb{P}(s\ \mathcal{AN}\ v) \geq \frac{1}{n^{16}}$.*

4.3 Probability of a Collision

In this section we analyze the probability of the event $(\mathcal{S}\ (\geq 2\mathcal{C})\ v)$ for a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$, and a vertex $v \in M_{\tilde{e}}(j,q)$, $\tilde{e} = (A,B)$.

Consider a vertex x of the instance of the MAX-REP problem, and let $\mathcal{S}(j) = \mathcal{S} \cap \chi_x(j)$ be a subset of the set $\mathcal{S}$ that contains the j-relevant copies of the vertex x. By Lemma 4.2, for any index $i \in \{1,2\}$,

$$\mathbb{P}^{\chi}(|\mathcal{S}(j) \cap \chi_x^{(i)}(j)| \geq |\mathcal{S}(j)|/(2+\epsilon)) \ \geq \ 1 - exp(-\Omega(|\mathcal{S}(j)|)) \ . \tag{4}$$

Consider a subset $\mathcal{S} \subseteq \chi_x$. Note that

$$|\mathcal{S} \setminus \mathcal{S}(j)| + |\mathcal{S}(j)|/(2+\epsilon) \ \geq \ (|\mathcal{S} \setminus \mathcal{S}(j)| + |\mathcal{S}(j)|)/(2+\epsilon) \ = \ |\mathcal{S}|/(2+\epsilon) \ . \tag{5}$$

By inequality (4), $\mathbb{P}^{\chi}(|\mathcal{S} \setminus \mathcal{S}(j)| + |\mathcal{S}(j) \cap \chi_x^{(i)}(j)| \geq |\mathcal{S} \setminus \mathcal{S}(j)| + |\mathcal{S}(j)|/(2+\epsilon)) \geq 1 - exp(-\Omega(|\mathcal{S}(j)|))$. In other words, by inequality (5),

$$\mathbb{P}^{\chi}(|\mathcal{S} \setminus \mathcal{S}(j)| + |\mathcal{S}(j) \cap \chi_x^{(i)}(j)| \geq |\mathcal{S}|/(2+\epsilon)) \geq 1 - exp(-\Omega(|\mathcal{S}(j)|)) \ . \tag{6}$$

Consider two cases. First, if $|\mathcal{S}(j)| \geq \frac{1+\epsilon}{2+\epsilon}|\mathcal{S}|$ then by inequality (6),

$$\mathbb{P}^{\chi}(|\mathcal{S} \setminus \mathcal{S}(j)| + |\mathcal{S}(j) \cap \chi_x^{(i)}(j)| \geq |\mathcal{S}|/(2+\epsilon)) \geq 1 - exp(-\Omega(|\mathcal{S}|)) \ . \tag{7}$$

Second, if $|\mathcal{S}(j)| < \frac{1+\epsilon}{2+\epsilon}|\mathcal{S}|$, then $|\mathcal{S} \setminus \mathcal{S}(j)| \geq |\mathcal{S}|/(2+\epsilon)$, and then with probability 1, $|\mathcal{S} \setminus \mathcal{S}(j)| + |\mathcal{S}(j) \cap \chi_x^{(i)}(j)| \geq |\mathcal{S}|/(2+\epsilon)$. It follows that inequality (7) holds in both cases.

To summarize,

Lemma 4.11 *For a vertex $x \in A \cup B$, a subset $\mathcal{S} \subseteq \chi_x$ of the copies of x, and for a pair of indices $j \in \hat{J}$, $i \in \{1,2\}$,*

$$\mathbb{P}^{\chi}(|\mathcal{S} \setminus \mathcal{S}(j)| + |\mathcal{S}(j) \cap \chi_x^{(i)}(j)| \geq |\mathcal{S}|/(2+\epsilon)) \geq 1 - exp(-\Omega(\mathcal{S}|)) \ .$$

Consider a super-edge $\tilde{e} = (A,B)$, and a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$. Let $S = \{x \mid cp(x) \in \mathcal{S}\}$ be the set of vertices that have at least one copy in the set $\mathcal{S}$. For a vertex $x \in S$, and an index $j \in \hat{J}$, let $\mathcal{S}_x(j)$ denote the subset of the j-relevant copies of the vertex x that appear in the set $\mathcal{S}$. Let τ be the partition of the set S into stars with respect to the super-edge $\tilde{e}$. In other words, $\tau = (T_1, T_2, \ldots, T_t)$, where for each star $T \in \tau$, $T = (a, b_1, b_2, \ldots, b_{p(T)})$, with $a \in A$, and for every index $\ell \in [p(T)]$, $b_\ell \in B$ and $\tilde{e}(b_\ell) = a$ (and $p(T)$ denotes the number of leaves of the star T). Also, $\bigcup_{T \in \tau} T = S$, and for every pair of stars $T, T' \in \tau$, $T \cap T' = \emptyset$.

Consider a partition of the set τ of stars into two disjoint subsets α and β, $\alpha \cup \beta = \tau$. Fix an index $j \in \hat{J}$. For each star $T \in \alpha$, fix an index $i(T) \in \{1,2\}$. For each star $T \in \beta$, $T = (a, b_1, b_2, \ldots, b_p)$, fix a sequence of indices $i_1(T), i_2(T), \ldots, i_{p(T)}(T) \in \{1,2\}$. Then

Lemma 4.12 *For a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$, a partition (α, β) of the set of stars $\tau(\mathcal{S})$, an index $j \in \hat{J}$, a choice of indices $i(T) \in \{1,2\}$ for stars $T \in \alpha$, and a choice of indices $i_1(T), i_2(T), \ldots, i_{p(T)} \in \{1,2\}$ for stars $T \in \beta$,*

$$\mathbb{P}^{\chi}(|\mathcal{S} \setminus \bigcup_{x \in S} \mathcal{S}_x(j)| + \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \alpha} \left(|\mathcal{S}_a(j) \cap \chi_a^{(i(T))}(j)| + \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j)| \right) \tag{8}$$
$$+ \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \beta} \left(|\mathcal{S}_a(j)| + \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j) \cap \chi_{b_\ell}^{(i_\ell(T))}| \right) \;\geq\; |\mathcal{S}|/(2+\epsilon)) \;\geq\; 1 - exp(-\Omega(|\mathcal{S}|)) \;.$$

Remark: This is a generic lemma which will be used later on to show that for any set $\mathcal{S}$ and vertex v as above, the set $\mathcal{S}$ contains many vertices that can be connected to the vertex v only by the mixing step of the reduction.

Proof: Regrouping the terms in the left-hand side of the inequality, we get the expression

$$\left(|\mathcal{S} \setminus \bigcup_{x \in S} \mathcal{S}_x(j)| + \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \alpha} \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j)| + \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \beta} |\mathcal{S}_a(j)| \right)$$
$$+ \left(\sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \alpha} |\mathcal{S}_a(j) \cap \chi_a^{(i_\ell(T))}(j)| + \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \beta} \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j) \cap \chi_{b_\ell}^{(i_\ell(T))}(j)| \right) \;.$$

Denote the first term by F and the second term by X. I.e.,

$$\begin{aligned} F &= |\mathcal{S} \setminus \bigcup_{x \in S} \mathcal{S}_x(j)| + \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \alpha} \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j)| + \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \beta} |\mathcal{S}_a(j)| \;, \\ X &= \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \alpha} |\mathcal{S}_a(j) \cap \chi_a^{(i_\ell(T))}(j)| + \sum_{T=(a,b_1,b_2,\ldots,b_{p(T)}) \in \beta} \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j) \cap \chi_{b_\ell}^{(i_\ell(T))}(j)| \;. \end{aligned}$$

Note that F is independent of the choices of indices $\{i(T) \mid T \in \alpha\}$, $\{i_\ell(T) \mid T \in \beta, \ell \in [p(T)]\}$, and its value is fixed whenever the set $\mathcal{S}$ and the index j are fixed. Consider two cases. If $F \geq |\mathcal{S}|/(2+\epsilon)$ then the probability of the event in the assertion of the lemma is 1, and we are done. Otherwise, note that, by construction, for every star $T = (a, b_1, b_2, \ldots, b_{p(T)}) \in \alpha$, and every vertex $y \in \mathcal{S}_a(j) \subseteq \chi_a(j)$, and every index $i(T) \in \{1,2\}$, the equality $\mathbb{P}^{\chi}(y \in \chi_a^{(i(T))}(j)) = 1/2$ holds, independently of all the events that involve other vertices $y' \in \mathcal{S}$. Similarly, by construction, for every star $T = (a, b_1, b_2, \ldots, b_{p(T)}) \in \beta$, and every vertex $y \in \mathcal{S}_{b_\ell}(j) \subseteq \chi_{b_\ell}(j)$, and every index $i_\ell(T) \in \{1,2\}$, the equality $\mathbb{P}^{\chi}(y \in \chi_{b_\ell}^{(i_\ell(T))}(j)) = 1/2$ holds, independently of all the events that involve other vertices. (Note that unlike the case with the *exact* partitions, the probability of a copy $y = cp(x) \in \chi_x(j)$ of the vertex $x \in S$ to belong to the subset $\chi_x^{(i)}(j)$, for some index $i \in \{1,2\}$, is affected neither by the binomial partitions of the vertices $x' \in S$, $x' \neq x$, nor by the events of the form $(y' \in \chi_x^{(i)}(j))$, where $y' \neq y$ is a different copy of the vertex x.)

Hence, the random variable X can be represented as the sum of $|\mathcal{S}| - F$ independent indicator random variables X_i, with $\mathbb{P}^{\chi}(X_i = 1) = \mathbb{P}^{\chi}(X_i = 0) = 1/2$. Hence, by Chernoff inequality, $\mathbb{P}^{\chi}(X \geq \frac{1}{2+\epsilon}(|\mathcal{S}| - F)) \geq 1 - exp(-\Omega(|\mathcal{S}| - F))$. Since $|\mathcal{S}| - F \geq \frac{1+\epsilon}{2+\epsilon}|\mathcal{S}|$, it follows that $\mathbb{P}^{\chi}(X \geq \frac{1}{2+\epsilon}(|\mathcal{S}| - F)) \geq 1 - exp(-\Omega(|\mathcal{S}|))$. Finally, since F and X and $\mathcal{S}$ are all non-negative, $\mathbb{P}^{\chi}(F + X \geq \frac{1}{2+\epsilon}|\mathcal{S}|) = \mathbb{P}^{\chi}(X \geq \frac{1}{2+\epsilon}|\mathcal{S}| - F) \geq \mathbb{P}^{\chi}(X \geq \frac{1}{2+\epsilon}(|\mathcal{S}| - F))$, and so $\mathbb{P}^{\chi}(F + X \geq \frac{1}{2+\epsilon}|\mathcal{S}|) \geq 1 - exp(-\Omega(|\mathcal{S}|))$. ∎

For some super-edge $\tilde{e} = (A, B)$ let $\mathcal{S} \subseteq \chi_A \cup \chi_B$ be a subset of vertices. Let $j \in \hat{J}$ and $q \in [L]$ be a pair of indices, and let v be some fixed vertex in $M_{\tilde{e}}(j, q)$. The collection $\tau = \{T_1, T_2, \ldots, T_t\}$ of subsets of $\mathcal{S}$ is called a *$\tilde{e}$-decomposition* of the set $\mathcal{S}$ into stars, if $\bigcup_{T \in \tau} T = S(\mathcal{S})$, and for each subset $T \in \tau$ and every two vertices $x, y \in T$, x and y belong to the same star with respect to the super-edge $\tilde{e}$. For a fixed super-edge $\tilde{e}$ and a subset $\mathcal{S}$ as above the $\tilde{e}$-decomposition τ of the subset $\mathcal{S}$ is unique, and is denoted $\tau(\mathcal{S})$ (when the super-edge $\tilde{e}$ is clear from the context).

Let $\alpha = \alpha(v) \subseteq \tau$ be the subset of stars $T = (a, b_1, b_2, \ldots, b_{p(T)})$ such that $v \in H_{\tilde{e},a}(j, q)$, and $\beta = \beta(v) = \tau \setminus \alpha(v)$ be the subset of stars $T = (a, b_1, b_2, \ldots, b_{p(T)})$ such that $v \in M_{\tilde{e}}(j, q) \setminus H_{\tilde{e},a}(j, q)$. Note that it may happen that either α or β is an empty set.

For each star $T = (a, b_1, b_2, \ldots, b_{p(T)}) \in \alpha$, let $i'(T) = i'(v, T) \in \{1, 2\}$ be the index such that $v \in H_{\tilde{e},a}^{(i'(T))}(j, q)$, and let $i(T) = \bar{i}'(T)$ be the other index. Analogously, for each star $T = (a, b_1, b_2, \ldots, b_{p(T)}) \in \beta$, and for each index $\ell \in [p(T)]$, let $i'_\ell(T) = i'_\ell(v, T) \in \{1, 2\}$ be the index such that $v \in H_{\tilde{e},b_\ell}^{(i'_\ell(T))}(j, q)$, and let $i_\ell(T)$ be the other index.

Now, Lemma 4.12 is applicable to the set $\mathcal{S} \subseteq \chi_A \cup \chi_B$, the partition (α, β) of the set $\tau = \tau(\mathcal{S})$ of the $\tilde{e}$-decomposition of the set $\mathcal{S}$, and the choice of indices $i(v, T)$ for every star $T \in \alpha$, and $(i_1(v, T), i_2(v, T), \ldots, i_{p(T)}(v, T))$ for every star $T \in \beta$. Hence, inequality (8) holds. I.e.,

$$\mathbb{P}^{\chi}(|\mathcal{S} \setminus (\bigcup_{x \in S} \mathcal{S}_x(j))| + \sum_{T \in \alpha} \left(|\mathcal{S}_a(j) \cap \chi_a^{(i(T))}(j)| + \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j)| \right)$$
$$+ \sum_{T \in \beta} \left(|\mathcal{S}_a(j)| + \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j) \cap \chi_{b_\ell}^{(i_\ell(T))}(j)| \right) \geq \frac{|\mathcal{S}|}{2+\epsilon}) \geq 1 - exp(-\Omega(|\mathcal{S}|)) .$$

Note that by construction (the mixing step), every vertex $z \in \mathcal{S} \setminus (\bigcup_{x \in S} \mathcal{S}_x(j))$ (this is a copy of some vertex $x \in S$, but its index is greater than 2^j) is connected to the vertex v with probability $1/2^j$, independently of all the other events. In other words, for every vertex $z \in \mathcal{S} \setminus (\bigcup_{x \in S} \mathcal{S}_x(j))$, $\mathbb{P}^{mix}(z \; \mathcal{C} \; v) = 1/2^j$.

Consider some star $T \in (a, b_1, b_2, \ldots, b_{p(T)}) \in \alpha$. Note that $v \in H_{\tilde{e},a}(j,q)$. For every vertex $b \in T$, and every copy $cp(b) \in \chi_b(j)$, by construction (the mixing step), the vertex $cp(b)$ is connected to v with probability $1/2^j$, independently of all the other events. In particular, it follows that for every vertex $z \in \bigcup_{b \in T} \mathcal{S}_b(j)$, $\mathbb{P}^{mix}(z \; \mathcal{C} \; v) = 1/2^j$. Note also that $|\bigcup_{b \in T} \mathcal{S}_b(j)| = \sum_{\ell=1}^{p(T)} |\mathcal{S}_{b_\ell}(j)|$.

Also recall that by definition of the index $i(T)$, the vertex v does not belong to the set $H_{\tilde{e},a}^{(i(T))}(j,q)$ (i.e., it belongs to the set $H_{\tilde{e},a}^{(i'(T))}(j,q)$). Hence, for a vertex $z \in \chi_a^{(i(T))}(j)$, by construction (the mixing step), $\mathbb{P}^{mix}(z \; \mathcal{C} \; v) = 1/2^j$, independently of all the other events. In particular, it follows that for every vertex $z \in \mathcal{S}_a(j) \cap \chi_a^{(i(T))}(j)$, $\mathbb{P}^{mix}(z \; \mathcal{C} \; v) = 1/2^j$.

Similar analysis applies to all the vertices $z \in \mathcal{S}_a(j) \cup \bigcup_{b_\ell \in T} (\mathcal{S}_{b_\ell}(j) \cap \chi^{(i_\ell(T))}(j))$. Specifically, consider the set

$$\begin{aligned} Z_v \;=\; & (\mathcal{S} \setminus \bigcup_{x \in S} \mathcal{S}_x(j)) \cup \bigcup_{T \in \alpha} \left((\mathcal{S}_a(j) \cap \chi^{(i(T))}(j)) \cup \bigcup_{b \in T} \mathcal{S}_b(j) \right) \\ \cup \; & \bigcup_{T \in \beta} \left(\mathcal{S}_a(j) \cup \bigcup_{\ell=1}^{p(T)} (\mathcal{S}_{b_\ell}(j) \cap \chi_{b_\ell}^{(i_\ell(T))}(j)) \right) . \end{aligned} \tag{9}$$

Every vertex $z \in Z_v$ satisfies that $\mathbb{P}^{mix}(z \; \mathcal{C} \; v) = 1/2^j$ independently of all the other events. Note also that a vertex $z \in Z_v$ can only be connected to the vertex v by the mixing step.

To summarize,

Lemma 4.13 *For a super-edge $\tilde{e} = (A, B)$, a subset $\mathcal{S} = \chi_A \cup \chi_B$, an index $j \in \hat{J}$, and a vertex $v \in M_{\tilde{e}}(j)$, the set $Z_v = Z_v(\mathcal{S}, j)$ that is given by (9) satisfies*

1. $\mathbb{P}(|Z_v| \geq |\mathcal{S}|/(2+\epsilon)) \geq 1 - exp(-\Omega(|\mathcal{S}|))$.
2. *For every vertex $z \in Z_v$, $\mathbb{P}^{mix}(z \; \mathcal{C} \; v) = 1/2^j$, and $\mathbb{P}^{nmix}(z \; \mathcal{C} \; v) = 0$, where the notation $\mathbb{P}^{mix}$ (respectively, $\mathbb{P}^{nmix}$) refers to the sample space that is defined by the mixing step (resp., all the other steps) of the reduction.*

Recall that the notation $(\mathcal{S} \; (\leq 1\mathcal{C}) \; v)$ denotes the event that there is at most one edge (z, v) for some vertex $z \in \mathcal{S}$.

