On Hardness and Approximation of Broadcasting in Sparse Graphs

Jeffrey Bringolf[1], Hovhannes A. Harutyunyan[1],
Shahin Kamali[2], Seyed-Mohammad Seyed-Javadi[2]

[1]Concordia University, Montreal, Canada[*]
[2]York University, Toronto, Canada[†]

Abstract

We study the Telephone Broadcasting problem in sparse graphs. Given a designated source in an undirected graph, the task is to disseminate a message to all vertices in the minimum number of rounds, where in each round every informed vertex may inform at most one uninformed neighbor. For general graphs with n vertices, the problem is NP-hard. Recent work shows that the problem remains NP-hard even on restricted graph classes such as cactus graphs of pathwidth 2 [Aminian et al., ICALP 2025] and graphs at distance-1 to a path forest [Egami et al., MFCS 2025].

In this work, we investigate the problem in several sparse graph families. We first prove NP-hardness for k-cycle graphs, namely graphs formed by k cycles sharing a single vertex, as well as k-path graphs, namely graphs formed by k paths with shared endpoints. Despite multiple efforts to understand the problem in these simple graph families, the computational complexity of the problem had remained unsettled, and our hardness results answer open questions by Bhabak and Harutyunyan [CALDAM 2015] and Harutyunyan and Hovhannisyan [COCAO 2023] concerning the problem's complexity in k-cycle and k-path graphs, respectively.

On the positive side, we present Polynomial-Time Approximation Schemes (PTASs) for k-cycle and k-path graphs, improving over the best existing approximation factors of 2 for k-cycle graphs and an approximation factor of 4 for k-path graphs. Moreover, we identify a structural frontier for tractability by showing that the problem is solvable in polynomial time on graphs of bounded bandwidth. This result generalizes existing tractability results for special sparse families such as necklace graphs.

[*]Emails: bringolfj@gmail.com, and hovhannes.harutyunyan@concordia.ca
[†]Emails: kamalis@yorku.ca, and mohammad.sj2@gmail.com

1 Introduction

In the TELEPHONE BROADCASTING problem, the objective is to spread a message from a designated source vertex to all vertices of a network via a sequence of *telephone calls*. The network is modeled as an undirected, unweighted graph on n vertices, and communication proceeds in synchronous rounds. Initially, only the source knows the message. In each round, every informed vertex may contact at most one uninformed neighbor to pass along the message. The goal is to minimize the total number of rounds until all vertices are informed.

The problem is known to be computationally difficult. Slater *et al.* [19] proved that TELEPHONE BROADCASTING is NP-complete. Elkin and Kortsarz [7] established that approximating TELEPHONE BROADCASTING within a factor of $3 - \epsilon$ for any $\epsilon > 0$ is NP-hard. On the other hand, Elkin and Kortsarz [8] designed an $\mathcal{O}(\log n/ \log\log n)$-approximation algorithm for general graphs, which is the best existing approximation algorithm. Whether a constant-factor approximation exists for general graphs remains open, though such results are known for specific classes, including unit disk graphs [18] and cactus graphs [1].

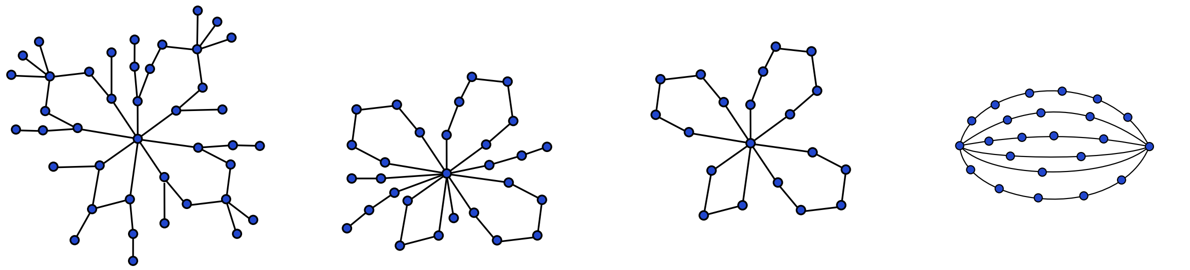
A substantial body of work has investigated the complexity of TELEPHONE BROADCASTING on restricted graph classes. On the positive side, polynomial-time algorithms are known for several families, including trees [10], grids, stars of cliques, fully connected trees, chains of rings, necklace graphs, and certain subclasses of cactus graphs; see, e.g., [5, 9, 12, 17, 20]. On the negative side, hardness results persist even for structurally simple graphs: the problem is NP-hard in planar cubic graphs and in graphs of bounded treewidth. More recently, Tale showed NP-hardness for graphs of pathwidth 3, while Aminian *et al.* [1] established hardness for so-called *snowflake graphs* (cactus graphs of pathwidth 2), resolving a long-standing open question about the complexity of broadcasting in cacti. Even stronger results were obtained by Egami *et al.* [6], who proved NP-hardness for graphs at distance 1 from path forests, i.e., graphs in which deleting a single vertex yields a path forest. See Figure 1 for illustrations of snowflake graphs and graphs at distance 1 from path forests.