Note that the event $(Z_v(\mathcal{S}) \; (\leq 1\mathcal{C}) \; v)$ contains the event $(\mathcal{S} \; (\leq 1\mathcal{C}) \; v)$, and so, for any probability measure $\mathbb{P}$, $\mathbb{P}(\mathcal{S} \; (\leq 1\mathcal{C}) \; v) \leq \mathbb{P}(Z_v(\mathcal{S}) \; (\leq 1\mathcal{C}) \; v)$. Also,

$$\mathbb{P}(Z_v(\mathcal{S}) \; (\leq 1\mathcal{C}) \; v) = \mathbb{P}(Z_v(\mathcal{S}) \; \mathcal{AN} \; v) + \mathbb{P}(\exists! z \in Z_v(\mathcal{S}) \;\; \text{s.t.} \;\; (z \; \mathcal{C} \; v)) \, .$$

By Lemma 4.13, $\mathbb{P}^{mix}(Z_v(\mathcal{S})\ \mathcal{AN}\ v) = (1-1/2^j)^{|Z_v(\mathcal{S})|}$, and $\mathbb{P}^{mix}(\exists! z \in Z_v(\mathcal{S}) \text{ s.t. } (z\ \mathcal{C}\ v)) \leq |Z_v(\mathcal{S})| \cdot \frac{1}{2^j} \cdot (1-1/2^j)^{|Z_v(\mathcal{S})|-1}$. Hence,

$$\begin{aligned} \mathbb{P}^{mix}(Z_v(\mathcal{S})\ (\leq 1\mathcal{C})\ v) &\leq (1-1/2^j)^{|Z_v(\mathcal{S})|} + (1-1/2^j)^{|Z_v(\mathcal{S})|-1}\frac{|Z_v(\mathcal{S})|}{2^j} \\ &\leq (1-1/2^j)^{|Z_v(\mathcal{S})|-1}(1+\frac{|Z_v(\mathcal{S})|}{2^j})\ , \end{aligned} \tag{10}$$

and so,

$$\begin{aligned} \mathbb{P}^{mix}(Z_v(\mathcal{S})\ (\geq 2\mathcal{C})\ v) &\geq 1-(1-1/2^j)^{|Z_v(\mathcal{S})|} - (1-1/2^j)^{|Z_v(\mathcal{S})|-1}\frac{|Z_v(\mathcal{S})|}{2^j} \\ &\geq 1-(1-1/2^j)^{|Z_v(\mathcal{S})|-1}(1+\frac{|Z_v(\mathcal{S})|}{2^j})\ . \end{aligned} \tag{11}$$

Also, denoting by $\mathbb{P}$ the probability measure in the sample space that is defined by the entire reduction, observe that by Bayes formula,

$$\mathbb{P}(\mathcal{S}\ (\leq 1\mathcal{C})\ v)\ \leq\ \mathbb{P}(\mathcal{S}\ (\leq 1\mathcal{C})\ v \mid |Z_v(\mathcal{S})| \geq \frac{|Z_v(\mathcal{S})|}{2+\epsilon}) + \mathbb{P}(|Z_v(\mathcal{S})| < \frac{|\mathcal{S}|}{2+\epsilon})\ .$$

Note that by definition of the set $Z_v(\mathcal{S})$, $\mathbb{P}(|Z_v(\mathcal{S})| < \frac{|\mathcal{S}|}{2+\epsilon}) = \mathbb{P}^{\chi}(|Z_v(\mathcal{S})| < \frac{|\mathcal{S}|}{2+\epsilon})$, and, by Chernoff inequality, the latter probability is at most $exp(-\Omega(|\mathcal{S}|))$. Also,

$$\mathbb{P}((\mathcal{S}\ (\leq 1\mathcal{C})\ v) \mid |Z_v(\mathcal{S})| \geq \frac{|\mathcal{S}|}{2+\epsilon})\ \leq\ \mathbb{P}(Z_v(\mathcal{S})\ (\leq 1\mathcal{C})\ v) \mid |Z_v(\mathcal{S})| \geq \frac{|\mathcal{S}|}{2+\epsilon})\ .$$

By Lemma 4.13(2),

$$\mathbb{P}(Z_v(\mathcal{S})\ (\leq 1\mathcal{C})\ v) \mid |Z_v(\mathcal{S})| \geq \frac{|\mathcal{S}|}{2+\epsilon})\ =\ \mathbb{P}^{mix}(Z_v(\mathcal{S})\ (\leq 1\mathcal{C})\ v) \mid |Z_v(\mathcal{S})| \geq \frac{|\mathcal{S}|}{2+\epsilon})\ \leq$$
$$(1-1/2^j)^{\frac{|\mathcal{S}|}{2+\epsilon}} + (1-1/2^j)^{\frac{|\mathcal{S}|}{2+\epsilon}-1}\frac{|\mathcal{S}|}{(2+\epsilon)2^j}\ \leq\ (1-1/2^j)^{\frac{|\mathcal{S}|}{2+\epsilon}-1}(1+\frac{|\mathcal{S}|}{(2+\epsilon)2^j})\ .$$

(For the first inequality note that since the function $f(x) = (1+c\cdot x)\cdot a^{x-1}$ is monotonically decreasing with x whenever $0 < a < 1$, the right-hand side expression of inequality (10) is maximized by substituting $|Z_v(\mathcal{S})| = \frac{|\mathcal{S}|}{2+\epsilon}$.) Hence,

$$\mathbb{P}(\mathcal{S}\ (\geq 2\mathcal{C})\ v)\ \geq\ 1-(1-1/2^j)^{\frac{|\mathcal{S}|}{2+\epsilon}} - (1-1/2^j)^{\frac{|\mathcal{S}|}{2+\epsilon}-1}\frac{|\mathcal{S}|}{(2+\epsilon)2^j} - exp(-\Omega(|\mathcal{S}|))\ \geq$$
$$1-(1-\frac{1}{2^j})^{\frac{|\mathcal{S}|}{2+\epsilon}-1}\cdot(1+\frac{|\mathcal{S}|}{(2+\epsilon)2^j}) - exp(-\Omega(|\mathcal{S}|))\ . \tag{12}$$

Also, analogously, by inequality (11),

$$\mathbb{P}(Z_v(\mathcal{S})\ (\geq 2\mathcal{C})\ v)\ \geq\ 1-(1-1/2^j)^{\frac{|\mathcal{S}|}{2+\epsilon}} - (1-1/2^j)^{\frac{|\mathcal{S}|}{2+\epsilon}-1}\frac{|\mathcal{S}|}{(2+\epsilon)2^j} - exp(-\Omega(|\mathcal{S}|))\ \geq$$
$$1-(1-\frac{1}{2^j})^{\frac{|\mathcal{S}|}{2+\epsilon}-1}\cdot(1+\frac{|\mathcal{S}|}{(2+\epsilon)2^j}) - exp(-\Omega(|\mathcal{S}|))\ . \tag{13}$$

4.4 Large Rounds and their Effect

In this section we consider the probability space that is determined by the random permutation and mixing steps (steps 2d - 2e) of the reduction, and, in particular, the partitions of the sets $\chi_x(j)$ are fixed for all the vertices $x \in A \cup B$ (and the index $j \in \hat{J}$ is fixed as well).

Definition 4.14 *We say that a round R is j-*large *with respect to a super-edge $\tilde{e} = (A, B)$ if $R_{\tilde{e}} = R \cap (\chi_A \cup \chi_B)$ is of size greater or equal to $c \cdot 2^{j+1} \ln n$, where $c = 2c_0 + 2$ (c_0 is a universal constant mentioned in Section 3; it will be determined later). Otherwise, the round is called j-*small *with respect to the super-edge $\tilde{e}$.*

A schedule Π is called short *if its length is at most $\log^2 n/100$.*

Consider a short schedule Π. Let $\pi_{\tilde{e}}(j)$ (respectively, $\Pi_{\tilde{e}}(j)$) be the subschedule of Π that contains only j-small (resp, j-large) rounds with respect to the super-edge $\tilde{e}$. Note that a vertex $v \in M_{\tilde{e}}(j)$ is informed by the schedule Π if and only if either the subschedule $\pi_{\tilde{e}}(j)$ or the subschedule $\Pi_{\tilde{e}}(j)$ informs v (or if both do).

Let $V_{\tilde{e}}(\pi_{\tilde{e}}(j)) = \bigcup_{R \in \pi_{\tilde{e}}(j)} R \cap (\chi_A \cup \chi_B) = \bigcup_{R \in \pi_{\tilde{e}}(j)} R_{\tilde{e}}$, and, analogously, $V_{\tilde{e}}(\Pi_{\tilde{e}}(j)) = \bigcup_{R \in \Pi_{\tilde{e}}(j)} R_{\tilde{e}}$.

Next, we show that the probability that the subschedule $\Pi_{\tilde{e}}(j)$ informs a significant number of vertices is negligibly small.

For a fixed super-edge $\tilde{e} = (A, B) \in \tilde{E}$, and a fixed index $j \in \hat{J}$, consider a subset $M' \subseteq M_{\tilde{e}}(j)$ such that for every index $q \in [L]$, $|M' \cap M_{\tilde{e}}(j, q)| \le 1$. Such subsets are particularly convenient for the analysis, since for a pair of distinct indices $q_1, q_2 \in [L]$, and a pair vertices $v_1 \in M_{\tilde{e}}(j, q_1)$, $v_2 \in M_{\tilde{e}}(j, q_2)$, the events $(x\ \mathcal{C}\ v_1)$ and $(x\ \mathcal{C}\ v_2)$ are independent.

Lemma 4.15 *For a fixed short schedule Π, with probability at least $1 - \frac{1}{2^{\Omega(n \cdot \log n)}}$, there is no subset $M' \subseteq M_{\tilde{e}}(j)$ of size $N = n$ that satisfies $|M_{\tilde{e}}(j, q) \cap M'| \le 1$ for every index $q \in [L]$, such that the subschedule $\Pi_{\tilde{e}}(j)$ informs all the vertices of the subset M'.*

Proof: Consider a fixed subset M' with these properties, and a fixed round $R_{\tilde{e}} = R \in \Pi_{\tilde{e}}(j)$. By definition, $|R_{\tilde{e}}| = |R \cap (\chi_A \cup \chi_B)| \ge c \cdot 2^{j+1} \ln n$. By inequality (12), the probability that the round R does not (≥ 2)-covers the vertex $v \in M'$ is at most

$$(1 - \frac{1}{2^j})^{\frac{|R_{\tilde{e}}|}{2+\epsilon}} + \frac{|R_{\tilde{e}}|}{2^j (2+\epsilon)} \cdot (1 - \frac{1}{2^j})^{\frac{|R_{\tilde{e}}|}{2+\epsilon} - 1} + exp(-\Omega(|R_{\tilde{e}}|)) \ .$$

Since $c \cdot 2^{j+1} \ln n \le |R_{\tilde{e}}| = |R \cap (\chi_A \cup \chi_B)| \le |\chi_A \cup \chi_B| = n$, it follows that this probability is at most

$$(1 - \frac{1}{2^j})^{\frac{c \cdot 2^{j+1} \ln n}{2+\epsilon}} + \frac{n}{(2+\epsilon) n^{0.4}} (1 - \frac{1}{2^j})^{\frac{c \cdot 2^{j+1} \ln n}{2+\epsilon} - 1} + exp(-\Omega(2^{j+1} \ln n)) \ \le$$

$$e^{-\frac{2c}{2+\epsilon}\cdot\ln n} + 3\cdot n^{0.6} e^{-\frac{2c}{2+\epsilon}\cdot\ln n - 1} + exp(-\Omega(n^{0.4})) \leq O(\frac{1}{n^{\frac{2c}{2+\epsilon}-0.6}}) + exp(-\Omega(n^{0.4})) .$$

So, altogether, for a fixed round $R \in \Pi_{\tilde{e}}(j)$ and vertex $v \in M'$, $\mathbb{P}(R$ does not (≥ 2)-cover $v) = O(\frac{1}{n^{\frac{2c}{2+\epsilon}-0.6}})$. Since $|\Pi_{\tilde{e}}(j)| = O(\log^2 n)$, it follows that

$$\mathbb{P}(\exists R \in \Pi_{\tilde{e}}(j) \text{ s.t. } R \text{ does not } (\geq 2)\text{-cover } v) = O\left(\frac{\log^2 n}{n^{\frac{2c}{2+\epsilon}-0.6}}\right) .$$

Observe also that

$$\begin{aligned}
\mathbb{P}(\Pi_{\tilde{e}}(j)\ \mathcal{INF}\ v) &= \mathbb{P}(\exists R \in \Pi_{\tilde{e}}(j) \text{ s.t. } (R\ \mathcal{INF}\ v)) \\
&\leq \mathbb{P}(\exists\ R \in \Pi_{\tilde{e}}(j) \text{ s.t. } R \text{ does not } (\geq 2)\text{-cover } v) \leq O\left(\frac{\log^2 n}{n^{\frac{2c}{2+\epsilon}-0.6}}\right) .
\end{aligned}$$

Since, by the assumption of the lemma, for any two vertices $v, v' \in M'$, $v \in M_{\tilde{e}}(j,q)$, $v' \in M_{\tilde{e}}(j,q')$, necessarily $q \neq q'$, it follows that the events $(\Pi_{\tilde{e}}(j)\ \mathcal{INF}\ v)$ and $(\Pi_{\tilde{e}}(j)\ \mathcal{INF}\ v')$ are independent. Hence, setting $\epsilon < 1/(10c)$ implies that

$$\mathbb{P}(\Pi_{\tilde{e}}(j)\ \mathcal{INF}\ M') \leq \left(O\left(\frac{\log^2 n}{n^{(2c/(2+\epsilon))-0.6}}\right)\right)^{|M'|} \leq \frac{1}{n^{n(c-1)}} .$$

This proves that the probability that the subschedule $\Pi_{\tilde{e}}(j)$ informs a fixed subset $M' \subseteq M_{\tilde{e}}(j,q)$ is negligibly small, whenever the subset M' satisfies the aforementioned conditions.

Recall that $|M_{\tilde{e}}(j,q)| = |\bigcup_{q=1}^{L} M_{\tilde{e}}(j)| = L \cdot M = 2 \cdot n^{2c_0}$. The number of ways to choose a subset of cardinality n out of a set of cardinality $2 \cdot n^{2c_0}$ is at most $(2 \cdot n^{2c_0})^n = 2^{2c_0 \cdot n \cdot \log n (1+o(1))}$. Hence, by union bound,

$$\begin{aligned}
&\mathbb{P}(\exists M' \subseteq M_{\tilde{e}}(j) \text{ with } |M'| = n \text{ and } \forall q \in [L], |M' \cap M_{\tilde{e}}(j,q)| \leq 1 \text{ s.t. } (\Pi_{\tilde{e}}(j)\ \mathcal{INF}\ M')) \\
&\leq \frac{2^{2c_0 \cdot n \cdot \log n (1+o(1))}}{2^{(c-1) n \cdot \log n}} = \frac{1}{2^{n \log n (c-1-2c_0(1+o(1)))}} .
\end{aligned}$$

Recall that c was set to $2c_0 + 2$, and so the above probability is at most $\frac{1}{2^{c' n \cdot \log n}}$, where $c' = c - 1 - 2c_0(1+o(1)) = 1 - o(1) > 0$ is some universal positive constant. ∎

4.5 Small Rounds: Generic Analysis

We start with describing a combinatorial lemma from [ABLP89]. We will use this lemma in the rest of our analysis.

Lemma 4.16 *[ABLP89] Let V_1 be a subset of n elements, and let $\Pi = (R_1, R_2, \ldots, R_t)$ be a collection of at most $\frac{\log^2 n}{100}$ subsets of V_1. Then there exists a subset $\mathcal{S} \subseteq V_1$, and an index $j \in \hat{J}$, such that*

1. $|\mathcal{S}| \leq 2^j \cdot \log n$.

2. *Let* $\Pi' = (R'_1, R'_2, \ldots) = \Pi \setminus \mathcal{S}$. *Then for every subset* $R' \in \Pi'$, $|R'| \geq 2^j$.

3. *Let* f_k *be the number of subsets* R' *with cardinality* $2^{j+k} \leq |R'| < 2^{j+k+1}$. *Then*

$$\sum_{k \geq 0} \frac{f_k}{2^k} \leq \log n \ .$$

Fix a super-edge $\tilde{e} = (A, B)$, and consider the subset $V_1 = \chi_A \cup \chi_B$ of vertices. Let $\Pi|_{\tilde{e}}$ be the schedule Π restricted to the set $\chi_A \cup \chi_B$. Apply Lemma 4.16 to this subset and the collection $\Pi|_{\tilde{e}}$ of subsets of the set $\chi_A \cup \chi_B$, and let $\mathcal{S}$ and j be the subset and the index whose existence are guaranteed by the lemma. (Observe that $|\chi_A \cup \chi_B| = n^{0.6} \cdot |A \cup B| = n$.)

Definition 4.17 1. *The index* j *as above is called the* pivot *of the super-edge* $\tilde{e}$ *with respect to the schedule* Π.

2. *The schedule* Π *is called* $\tilde{e}$-proper *if* $V_{\tilde{e}}(\pi_{\tilde{e}}(j))$ *is an* $(\tilde{e}, j)$*-partial set, where the index* $j \in \hat{J}$ *is the pivot of the super-edge* $\tilde{e}$ *with respect to* Π.

Consider an $\tilde{e}$-proper schedule Π, and the subschedule $\pi_{\tilde{e}}(j)$ for the pivot j of the super-edge $\tilde{e}$. Let v be some fixed vertex in $M_{\tilde{e}}(j)$, and let $\mathcal{S}' = \mathcal{S} \cap V_{\tilde{e}}(\pi_{\tilde{e}}(j))$.

Lemma 4.18 $\mathbb{P}(\mathcal{S}' \ \mathcal{AN} \ v) \ \geq \ \frac{1}{n^{16}}$.

Proof: Note that $\mathcal{S}' \subseteq V_{\tilde{e}}(\pi_{\tilde{e}}(j))$, and Π is an $\tilde{e}$-proper schedule, and so $V_{\tilde{e}}(\pi_{\tilde{e}}(j))$ is an $(\tilde{e}, j)$-partial set. Hence, the subset $\mathcal{S}' \subseteq V_{\tilde{e}}(\pi_{\tilde{e}}(j))$ is an $(\tilde{e}, j)$-partial set as well. Also, note that $|\mathcal{S}'| \leq |\mathcal{S}| \leq 2^j \log n$ (by the choice of the subset $\mathcal{S}$, see Lemma 4.16(1). Now the assertion of the lemma follows from Corollary 4.10. ∎

Next, we analyze the probability that for a fixed vertex $v \in M_{\tilde{e}}(j)$, for every round $R \in \pi_{\tilde{e}}(j)$, the set $R' = R \setminus \mathcal{S}'$ (≥ 2)-covers v.

Suppose we have a collection of subsets $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_t\}$ of the set $\chi_A \cup \chi_B$ of size at least 2^j each. For a fixed vertex v, for each index $\ell \in [t]$, by Lemma 4.13, $\mathbb{P}^{\chi}(|Z_v(\mathcal{S}_\ell)| \geq \frac{|\mathcal{S}_\ell|}{2+\epsilon}) \geq 1 - exp(-\Omega(|\mathcal{S}_\ell|))$. Hence, by union bound,

$$\mathbb{P}^{\chi}\left(\forall \ell \in [t], |Z_v(\mathcal{S}_\ell)| \geq \frac{|\mathcal{S}_\ell|}{2+\epsilon}\right) \ \geq \ 1 - \sum_{\ell=1}^{t} exp(-\Omega(|\mathcal{S}_\ell|)) \ .$$

Suppose that $t = O(\log^2 n)$, and that for every $\ell \in [t]$, $|\mathcal{S}_\ell| \geq n^{0.4}$. Then

$$\mathbb{P}^{\chi}\left(\forall \ell \in [t], |Z_v(\mathcal{S}_\ell)| \geq \frac{|\mathcal{S}_\ell|}{2+\epsilon}\right) \ \geq \ 1 - exp(-\Omega(n^{0.4})) \ . \tag{14}$$

Note that all the steps of the reduction other than the binomial partition step have no effect on the cardinality of the set $Z_v(\mathcal{S})$, and, therefore, inequality (14) holds also when replacing the probability measure $\mathbb{P}^{\chi}$ by the probability measure $\mathbb{P}$ (that refers to the sample space that is defined by the entire reduction).

Consider the probability $\mathbb{P}(\exists \ell \in [t], (\mathcal{S}_\ell\ (\leq 1\mathcal{C})\ v))$. By Bayes formula,

$$\begin{aligned}
&\mathbb{P}(\exists \ell \in [t], (\mathcal{S}_\ell\ (\leq 1\mathcal{C})\ v)) \ = \\
&\mathbb{P}(\exists \ell \in [t], (\mathcal{S}_\ell\ (\leq 1\mathcal{C})\ v) \mid \exists \ell \in [t], |Z_v(\mathcal{S}_\ell)| < \frac{|\mathcal{S}_\ell|}{2+\epsilon}) \cdot \mathbb{P}(\exists \ell \in [t], |Z_v(\mathcal{S}_\ell)| < \frac{|\mathcal{S}_\ell|}{2+\epsilon}) \\
&\quad + \ \mathbb{P}(\exists \ell \in [t], (\mathcal{S}_\ell\ (\leq 1\mathcal{C})\ v) \mid \forall \ell \in [t], |Z_v(\mathcal{S}_\ell)| \geq \frac{|\mathcal{S}_\ell|}{2+\epsilon}) \cdot \mathbb{P}(\forall \ell \in [t], |Z_v(\mathcal{S}_\ell)| \geq \frac{|\mathcal{S}_\ell|}{2+\epsilon}) \ .
\end{aligned}$$

By inequality (14), the first term is at most $exp(-\Omega(n^{0.4}))$. The second term is at most $\mathbb{P}(\exists \ell \in [t], (\mathcal{S}_\ell\ (\leq 1\mathcal{C})\ v) \mid \forall \ell \in [t], |Z_v(\mathcal{S}_\ell)| \geq \frac{|\mathcal{S}_\ell|}{2+\epsilon})$. We next analyze this probability. (Note that the events $(\mathcal{S}_\ell\ (\leq 1\mathcal{C})\ v)$ and $(\mathcal{S}_k\ (\leq 1\mathcal{C})\ v)$ may be dependent, as $\mathcal{S}_\ell, \mathcal{S}_k \subseteq \chi_A \cup \chi_B$ are arbitrary subsets that are allowed to intersect.) Recall that by inequality (13),

$$\begin{aligned}
&\mathbb{P}(Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v \mid \forall h \in [t], |Z_v(\mathcal{S}_h)| \geq \frac{|\mathcal{S}_h|}{2+\epsilon}) \ \geq \\
&1 - (1 - \frac{1}{2^j})^{\frac{|\mathcal{S}_\ell|}{2+\epsilon} - 1}(1 + \frac{|\mathcal{S}_\ell|}{(2+\epsilon)2^j}) - exp(-\Omega(n^{0.4})) \ .
\end{aligned} \tag{15}$$

(Strictly speaking, inequality (13) implies that $\mathbb{P}(Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v)$ is greater or equal than the right-hand side expression of (15). However, it is easy to see that precisely the same considerations that were used to establish inequality (15) apply here.)

Definition 4.19 *For a random function that returns a set, a collection of values of this random function is an event in the given probability space. Each value in this collection is a set, and thus, an event is a collection of sets. Such a collection is* monotonically increasing *if whenever it contains a set T, it contains all the supersets of T. An event that corresponds to a collection of sets is* monotonically increasing*, whenever the corresponding collection of sets is monotonically increasing.*

Note the collection of all sets T that satisfy $(T\ (\geq 2\mathcal{C})\ v)$ (for some fixed vertex $v \in M_{\tilde{e}}(j)$) is a monotonically increasing collection. Since for every index $\ell \in [t]$, $Z_v(\mathcal{S}_\ell)$ is a random function that returns a set, the event $(Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v)$ is monotonically increasing. The same is true for the events $(Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v) \mid \forall h \in [t], |Z_v(\mathcal{S}_h)| \geq \frac{|\mathcal{S}_h|}{2+\epsilon})$. (For events, the notation $(A \mid B)$ is equivalent to the notation $(A \cap B)$.)

Recall also that every vertex $s \in \bigcup_{\ell=1}^{t} Z_v(\mathcal{S}_\ell)$ is connected to the vertex v with probability $1/2^j$, independently of other vertices, and therefore, the FKG inequality is applicable (see, e.g., [AS92], ch. 7). Hence,

$$\begin{aligned}
&\mathbb{P}(\forall \ell \in [t], (Z_v(\mathcal{S}_\ell) \ (\geq 2\mathcal{C}) \ v) \mid \forall h \in [t], |Z_v(\mathcal{S}_h)| \geq \frac{|\mathcal{S}_h|}{2+\epsilon}) \ \geq \\
&\Pi_{\ell=1}^{t} \mathbb{P}((Z_v(\mathcal{S}_\ell) \ (\geq 2\mathcal{C}) \ v) \mid \forall h \in [t], |Z_v(\mathcal{S}_h)| \geq \frac{|\mathcal{S}_h|}{2+\epsilon}) \ \geq \\
&\Pi_{\ell=1}^{t}(1 - (1 + \frac{|\mathcal{S}_\ell|}{2^j(2+\epsilon)}) \cdot (1 - 1/2^j)^{\frac{|\mathcal{S}_\ell|}{2+\epsilon} - 1}) \ .
\end{aligned}$$

We next group the indices $\ell \in [t]$ by the sizes of the sets $\mathcal{S}_\ell$. For each index $k = 0, 1, 2, \ldots$, there are f_k indices ℓ such that $2^{j+k} \leq |\mathcal{S}_\ell| \leq 2^{j+k+1}$. The function $g(x) = (x \cdot b + 1)a^{\frac{x-1}{2+\epsilon}}$ with $0 < a < 1$ is monotonically decreasing with x for sufficiently large x, and in our case $|\mathcal{S}_\ell| \geq 2^j \geq n^{0.4}$ are all sufficiently large. Hence,

$$\begin{aligned}
&(1 - \frac{1}{2^j})^{\frac{|\mathcal{S}_\ell|}{2+\epsilon} - 1} \cdot (1 + \frac{|\mathcal{S}_\ell|}{(2+\epsilon)2^j}) \ \leq \ (1 + o(1)) \cdot (1 - \frac{1}{2^j})^{\frac{2^{j+k}}{2+\epsilon}} \cdot (1 + \frac{2^k}{2+\epsilon}) \ \leq \\
&(1 + o(1)) \cdot e^{-\frac{2^k}{2+\epsilon}}(1 + \frac{2^k}{2+\epsilon}) \ .
\end{aligned}$$

Hence,

$$\Pi_{\ell=1}^{t}(1 - (1 - 1/2^j)^{\frac{|\mathcal{S}_\ell|}{2+\epsilon} - 1} \cdot (1 + \frac{|\mathcal{S}_\ell|}{(2+\epsilon)2^j})) \ \geq \ \Pi_{k \geq 0}(1 - (1 + o(1))e^{-2^k/(2+\epsilon)} \cdot (1 + \frac{2^k}{2+\epsilon}))^{f_k} \ .$$

For a sufficiently large k (i.e., there exists K such that for every $k \geq K$), $(1+o(1))e^{-2^k/(2+\epsilon)} < \frac{3}{2 \cdot 2^k}$. Note also that for every positive k, $(1 + o(1))e^{-2^k/(2+\epsilon)}(1 + \frac{2^k}{2+\epsilon}) < 1$.

Recall that $\sum_{k \geq 0} f_k/2^k \leq \log n$. Hence, $\sum_{k \geq 0}^{K-1} f_k/2^k, \sum_{k \geq K} f_k/2^k \leq \log n$. It follows that $\sum_{k \geq 0}^{K-1} \leq 2^{K-1} \cdot \log n$. Thus,

$$\Pi_{k \geq 0}(1 - (1 + o(1)) \cdot exp(-\frac{2^k}{2+\epsilon}) \cdot (1 + \frac{2^k}{2+\epsilon}))^{f_k} \ \geq \ \Pi_{k \geq 0}^{K-1} \rho^{f_k} \cdot \Pi_{k \geq K}(1 - \frac{3}{2 \cdot 2^k})^{f_k} \ ,$$

where $\rho = \min\{1 - (1 + o(1)) \cdot e^{-\frac{2^k}{2+\epsilon}} \cdot (1 + \frac{2^k}{2+\epsilon}) \mid k = 0, 1, 2, \ldots, K-1\} > 0$ is some universal constant (since it is the minimum of a finite number of elements).

For sufficiently small real $x > 0$, $1 - x \geq e^{-2x}$. Hence,

$$\begin{aligned}
&\Pi_{k \geq K}(1 - \frac{3}{2 \cdot 2^k})^{f_k} \ \geq \ \Pi_{k \geq K} exp(-3/2^k)^{f_k} \ = \ exp(-3 \sum_{k \geq K} f_k/2^k) \ \geq \\
&exp(-3 \log n) \ = \ 1/n^{3 \log e} \ .
\end{aligned}$$

Also,

$$\Pi_{k \geq 0}^{K-1} \rho^{f_k} \ = \ \rho^{\sum_{k=0}^{K-1} f_k} \ \geq \ \rho^{2^{K-1} \cdot \log n} \ = \ \frac{1}{n^{\log(1/\rho) \cdot 2^{K-1}}} \ .$$

Hence,

$$\mathbb{P}(\forall \ell \in [t], (Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v) \mid \forall h \in [t], |Z_v(\mathcal{S}_h)| \geq \frac{|\mathcal{S}_h|}{2+\epsilon}) \ \geq \ \frac{1}{n^{(\log(1/\rho)\cdot 2^{K-1}+3\cdot\log e)}} \ .$$

Hence,

$$\begin{aligned}
&\mathbb{P}(\forall \ell \in [t], (Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v)) \ = \\
&\mathbb{P}(\forall \ell \in [t], (Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v) \mid \forall h \in [t], |Z_v(\mathcal{S}_h)| \geq \frac{|\mathcal{S}_h|}{2+\epsilon}) \cdot \mathbb{P}(\forall h \in [t], |Z_v(\mathcal{S}_h)| \geq \frac{|\mathcal{S}_h|}{2+\epsilon}) \ + \\
&\mathbb{P}(\forall \ell \in [t], (Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v) \mid \exists h \in [t] \ \text{ s.t. } \ |Z_v(\mathcal{S}_h)| < \frac{|\mathcal{S}_h|}{2+\epsilon}) \cdot \\
&\mathbb{P}(\exists h \in [t] \ \text{ s.t. } \ |Z_v(\mathcal{S}_h)| < \frac{|\mathcal{S}_h|}{2+\epsilon}) \ \geq \\
&\mathbb{P}(\forall \ell \in [t], (Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v) \mid \forall h \in [t], |Z_v(\mathcal{S}_h)| \geq \frac{|\mathcal{S}_h|}{2+\epsilon}) \cdot (1 - exp(-\Omega(n^{0.4}))) \ \geq \\
&\frac{1}{n^{(\log(1/\rho)\cdot 2^{K-1}+3\cdot\log e)}} \cdot (1 - exp(-\Omega(n^{0.4}))) \ \geq \ \frac{1}{n^{c_1}} \ ,
\end{aligned}$$

where c_1 is an appropriate universal constant. I.e.,

$$c_1 = \log(1/\rho) \cdot 2^{K-1} + 3 \cdot \log e + 1 \ . \tag{16}$$

Hence,

$$\mathbb{P}(\forall \ell \in [t], (\mathcal{S}_\ell\ (\geq 2\mathcal{C})\ v)) \ \geq \ \mathbb{P}(\forall \ell \in [t], (Z_v(\mathcal{S}_\ell)\ (\geq 2\mathcal{C})\ v)) \ \geq \ \frac{1}{n^{c_1}} \ .$$

4.6 Small Rounds: Specific Analysis

Consider a schedule $\Pi = (R_1, R_2, \ldots, R_t)$, $t \leq (\log^2 n)/100$, and a super-edge $\tilde{e} = (A, B)$. Let $\Pi|_{\tilde{e}} = (R_1|_{\tilde{e}}, R_2|_{\tilde{e}}, \ldots, R_t|_{\tilde{e}})$, $R_\ell|_{\tilde{e}} = R_\ell \cap (\chi_A \cup \chi_B)$, be the subschedule of Π restricted to the super-edge $\tilde{e}$ (for simplicity of notation we assume that for every index $\ell \in [t]$, the set $R_\ell|_{\tilde{e}}$ is not empty; otherwise the number of rounds in the restricted schedule would be smaller than in the original one). Now, apply Lemma 4.16 to the schedule $\Pi|_{\tilde{e}} = (R_1|_{\tilde{e}}, R_2|_{\tilde{e}}, \ldots, R_t|_{\tilde{e}})$. Observe that for every index $\ell \in [t]$, $R_\ell|_{\tilde{e}} \subseteq \chi_A \cup \chi_B$, and that $|\chi_A \cup \chi_B| = |A \cup B| \cdot n^{0.6} = n$. By Lemma 4.16, there exists a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$, and an index $j \in \hat{J}$ such that

1. $|\mathcal{S}| \leq 2^j \log n$.

2. Let $\Pi' = (R'_1, R'_2, \ldots, R'_t)$, and for every index $\ell \in [t]$, $R'_\ell = R_\ell|_{\tilde{e}} \setminus \mathcal{S}$ (again, for simplicity of notation we assume that for every index $\ell \in [t]$, the set R'_ℓ is not empty). Then for every index $\ell \in [t]$, $|R'_\ell| \geq 2^j$.

3. Let $f_k = |\{R'_\ell \mid 2^{j+k} \leq |R'_\ell| < 2^{j+k+1}\}|$. Then $\sum_{k\geq 0} f_k/2^k \leq \log n$.

Now, when we are already equipped with the index j, consider the subschedules $\Pi_{\tilde{e}}(j)$ and $\pi_{\tilde{e}}(j)$ of the schedule Π (defined in the beginning of Section 4.4). By Lemma 4.15, with probability at least $1 - \frac{1}{2^{\Omega(n\cdot\log n)}}$ there is no subset $M' \subseteq M_{\tilde{e}}(j)$ of size $|M'| = n$ that satisfies $|M' \cap M_{\tilde{e}}(j,q)| \leq 1$ for every index $q \in [L]$ and such that the subschedule $\Pi_{\tilde{e}}(j)$ informs all the vertices of the set M'.

Recall that the index j as above is called the *pivot* of the super-edge $\tilde{e}$ with respect to the schedule Π. Suppose that Π is an $\tilde{e}$-proper schedule (see Definition 4.17(2), i.e., the set $V_{\tilde{e}}(\pi_{\tilde{e}}(j))$ is $(\tilde{e},j)$-partial. Consider the subset $\mathcal{S}' \subseteq \mathcal{S}$ given by $\mathcal{S}' = \mathcal{S} \cap V_{\tilde{e}}(\pi_{\tilde{e}}(j))$. Since $|\mathcal{S}'| \leq |\mathcal{S}| \leq 2^j \log n$, by Lemma 4.18, for each fixed vertex $v \in M_{\tilde{e}}(j)$, $\mathbb{P}(\mathcal{S}' \ \mathcal{AN} \ v) \geq \frac{1}{n^{16}}$.

Now consider the subschedule $\pi|_{\tilde{e}}$ of the schedule $\Pi|_{\tilde{e}} = (R_1|_{\tilde{e}}, R_2|_{\tilde{e}}, \ldots, R_t|_{\tilde{e}})$ that contains only rounds that are j-small with respect to the super-edge $\tilde{e}$. In other words, let $\pi|_{\tilde{e}} = (\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_{t'})$, such that for every index $\ell \in [t']$, there exists an index $i \in [t]$ such that $\mathcal{S}_\ell = R_i|_{\tilde{e}}$; note that the subschedule $\pi|_{\tilde{e}}$ is different from the subschedule $\pi_{\tilde{e}}(j)$ because in $\pi_{\tilde{e}}(j)$ the rounds R appear precisely in the same form as they appear in the original schedule Π, while in $\pi|_{\tilde{e}}$ they are intersected with the set $\chi_A \cup \chi_B$. The precise relation between these two subschedules is given by $\pi|_{\tilde{e}} = \pi_{\tilde{e}}(j)|_{\tilde{e}}$.

Note that by definition of j-small rounds, for every index $\ell \in [t']$, $\mathcal{S}_\ell \cap (\chi_A \cup \chi_B)| = |\mathcal{S}_\ell| \leq 2^{j+1} \cdot c \cdot \ln n$. Observe that by property 2 of the set $\mathcal{S}$ and the index $j \in \hat{J}$, for every index $\ell \in [t']$, $|\mathcal{S}'_\ell| = |\mathcal{S}_\ell \setminus \mathcal{S}| = |R_i|_{\tilde{e}} \setminus \mathcal{S}| \geq 2^j$, where $i \in [t]$ is the index such that $\mathcal{S}_\ell = R_i|_{\tilde{e}}$.

Now, following the considerations of Section 4.5, it follows that

$$\mathbb{P}(\forall \ell \in [t'], \mathcal{S}'_\ell \ (\geq 2\mathcal{C}) \ v) \ \geq \ \frac{1}{n^{c_1}} \ .$$

In other words, for every super-edge $\tilde{e} = (A,B)$, for every $\tilde{e}$-proper schedule Π of length $O(\log^2 n)$, there exists a pivot index $j \in \hat{J}$ and a subset $\mathcal{S} \subseteq \chi_A \cup \chi_B$ such that

1. The subschedule $\Pi_{\tilde{e}}(j)$ "hardly informs anything" (in the sense of Lemma 4.15).

2. For each fixed vertex $v \in M_{\tilde{e}}(j)$, and the set $\mathcal{S}' = \mathcal{S} \cap V_{\tilde{e}}(\pi_{\tilde{e}}(j))$, $\mathbb{P}(\mathcal{S}' \ \mathcal{AN} \ v) \geq 1/n^{16}$.

3. For each fixed vertex $v \in M_{\tilde{e}}(j)$, with probability at least $1/n^{c_1}$, for every round $R \in \pi_{\tilde{e}}(j)$, the subset $R|_{\tilde{e}} \setminus \mathcal{S}' = R|_{\tilde{e}} \setminus \mathcal{S}$ satisfies $((R|_{\tilde{e}} \setminus \mathcal{S}') \ (\geq 2\mathcal{C}) \ v)$.

Note that the analysis of the probability $\mathbb{P}(\forall R \in \pi_{\tilde{e}}(j), ((R|_{\tilde{e}} \setminus \mathcal{S}') \ (\geq 2\mathcal{C}) \ v))$, for $\mathcal{S}' = \mathcal{S} \cap V_{\tilde{e}}(\pi_{\tilde{e}}(j))$, can be carried through without any change for the conditional probability

$\mathbb{P}(\forall R \in \pi_{\tilde{e}}(j), ((R|_{\tilde{e}} \setminus \mathcal{S}')\ (\geq 2\mathcal{C})\ v) \mid (\mathcal{S}'\ \mathcal{AN}\ v))$. To see it, recall that the analysis considers the sets $Z_v(R|_{\tilde{e}} \setminus \mathcal{S})$, and all the vertices $z \in Z_v(R|_{\tilde{e}} \setminus \mathcal{S})$ are connected to v with probability $1/2^j$ independently of all the other events. The set $\mathcal{S}'$ intersects with no set $Z_v(R|_{\tilde{e}} \setminus \mathcal{S}')$, proving this claim.

Hence,

$$\begin{aligned}&\mathbb{P}(\forall R \in \pi_{\tilde{e}}(j), ((R|_{\tilde{e}} \setminus \mathcal{S}')\ (\geq 2\mathcal{C})\ v), \text{ and } (\mathcal{S}'\ \mathcal{AN}\ v)) \ = \\ &\mathbb{P}(\forall R \in \pi_{\tilde{e}}(j), ((R|_{\tilde{e}} \setminus \mathcal{S}')\ (\geq 2\mathcal{C})\ v) \mid (\mathcal{S}'\ \mathcal{AN}\ v)) \cdot \mathbb{P}((\mathcal{S}'\ \mathcal{AN}\ v)) \ \geq \ \frac{1}{n^{c_1+16}}\ .\end{aligned}$$

Hence,

$$\mathbb{P}(\pi_{\tilde{e}}(j)\ \mathcal{NINF}\ v) \ \geq \ \mathbb{P}(\forall R \in \pi_{\tilde{e}}(j),\ (R|_{\tilde{e}} \setminus \mathcal{S}')\ (\geq 2\mathcal{C})\ v), (\mathcal{S}'\ \mathcal{AN}\ v)) \ \geq \ \frac{1}{n^{c_1+16}}\ . \quad (17)$$

Set

$$c_0 = c_1 + 17 \quad (18)$$

(recall that $L = 2n^{c_0}$, and $M = n^{c_0}$).

Lemma 4.20 *For a fixed super-edge $\tilde{e} \in \tilde{E}$, and a fixed $\tilde{e}$-proper schedule Π, consider the L sets $M_{\tilde{e}}(j,q)$, $q \in [L]$. For each index $q \in [L]$, let X_q denote the indicator random variable of the event $(\pi_{\tilde{e}}(j)\ \mathcal{INF}\ M_{\tilde{e}}(j,q))$. Let $X = \sum_{q=1}^{L} X_q$. Then $\mathbb{P}(X \leq 2n^{c_0} - n) \geq 1 - exp(-\Omega(n^{c_0}))$.*

Proof: Consider a fixed index $q \in [L]$. Let $Y = Y(q)$ denote the random variable that counts the number of vertices of the set $M_{\tilde{e}}(j,q)$ that are informed by the schedule $\pi_{\tilde{e}}(j)$. Let Y_v denote the indicator random variable of the event $(\pi_{\tilde{e}}(j)\ \mathcal{INF}\ v)$. Then $Y = \sum_{v \in M_{\tilde{e}}(j,q)} Y_v$. By linearity of expectation, $\mathbb{E}(Y) = \sum_{v \in M_{\tilde{e}}(j,q)} \mathbb{E}(Y_v)$. Note that by inequality (17), $\mathbb{E}(Y_v) = \mathbb{P}(Y_v = 1) \leq 1 - \frac{1}{n^{c_1+16}}$. Hence, $\mathbb{E}(Y) = \sum_{v \in M_{\tilde{e}}(j,q)} \mathbb{E}(Y_v) \ \leq \ M(1 - \frac{1}{n^{c_1+16}}) = n^{c_0} - n$. By Markov inequality,

$$\mathbb{P}(Y(q) \geq n^{c_0}) \ = \ \mathbb{P}(Y(q) \geq \frac{n^{c_0}}{\mathbb{E}(Y(q))} \cdot \mathbb{E}(Y(q))) \ \leq \ \frac{\mathbb{E}(Y(q))}{n^{c_0}} \ \leq \ 1 - \frac{1}{n^{c_0-1}}\ .$$

Note that $\mathbb{P}(X_q = 1) = \mathbb{P}(Y(q) \geq M = n^{c_0})$, and so $\mathbb{P}(X_q = 1) \leq 1 - \frac{1}{n^{c_0-1}}$.