These results highlight intriguing frontier cases where the complexity of the problem remains unresolved. One such case is that of *k-cycle graphs* (also called *flower graphs*), obtained by joining k cycles at a single vertex. Despite their simple definition, these graphs exhibit surprisingly rich behavior. Broadcasting on k-cycle graphs has been studied from an approximation perspective: the best known algorithm achieves a factor of 2 [2], yet the exact complexity remains open. A closely related family is that of *k-path graphs*, formed by joining k paths at their endpoints. Several approximation algorithms are known for k-path graphs [3, 14], with the current best result being a 2-approximation [14]. Nevertheless, the precise computational complexity of the problem for this family also remains open.

1.1 Contribution

Our contributions in this paper are as follows. In a nutshell, we tighten the frontier between the tractable and intractable cases of the TELEPHONE BROADCASTING problem.

- We prove that TELEPHONE BROADCASTING is NP-Complete in two of the simplest sparse graph families: *k-cycle graphs* (Theorem 3.4) and *k-path graphs* (Theorem 3.5). Both families have been studied in the broadcasting literature, yet their complexity status remained open. Our hardness results also imply hardness for broader classes, including series–parallel graphs (SP-graphs), for which no hardness was previously known.



(a) A snowflake graph (cactus of pathwdith 2) [1]

(b) Graph of distance 1 to path forest [6]

(c) A k-cycle graph (Theorem 3.4)

(d) A k-path graph (Theorem 3.5)

Figure 1: Sparse graph families for which telephone broadcasting is proven to be NP-hard.

- We design polynomial-time approximation schemes (PTAS) for both k-cycle graphs (Corollary 4.7) and k-path graphs (Corollary 4.14). To the best of our knowledge, these are the first graph families for which Telephone Broadcasting admits both a hardness result and a matching PTAS, highlighting their role as boundary cases between tractability and intractability.

- We present a polynomial-time algorithm for optimal broadcasting in graphs of bounded bandwidth (Theorem 5.2). This result generalizes several known tractable cases, such as chains of rings and related structures. Notably, Telephone Broadcasting is among the few problems exhibiting a complexity gap between graphs of bounded bandwidth (where the problem is tractable) and graphs of bounded pathwidth (where the problem is NP-hard). In particular, k-cycle and k-path graphs both have pathwidth 2 but unbounded bandwidth, placing them precisely on the intractable side of this gap.

1.2 Overview of Techniques

Our results combine hardness reductions, approximation schemes, and dynamic programming, each tailored to the structural properties of the graph families under consideration. We give a brief outline here.

- For the negative results, we design reductions that encode scheduling constraints directly into the graph structure. In particular, the NP-hardness proofs for k-cycle and k-path graphs exploit the interaction between cycle (or path) lengths and the timing of broadcast calls. The shared vertex in k-cycles (or the pair of endpoints in k-paths) acts as the originator, and we show that deciding feasibility becomes equivalent to a combinatorial partitioning problem whose complexity propagates to the broadcasting instance. Our reductions build a more restricted variant of *Numerical 3-Dimensional Matching (RN3DM)* (Definition 3.1), previously employed in hardness proofs for telephone broadcasting. This restricted variant, which is known to be still NP-hard [22], makes it possible to encode number-theoretic constraints using only these simple graph families.