It is rather easy to see that the proof of inequality (17) carries through also when the probabilities of the events of the form $(\pi_{\tilde{e}}(j)\ \mathcal{NINF}\ v)$ for a vertex $v \in M_{\tilde{e}}(j,q)$ are conditioned on a collection of the events of the form $(x\ \mathcal{C}\ v')$ for some vertex $x \in \mathcal{V}_1$ and a vertex $v' \in M_{\tilde{e}}(j,q')$ for $q \neq q'$ (or on $v' \in M_{\tilde{e}'}(j,q)$ for a different super-edge $\tilde{e}' \neq \tilde{e}$). Hence, for every index $q \in [L]$, and any choice of $L-1$ binary values $b_1, b_2, \ldots, b_{q-1}, b_{q+1}, \ldots, b_L \in \{0,1\}$,

$$\mathbb{P}(X_q = 1 \mid \forall p \in [L] \setminus \{q\}, X_p = b_p) \ \leq \ 1 - \frac{1}{n^{c_1+16}}\ .$$

For every index $q \in [L]$, let X'_q denote the random variable $1 - X_q$. Then

$$\mathbb{P}(X'_q = 1 \mid \forall p \in [L] \setminus \{q\}, X_p = b_p) \ \geq \ \frac{1}{n^{c_1+16}} \ .$$

For every index $q \in [L]$, let

$$Z_q \ = \ \begin{cases} 1, & \text{with probability } \frac{1}{n^{c_1+16}}, \\ 0, & \text{with probability } 1 - \frac{1}{n^{c_1+16}}. \end{cases}$$

(The random variables $Z_1, Z_2, \ldots, Z_L$ are identically distributed and independent.)

Then $\mathbb{P}(\sum_{q=1}^{L} X'_q \geq n) \geq \mathbb{P}(\sum_{q=1}^{L} Z_q \geq n)$. Let $Z = \sum_{q=1}^{L} Z_q$. Note that $\mathbb{E}(Z) = \sum_{q=1}^{L} \mathbb{E}(Z_q) = \frac{L}{n^{c_0-1}} = 2n$. Hence, by Chernoff inequality, $\mathbb{P}(X' \geq n) \geq \mathbb{P}(Z \geq n) \geq 1 - exp(-\Omega(n^{c_0}))$. In other words, $\mathbb{P}(X \leq 2 \cdot n^{c_0} - n) \leq 1 - exp(-\Omega(n^{c_0}))$. ∎

Lemma 4.21 *For a fixed super-edge $\tilde{e}$, and a fixed short schedule Π, $\mathbb{P}(\Pi \ \mathcal{INF} \ M_{\tilde{e}}(j)) \leq exp(-\Omega(n^{c_0}))$, where j is the pivot of the super-edge $\tilde{e}$ with respect to the schedule Π.*

Proof: By Bayes formula,

$$\begin{aligned} &\mathbb{P}(\Pi \ \mathcal{INF} \ M_{\tilde{e}}(j)) \ \leq \ \mathbb{P}(\Pi \ \mathcal{INF} \ M_{\tilde{e}}(j) \mid X \leq 2n^{c_0} - n) + \mathbb{P}(X > 2n^{c_0} - n) \ \leq \\ &\mathbb{P}(\Pi \ \mathcal{INF} \ M_{\tilde{e}}(j) \mid X \leq 2n^{c_0} - n) + exp(-\Omega(n^{c_0})) \ . \end{aligned}$$

Let $\alpha = \alpha(X) \in [L]$ denote the subset of indices $q \in [L]$ such that $X_q = 1$ (i.e., the event $(\pi_{\tilde{e}}(j) \ \mathcal{INF} \ M_{\tilde{e}}(j,q))$ holds), and $\beta = \beta(X) = [L] \setminus \alpha$ denote the complimentary subset of indices. Note that whenever $X \leq 2 \cdot n^{c_0} - n$, necessarily $|\beta(X)| \geq n$. Observe also that by definition of the subset $\alpha(X)$,

$$\mathbb{P}(\Pi \ \mathcal{INF} \ M_{\tilde{e}}(j) \mid X \leq 2 \cdot n^{c_0} - n) \ = \ \mathbb{P}(\forall \ q \in \beta(X), (\Pi \ \mathcal{INF} \ M_{\tilde{e}}(j,q)) \mid X \leq 2 \cdot n^{c_0} - n) \ .$$

Recall that by definition of the subset $\beta(X)$, for each index $q \in \beta(X)$, the event
$(\pi_{\tilde{e}}(j) \ \mathcal{NINF} \ M_{\tilde{e}}(j,q))$ holds. Hence,

$$\begin{aligned} &\mathbb{P}(\forall \ q \in \beta(X), (\Pi \ \mathcal{INF} \ M_{\tilde{e}}(j,q)) \mid X \leq 2 \cdot n^{c_0} - n) \ = \\ &\mathbb{P}(\forall \ q \in \beta(X), (\Pi_{\tilde{e}}(j) \ \mathcal{INF} \ M_{\tilde{e}}(j,q)) \mid X \leq 2 \cdot n^{c_0} - n) \ . \end{aligned}$$

Consider an arbitrary subset $M'' \subseteq M_{\tilde{e}}(j)$ that is formed by picking one arbitrary element v_q from the subset $M_{\tilde{e}}(j,q)$ for each index $q \in [L]$ and inserting v_q into the subset M''. Note that $|M''| = |\beta(X)| \geq n$. Let $M' \subseteq M''$ be a subset of M' obtained by choosing some arbitrary n elements of the subset M'. Note that

$$\mathbb{P}(\forall \ q \in \beta(X), (\Pi_{\tilde{e}}(j) \ \mathcal{INF} \ M_{\tilde{e}}(j,q)) \mid X \leq 2 \cdot n^{c_0} - n) \ \leq \ \mathbb{P}(\Pi_{\tilde{e}}(j) \ \mathcal{INF} \ M') \ .$$

Note that the subset $M' \subseteq M_{\tilde{e}}(j)$ satisfies the conditions of Lemma 4.15. Hence, by Lemma 4.15, $\mathbb{P}(\Pi_{\tilde{e}}(j)\ \mathcal{INF}\ M') \le \frac{1}{2^{\Omega(n \cdot \log n)}}$. To conclude, $\mathbb{P}(\Pi\ \mathcal{INF}\ M_{\tilde{e}}(j)) \le \frac{1}{2^{\Omega(n \cdot \log n)}} + exp(-\Omega(n^{c_0})) \le exp(-\Omega(n^{c_0}))$. (The first term is "swallowed" by the Ω-notation in the exponent.) ▮

Definition 4.22 *A schedule Π is called* proper *if there exists a subset $\tilde{H} \subseteq \tilde{E}$ that contains at least one half of all the super-edges, and such that the schedule Π is $\tilde{e}$-proper with respect to every super-edge $\tilde{e} \in \tilde{H}$. Otherwise, the schedule Π is called* non-proper.

Let $\hat{\Pi}$ denote the collection of all short proper schedules.

Lemma 4.23 *With probability at least $1 - exp(-\Omega(n^{1.6} \log^3 n))$, there is no admissible schedule $\Pi \in \hat{\Pi}$.*

Remark: This lemma holds for instances $\mathcal{G}$ of the radio broadcast problem that are obtained via our reduction from *both* NO- and YES-instances $\mathcal{M}$ of the MAX-REP problem. However, whenever the reduction starts from a YES-instance $\mathcal{M}$ of the MAX-REP problem, the corresponding instance $(\mathcal{G}, s)$ of the radio broadcast problem has an admissible short *non-proper* schedule, and as we will later see, this is not the case when $\mathcal{M}$ is a NO-instance.

Proof: Consider a fixed schedule $\Pi \in \hat{\Pi}$. By definition, Π is a proper schedule, and $|\Pi| \le \log^2 n/100$. In other words, there exists a subset $\tilde{H} \subseteq \tilde{E}$ of super-edges, so that for every super-edge $\tilde{e} \in \tilde{H}$, the schedule Π is $\tilde{e}$-proper, and $|\tilde{H}| \ge |\tilde{E}|/2$.

Consider two distinct fixed super-edges $\tilde{e}_1, \tilde{e}_2 \in \tilde{E}$, and a schedule Π that is both $\tilde{e}_1$-proper and $\tilde{e}_2$-proper. As was already mentioned, it is easy to see that the proof of Lemma 4.21 can be carried through for conditional probabilities of certain types, and, in particular, it follows that

$$\mathbb{P}((\Pi\ \mathcal{INF}\ M_{\tilde{e}_1}(j)) \mid (\Pi\ \mathcal{INF}\ M_{\tilde{e}_2}(j))) \ \le\ exp(-\Omega(n^{c_0}))\ .$$

Also, since

$$\begin{aligned} &\mathbb{P}((\Pi\ \mathcal{INF}\ M_{\tilde{e}_1}(j)), (\Pi\ \mathcal{INF}\ M_{\tilde{e}_2}(j))) \ = \\ &\mathbb{P}((\Pi\ \mathcal{INF}\ M_{\tilde{e}_1}(j)) \mid (\Pi\ \mathcal{INF}\ M_{\tilde{e}_2}(j))) \cdot \mathbb{P}((\Pi\ \mathcal{INF}\ M_{\tilde{e}_2}(j)))\ , \end{aligned}$$

it follows that for a schedule Π that is $\tilde{e}$-proper for every super-edge $\tilde{e} \in \tilde{H}$,

$$\mathbb{P}(\Pi\ \mathcal{INF}\ \mathcal{V}_2) \ \le\ \mathbb{P}(\Pi\ \mathcal{INF}\ \bigcup_{\tilde{e} \in \tilde{H}} M_{\tilde{e}}(j)) \ \le\ exp(-\Omega(n^{c_0}))^{|\tilde{H}|}\ .$$

Recall that the set $\mathcal{V}_1$ contains $n^{0.6}$ copies of every vertex of the set V, and therefore, $|\mathcal{V}_1| = n^{0.6}|V| = n^{1.6}$. Each round R is a subset of the set $\mathcal{V}_1$, and so, there are at most

$2^{n^{1.6}}$ ways to choose one round. Since a short schedule contains at most $\log^2 n/100$ rounds, it follows that $|\hat{\Pi}| \leq 2^{n^{1.6} \cdot \log^2 n}$. Hence, by union bound,

$$\mathbb{P}(\exists \Pi \in \hat{\Pi} \text{ s.t. } (\Pi\ \mathcal{INF}\ \mathcal{V}_2)) \leq 2^{n^{1.6} \log^2 n - \Omega(n \cdot |\tilde{H}|)} .$$

Recall that by restriction 2 of the MAX-REP problem (see Section 2.3), $|\tilde{E}| = \Omega(n^{0.6} \log^3 n)$. Recall also that $|\tilde{H}| \geq |\tilde{E}|/2$, and so

$$\mathbb{P}(\exists \Pi \in \hat{\Pi} \text{ s.t. } (\Pi\ \mathcal{INF}\ \mathcal{V}_2)) \leq 2^{n^{1.6} \log^2 n - \Omega(n^{1.6} \log^3 n)} = 2^{-\Omega(n^{1.6} \log^3 n)} . \qquad \blacksquare$$

4.7 Deriving the Results

In this section we glue together all the pieces of the analysis and derive our main results.

Let $\mathcal{M} = (G = (V, E), \tilde{G})$, $|V| = n$, be a YES-instance of the MAX-REP problem, and let $(\mathcal{G}, s)$ be the instance of of the radio broadcast problem that is obtained via our reduction. Then

Lemma 4.24 *There is an admissible schedule* Π *of length* $|\Pi| \leq 4 \cdot \log n + 1$ *for the instance* $(\mathcal{G}, s)$ *of the radio broadcast problem (and this schedule is non-proper).*

Proof: Let C be a perfect MAX-cover, that is, a subset of the set $V_1 \cup V_2$ that covers all the super-edges and contains exactly one vertex from each super-vertex. (Recall that by definition of the YES-instance of the MAX-REP problem, there exists a perfect MAX-cover C for such an instance.) For each super-vertex $X \in \tilde{V}_1 \cup \tilde{V}_2$, let $C(X)$ denote the unique vertex $x \in X$ such that $x \in C \cap X$.

We next construct the schedule Π of the form $\Pi = \Pi_0 \cdot \Pi_{0.4 \log n} \cdot \Pi_{0.4 \log n + 1} \cdot \ldots \cdot \Pi_{0.6 \log n}$, where $\cdot$ stands for concatenation, the schedule Π_0 consists of a single singleton round $\{s\}$, and the schedules Π_j, $j \in \hat{J}$, are constructed in the following way. For each index $j \in \hat{J}$, the schedule Π_j consists of four rounds. On the first round of the schedule Π_j, the vertices of the set $\bigcup_{A \in \tilde{V}_1} \chi^{(1)}_{C(A)}(j)$ broadcast. On the second round of Π_j, the vertices of the set $\bigcup_{A \in \tilde{V}_1} \chi^{(2)}_{C(A)}(j)$ broadcast. On the third round of Π_j, the vertices of the set $\bigcup_{B \in \tilde{V}_2} \chi^{(1)}_{C(B)}(j)$ broadcast. On the fourth round of Π_j, the vertices of the set $\bigcup_{B \in \tilde{V}_2} \chi^{(2)}_{C(B)}(j)$ broadcast.

We next argue that the first round of the schedule Π_j informs the set $\bigcup_{\tilde{e}=(A,B) \in \tilde{E}} \bigcup_{q=1}^{L} H^{(1)}_{\tilde{e},C(A)}(j, q)$. It is easy to see that essentially the same argument shows also that the second round of the schedule Π_j informs the set $\bigcup_{\tilde{e}=(A,B) \in \tilde{E}} \bigcup_{q=1}^{L} H^{(2)}_{\tilde{e},C(A)}(j, q)$, the third round informs the set $\bigcup_{\tilde{e}=(A,B) \in \tilde{E}} \bigcup_{q=1}^{L} H^{(1)}_{\tilde{e},C(B)}(j, q)$, and the fourth round informs the set $\bigcup_{\tilde{e}=(A,B) \in \tilde{E}} \bigcup_{q=1}^{L} H^{(2)}_{\tilde{e},C(B)}(j, q)$.

Consider a round on which all the vertices of the set $\bigcup_{A \in \tilde{V}_1} \chi^{(1)}_{C(A)}(j)$ broadcast, and no other vertex does. We argue that such a round informs the set $\bigcup_{\tilde{e}=(A,B) \in \tilde{E}} \bigcup_{q=1}^{L} H^{(1)}_{\tilde{e},C(A)}(j, q)$.

Observe that, by construction, for a fixed index $q \in [L]$, a super-edge $\tilde{e} = (A, B)$, and a super-vertex $A' \in \tilde{V}_1$, $A' \neq A$, there are no edges between the set $\chi^{(1)}_{A'}(j)$ and the set $H^{(1)}_{\tilde{e},A}(j, q)$. Hence, it is sufficient to argue that for every index $q \in [L]$, and for every super-vertex $A \in \tilde{V}_1$, and a super-edge $\tilde{e}$ that is adjacent to the super-vertex A, the set $\chi^{(1)}_{C(A)}(j)$ informs the set $H^{(1)}_{\tilde{e},C(A)}(j, q)$.

We actually prove a stronger claim. Specifically, we next show the above claim not just for the vertex $C(A) \in A$, but rather for an arbitrary vertex $a \in A$.

Recall that by construction (the random permutation step), the set $H^{(1)}_{\tilde{e},a}(j, q)$ is divided into $\bar{\chi} = |\chi^{(1)}_a(j)|$ arbitrary subsets $P^{(1)}_{\tilde{e},a}(j, q, \ell)$, $\ell \in [\bar{\chi}]$ of equal size. Then a random permutation $\sigma \in S_{\bar{\chi}}$ ($S_{\bar{\chi}}$ stands for the set of all permutations of the set $[\bar{\chi}]$) is applied to the set $\chi^{(1)}_a(j) = \{cp_1(a), cp_2(a), \ldots, cp_{\bar{\chi}}(a)\}$, and for every index $k \in [\bar{\chi}]$, the vertex $cp_k(a)$ is connected to all the vertices of the set $P^{(1)}_{\tilde{e},a}(j, q, \sigma(k))$. Note also that for a vertex $v \in P^{(1)}_{\tilde{e},a}(j, q, \sigma(k))$, and a vertex $cp_\ell(a) \in \chi^{(1)}_a(j)$ with $\ell \neq k$, by construction, these two vertices are necessarily not connected. It follows that the set $\chi^{(1)}_a(j) = \{cp_1(a), cp_2(a), \ldots, cp_{\bar{\chi}}(a)\}$ informs the set $\bigcup_{k=1}^{\chi} P^{(1)}_{\tilde{e},a}(j, q, \sigma(k)) = H^{(1)}_{\tilde{e},a}(j, q)$, as required.

Note that $\mathcal{V}_2 = \bigcup_{j \in \hat{J}} \left(\bigcup_{\tilde{e} \in \tilde{E}} M_{\tilde{e}}(j) \right)$, and for a super-edge $\tilde{e} = (A, B)$,

$$
\begin{aligned}
M_{\tilde{e}}(j) &= \bigcup_{q=1}^{L} (H^{(1)}_{\tilde{e},C(A)}(j,q) \cup H^{(2)}_{\tilde{e},C(A)}(j,q) \cup H^{(1)}_{\tilde{e},C(B)}(j,q) \cup H^{(2)}_{\tilde{e},C(B)}(j,q)) \\
&= \bigcup_{q=1}^{L} (H_{\tilde{e},C(A)}(j,q) \cup H_{\tilde{e},C(B)}(j,q)) \ .
\end{aligned}
$$

(Since the set C is a perfect MAX-cover, for every super-edge $\tilde{e} = (A, B) \in \tilde{E}$, the pair $(C(A), C(B))$ is an $\tilde{e}$-m.p., and so $H_{\tilde{e},C(A)}(j, q) \cup H_{\tilde{e},C(B)}(j, q) = M_{\tilde{e}}(j, q)$.)

Hence, for every index $j \in \hat{J}$, the set $\left(\bigcup_{\tilde{e} \in \tilde{E}} M_{\tilde{e}}(j) \right)$ is informed by the subschedule Π_j, and consequently, the set $\mathcal{V}_2$ is informed by the schedule Π. ▮

Lemma 4.25 *For a NO-instance $\mathcal{M}$ of the MAX-REP problem, and the instance $\mathcal{G}$ of the radio broadcast problem that is obtained from the instance $\mathcal{M}$ via our reduction, with probability 1 there is no non-proper admissible short schedule Π for the instance $\mathcal{G}$.*

Proof: Suppose for contradiction that there exists a schedule Π as above. Assume, without loss of generality, that the first round of the schedule Π is the set $\{s\}$, that all the other rounds $R \in \Pi$ are subsets of the set $\mathcal{V}_1$. Let $\tilde{H} \subseteq \tilde{E}$ be a subset of super-edges, such that for every super-edge $\tilde{e} \in \tilde{H}$, the schedule Π is $\tilde{e}$-proper, and let $\tilde{H}' = \tilde{E} \setminus \tilde{H}$ be the complimentary set (i.e., for every super-edge $\tilde{e} \in \tilde{H}'$, the schedule Π is not $\tilde{e}$-proper). Since the schedule Π is non-proper, $|\tilde{H}| < |\tilde{E}|/2$, and so $|\tilde{H}'| > |\tilde{E}|/2$.