- We design polynomial-time approximation schemes for k-cycle and k-path graphs by reducing the broadcasting problem to covering problems over integers. For k-cycle graphs, the task reduces to covering a multiset of cycle lengths $\{\ell_1, \ldots, \ell_k\}$ with the smallest possible broadcast round β using pairs of numbers from $[\,\beta\,]$, so that for each ℓ_i there exist $a, b \in [\,\beta\,]$ associated with ℓ_i such that $a + b \geq \ell_i$ (See Definition 4.1). For k-path graphs, the covering problem is slightly different: we seek the smallest β such that each ℓ_i can be covered by one number from $[\,\beta\,]$ and one from $[\,\beta - d\,]$, where d is the length of the shortest path (see Definition 4.9). In both cases, approximate solutions to the number-theoretic covering problems translate into near-optimal broadcasting schedules, yielding PTASs for k-cycle and k-path graphs.

- For finding optimal broadcast schemes in graphs of bounded bandwidth, we exploit the fact that a bandwidth-k ordering restricts each vertex to neighbors within a sliding window of $\mathcal{O}(k)$ vertices. This allows us to describe the status of any partial broadcast tree using a compact boundary state and use a dynamic programming approach which has a table with n rows (one for each prefix of the ordering) and at most $n^{\mathcal{O}(k^2)}$ states per row, giving a total size bounded by $f(k) \cdot n^{\mathcal{O}(k^2)}$.

2 Preliminaries

We use notation $[n]$ and $[n, m]$ for positive integers n and m for the set of all natural numbers less than or equal to n and all natural numbers between n and m (inclusive), respectively (i.e. $[n] = \{1, 2, \ldots, n\}$ and $[n, m] = \{n, n+1, \ldots, m\}$). Also, we refer to the number of distinct elements in a multiset as its "support".

Definition 2.1 (Telephone Broadcasting [16]). *An instance (G, s) of the* Telephone Broadcasting *problem is defined by a connected, undirected, and unweighted graph $G = (V, E)$ and a vertex $s \in V$, where s is the only informed vertex. The broadcasting protocol is synchronous and occurs in discrete rounds. In each round, an informed vertex can inform at most one of its uninformed neighbors. The goal is to broadcast the message as quickly as possible so that all vertices in V get informed in the minimum number of rounds.*

Given that the problem can be solved in linear time on trees [10], it is common to represent a *broadcast scheme*—that is, a feasible solution to the problem—by a spanning tree of G rooted at s. We refer to such a tree as a *broadcast tree*. In this tree, an edge (u, v) in the tree indicates that vertex v receives the message from u in the broadcast process. Thus, throughout the paper, the terms "broadcast tree" and "broadcast scheme" are used interchangeably. Likewise, a *broadcast subtree* denotes a partial solution, i.e., a subtree of a broadcast tree capturing how a subset of vertices is informed. Throughout, we use $\boldsymbol{br}^*(G, s)$ to denote the optimal broadcast time of graph G when the originator is s. When the context is clear, we simply write $\boldsymbol{br}^*$ in place of $\boldsymbol{br}^*(G, s)$.

Definition 2.2 (k-cycle and k-path graphs). *A* k-cycle *graph G contains k cycles with exactly one common vertex r. A* k-path *graph G contains k paths with shared endpoints s and t.*

See Figure 1 (parts (c) and (d)) for illustrations of k-cycle and k-path graphs. Observe that any k-cycle (respectively, k-path) graph can be fully described by a set of integers specifying the number of vertices on each cycle (respectively, path). This representation allows us to relate such graphs directly to integer sets, a connection that underlies both our hardness proofs and our algorithmic techniques. These graph families are also simple representatives of well-studied classes: both have bandwidth 2, k-cycle graphs form a subclass of cactus graphs, and k-path graphs form a subclass of series–parallel graphs. Consequently, our hardness results extend beyond k-cycles and k-paths to broader sparse families, including series–parallel graphs, for which no hardness results were previously known.