For a super-edge $\tilde{e} \in \tilde{H}'$, let $j = j(\tilde{e})$ be the pivot of the super-edge $\tilde{e}$ with respect to the schedule Π. By definition, for every super-edge $\tilde{e} \in \tilde{H}'$, the set $V_{\tilde{e}}(\pi_{\tilde{e}}(j(\tilde{e})))$ is not $(\tilde{e}, j)$-partial, and so, there exists an $\tilde{e}$-m.p. $(a, b) \in (A \times B) \cap E$ such that

$$|V_{\tilde{e}}(\pi_{\tilde{e}}(j(\tilde{e}))) \cap \chi_a(j)|, |V_{\tilde{e}}(\pi_{\tilde{e}}(j(\tilde{e}))) \cap \chi_b(j)| \ \geq \ 2^{j(\tilde{e})-3} \ .$$

Pick one such $\tilde{e}$-m.p. for every super-edge $\tilde{e} \in \tilde{H}'$, and denote it $(a(\tilde{e}), b(\tilde{e}))$. For each index $j \in \hat{J}$, let $\tilde{H}'_j$ be the subset of set $\tilde{H}'$ that contains all the super-edges whose pivot with respect to the schedule Π is equal to j. Note that $\tilde{H}' = \bigcup_{j \in \hat{J}} \tilde{H}'_j$. For each super-vertex $X \in \tilde{V}_1 \cup \tilde{V}_2$, let $\tilde{H}'_j(X)$ denote the subset of $\tilde{H}'_j$ that contains all the super-edges $\tilde{e} \in \tilde{H}'_j$ that are adjacent to the super-vertex X. Note that $\tilde{H}'_j = \bigcup_{A \in \tilde{V}_1} \tilde{H}'_j(A) = \bigcup_{B \in \tilde{V}_2} \tilde{H}'_j(B)$.

We next form a partial cover C_j for the MAX-REP instance $\mathcal{M}$ in the following way. For each super-vertex $A \in \tilde{V}_1$, set $C_j(A) = \{a(\tilde{e}) \mid \tilde{e} \in \tilde{H}'_j(A)\}$, and, analogously, for each super-vertex $B \in \tilde{V}_2$, set $C_j(B) = \{b(\tilde{e}) \mid \tilde{e} \in \tilde{H}'_j(B)\}$. Now, set $C_j = \bigcup_{X \in \tilde{V}_1 \cup \tilde{V}_2} C_j(X)$, and $C = \bigcup_{j \in \hat{J}} C_j$. Also, for a super-vertex $X \in \tilde{V}_1 \cup \tilde{V}_2$, let $C(X)$ denote the set $C \cap X = \bigcup_{j \in \hat{J}} C_j(X)$.

Note that for for every super-edge $\tilde{e} = (A, B) \in \tilde{H}'$, necessarily $a(\tilde{e}) \in C_{j(\tilde{e})}(A)$, and $b(\tilde{e}) \in C_{j(\tilde{e})}(B)$, and so $a(\tilde{e}), b(\tilde{e}) \in C_{j(\tilde{e})} \subseteq C$. Hence, the set C covers all the super-edges of the set $\tilde{H}'$, and since $|\tilde{H}'| \geq |\tilde{E}|/2$, it follows that the set C covers at least one half of all the super-edges of the instance $\mathcal{M}$ of the MAX-REP problem.

We next provide an upper bound on the size of each set $C_j(X)$ for each index $j \in \hat{J}$ and super-vertex $X \in \tilde{V}_1 \cup \tilde{V}_2$. Consider first $C_j(A)$ for some super-vertex $A \in \tilde{V}_1$ and index $j \in \hat{J}$. By definition, $C_j(A) = \{a(\tilde{e}) \mid \tilde{e} \in \tilde{H}'_j(A)\}$, and $\tilde{H}'_j(A)$ is the subset of super-edges that are adjacent to $\tilde{e}$ and such that $j(\tilde{e}) = j$. For every super-edge $\tilde{e} = (A, B) \in \tilde{H}'_j(A)$, consider the subschedule $\pi_{\tilde{e}}(j)$ of the schedule Π. Recall that the subschedule $\pi_{\tilde{e}}(j)$ contains only the rounds R that are j-small with respect to the super-edge $\tilde{e}$, i.e., the rounds R that satisfy $|R \cap (\chi_A \cup \chi_B)| \leq 2^{j+1} \cdot c \cdot \ln n = O(2^j \cdot \log n)$. Note also that $|\pi_{\tilde{e}}(j)| \leq |\Pi| = O(\log^2 n)$. Hence, $|V_{\tilde{e}}(\pi_{\tilde{e}}(j)) \cap (\chi_A \cup \chi_B)| = O(2^j \cdot \log^3 n)$. However, observe that as $\tilde{e} = (A, B) \in \tilde{H}_j(A)$, it follows that $j = j(\tilde{e})$ is the pivot of the super-edge $\tilde{e}$ with respect to the schedule Π. Also, the schedule Π is not $\tilde{e}$-proper, i.e., the set $V_{\tilde{e}}(\pi_{\tilde{e}}(j))$ is not $(\tilde{e}, j)$-partial. By the choice of the vertices $a(\tilde{e})$, $b(\tilde{e})$, they satisfy the inequality

$$|V_{\tilde{e}}(\pi_{\tilde{e}}(j(\tilde{e}))) \cap \chi_{a(\tilde{e})}(j(\tilde{e}))|, |V_{\tilde{e}}(\pi_{\tilde{e}}(j(\tilde{e}))) \cap \chi_{b(\tilde{e})}(j(\tilde{e}))| \ \geq \ 2^{j(\tilde{e})-3} \ .$$

Since for every super-edge $\tilde{e} \in \tilde{H}'_j(A)$, $j(\tilde{e}) = j$, it follows that the two following conditions also hold for every super-edge $\tilde{e} \in \tilde{H}'_j(A)$.

1. $|V_{\tilde{e}}(\pi_{\tilde{e}}(j)) \cap \chi_A| = O(2^j \cdot \log^3 n)$.

2. $|V_{\tilde{e}}(\pi_{\tilde{e}}(j)) \cap \chi_{a(\tilde{e})}(j)| \geq 2^{j-3}$.

Note that for different vertices $a_1, a_2 \in A$, the sets $(V_{\tilde{e}}(\pi_{\tilde{e}}(j)) \cap \chi_{a_1}(j))$ and $(V_{\tilde{e}}(\pi_{\tilde{e}}(j)) \cap \chi_{a_2}(j))$ are disjoint. Let A' denote the set $\{a \mid \exists \tilde{e} \in \tilde{H}'_j(A) \text{ s.t. } a = a(\tilde{e})\}$. Hence, the conditions (1) and (2) imply that

$$2^j \cdot \log^3 n \;\geq\; |V_{\tilde{e}}(\pi_{\tilde{e}}(j)) \cap \chi_A| \;=\; |\bigcup_{a \in A'} (V_{\tilde{e}}(\pi_{\tilde{e}}(j)) \cap \chi_a)| \;\geq$$
$$|\bigcup_{a \in A'} (V_{\tilde{e}}(\pi_{\tilde{e}}(j)) \cap \chi_a(j))| \;\geq\; 2^{j-3} \cdot |A'| \;.$$

Hence,

$$|A'| \;=\; |\{a \in A \mid a = a(\tilde{e}) \text{ for some } \tilde{e} \in \tilde{H}'_j(A)\}| \;=\; O(\log^3 n) \;. \tag{19}$$

Note, however, that

$$C_j(A) \;=\; \{a(\tilde{e}) \mid \tilde{e} \in \tilde{H}'_j(A)\} \;=\; \{a \mid \exists \tilde{e} \in \tilde{H}'_j(A) \text{ s.t. } a = a(\tilde{e})\} \;=\; A' \;.$$

Hence, $|C_j(A)| = O(\log^3 n)$.

Analogous considerations show that for every super-vertex $B \in \tilde{V}_2$ and index $j \in \hat{J}$, necessarily $|C_j(B)| = O(\log^3 n)$. Hence, for every index $j \in \hat{J}$, $|C_j| = O(\log^3 n)|\tilde{V}|$, and so $|C| = \sum_{j \in \hat{J}} |C_j| = O(\log^4 n)|\tilde{V}|$. Actually, we have shown a stronger claim, specifically, that for every super-vertex $X \in \tilde{V}_1 \cup \tilde{V}_2$, $|C \cap X| = |C(X)| = |\bigcup_{j \in \hat{J}} C_j(X)| = O(\log^4 n)$.

In other words, we have shown that the set C covers at least one half of all the super-edges of the instance $\mathcal{M}$ of the MAX-REP problem, and contains $O(\log^4 n)$ representative vertices from each super-vertex. Consider a subset $C' \subseteq C$ formed by picking one representative x from $C \cap X$ for each super-vertex $X \in \tilde{V}_1 \cup \tilde{V}_2$ independently at random. Observe that the expected number of super-edges that are covered by C' is at least $\frac{|\tilde{E}|}{2 \cdot \log^8 n}$. Therefore, there exists a subset $C^* \subseteq C$ that covers at least $\frac{|\tilde{E}|}{2 \cdot \log^8 n}$ super-edges, and contains exactly one representative vertex from each super-vertex (i.e., for every super-vertex $X \in \tilde{V}_1 \cup \tilde{V}_2$, $|C^* \cap X| = 1$). However, this implies that the instance $\mathcal{M}$ is a YES-instance of the MAX-REP problem, contradiction.

Therefore, there is no non-proper admissible short schedule Π for the instance $\mathcal{G}$ of the radio broadcast problem. ∎

Corollary 4.26 *Consider a NO-instance $\mathcal{M}$ of the MAX-REP problem, and the instance $\mathcal{G}$ of the radio broadcast problem that is obtained from the instance $\mathcal{M}$ via our reduction. With probability at least $1 - exp(-\Omega(n^{1.6} \log^3 n))$ there exists no admissible short schedule Π for the instance $\mathcal{G}$.*

Proof: By Lemma 4.25, under these assumptions, with probability 1 there is no non-proper admissible short schedule for the instance $\mathcal{G}$. By Lemma 4.23, with probability at least $1 - exp(-\Omega(n^{1.6}\log^3 n))$ there is no proper admissible short schedule for the instance $\mathcal{G}$. Since any schedule is either proper or non-proper, the assertion follows. ∎

Theorem 4.27 *Unless $NP \subseteq BPTIME(n^{O(\log\log n)})$, there exists a universal constant c such that no (deterministic or randomized) polynomial time algorithm may provide an additive $(c \cdot \log^2 n)$-approximation guarantee for the radio broadcast problem.*

Proof: Note that our reduction can be implemented in the straightforward way in probabilistic polynomial time, and the size of the instance $\mathcal{G}$ of the radio broadcast problem that is returned by the reduction whenever it is invoked on an instance $\mathcal{M}$ of size n of the MAX-REP problem is polynomial in n (specifically, $O(n^{c_0})$, where c_0 is given by (18) and (16)). Hence, the assertion of the theorem with $c = \frac{1}{100 \cdot c_0^2}$ follows directly from Lemma 4.24, Corollary 4.26 and Theorem 2.5. ∎

Observe that our reduction shows that (under a certain complexity-theoretic assumption) no algorithm that has polynomial running time may distinguish between instances of the radio broadcast problem that have value at most $4\log n + 1$ and instances that have value $\Omega(\log^2 n)$. However, it is easy to generalize the reduction in such a way that it will show that under the same assumption no algorithm that has polynomial running time may distinguish between instances of value at most $4\log n + T$ and instances of value $\Omega(\log^2 n) + T$, where T is a parameter of the reduction. To this end, introduce T vertices $s = s_1, s_2, \ldots, s_T$ into the vertex set $\bar{V}$ of the graph $\bar{G} = (\bar{V}, \bar{E})$ (where $\mathcal{G} = (\bar{G} = (\bar{V}, \bar{E}), s)$, $\bar{V} = \{s_1, s_2, \ldots, s_T\} \cup \mathcal{V}_1 \cup \mathcal{V}_2$, is the instance of the radio broadcast problem), insert the edges $(s_1, s_2), (s_2, s_3), \ldots, (s_{T-1}, s_T)$ into the edge set $\bar{E}$ of the graph $\bar{G}$, and connect the vertex s_T to all the vertices of the set $\mathcal{V}_1$. The other steps of the reduction, as well as its analysis, stay intact.

Acknowledgements

The first-named author is grateful to Oded Regev for helpful discussions.

References

[AB+92] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Fast network decomposition. In *Proc. 11th ACM Symp. on Principles of Distr. Comp.*, pp. 161-173, Aug. 1992.

[ABLP89] N. Alon, A. Bar-Noy, N. Linial, D. Peleg. A lower bound for radio broadcast. In *Journal of Computer and System Sciences 43*, pp. 290-298, 1991. Preliminary version appeared in *Proc. 21st Ann. Symp. on Theory of Comp.*, Seattle, WA, May 1989.

[ABSS93] S. Arora, L. Babai, J. Stern, and Z.Swedyk. The hardness of approximate optima in lattices, codes and linear equations. In *Proc. 34th IEEE Symp. on Foundations of Computer Science*, pp. 724-733, 1993.

[AL96] S. Arora and K. Lund. In *Approximation Algorithms for NP-hard Problems*, D. Hochbaum, ed., PWS Publishing, 1996.

[AS92] N. Alon and J. Spencer. *The Probabilistic Method.* Wiley, 1992.

[B01] R. Bar-Yehuda. Private communication.

[BGI86] R. Bar-Yehuda, R., O. Goldreich, and A. Itai. Efficient emulation of single-hop radio network with collision detection on multi-hop radio network with no collision detection. In *Distributed Computing*, 5(2):67–72, 1991. Preliminary version appeared in *Proc. 4th ACM Symp. on Principles of Distr. Comp.*, pp.98-107, 1986.

[C52] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Annals of Mathematical Statistics*, 23:493–509, 1952.

[CG+02] B. S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, W. Rytter. Deterministic broadcasting in ad hoc radio networks. Distributed Computing 15(1): 27-38 (2002)

[CGR00] M. Chrobak, L. Gasieniec, W. Rytter. Fast Broadcasting and Gossiping in Radio Networks. In *Proc. 41st Symp. on Foundations of Computer Science*, pp. 575-581, Redondo Beach, CA, Nov. 2000.

[CK85] I. Chlamtac and S. Kutten. A spatial-reuse TDMA/FDMA for mobile multihop radio networks. In *Proc. IEEE INFOCOM*, pp. 389-94, 1985.

[CMS03] A. F. Clementi, A. Monti, R. Silvestri. Distributed broadcast in radio networks of unknown topology. TCS 1-3(302): 337-364 (2003), Preliminary version appeared in *Proc. of the 14th ACM-SIAM Symp. on Discr. Algorithms*, pp. 709-718, Washington, DC, Jan. 2001.

[CR03] A. Czumaj and W. Rytter. Broadcasting Algorithms in Radio Networks with Unknown Topology. in *Proc. 44th Symp. on Foundations of Computer Science*, pp. 492-501, 2003.

[CW87] Chlamtac and Weinstein The wave expansion approach to broadcasting in multihop radio networks, In *Proc. IEEE INFOCOM*, pp. 874-881, 1987.

[DGKR03] I. Dinur, V. Guruswami, S. Khot, and O. Regev. A New Multilayered PCP and the Hardness of Hypergraph Vertex Cover, In *Proc. 35th Symp. on Theory of Computing*, pp. 595-601, 2003.

[EK02] M. Elkin and G. Kortsarz. Approximation Algorithm for Directed Telephone Multicast Problem. In *Proc. 34th Annual ACM Symp. on Theory of Computing*, pp. 212-223, Canada, Montreal, May 2002.

[FR94] M. Furer and B. Raghavachari. Approximating the Minimum-Degree Steiner Tree to within One of Optimal. J. Algorithms 17(3), pp. 409-423, 1994.

[GKR00] N. Garg, G. Konjevod, R. Ravi. A Polylogarithmic Approximation Algorithm for the Group Steiner Tree Problem. J. Algorithms 37(1), pp. 66-84, 2000.

[GM95] I. Gaber and Y. Mansour. Broadcast in radio networks. In *Proc. of the 6th ACM-SIAM Symp. on Discr. Algorithms*, pp. 577–585, 1995.

[HK03] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability, in *Proc. of the 35th Symp. on Theory of Computing*, pp. 585-594, 2003.

[HK+03] E. Halperin, G. Kortsarz, R. Krauthgamer, A. Srinivasan and N. Wang, Integrality ratio for Group Steiner Trees and Directed Steiner Trees, In *Proc. of Symp. on Discr. Algorithms*, pp. 275-284, 2003.

[I02] P. Indyk. Explicit constructions of selectors and related combinatorial structures, with applications. In *Proc. 15th Symp. on Discr. Algorithms*, pp. 697-704, 2002.

[KM98] Kushilevitz and Mansour. An $\Omega(D \log(n/D))$ lower bound for broadcast in radio networks. In *SIAM J. of Computing*, Vol. 27, No. 3, pp. 702-712, June 1998.

[KP02] D. Kowalski and A. Pelc, Deterministic broadcasting time in radio networks of unknown topology, In *Proc. 43rd Ann. IEEE Symposium on Foundations of Computer Science*, pp. 63-72, 2002.

[KP03] D. Kowalski and A. Pelc, Broadcasting in undirected ad hoc radio networks In *Proc. 16th ACM Symp. on Principles of Distr. Comp.*, pp. 73 - 82, 2003.

[LS91] N. Linial and M. Saks. Decomposing graphs into regions of small diameter. In *Proc. 2nd ACM-SIAM Symp. on Discr. Alg.*, pp. 320-30, 1991.

[LY94] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. Assoc. Comput. Mach.*, 41(5):960–981, 1994.

[MR95] R. Motwani, P. Raghavan. Randomized Algorithms. *Cambridge University Press.*, 1995.

[R98] R. Raz. A Parallel Repetition Theorem, *SIAM J. of Computing*, 27(3) pp. 763-803, 1998.

A Upper Bound

A.1 Preliminaries

Definition A.1 *For a rooted tree $(T = (V_T, E_T), s)$, $s \in V_T$, the* parent *of a vertex $v \in V_T$, $v \neq s$, is the unique vertex u such that $(u, v) \in E_T$, and u belongs to the path between s and v in T. The vertex u is denoted $parent_T(u)$. The vertex v is also said to be a* child *of the vertex u in the rooted tree (T, s).*

For a rooted tree (T, s), the rank *of a vertex $v \in V_T$ is defined as follows. If v is a leaf, its rank is 0. For an internal vertex v, let R be the set of ranks of all the children of the vertex v, and let r be the maximum value in R. If there is exactly one child w of v with rank r, then rank of the vertex v is defined to be r; otherwise, it is defined to be $r + 1$.*

For a vertex $v \in V_T$, let $\ell_T(v)$ (or $\ell(v)$, when the tree T can be deduced from the context) denote the level *of the vertex v in the rooted tree (T, s), that is, the distance between the vertices s and v in T.*

The radius *of a rooted tree (T, s), denoted $Rad(T, s)$, is $\max\{\ell(v) \mid v \in V_T\}$. For a graph $G = (V, E)$, and a designated vertex s, the* radius of the graph G with respect to the vertex s*, denoted $Rad(G, s)$, is the radius of any breadth-first-search (henceforth, BFS) spanning tree of the graph G rooted in the vertex s. The* diameter *of a graph $G = (V, E)$, denoted $Diam(G)$, is the maximal distance between a pair of vertices $u, w \in V$. For a subset $C \subseteq V$ of the vertices, the* diameter *of C, denoted $Diam_G(C)$, is the maximum distance in $G(C)$*

between a pair of vertices in C*, where* $G(C) = (C, E(C))$*,* $E(C) = \{(u, w) \in E \mid u, w \in C\}$ *is the* subgraph *of* G induced *by the subset* C*.*

Lemma A.2 *For an* n*-vertex graph rooted tree* (T, s) *and a vertex* $v \in V_T$*,* $0 \leq rank(v) \leq \log n$*.*

Proof: Follows directly from the definition. ▮

Definition A.3 *For a graph* $G = (V, E)$ *and a pair of positive integer numbers* κ *and* α*, a* (κ, α)-partition $\mathcal{C}$ *of the graph* G *is a collection of subsets of* V*, called* clusters*, that satisfy*

1. *For every cluster* $C \in \mathcal{C}$*,* $Diam(C) \leq \kappa$*.*
2. *Consider the super-graph* $\tilde{G} = (\tilde{V}, \tilde{E})$*,* $\tilde{V} = \mathcal{C}$*,* $\tilde{E} = \{(C, C') \mid \exists v \in C,\ v' \in C'\ \ s.t.\ \ (v, v') \in E\}$*. The super-graph* $\tilde{G}$ *is* α-colorable*, i.e., there exists a coloring* $\chi : \mathcal{C} \to [\alpha]$ *such that for every super-edge* $(C, C') \in \tilde{E}$*, necessarily* $\chi(C) \neq \chi(C')$*.*

Theorem A.4 *[LS91, AB+92] For every* n*-vertex graph* $G = (V, E)$*, there exists an* $(O(\log n), O(\log n))$*-partition* $\mathcal{C}$ *of the graph* G*. Furthermore, this partition is computable deterministically within time polynomial in* n*, and also, the coloring* χ *that is associated with the partition* $\mathcal{C}$ *and satisfies the property 2 of Definition A.3 is also computable within the same time bounds.*

Definition A.5 *For an* n*-vertex rooted tree* $(T = (V_T, E_T), s)$*, a* semi-spanning path-forest F *of* T *is a collection of paths that satisfy*

1. *Every path* $P \in F$*,* $P = (V_P, E_P)$*, is contained in the tree* T*, i.e.,* $V_P \subseteq V_T$*,* $E_P \subseteq E_T$*, and, furthermore, if* $V_P = (v_1, v_2, \ldots, v_k)$ *then either* v_1 *is an ancestor of* v_k *in the tree* T *or vice versa.*
2. *The paths of the forest* F *are vertex-disjoint.*
3. *For every vertex* $v \in V_T$*, the unique path that connects the vertices* s *and* v *in the tree* T *contains at most* $\log n$ *edges that do not belong to the set* $\bigcup_{P \in F} E_P$ *(such edges will be henceforth referred to as the* non-edges of the forest F*).*

Lemma A.6 *For every* n*-vertex rooted tree* (T, s)*, a semi-spanning path-forest* F *of the tree exists, and can be constructed within time polynomial in* n*.*

Proof: Consider the edge set $E' = \{(v, w) \in E_T \mid rank(v) = rank(w)\}$. Observe that the edge set E' decomposes into a collection of vertex-disjoint paths. Denote this collection by F. It is easy to see that F is a semi-spanning path-forest of the input tree. ▮

Definition A.7 *For a schedule* $\Pi = (R_1, R_2, \ldots, R_t)$*, a vertex* v *is said to be* scheduled to broadcast on round $\ell \in [t]$ by the schedule Π*, if* $v \in R_\ell$*.*

A.2 Preprocessing

Our algorithm consists of two major parts, the *preprocessing* algorithm, and the algorithm that actually constructs the schedule. While neither of the algorithms is executed offline, and running them efficiently is equally important, the preprocessing algorithm is somewhat detached from the specifics of the radio broadcast problem, and this motivates its name.

In this section we describe our preprocessing algorithm, which is essentially a variant of the analogous part of the algorithm due to Gaber and Mansour [GM95]. However, unlike the algorithm of [GM95] our algorithm forms *overlapping layers* and not disjoint ones, and this difference affects the implementation. In terms of its effect on the length of the resulting schedule, this different way of implemeting the preprocessing algorithm enables us to reduce the additive term from $O(\log^5 n)$ in the result of [GM95] to $O(\log^4 n)$.

The input of the radio broadcast problem is a graph $G = (V, E)$, and a source vertex $s \in V$. Let *Rad* denote $Rad(G, s)$. Assume that *Rad* is at least a sufficiently large constant. For graphs with smaller radius optimal up to a constant factor schedules are provided by the algorithms of Chlamtac and Weinstein [CW87] and Bar-Yehuda et al. [BGI86].

The first step of the preprocessing algorithm is to construct a BFS spanning tree T of the graph G, rooted in the source s. The second step of the algorithm is to cover the vertex set V by x or $x+1$ subsets, where $x = 1, 2, \ldots$ is a positive integer parameter, $x \le Rad/3$. The specific value of x will be determined later. Let d denote $\lfloor Rad/x \rfloor$. Note that $d \ge 3$. The vertex set V is covered by the subsets $V_1, V_2, \ldots, V_x$, (and, possibly V_{x+1} if *Rad* is not divisible by x), where $V_j = \{v \mid (j-1)d + 1 \le \ell(v) \le jd\}$, and $\ell(v) = d_G(s, v)$ is the *level* of the vertex v. Set $y = x$ if *Rad* is divisible by x, and $y = x + 1$ otherwise. Note that for every index $j \in [y-1]$, the subsets V_j and V_{j+1} intersect.

The third step of the algorithm is to form a set $S(v)$ of vertices for every vertex v that satisfies $\ell(v) \equiv 0 \pmod{d}$. The set $S(v)$ contains all the descendents w of the vertex v in the tree T that satisfy $\ell_T(w) \le \ell_T(v) + d$.

Let $\tilde{V} = \{S(v) \mid \ell(v) \equiv 0 \pmod{d}\}$, and $\tilde{E} = \{(S(v), S(v')) \mid (S(v) \cap S(v') \neq \emptyset)$ or $(\exists w \in S(v),\ w' \in S(v')$ s.t. $(w, w') \in E$ and $\ell_T(v) = \ell_T(v'))\}$. Let $\tilde{G} = (\tilde{V}, \tilde{E})$. Note that while two subsets $S(v), S(v')$ with $\ell_T(v) = \ell_T(v')$ share a super-edge if $S(v) \cap S(v') \neq \emptyset$, *or* if there exists and edge $(w, w') \in E$, $w \in S(v)$, $w' \in S(v')$, two subsets $S(v)$, $S(v')$ with $\ell(v) \neq \ell(v')$ share a super-edge only if they intersect.

For every index $j \in [y]$, let $\tilde{V}_j$ denote the subset of $\tilde{V}$ that contains only the sets $S(v) \in \tilde{V}$ that satisfy $\ell_T(v) = (j-1)d$. Let $\tilde{E}_j = \tilde{E}(\tilde{V}_j)$ denote the subset of the super-edges that is induced by the subset $\tilde{V}_j$ of the super-vertices. Denote $\tilde{G}_j = (\tilde{V}_j, \tilde{E}_j)$. Next, for every index $j \in [y]$, the preprocessing algorithm forms an $(O(\log n), O(\log n))$-partition $\mathcal{C}_j$ of the super-graph $\tilde{G}_j$. Let $\mathcal{C} = \bigcup_{j=1}^{y} \mathcal{C}_j$. Note that for an index $j \in [y]$, and a cluster $C \in \mathcal{C}_j$, the diameter of the cluster C with respect to the super-graph $\tilde{G}_j$ is $O(\log n)$, and so, the diameter of the cluster $C_G = \bigcup_{S(v) \in C} S(v)$ is $O(d \log n)$ (since $Diam_G(S(v)) \leq 2d$). With a slight abuse of notation we will henceforth denote by C both the collection of subsets $S(v)$ that were used to form the cluster C, and the set $C_G = \bigcup_{S(v) \in C} S(v)$.

Note that every cluster $C \in \mathcal{C}_j$ contains one or more vertex v of level $\ell_T(v) = (j-1)d$, and one or more vertex w of level $l_T(w) = \min\{jd, Rad\}$. Note also that viewing clusters as collections of subsets $S(v)$, different clusters $C, C' \in \mathcal{C}_j$ cannot possibly intersect in a subset $S(v)$, because $\mathcal{C}_j$ is a *partition* of the super-graph $\tilde{G}_j$. However, viewing clusters as collections of vertices of the graph G, since the sets $S(v)$ do intersect, the clusters $C, C' \in \mathcal{C}_j$ may intersect in a subset $S = C \cap C' \subseteq V$. For a pair of clusters $C_j \in \mathcal{C}_j$, $C_{j+1} \in \mathcal{C}_{j+1}$, they may intersect in a subset $S = C_j \cap C_{j+1} \subseteq V$ of vertices of level jd. Since $d \geq 3$, it is easy to see that two clusters $C_j \in \mathcal{C}_j$, $C_{j'} \in \mathcal{C}_{j'}$ with $|j - j'| \geq 2$ cannot possibly intersect.

The partition $\mathcal{C}_j$ will be referred to as the *jth layer* of the super-graph $\tilde{G}$. Note that the first layer $\mathcal{C}_1$ of the super-graph $\tilde{G}$ contains only one single cluster $C(s) = S(s)$.

Next, the preprocessing algorithm assigns *ranks* to all the clusters of the collection $\mathcal{C}$, constructs a BFS spanning tree τ rooted in the cluster $C(s)$ of the graph $\tilde{G}$, and chooses a pair of designated vertices, called the *messenger* and the *representative* for every cluster $C \in \mathcal{C}$. These three tasks are performed simultaneously in the following way.

The algorithm initializes the tree τ to be an empty tree. Then, it assigns rank 0 to every cluster $C \in \mathcal{C}_y$. Next, in every cluster $C \in \mathcal{C}_y$, the algorithm picks an arbitrary leaf $w \in C$ of the tree T and designates it to serve as the *representative* of the cluster C. In addition, in every cluster $C \in \mathcal{C}_y$, the algorithm designates one of the ancestors u of the representative w that has level $\ell_T(u) = (y-1) \cdot d$ to serve as the *messenger* of the cluster C. Then the algorithm initializes the set $\mathcal{U}$ of *uncovered* clusters to be equal to $\mathcal{C}_y$.

After the algorithm finishs with the layer $j = 2, 3, \ldots, y$, it proceeds to layer $j-1$. Specifically, it passes over the clusters $C_{j-1} \in \mathcal{C}_{j-1}$ in an arbitrary order. For every cluster $C_{j-1} \in \mathcal{C}_{j-1}$ it does the following. Let $\mathcal{N} \subseteq \mathcal{U}$ be the subset of clusters C whose intersection with the cluster C_{j-1} contains the messenger u of the cluster C. For each cluster $C \in \mathcal{N}$, the algorithm adds the super-edge (C_{j-1}, C) into the tree τ, and sets C_{j-1} to be the parent of C in the tree τ. Let $\mathcal{R}$ be the set of ranks of all the children of the cluster C_{j-1}. Let $r = \max\{r' \in \mathcal{R}\}$. If there exists exactly one cluster $C \in \mathcal{N}$ with rank r, then the messenger w of the cluster C becomes the *representative* of the cluster C_{j-1}. In this case the algorithm assigns the cluster C_{j-1} rank r. Otherwise, an arbitrary cluster $C \in \mathcal{N}$ is chosen, and its messenger w becomes the representative of the cluster C_{j-1}. In this case the algorithm assigns the cluster C_{j-1} rank $r+1$. Now, the algorithm designates one of the ancestors $u \in C_{j-1}$ of the representative w of the cluster C_{j-1} that satisfies $\ell_T(u) = (j-2) \cdot d$ to serve as the messenger of the cluster C_{j-1}. Finally, the set $\mathcal{N}$ is removed from the set $\mathcal{U}$ (i.e., $\mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{N}$), and the algorithm enters into another iteration of the loop by picking another cluster of layer $j-1$.

Observe that this procedure indeed builds a tree τ rooted in the cluster $C(s)$, and this tree spans the entire collection $\mathcal{C}$ of clusters. Furthermore, note that τ is a BFS spanning tree of the super-graph $\tilde{G}$. Note also that for every cluster $C \in \mathcal{C}$, the rank of the cluster C in the tree τ is exactly equal to the rank that was assigned by this procedure. Note also that the messenger of the cluster $C(s) = S(s)$ is inevitably the source s.

Let $\tilde{E}_{\mathcal{F}}$ be the subset of super-edges $\tilde{e} = (C, C')$ of the tree τ that satisfy $rank(C) = rank(C')$. Note that the set $\tilde{E}_{\mathcal{F}}$ decomposes into a path-forest $\mathcal{F}$, and, furthermore, $\mathcal{F}$ is a semi-spanning path-forest. In addition, the path-forest $\mathcal{F}$ satisfies one more important property.

Lemma A.8 *Consider a path $\mathcal{P} \in \mathcal{F}$, $\mathcal{P} = (C_1, C_2, \ldots, C_k)$, where C_1 is the ancestor of C_k in the tree τ. For each index $i \in [k]$, let u_i and w_i denote the messenger and the representative of the cluster C_i, respectively, and let P_i denote the path between them in the tree T. Then the paths P_i can be concatenated to form a single path $P = P_1 \cdot P_2 \cdot \ldots \cdot P_k$ in the tree T, the vertex u_1 is an ancestor of the vertex w_k in the tree T, and $u_i = w_{i-1}$ for every index $i \in [k] \setminus \{1\}$.*

Note that the number of distinct subsets $S(v)$ is at most n, and therefore, $|\mathcal{C}| \le |\tilde{V}| \le n$, as well. Note also that for every cluster $C \in \mathcal{C}$, the path between the root cluster $C(s)$ and the cluster C in the tree τ contains at most $\log |\mathcal{C}| \le \log n$ non-edges of the forest $\mathcal{F}$.

Also, consider two clusters $C, C' \in \mathcal{C}_j$, for some index $j \in [y]$. Let u (resp., u') be the messenger of the cluster C (resp., C'), and w (resp., w') be its representative. Recall that $\mathcal{C}_j$ is an $(O(\log n), O(\log n))$-partition of the graph $\tilde{G}_j = (\tilde{V}_j, \tilde{E}_j)$, and $C, C' \in \tilde{V}_j$. Let χ_j be the coloring associated with this partition, and suppose that $\chi_j(C) = \chi_j(C')$. Note that it implies that $C \cap C' = \emptyset$, and, furthermore, there exists no edge $e = (z, z') \in E$, with $z \in C$, $z' \in C'$. Recall also that $|\{\chi_j(C) \mid C \in \mathcal{C}_j\}| \le \alpha = O(\log n)$.

A.3 Constructing the Schedule

Our algorithm for constructing the schedule is given as input the description of the layers $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_y$, the colorings $\chi_1, \chi_2, \ldots, \chi_y$ of the layers, the tree τ, the path-forest $\mathcal{F}$, and the messenger-representative pair of vertices for every cluster. All this is the output of the preprocessing algorithm.

The algorithm for constructing the schedule uses two major building blocks, namely, Procedures *Vertical_Broadcast* and *Horizontal_Broadcast*. These procedures are described in Sections A.4 and A.5. The two procedures are analogous to Procedures *Broadcast_Through* and *Broadcast_All* of [GM95]. However, our algorithm does not have an analogue to Procedure *Superlevel_to_Superlevel* of [GM95], because our preprocessing algorithm forms *overlapping layers*, and not disjoint ones as the algorithm of [GM95] does. Therefore, in our algorithm the delivery of the message between consequent layers is done via the boundary vertices that participate in both layers.

We remark that while our preprocessing algorithm and the analogous part of the algorithm of [GM95] have a similar nature, our algorithm for constructing the schedule is substantially different from the analogous part of the algorithm [GM95], which is called there the "main algorithm". The difference concerns mostly the way that different building blocks are combined into a single schedule. The main algorithm of [GM95] uses a round-robin technique of time sharing for running different procedures, and this way it ensures that there is no collisions between the executions of different procedures. This technique gives rise to a relatively simple analysis, but it carries a price of increasing the length of the schedule by a constant multiplicative factor. Our algorithm is geared to avoid this unnecessary multiplicative factor in order to achieve a *purely additive* approximation of the optimal schedule. To this end, our algorithm carefully combines different building blocks in such a manner that they cause no collision even though they employ *the same rounds.*

A.4 Procedure *Vertical_Broadcast*

The simplest setting in which Procedure *Vertical_Broadcast* can be invoked is on a cluster $C \in \mathcal{C}$, assuming that the messenger u of C is informed. In this case the procedure propagates the message along one of the shortest paths in the cluster C between the messenger u and the representative w of C. Observe that this process of propagating the message results in a schedule of length at most d.

The more elaborated, and by far more common, scenario in which Procedure *Vertical_Broadcast* is invoked is as follows. Let $\mathcal{S} = \{C^{(1)}, C^{(2)}, \ldots, C^{(k)}\} \subseteq \mathcal{C}_j$ be a collection of clusters of layer j, for some index $j \in [y]$. Let $u^{(i)}, w^{(i)} \in C^{(i)}$ be the messenger and the representative of the cluster $C^{(i)}$, respectively, for each index $i \in [k]$. Suppose that all the messengers $u^{(1)}, u^{(2)}, \ldots, u^{(k)}$ are informed. Procedure *Vertical_Broadcast*, if it is invoked in this setting, needs to propagate the message to the representatives $w^{(1)}, w^{(2)}, \ldots, w^{(k)}$. For each index $i \in [k]$, the procedure picks some shortest path $P^{(i)}$ between the messenger $u^{(i)}$ and the representative $w^{(i)}$. This path $P^{(i)}$, denoted also as $P(C^{(i)})$, will be henceforth referred to as the *chosen path* of the cluster $C^{(i)}$. Let χ_j be the coloring associated with layer j of the collection $\mathcal{C}$. Let $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_\alpha$ be the partition of the subset $\mathcal{S}$ into a disjoint union of color classes, that is, $\bigcup_{p=1}^{\alpha} \mathcal{S}_p = \mathcal{S}$, with $\mathcal{S}_p = \{C \in \mathcal{S} \mid \chi_j(C) = p\}$ for each index $p \in [\alpha]$. Note that some color classes $\mathcal{S}_p$ may be empty. Observe that for an index $p \in [\alpha]$, and for a pair of distinct clusters $C, C' \in \mathcal{S}_p$, necessarily $C \cap C' = \emptyset$, and, furthermore, the edge set E of the graph G contains no edge (z, z') with $z \in C$ and $z' \in C'$. In particular, the same property holds for the vertex sets V_P and $V_{P'}$ of the paths P and P' respectively. Recall that the number α of color classes is $O(\log n)$.

For a path $P = (u = u_1, u_2, \ldots, u_{d'} = w)$, $d' \leq d$, between the messenger u and the representative w of some cluster C, and an index $\ell \in [d']$, the vertex u_ℓ is called the *ℓth vertex* of the path P.

We next describe the structure of the schedule that Procedure *Vertical_Broadcast* forms whenever it is invoked in this scenario. On the first round of this schedule, the first vertices of the paths $P(C)$ for different clusters $C \in \mathcal{S}_1$ broadcast. (In a situation like this, when the broadcast of the vertex u_1 was originated as part of processing the cluster C, we say that the vertex u_1 *C-broadcasts*. For a vertex v that belongs to an intersection of two clusters C and C', in principle, it may happen that the vertex v both C- and C'-broadcasts in the same round. This, of course, would have the same effect as if it would be just C-broadcast, or just C'-broadcast, and the distinction between these situations is needed only for the analysis.)

More generally, for a round $\ell \in [d]$, on the ℓth round of the schedule that is constructed by Procedure *Vertical_Broadcast*, the ℓth vertices u_ℓ of the chosen paths $P = P(C)$ broadcast, for every cluster $C \in \mathcal{S}_1$. (If one or more of these paths are not long enough to have the ℓth vertex, the set of broadcasting vertices on round ℓ will not contain a vertex from this specific path; if none of these paths contains ℓth vertex, the round will be replaced by a *void* (that is, empty) round.) In addition, for a round $\ell = 4, 5, \ldots, d+3$, on the ℓth round of the schedule that is constructed by Procedure *Vertical_Broadcast*, the $(\ell-3)$rd vertices $u_{\ell-3}$ of the chosen paths $P(C)$ broadcast, for all the clusters $C \in \mathcal{S}_2$. More generally, for an index $p \in [\alpha]$, a round $\ell = 3(p-1)+1, 3(p-1)+2, \ldots, 3(p-1)+d$, on the ℓth round of this schedule the $(\ell - 3(p-1))$th vertices $u_{\ell-3(p-1)}$ of the chosen paths $P(C)$ broadcast, for all the clusters $C \in \mathcal{S}_p$.

This completes the description of the schedule that is constructed by Procedure *Vertical_Broadcast*. We next turn to its analysis.