Definition 2.3 (Graph Bandwidth). *Given a graph G, an* ordering with bandwidth k *of G is a permutation $v_1, \ldots, v_n$ of vertices of G such that for each edge $v_i \sim v_j$ we have $|i - j| \leq k$. The* bandwidth *of G is the minimum k such that such an ordering exists.*

Certain graphs of bounded bandwidth have also been studied in the context of telephone broadcasting. For instance, necklace graphs—collections of cycles where each consecutive pair shares a vertex—have bandwidth $k = 2$ and the problem can be optimally solved for them [13, 15].

3 Hardness Results

In this section, we present our hardness results for TELEPHONE BROADCASTING in k-cycle graphs and k-path graphs. These results extend the existing hardness landscape by showing that even some of the simplest graph families exhibit computational intractability.

Recent hardness results for broadcasting typically rely on reductions from variants of the *Numerical Matching with Target Sums* (NMTS) problem, which is known to be strongly NP-hard [11] (Problem SP17). An instance of NMTS consists of three multisets U, V, and W, each containing m positive integers. The decision problem asks whether there exists a partition of $U \cup V \cup W$ into m disjoint sets $A_1, \ldots, A_m$, such that each A_i contains exactly one element $u_i \in U$, one element $v_i \in V$, and one element $w_i \in W$, with the property that $u_i + v_i = w_i$ for all $i \in [m]$.

Tale [21] introduced a variant of NMTS, called *Numerical 3-Dimensional (Almost) Matching*, in which no repeated entries appear in U and V, and all elements of $U \cup V \cup W$ are multiples of m and two integers T and V. As in NMTS, the objective is to partition $U \cup V$ into m disjoint sets $A_1, \ldots, A_m$, each containing exactly one element from U and one element from V. However, instead of requiring $\sum_{a \in A_i} a = w_i$, the condition is relaxed to $\sum_{a \in A_i} a \geq T - \lambda$. Tale showed that this problem is strongly NP-hard via a reduction from NMTS, and subsequently reduced it to Telephone Broadcasting in graphs of pathwidth at most 3. Egami et al. [6] studied another restricted variant of NMTS, in which all members of U are even while all members of $V \cup W$ are odd. They proved that this variant remains strongly NP-hard and provided a reduction to broadcasting in graphs that are distance-1 from a path forest.

To establish our hardness results, we rely on another restricted version of NMTS in which $U = V = [m]$. This problem, referred to as *Restricted Numerical 3-Dimensional Matching (RN3DM)*, was studied by Yu et al. [22], who established its strong NP-hardness. Their proof relies on a reduction from the *two-machine flowshop scheduling problem with delays*, which they first showed to be NP-hard. Formally, RN3DM is defined as follows.

Definition 3.1 (RESTRICTED NUMERICAL 3-DIMENSIONAL MATCHING (RN3DM) [22])**.** *Given a multiset $W = \{w_1, w_2, \ldots, w_m\}$ of positive integers such that $\sum_{i=1}^{m} w_i + m(m+1) = me$ for some integer e, do there exist two permutations λ and μ of $[m]$ such that $\lambda(i) + \mu(i) + w_i = e \quad$ for all $1 \leq i \leq m$?*

Example 1 (Yes-instance of RN3DM)**.** *Consider the instance of RN3DM with $W = \{1, 3, 4, 4\}$, so $m = 4$. Note that we have $\sum_{i=1}^{m} w_i + m(m+1) = 32 = me$ for $e = 8$. Let $\lambda = (4, 1, 3, 2)$ and $\mu = (3, 4, 1, 2)$. One can check that for every $i \in [4]$, $\lambda(i) + \mu(i) + w_i = e$. For instance, $\lambda(1) + \mu(1) + w_1 = 4 + 3 + 1 = 8$, and $\lambda(2) + \mu(2) + w_2 = 1 + 4 + 3 = 8$. The same holds for $i = 3, 4$. Hence λ and μ certify that W is a* yes-instance *of RN3DM.*

Given that RN3DM is strongly NP-hard [22], we assume that the numbers in W are encoded in unary. We present reductions from RN3DM to broadcasting in k-cycle and k-path graphs. For k-cycle graphs, our approach introduces an intermediate problem as part of the reduction series, whereas for k-path graphs, we directly reduce RN3DM to telephone broadcasting.

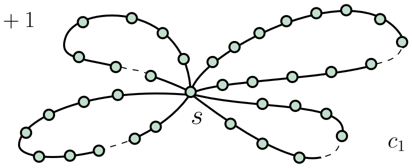


Figure 2: An instance $C = \{7, 7, 9, 13\}$ of Even-Odd RN3DM reduces to the above instance of telephone broadcasting in k-cylce graphs (with s as the originator and $br = 8$). The broadcast tree associated with with solution $\alpha = (4, 6, 2, 8)$ and $\beta = (3, 1, 7, 5)$ is highlighted.

3.1 Hardness of Telephone Broadcasting in k-cycle graphs

We prove that the k-cycle broadcasting decision problem is NP-hard through a series of reductions starting from RN3DM. As a first step, we define an intermediate problem, called *Even-Odd RN3DM*, which is a variant of RN3DM. In this variant, instead of using two permutations λ and μ of $[m]$ (matched with the input set W), we use α as a permutation of the even integers in $[2m]$ and β as a permutation of the odd integers in $[2m]$ to be matched with an input set that we call C. Formally, Even-Odd RN3DM is defined as follows.

Definition 3.2 (Even-Odd Restricted Numerical 3-Dimensional Matching (Even-Odd RN3DM))**.** *Given a multiset $C = \{c_1, c_2, \ldots, c_m\}$ of positive integers such that $\sum_{i=1}^{m} c_i = m(2m+1)$, does there exist a permutation α of $2, 4, 6, \ldots, 2m$ and a permutation β of $1, 3, 5, \ldots, 2m-1$ such that $\alpha(i) + \beta(i) = c_i \quad$ for all $1 \leq i \leq m$?*

Observe that any instance of Even-Odd RN3DM in which C contains an even element is trivially a No-instance, since each c_i must be represented as the sum of one even and one odd number, which always yields an odd value. Hence, without loss of generality, we restrict attention to instances where all elements of C are odd.

Example 2 (Yes-instance of Even-Odd RN3DM)**.** *Consider the instance of Even-Odd RN3DM with $C = \{7, 7, 9, 13\}$, where $m = 4$. Note that $\sum_{i=1}^{4} c_i = 36 = m(2m+1)$. Now take the permutations $\alpha = (4, 6, 2, 8)$ and $\beta = (3, 1, 7, 5)$. One can check for every $i \in [4]$, $\alpha(i) + \beta(i) = c_i$. For instance $\alpha(1)+\beta(1) = 4+3 = 7 = c_1$, and $\alpha(2)+\beta(2) = 6+1 = 7 = c_2$. The same holds for $i = 3, 4$. Thus, α and β certify that C is a* yes-instance *of Even-Odd RN3DM.*

Lemma 3.3. *Even-Odd RN3DM is strongly NP-hard.*

Proof Sketch: We use a reduction from RN3DM, which is strongly NP-Hard [22]. Given an instance $W = \{w_1, \ldots, w_m\}$ of RN3DM with target sum e, we construct an instance $C = \{c_1, \ldots, c_m\}$ of Even-Odd RN3DM by setting $c_i = 2e - 2w_i - 1 \quad$ for each $i \in [m]$. It is straightforward to verify that W is a Yes-instance of RN3DM if and only if C is a Yes-instance of Even-Odd RN3DM. The transformation uses unary encodings, so the reduction is strong. □

Proof. We give a polynomial-time reduction from RN3DM. Let $W = \{w_1, \ldots, w_m\}$ be an arbitrary instance of RN3DM. By definition, we have $\sum_{i=1}^{m} w_i + m(m+1) = me$ for some integer e. Construct an instance $C = \{c_1, \ldots, c_m\}$ of Even-Odd RN3DM, where $c_i = 2e - 2w_i - 1 \quad$ for each $i \in [m]$. To illustrate, the instance of RN3DM in Example 1 reduces to the instance of Even-Odd RN3DM in Example 2. First, we verify that C satisfies the side condition of Even-Odd RN3DM:

$$\sum_{i=1}^{m} c_i = \sum_{i=1}^{m} (2e - 2w_i - 1) = 2em - 2(\sum_{i=1}^{m} w_i) - m = 2em - 2(me - m(m+1)) - m = m(2m+1).$$

In the above equalities, we used the definition of C and the side condition that holds for the instance W of RN3DM. Thus, C is a valid instance of Even-Odd RN3DM.