Observe that the broadcast in clusters of the same color cannot possibly cause a collision. Consider a pair of clusters $C, C' \in \mathcal{C}_j$ with $\chi_j(C) = p$, $\chi_j(C') = q$, and $p \neq q$. Consider some round $\ell = 1, 2, \ldots, d + 3(\alpha - 1)$. Let $v \in C$, $v' \in C'$ be a pair of vertices that are scheduled to C-broadcast and C'-broadcast on round ℓ, respectively. It follows that v is the $(\ell - 3(p-1))$th vertex of the path $P(C)$, and that v' is the $(\ell - 3(q-1))$th vertex of the path $P(C')$. Recall that $C, C' \in \mathcal{C}_j$. It follows that $\ell_T(v) = (j-1)d + \ell - 3(p-1)$, and $\ell_T(v') = (j-1)d + \ell - 3(q-1)$, and so, since $p \neq q$ are both integer, $|\ell_T(v) - \ell_T(v')| \geq 3$. In other words, $|d_G(s,v) - d_G(s,v')| \geq 3$, and so $d_G(v,v') \geq 3$. Hence, the broadcasts of the vertices v and v' do not cause a collision, as required.

Hence, executions of Procedure *Vertical_Broadcast* do not cause collisions between vertices that participate in the same execution. The length of the schedule that is formed by an execution of Procedure *Vertical_Broadcast* is $\tilde{d} = d + 3(\alpha - 1)$. Note that by padding the schedules with as many void rounds as required we guarantee that every execution of Procedure *Vertical_Broadcast* forms a schedule of length precisely $\tilde{d}$. This homogenity of the lengths of schedules is crucial for our analysis. Note also that our description of Procedure *Vertical_Broadcast* assumes that all the clusters in which the broadcast should be conducted are known in advance, and no cluster ever joins "on the fly". In our analysis we will show that our algorithm invokes Procedure *Vertical_Broadcast* in such a way that this property is indeed satisfied.

We also remark that all the procedures that our algorithm employs accept as input an additional input parameter, called the *starting round*. This parameter is a positive integer number that determines the index of the first non-empty round of the schedule that the procedure should construct. In other words, if according to the above description Procedure *Vertical_Broadcast* is supposed to return a schedule $\Pi = (R_1, R_2, \ldots, R_t)$ on a certain input, then whenever it is invoked on the same input with a starting round parameter t' it would return the schedule $\epsilon_{t'-1} \cdot \Pi$, where $\epsilon_{t'-1}$ stands for a schedule that contains $t'-1$ void rounds, and $\cdot$ stands for concatenation. For an execution χ of one of the procedures that are used by our algorithm, its starting round will be denoted by $I(\chi)$.

A.5 Procedure *Horizontal_Broadcast*

Consider a collection of clusters $\mathcal{S} = \{C^{(1)}, C^{(2)}, \ldots, C^{(k))}\} \subseteq \mathcal{C}_j$, where $j \in [y]$ is an index. Suppose that all the representatives $w^{(1)}, w^{(2)}, \ldots, w^{(k)}$ of these clusters are informed, and the goal is to inform all the vertices of the set $\bigcup_{C \in \mathcal{S}} C$. This is done using Algorithm *Anonymous_Broadcast* due to Bar-Yehuda et al. [BGI86]. Specifically, we use the following result.

Theorem A.9 *[BGI86, CW87] (see also [GM95]) Given an n-vertex graph $\hat{G} = (\hat{V}, \hat{E})$, and a subset $\hat{U} \subseteq \hat{V}$ of informed vertices, there exists a schedule of length $K \cdot (D \log n + \log^2 n)$ that delivers the message to all the vertices of the graph, where K is a universal constant and $D = \max_{v \in V} \min_{u \in U} d_G(u, v)$. Furthermore, this schedule can be computed in randomized polynomial time. In deterministic polynomial time a schedule of length $KD \log^2 n$ can be computed, and the latter schedule also delivers the message from the informed vertices to all the other vertices of the graph.*

Observe that for a cluster $C \in \mathcal{S}$, its diameter in the super-graph $\tilde{G}$ is $O(\log n)$, and each vertex $S(v)$ from which the super-vertices of $\tilde{G}$ were formed is itself a subset of diameter at most $2d$ (in the graph G). Hence, $D \leq \max\{Diam_G(C) \mid C \in \mathcal{S}\} = O(d \log n)$, and so the length of the schedule that is constructed by Algorithm *Anonymous_Broadcast* is $O(d \log^2 n)$. Let c denote the universal constant hidden by the O-notation. For the sake of homogenity of the lengths of schedules constructed by Procedure *Horizontal_Broadcast*, whenever the resulting schedule is of length $L < cd \log^2 n$, it is padded with $cd \log^2 n - L$ empty rounds. By Theorem A.9, this schedule informs all the vertices of the set $\bigcup_{C \in \mathcal{S}} C$.

To prevent collisions between simultaneous executions of Procedure *Horizontal_Broadcast* in adjacent layers we modify the procedure in the following way. Whenever the input collection of clusters $\mathcal{S}$ is contained in an *even* (resp., *odd*) layer $\mathcal{C}_j$ (i.e., its index j is even (resp., odd)), the procedure constructs a schedule that employs only even (resp., odd) rounds. The odd (resp., even) rounds are left empty. This raises the length of the schedules constructed by Procedure *Horizontal_Broadcast* to $2c \cdot d \log n$. Let a denote the constant $2c$. (Note that this is an example of using the round-robin time sharing technique. In the algorithm of [GM95] it is used also for avoiding collisions between executions of different procedures, and between different executions of the same procedure that accept as input collections of clusters of the same layer.)

A.6 Procedure *Period*

In this section we combine Procedures *Vertical_Broadcast* and *Horizontal_Broadcast* described in the previous sections, and describe Procedure *Period*, which is the major component of our algorithm for constructing the schedule.

This procedure is invoked in the same scenario as Procedure *Vertical_Broadcast*, i.e., when it is given a collection $\mathcal{S} \subseteq \mathcal{C}_j$ of clusters of layer j, for some index $j \in [y]$, whose messengers are all informed. Let I denote its starting round parameter.

Procedure *Period* starts with running Procedure *Vertical_Broadcast* on the collection $\mathcal{S}$ with the same starting round parameter I. Let Π_{VL} denote the resulting schedule. Then it forms $\tilde{d}$ void rounds, and then it runs Procedure *Horizontal_Broadcast* on the same collection $\mathcal{S}$, this time with starting round parameter $I + 2\tilde{d}$. Let Π_{HL} denote the resulting schedule. (Note that at this point all the representatives of the clusters $C \in \mathcal{S}$ are informed.) Finally, it forms $2\tilde{d}$ additional void rounds. In addition, Procedure *Period* invokes itself recursively twice, as will be described below.

Note that Procedure *Period* constructs a schedule of length $4\tilde{d} + a \cdot d \log^2 n$ that delivers the message from the set of the messengers of the clusters $C \in \mathcal{S}$ to all the vertices of the set $\bigcup_{C \in \mathcal{S}} C$.

Let $\mathcal{S}'' \subseteq \mathcal{S}$ be a collection of clusters $C \in \mathcal{C}_j$ that have a child (with respect to the tree τ) C' such that the super-edge (C, C') belongs to the path-forest $\mathcal{F}$. Let $\mathcal{S}' = \{C' \mid \mathit{parent}_\tau(C') \in \mathcal{S}''\}$. Note that $\mathcal{S}' \subseteq \mathcal{C}_{j+1}$. Let W'' be the set of representatives of the clusters from the collection $\mathcal{S}''$. Observe that, by construction, each representative $w \in W''$ of a cluster $C \in \mathcal{S}''$ is the messenger of the cluster $C' \in \mathcal{S}'$ that satisfies $\mathit{parent}_\tau(C') = C$. It follows that after the schedule that is returned by the invocation of Procedure *Vertical_Broadcast* terminates, all the messengers of the clusters $C' \in \mathcal{S}'$ are informed.

Procedure *Period* invokes itself recursively on the collection $\mathcal{S}' \subseteq \mathcal{C}_{j+1}$, and sets the starting round parameter of this recursive invocation to be $I + \tilde{d}$.

Let $\hat{\mathcal{S}}$ be the collection of clusters of layer $j + 1$ that do not belong to the collection $\mathcal{S}'$, and whose messengers are informed by the schedule that is constructed by the execution of Procedure *Horizontal_Broadcast* on input $\mathcal{S}$. The last step of Procedure *Period* is to invoke itself recursively on the collection $\hat{\mathcal{S}} \subseteq \mathcal{C}_{j+1}$ with starting round parameter $I + 4\tilde{d} + a \cdot d \log^2 n$.

Note that one of the collections $\hat{\mathcal{S}}$, $\mathcal{S}'$, or both of them, could be empty. This would lead to an invocation of Procedure *Period* on an empty collection; in this case it returns an empty schedule. This is, in fact, the termination condition of the recursion.

After the recursive invocations are over, the original execution of Procedure *Period* has three schedules Π_0, Π_1 and Π_2. The schedule Π_0 is obtained by processing the collection $\mathcal{S}$ directly, that is, invoking on it Procedures *Vertical_Broadcast* and *Horizontal_Broadcast* and concatenating them in the way that was described above (along with introducing the void rounds). The schedules Π_1 and Π_2 are returned by the first and the second recursive invocations of Procedure *Period*, respectively (on collections $\mathcal{S}'$ and $\hat{\mathcal{S}}$, respectively). This three schedules are unified into a single schedule Π that the execution returns in a straightforward way. Specifically, let $L = \max\{|\Pi_0|, |\Pi_1|, |\Pi_2|\}$, and let $L_i = |\Pi_i|$, for $i \in \{0,1,2\}$. For each index $i \in \{0,1,2\}$ such that $L_i < L$, pad the schedule Π_i with $L - L_i$ void rounds. For $i \in \{0,1,2\}$, let Π'_i denote the obtained schedules. Now set Π to be the *round-wise* union of the schedules Π'_0, Π'_1, and Π'_2. I.e., let $\Pi'_i = (R_1^{(i)}, R_2^{(i)}, \ldots, R_L^{(i)})$, $i \in \{0,1,2\}$. Then $\Pi = (\bigcup_{i=0}^2 R_1^{(i)}, \bigcup_{i=0}^2 R_2^{(i)}, \ldots, \bigcup_{i=0}^2 R_L^{(i)})$ is the schedule returned by the execution. (Henceforth we refer to this operation of unifying schedules of not necessarily equal lengths as *schedule-union*. In this terminology, the schedule Π is just the schedule-union of schedules Π_0, Π_1 and Π_2. Also, the schedule Π_0 is the schedule-union of the schedules Π_{VL} and Π_{HL}.)

A.7 The Combined Algorithm

Consider the execution of Procedure *Period* on the input collection that contains only the single cluster $C(s) \in \mathcal{C}_1$ with starting round parameter equal to 1. We next modify the schedule Π that is obtained by this invocation to construct the resulting schedule Π^* of our algorithm.

Let Φ be the set of all the recursive invocations of Procedure *Period* that were originated directly or indirectly by the execution of Procedure *Period* on the input collection $\{C(s)\}$. Let E_Φ be the set of pairs of executions $(\varphi, \varphi') \in \Phi^2$, $\varphi \neq \varphi'$, such that either the execution φ recursively invokes the execution φ' or vice versa. Observe that the graph $T_\Phi = (\Phi, E_\Phi)$ is a binary tree. Consider an internal vertex (that is, not a leaf) $\varphi \in \Phi$ of this tree. This vertex corresponds to an execution of Procedure *Period* on an input collection $\mathcal{S} \subseteq \mathcal{C}_j$ of clusters, for some index $j \in [y-1]$. Since φ is not a leaf of the tree T_Φ, it follows that φ has two children in T_Φ. These children correspond to the executions φ' and $\hat{\varphi}$ of Procedure *Period* on input collections $\mathcal{S}'$ and $\hat{\mathcal{S}}$, respectively. The execution φ' (resp., $\hat{\varphi}$) will be referred as the *first* (resp., *second*) *child-invocation of the execution* φ.

For a vertex v that belongs to a cluster C of layer 1 or 2, its behavior in the schedule Π^* is identical to its behavior in the schedule Π. For a layer $j = 3, 4, \ldots, y$, let $\{\varphi^{(1)}, \varphi^{(2)}, \ldots, \varphi^{(k)}\} = \Phi(j)$ be the set of all different recursive invocations of Procedure

Period on collections $\mathcal{S} \subseteq \mathcal{C}_j$ of clusters of layer j. In other words, $\Phi(j)$ is the set of vertices φ of the tree T_Φ that have level j, i.e., satisfy $\ell_{T_\Phi}(\varphi) = j$.

Recall that each such execution φ has an attached starting round parameter. For a positive integer $\ell = 1, 2, \ldots$, let Φ_ℓ be the subset of the set Φ that contains all the invocations with starting rounds equal to ℓ. For each non-empty subset Φ_ℓ, let $\mathcal{S}'_\ell = \bigcup\{\mathcal{S} \mid \mathcal{S}$ is the input collection of $\varphi \in \Phi_\ell\}$. Let φ'_ℓ be the execution of Procedure *Period* on the input collection $\mathcal{S}'_\ell$ with starting round parameter equal to ℓ. Now the algorithm replaces the partial schedules that were obtained by running the executions $\varphi^{(1)}, \varphi^{(2)}, \ldots, \varphi^{(k)}$ by the partial schedules obtained by running the executions $\varphi'_1, \varphi'_2, \ldots$ (only those of them for which $\mathcal{S}'_\ell \neq \emptyset$).

The last step of the algorithm is to find the smallest index t such that the first t rounds of the schedule inform all the vertices of the graph, and to remove all the rounds with index greater than t from the schedule. This completes the description of our algorithm.

The following lemma shows that the executions φ'_ℓ are valid executions of Procedure *Period*.

Lemma A.10 *All the executions φ'_ℓ are within the framework of the prescribed scenario for executing Procedure Period.*

Proof: Observe that for every execution $\varphi \in \Phi_\ell$, its starting round is ℓ, and its input collection is $\mathcal{S} \subseteq \mathcal{C}_j$, where j is the index such that $\varphi \in \Phi(j)$. Since the execution φ is within the framework of the prescribed scenario for executing Procedure *Period*, it follows that all the messengers of the cluster $C \in \mathcal{S}$ are informed before the starting round ℓ of the exectuion φ.

Hence, all the messengers of the clusters $C \in \mathcal{S}'_\ell$ are informed before the beginning of the starting round ℓ of the execution φ'_ℓ. ▮

Lemma A.11 *The running time of the algorithm is polynomial in n.*

Proof: Consider the child-invocations φ' and $\hat{\varphi}$ of an execution φ of Procedure *Period* on a collection $\mathcal{S} \subseteq \mathcal{C}_j$, for some index $j \in [y-1]$. Consider the input collections $\mathcal{S}', \hat{\mathcal{S}} \subseteq \mathcal{C}_{j+1}$ of the executions φ' and $\hat{\varphi}$, respectively, of Procedure *Period*. Note that by construction, $\mathcal{S}' \cap \hat{\mathcal{S}} = \emptyset$.

A straightforward induction on the level of the specific cluster C in the tree τ shows that every cluster $C \in \mathcal{C} = \bigcup_{j=1}^{y} \mathcal{C}_j$, there is at most one execution φ of Procedure *Period* whose input collection $\mathcal{S}$ contains the cluster C. Therefore, the overall number of (recursive) invocations of Procedure *Period* throughout the execution *Period*$(\{C(s)\})$ is at most $|\mathcal{C}|$.

Since every cluster C of $\mathcal{C}$ contains at least one vertex that belongs to no other cluster $C' \in \mathcal{C}$ (recall that $d \geq 3$), it follows that $|\mathcal{C}| \leq n$, and so at most n recursive invocations of Procedure *Period* occur during the execution of the algorithm. Since every invocation of Procedure *Period* (disregarding its child-invocations) requires running time at most polynomial in n, it follows that the running time of the entire algorithm is polynomial in n as well. (Notice also that the step of our algorithm that merges some of the executions of Procedure *Period* can also be carried through in polynomial time.) ▮

A.8 Analysis

In this section we argue that our algorithm constructs a short broadcast schedule that informs all the vertices of the input graph G.

We start with the following lemma that is proven by induction on the indices $j \in [y]$ of layers $\mathcal{C}_j$ of the super-graph $\tilde{G}$.

Lemma A.12 *The starting rounds of executions φ and γ of Procedures Period and Vertical_Broadcast, respectively, on input collections $\mathcal{S} \subseteq \mathcal{C}_j$ are of the form*

$$I(\varphi) \;=\; I(\gamma) \;=\; 1 + \tilde{d} \cdot (j-1) + h(3 \cdot \tilde{d} + a \cdot d \log^2 n) \;,$$

for some positive integer $h \in [j-1]$. The starting rounds of executions δ of Procedure Horizontal_Broadcast on such a collection are of the form

$$I(\delta) \;=\; 1 + \tilde{d} \cdot (j+1) + h(3 \cdot \tilde{d} + a \cdot d \log^2 n) \;,$$

for some positive integer $h \in [j-1]$.

Proof:

The induction base: For layer $j = 1$, the only execution of Procedure *Vertical_Broadcast* (on the input collection $\{C(s)\}$) has starting round 1, and the same is true for Procedure *Period*. The only execution of Procedure *Horizontal_Broadcast* (on the same singleton input collection) starts at round $1 + 2\tilde{d}$, proving the induction base.

The induction step: For an index $j \in [y]$, let $Period_j$ (resp., $Vertical\_Broadcast_j$; $Horizontal\_Broadcast_j$) denote the set of all the executions of Procedure *Period* (resp., *Vertical_Broadcast*; *Horizontal_Broadcast*) on different collections $\mathcal{S} \subseteq \mathcal{C}_j$ that occur during the algorithm.

For an execution $\varphi \in Period_j$, let $\gamma(\varphi)$ (resp., $\delta(\varphi)$) denote the execution of Procedure *Vertical_Broadcast* (resp., *Horizontal_Broadcast*) that is invoked by φ. Let φ' and $\hat{\varphi}$ denote the first and second child-invocations of the execution φ (note that $\varphi', \hat{\varphi} \in Period_{j+1}$).

By the induction hypothesis, since $\varphi \in \mathit{Period}_j$,

$$I(\varphi) = 1 + \tilde{d} \cdot (j-1) + h(\varphi) \cdot (3 \cdot \tilde{d} + a \cdot d \log^2 n) \ ,$$

where $h(\varphi) \in [j-1]$. Also, by construction,

$$\begin{aligned}
I(\varphi') &= I(\varphi) + \tilde{d} \ , \ I(\hat{\varphi}) \ = \ I(\varphi) + 4\tilde{d} + a \cdot d \log^2 n \ , \\
I(\gamma(\varphi)) &= I(\varphi) \ , \ I(\delta(\varphi)) \ = \ I(\varphi) + 2 \cdot \tilde{d} \ , \\
I(\gamma(\varphi')) &= I(\varphi') \ , \ I(\delta(\varphi')) \ = \ I(\varphi') + 2 \cdot \tilde{d} \ , \\
I(\gamma(\hat{\varphi})) &= I(\hat{\varphi}) \ , \ I(\delta(\hat{\varphi})) \ = \ I(\hat{\varphi}) + 2 \cdot \tilde{d} \ .
\end{aligned}$$

Observe that $\varphi \in \mathit{Period}_j$, $\varphi', \hat{\varphi} \in \mathit{Period}_{j+1}$, $\gamma(\varphi'), \gamma(\hat{\varphi}) \in \mathit{Vertical\_Broadcast}_{j+1}$, and $\delta(\varphi'), \delta(\hat{\varphi}) \in \mathit{Horizontal\_Broadcast}_{j+1}$. Hence,

$$\begin{aligned}
I(\gamma(\varphi')) &= I(\varphi') \ = \ I(\varphi) + \tilde{d} \\
&= 1 + \tilde{d} \cdot j + h(\varphi) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ ,
\end{aligned}$$

with $h(\varphi) \in [j-1] \subseteq [j]$.