Next, we show the correctness of the reduction.

(Forward direction). Suppose W is a yes-instance of RN3DM. Then there exist permutations λ and μ of $[m]$ such that for each $i \in [m]$, $\lambda(i)+\mu(i)+w_i = e$. Define α as the permutation of $\{2, 4, \ldots, 2m\}$ given by $\alpha(i) = 2\lambda(i)$, and β as the permutation of $\{1, 3, \ldots, 2m-1\}$ given by $\beta(i) = 2\mu(i) - 1$. Then, for every i,

$$\alpha(i) + \beta(i) = 2\lambda(i) + (2\mu(i) - 1) = 2(\lambda(i) + \mu(i)) - 1 = 2(e - w_i) - 1 = 2e - 2w_i - 1 = c_i.$$

Hence, α and β certify that C is a yes-instance of Even-Odd RN3DM.

(Reverse direction). Suppose C is a yes-instance of Even-Odd RN3DM. Then there exist permutations α of $(2, 4, \ldots, 2m)$ and β of $(1, 3, \ldots, 2m-1)$ such that for each $i \in [m]$, $\alpha(i) + \beta(i) = c_i$. Define $\lambda(i) = \alpha(i)/2$ and $\mu(i) = (\beta(i) + 1)/2$. Both λ and μ are permutations of $[m]$. For each i,

$$\lambda(i) + \mu(i) + w_i = \frac{\alpha(i)}{2} + \frac{\beta(i) + 1}{2} + w_i = \frac{c_i + 1}{2} + w_i = \frac{(2e - 2w_i - 1) + 1}{2} + w_i = e.$$

Thus λ and μ certify that W is a yes-instance of RN3DM.

Since the reduction preserves yes-instances in both directions and can be carried out in polynomial time, Even-Odd RN3DM is strongly NP-hard. □

Theorem 3.4. *Broadcasting in k-cycle graphs is NP-complete.*

Proof. Telephone Broadcasting is in NP in general graphs (and hence also for k-cycle graphs). We prove NP-hardness by a reduction from the Even-Odd RN3DM problem, which is strongly NP-Hard by Lemma 3.3.

Let $C = \{c_1, c_2, \ldots, c_m\}$ be an instance of Even-Odd RN3DM. We construct an instance (G, s, t) of Telephone Broadcasting in k-cycle graphs as follows. Graph G consists of m cycles, containing $c_1, c_2, \ldots, c_m$, extra vertices in addition to a single vertex s that is shared between them. Now s serves as the originator of the broadcast. Thus, the number of vertices in G is $n = \sum_{i=1}^{m} c_i + 1 = m(2m + 1) + 1$, and the degree of s is $2m$. Since RN3DM is strongly NP-hard by Lemma 3.3, we may assume the values c_i are encoded in unary, so the reduction is polynomial time. The decision problem asks whether broadcasting completes within $t = 2m$ rounds. See Figure 2.

(Forward direction). Suppose the answer to instance C of Even-Odd RN3DM is yes. Then there exist permutations α of $(2, 4, \ldots, 2m)$ and β of $(1, 3, \ldots, 2m-1)$ such that $\alpha(i) + \beta(i) = c_i$ for all $i \in [m]$. Construct a broadcast scheme as follows: the source s informs its two neighbors in each cycle c_i at times $2m - \alpha(i) + 1$ and $2m - \beta(i) + 1$, respectively. Given that α and β are permutations of space $[2m]$, the source s informs exactly one vertex at each given round $i \in [2m]$. Each newly informed vertex continues by informing its uninformed neighbor in the subsequent round. Hence, by round $br = 2m$, exactly $\alpha(i) + \beta(i) = c_i$ vertices of the cycle (excluding s) are informed, and therefore the entire graph is informed. Thus, (G, s, br) is a yes-instance of broadcasting.