Analogously,

$$I(\gamma(\hat{\varphi})) \ = \ I(\hat{\varphi}) \ = \ I(\varphi) + 4\tilde{d} + a \cdot d \log^2 n \ = \ 1 + \tilde{d} \cdot j + h(\gamma(\hat{\varphi})) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ ,$$

with $h(\gamma(\hat{\varphi})) = h(\varphi) + 1$. Since $h(\varphi) \in [j-1]$, it follows that $h(\gamma(\hat{\varphi})) \in [j]$.

Also,

$$\begin{aligned}
I(\delta(\varphi')) &= I(\varphi') + 2\tilde{d} \ = \ I(\varphi) + 3\tilde{d} \\
&= 1 + \tilde{d} \cdot (j+2) + h(\varphi) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ ,
\end{aligned}$$

and $h(\varphi) = h(\delta(\varphi)) \in [j-1] \subseteq [j]$.

Finally,

$$\begin{aligned}
I(\delta(\hat{\varphi})) &= I(\hat{\varphi}) + 2\tilde{d} \ = \ I(\varphi) + 6\tilde{d} + a \cdot d \log^2 n \\
&= 1 + \tilde{d} \cdot (j+2) + (h(\varphi) + 1) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ ,
\end{aligned}$$

and so $h(\delta(\hat{\varphi})) = h(\varphi) + 1 \in [j]$, as required.

To summarize, $I(\gamma(\varphi'))$, $I(\gamma(\hat{\varphi}))$, $I(\delta(\varphi'))$ and $I(\delta(\hat{\varphi}))$ are all of required form. (Note that $\gamma(\varphi'), \gamma(\hat{\varphi}) \in \mathit{Vertical\_Broadcast}_{j+1}$, and $\delta(\varphi'), \delta(\hat{\varphi}) \in \mathit{Horizontal\_Broadcast}_{j+1}$, and the required forms for the staring rounds of executions of Procedures *Vertical_Broadcast* and *Horizontal_Broadcast* are different.) ∎

Notation A.13 *For an execution χ of one of the procedures that are used by our algorithm, let $\Pi(\chi)$ denote the schedule returned by this procedure.*

Definition A.14 *Consider two executions χ_1, χ_2 of procedures that are used in our algorithm. These two executions are said to* interfere one with another *if there exists an index $\ell = 1, 2, \ldots$, and a pair of (not necessarily distinct) vertices $u, w \in V$ at distance at most 2 one from another in the graph $G = (V, E)$, such that the vertex u is scheduled to broadcast on round ℓ by the schedule $\Pi(\chi_1)$ and the vertex w is scheduled to broadcast on round ℓ by the schedule $\Pi(\chi_2)$.*

Lemma A.15 *Different executions of Procedures Vertical_Broadcast and Horizontal_Broadcast do not interfere one to another.*

Proof: Consider two executions χ_1 and χ_2. For $i \in \{1, 2\}$, let j_i be the *layer* of the execution χ_i, that is, the index that satisfies $\chi_i \in \mathit{Vertical\_Broadcast}_{j_i} \cup \mathit{Horizontal\_Broadcast}_{j_i}$. Observe that if $|j_1 - j_2| \geq 2$ then the executions χ_1 and χ_2 cannot interfere one to another. (Recall that $d \geq 3$.) Hence, from now on we assume that either the layers of the executions χ_1 and χ_2, denoted $\ell(\chi_1)$ and $\ell(\chi_2)$, are equal, or they differ by 1.

The discussion splits into two cases. The first case is $\ell(\chi_1) = \ell(\chi_2) = j$, for some index $j \in [y]$, and the second case is $\ell(\chi_1) = j$, $\ell(\chi_2) = j + 1$ (without loss of generality), for some index $j \in [y - 1]$.

1. ($\ell(\chi_1) = \ell(\chi_2) = j$)
   The discussion splits again, this time into three cases, depending on whether χ_1, χ_2 are both executions of Procedure *Vertical_Broadcast*, or both of them are executions of Procedure *Horizontal_Broadcast*, or one of them is an execution of Procedure *Vertical_Broadcast* and the other is an execution of Procedure *Horizontal_Broadcast*.

   (a) ($\chi_1, \chi_2 \in \mathit{Vertical\_Broadcast}_j$)
   In this case the starting rounds $I(\chi_1)$, $I(\chi_2)$ of the two executions satisfy

   $$\begin{aligned} I(\chi_1) &= 1 + \tilde{d} \cdot (j-1) + h(\chi_1) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ , \\ I(\chi_2) &= 1 + \tilde{d} \cdot (j-1) + h(\chi_2) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ , \end{aligned}$$

   where $h(\chi_1), h(\chi_2) \in [j-1]$.
   If $h(\chi_1) = h(\chi_2)$ then $I(\chi_1) = I(\chi_2)$, but by construction, this cannot happen to two distinct executions of the same procedure that have input collections of the same layer. (Recall that in this case the input collections $\mathcal{S}(\chi_1), \mathcal{S}(\chi_2) \in \mathcal{C}_j$ of these two executions would be merged into a single collection, and Procedure

Vertical_Broadcast would be invoked on a unified collection that would contain the set $\mathcal{S}(\chi_1) \cup \mathcal{S}(\chi_2)$.)

If $h(\chi_1) \neq h(\chi_2)$ then since both $h(\chi_1)$ and $h(\chi_2)$ are integer, it follows that $|I(\chi_1) - I(\chi_2)| \geq 3\tilde{d} + a \cdot d \log^2 n$. Suppose, without loss of generality, that $I(\chi_1) < I(\chi_2)$. The length of the schedule $\Pi(\chi_1)$ that is constructed by the execution χ_1 is $\tilde{d} < I(\chi_2) - I(\chi_1)$, and so, the executions χ_1 and χ_2 do not interfere one to another.

(b) ($\chi_1, \chi_2 \in$ *Horizontal_Broadcast*$_j$)

In this case the starting rounds of the two executions satisfy

$$\begin{aligned} I(\chi_1) &= 1 + \tilde{d} \cdot (j+1) + h(\chi_1) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ , \\ I(\chi_2) &= 1 + \tilde{d} \cdot (j+1) + h(\chi_2) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ , \end{aligned}$$

where $h(\chi_1), h(\chi_2) \in [j-1]$.

If $h(\chi_1) = h(\chi_2)$ then $I(\chi_1) = I(\chi_2)$, and by the same argument as in Case 1a, this implies that $\chi_1 = \chi_2$, and we are done.

If $h(\chi_1) \neq h(\chi_2)$ then $|I(\chi_1) - I(\chi_2)| \geq 3\tilde{d} + a \cdot d \log^2 n$. Since $\max\{|\Pi(\chi_1)|, |\Pi(\chi_2)|\} = a \cdot d \log^2 n < |I(\chi_1) - I(\chi_2)|$, the executions χ_1 and χ_2 would not interfere one to another in this case as well.

(c) ($\chi_1 \in$ *Vertical_Broadcast*$_j$, $\chi_2 \in$ *Horizontal_Broadcast*$_j$)

In this case

$$\begin{aligned} I(\chi_1) &= 1 + \tilde{d} \cdot (j-1) + h(\chi_1) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ , \\ I(\chi_2) &= 1 + \tilde{d} \cdot (j+1) + h(\chi_2) \cdot (3\tilde{d} + a \cdot d \log^2 n) \ , \end{aligned}$$

where $h(\chi_1), h(\chi_2) \in [j-1]$.

If $h(\chi_1) = h(\chi_2)$ then $I(\chi_2) - I(\chi_1) = 2\tilde{d}$. The length of the schedule $\Pi(\chi_1)$ that is constructed by the execution χ_1 is $\tilde{d} < I(\chi_2) - I(\chi_1)$, and so, in this subcase we are done.

If $h(\chi_1) < h(\chi_2)$ then $I(\chi_1) < I(\chi_2)$ too, and the difference $I(\chi_2) - I(\chi_1)$ is even greater than in the subcase $h(\chi_1) = h(\chi_2)$.

Finally, consider the subcase $h(\chi_1) > h(\chi_2)$. Since $h(\chi_1)$ and $h(\chi_2)$ are both integer, it follows that

$$\begin{aligned} I(\chi_1) - I(\chi_2) &= (h(\chi_1) - h(\chi_2)) \cdot (3\tilde{d} + a \cdot d \log^2 n) - 2\tilde{d} \\ &\geq \tilde{d} + a \cdot d \log^2 n \ > \ 0 \ , \end{aligned}$$

and so, the starting round of the execution χ_2 is at least by $\tilde{d}+a\cdot d\log^2 n$ smaller than the starting round of the execution χ_1. The length of the schedule $\Pi(\chi_2)$ is $a\cdot d\log^2 n$, and so, this schedule terminates $\tilde{d}$ rounds before the starting round of the execution χ_1, implying that the two executions do not interfere one to another.

2. $(\ell(\chi_1)=j,\ \ell(\chi_2)=j+1)$

   (a) $(\chi_1 \in \mathit{Vertical\_Broadcast}_j,\ \chi_2 \in \mathit{Vertical\_Broadcast}_{j+1})$
   In this case

$$\begin{aligned} I(\chi_1) &= 1+\tilde{d}\cdot(j-1)+h(\chi_1)\cdot(3\tilde{d}+a\cdot d\log^2 n)\ ,\\ I(\chi_2) &= 1+\tilde{d}\cdot j+h(\chi_2)\cdot(3\tilde{d}+a\cdot d\log^2 n)\ , \end{aligned}$$

   where $h(\chi_1)\in[j-1]$, $h(\chi_2)\in[j]$.
   If $h(\chi_1)=h(\chi_2)$ then $I(\chi_2)-I(\chi_1)=\tilde{d}$, which is precisely the length of the schedule $\Pi(\chi_1)$, and so, the schedule $\Pi(\chi_1)$ terminates exactly one round before the starting round of the execution χ_2.
   If $h(\chi_1)\neq h(\chi_2)$ then

$$|I(\chi_2)-I(\chi_1)|\ \geq\ 2\tilde{d}+a\cdot d\log^2 n\ \geq\ \max\{|\Pi(\chi_1)|,|\Pi(\chi_2)|\}=\tilde{d}\ .$$

   and, therefore, the two executions do not interfere one to another.

   (b) $(\chi_1 \in \mathit{Vertical\_Broadcast}_j,\ \chi_2 \in \mathit{Horizontal\_Broadcast}_{j+1})$
   In this case

$$\begin{aligned} I(\chi_1) &= 1+\tilde{d}\cdot(j-1)+h(\chi_1)\cdot(3\tilde{d}+a\cdot d\log^2 n)\ ,\\ I(\chi_2) &= 1+\tilde{d}\cdot(j+2)+h(\chi_2)\cdot(3\tilde{d}+a\cdot d\log^2 n)\ , \end{aligned}$$

   where $h(\chi_1)\in[j-1]$, $h(\chi_2)\in[j]$.
   If $h(\chi_1)=h(\chi_2)$ then $I(\chi_2)-I(\chi_1)=3\tilde{d}>0$, and since $|\Pi(\chi_1)|=\tilde{d}$, and $I(\chi_1)<I(\chi_2)$, we are done.
   If $h(\chi_1)\neq h(\chi_2)$ then $|I(\chi_2)-I(\chi_1)|\geq a\cdot d\log^2 n$, and since $\max\{|\Pi(\chi_1)|,|\Pi(\chi_2)|\}=a\cdot d\log^2 n$, the schedule with smaller starting round will terminate at least one round before the beginning of the schedule with the larger starting round.

   (c) $(\chi_1 \in \mathit{Vertical\_Broadcast}_{j+1},\ \chi_2 \in \mathit{Horizontal\_Broadcast}_j)$
   In this case

$$\begin{aligned} I(\chi_1) &= 1+\tilde{d}\cdot j+h(\chi_1)\cdot(3\tilde{d}+a\cdot d\log^2 n)\ ,\\ I(\chi_2) &= 1+\tilde{d}\cdot(j+1)+h(\chi_2)\cdot(3\tilde{d}+a\cdot d\log^2 n)\ , \end{aligned}$$

where $h(\chi_1) \in [j]$, $h(\chi_2) \in [j-1]$.
If $h(\chi_1) = h(\chi_2)$ then $I(\chi_2) = I(\chi_1) + \tilde{d}$, and, in particular, $I(\chi_1) < I(\chi_2)$. Since $|\Pi(\chi_1)| = \tilde{d}$, the schedule $\Pi(\chi_1)$ terminates at least one round before $I(\chi_2)$, that is, the round when the schedule $\Pi(\chi_2)$ starts.
If $h(\chi_1) \neq h(\chi_2)$ then $|I(\chi_1) - I(\chi_2)| \geq 2\tilde{d} + a \cdot d \log^2 n \geq \max\{|\Pi(\chi_1)|, |\Pi(\chi_2)|\}$, and we are done.

(d) ($\chi_1 \in \mathit{Horizontal\_Broadcast}_j$, $\chi_2 \in \mathit{Horizontal\_Broadcast}_{j+1}$)
Recall that the schedule that is constructed by an execution $\delta \in \mathit{Horizontal\_Broadcast}_j$ employs only rounds ℓ that satisfy $\ell \equiv j \pmod 2$. Hence the execution χ_1 employs only rounds with even indices and the execution χ_2 employs only rounds with odd indices, or vice versa. Hence, the two executions do not interfere one to another.

▮

Lemma A.16 *The length of the schedule Π^* that is constructed by our algorithm is at most $Rad + O(d \cdot \log^3 n + x \cdot \log n)$, and it informs all the vertices of the graph.*

Proof: Consider a vertex $v \in V$. Let $C \in \mathcal{C}$ be one of the clusters that contain the vertex v. Let $\mathcal{P}$ be the (unique) path between the root cluster $C(s)$ and the cluster C in the tree τ. Observe that $\mathcal{P}$ contains at most y super-vertices (clusters), and, therefore, at most x super-edges. Recall that since $\mathcal{F}$ is a *semi-spanning* path-forest, the path $\mathcal{P}$ contains at most $\log n$ super-edges that do not belong to $\mathcal{F}$ (henceforth termed as *non-edges* of $\mathcal{F}$).

Consider the unique representation of the path $\mathcal{P}$ as the concatenation $\mathcal{P} = \mathcal{P}_1 \cdot \tilde{e}_1 \cdot \mathcal{P}_2 \cdot \tilde{e}_2 \cdot \ldots \cdot \mathcal{P}_r \cdot \tilde{e}_r \cdot \mathcal{P}_{r+1}$, where $r \in [\log n]$, and for each index $i \in [r]$, $\mathcal{P}_i$ is a (possibly empty) path path that is contained in one of the paths of the forest $\mathcal{F}$, and the super-edges $\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_r$ are non-edges of the path-forest $\mathcal{F}$. Also, the cluster $C(s)$ is one of the endpoints of the path $\mathcal{P}_1$, if $\mathcal{P}_1 \neq \emptyset$, and if $\mathcal{P}_1 = \emptyset$ then $C(s)$ is one of the endpoints of the super-edge $\tilde{e}_1$. Analogously, the cluster C is one of the endpoints of the path $\mathcal{P}_{r+1}$ if the latter is not empty, and if it is empty that the cluster C is one of the endpoints of the super-edge $\tilde{e}_r$.

For each index $i \in [r+1]$, let $p_i = |\mathcal{P}_i|$. Let $\mathcal{P}_i = (C_1^{(i)}, C_2^{(i)}, \ldots, C_{p_i}^{(i)})$, and for each index $j \in [p_i]$, let u_j, w_j denote the messenger and the representative of the cluster $C_j^{(i)}$, respectively. Observe that by Lemma A.8, $w_1 = u_2$, $w_w = u_3$, $\ldots, w_{p_i - 1} = u_{p_i}$, and the vertex w_{p_i} is a descendent of the vertex u_1 with respect to the BFS spanning tree T of the graph G.

Consider an invocation of Procedure *Period* on some collection $\mathcal{S}$ that contains the cluster $C_1^{(i)}$. Suppose that this invocation has starting round parameter I. By construction, the procedure invokes itself recursively with starting round parameter $I+\tilde{d}$ rounds on a collection $\mathcal{S}'$ of clusters that contains the cluster $C_2^{(i)}$. The latter recursive invocation again invokes itself recursively with starting round parameter $I + 2\tilde{d}$ on a collection $\mathcal{S}''$ of clusters that contains the cluster $C_3^{(i)}$, etc. To summarize, this line of recursive invocations constructs a partial schedule of length $p_i \cdot \tilde{d}$ that delivers the message from the messenger u_1 of the cluster $C_1^{(i)}$ to the messenger u_{p_i} of the cluster $C_{p_i}^{(i)}$.

For each non-edge $\tilde{e}_i = (C', C'')$, $i \in [r]$, of the path-forest $\mathcal{F}$, the message is delivered from the messenger of the cluster C' to all the other vertices of this cluster (including the messenger of the cluster C'' that belongs to the intersection $C' \cap C''$) via a schedule that is constructed by an execution of Procedure *Period* on a collection $\mathcal{S}$ of clusters that contains the cluster C'.

Procedure *Period* starts with invoking Procedure *Vertical_Broadcast* on the same input collection $\mathcal{S}$. This invocation constructs a schedule of length $\tilde{d}$ that delivers the message from the messenger u' of the cluster C' to its representative w'. Then this schedule is padded with $\tilde{d}$ empty rounds, and then Procedure *Period* concatenates the resulting schedule with the schedule that is returned by the invocation of Procedure *Horizontal_Broadcast* on the same input collection $\mathcal{S}$ (but with starting round parameter greater by $2\tilde{d}$ than the one that Procedure *Period* accepted as a part of its input). The latter schedule is of length $a \cdot d \log^2 n$, and, in particular, it delivers the message from the representative w' to all the other vertices of the cluster C' including the messenger u'' of the cluster C''. Finally, this schedule is padded with extra $2\tilde{d}$ empty rounds.

To summarize, for each non-edge $\tilde{e}_i = (C', C'')$, $i \in [r]$, of the path-forest $\mathcal{F}$, Procedure *Period* constructs a schedule of length $4\tilde{d} + a \cdot d \log^2 n$ that delivers the message from the messenger u' of the cluster C' to the messenger u'' of the cluster C''.

Altogether, it follows that our algorithm constructs a schedule that delivers the message from the messanger s of the cluster $C(s)$ to the messanger u of the cluster C, and this schedule is of length

$$\sum_{i=1}^{r+1} p_i\tilde{d} + r \cdot (4\tilde{d} + a \cdot d \log^2 n) \ = \ \tilde{d}\sum_{i=1}^{r+1} p_i + r(4\tilde{d} + a \cdot d \log^2 n)) \ .$$

Observe that $|\mathcal{P}| = \sum_{i=1}^{r+1} p_i + r$. Hence, the length of that schedule is equal to $\tilde{d}|\mathcal{P}| + r \cdot (3\tilde{d} + a \cdot d \log^2 n)$. Finally, the delivery of the message from the messenger u of the cluster C to the vertex v (and to all the other vertices of this cluster C) is done via a schedule of length $4\tilde{d} + a \cdot d \log^2 n$ that is constructed by an invocation of Procedure *Period* on a collection of clusters that contains the cluster C.

Hence, altogether, the schedule that is constructed by our algorithm delivers the message from the source s to the vertex v no later than on round $\tilde{d}\cdot(|\mathcal{P}|+1)+(r+1)\cdot(3\tilde{d}+a\cdot d\log^2 n)$. Observe that the tree τ is of length x, $r \le \log n$, $\alpha = O(\log n)$, and $\tilde{d} = d + O(\log n)$. Hence, the length of the schedule is at most

$$\begin{aligned}(d + O(\log n))(x+1) + O(\log n)(O(d) + O(\log n) + O(d\log^2 n)) \; &= \\ d\cdot x + O(x\cdot\log n + d\cdot\log^3 n) \;\le\; Rad + O(x\cdot\log n + d\cdot\log^3 n)\;.\end{aligned}$$

Recall that $d = \lfloor Rad/x \rfloor$, and the lemma follows. ▮

Corollary A.17 *Running our algorithm on input $(G = (V,E), s)$, $s \in V$, with input parameter $x = \sqrt{Rad}\cdot\log n$ results in a radio broadcast schedule of length $Rad + O(\sqrt{Rad}\cdot\log^2 n)$.*

Note that for graphs of radius $Rad = \Omega(\log^4 n)$, the algorithm provides a radio broadcast schedule of length $O(Rad)$.