(Reverse direction). Now suppose (G, s, br) is a yes-instance of broadcasting with $br = 2m$. So, there is a broadcast scheme that completes within $22m$ rounds. Without loss of generality, assume the scheme is non-lazy, i.e., every informed vertex calls an uninformed neighbor whenever possible. Let t_i denote the number of vertices newly informed in round i. At round 0, only s is informed: $t_0 = 1$. At each round i, only s and the vertices informed at time $i-1$ can inform new vertices (since other vertices of degree 2 have no additional uninformed neighbors left). Hence, $t_{i+1} \leq t_i + 1$. It

follows that the number of informed vertices by round i is at most $1+1+2+\cdots+i = 1+\frac{i(i+1)}{2}$. For $i = 2m$, this reaches exactly $1+m(2m+1) = n$, so the broadcast must finish in exactly $2m$ rounds (and not earlier). This forces the source s to make a call in every round, and since $\deg(s) = 2m$, s informs exactly one neighbor per round. Consequently, each cycle c_i must be partitioned into two paths, informed from s in opposite directions, of lengths $\alpha(i)$ and $\beta(i)$ with $\alpha(i) + \beta(i) = c_i$. Because c_i is odd (otherwise the Even-Odd RN3DM instance would be trivial), we conclude that one of $\alpha(i), \beta(i)$ is odd and the other is even. This provides the exact decomposition required for a yes-instance of Even-Odd RN3DM. □

3.2 Hardness of Telephone Broadcasting in k-Path Graphs

Next, we use a reduction from RN3DM to establish NP-completeness of broadcasting in k-path graphs.

Theorem 3.5. *Broadcasting in k-path graphs is NP-complete.*

Proof. Telephone Broadcasting belongs to NP for general graphs and hence also for k-path graphs. We prove NP-hardness by a reduction from the *Restricted Numerical 3-Dimensional Matching* (RN3DM) problem (Definition 3.1), which is strongly NP-Hard [22].

Let $W = \{w_1, w_2, \ldots, w_m\}$ be an instance of RN3DM. By definition, $\sum_{i=1}^{m} w_i + m(m+1) = me$ for some integer e. We construct an instance (G, s, br) of Telephone Broadcasting in k-path graphs as follows. The graph G consists of m internally disjoint paths with shared endpoints s and t, where the i-th path contains $\ell_i = e - w_i$ internal vertices in addition to s and t. Furthermore, we add an edge connecting s and t. The vertex s serves as the source of the broadcast. The total number of vertices in G is $n = 2 + \sum_{i=1}^{m}(e - w_i) = 2 + me - \sum_{i=1}^{m} w_i = 2 + m(m+1)$. Both s and t have degree $2m+1$. Since RN3DM is strongly NP-hard, we may assume that the values w_i are encoded in unary, ensuring that the reduction is polynomial time. The decision problem asks whether broadcasting completes within $br = m+1$ rounds (see Figure 3).

(Forward direction). Suppose W is a yes-instance of RN3DM. Then there exist permutations λ and μ of $[m]$ such that $\lambda(i) + \mu(i) = e - w_i = \ell_i$ for all $i \in [m]$. Construct a broadcast schedule as follows: in round 1, the source s calls t. Subsequently, in round $j \in [m]$, s and t inform their neighbors in the j'th path at times $m+1-\lambda(j)$ and $m+1-\mu(j)$, respectively. Since λ and μ are permutations, exactly one new vertex is informed from s and one from t in each round (i.e., this is a valid broadcast scheme). Each newly informed vertex continues the broadcast by informing its remaining neighbor in the next round. Hence, by round $br = m+1$, exactly $\lambda(j)$ vertices from the side of s and $\mu(j)$ vertices from the side of t from path j are informed; therefore, all ℓ_j internal vertices of every path are informed, and the entire graph is covered. Thus, (G, s, br) is a yes-instance of Telephone Broadcasting.

(Reverse direction). Suppose (G, s, br) is a yes-instance of Telephone Broadcasting with $br = m+1$. Without loss of generality, assume the scheme is *non-lazy*, i.e., every informed vertex calls an uninformed neighbor whenever possible. Let t_i denote the number of vertices newly informed in round i. At round 0, only s is informed, so $t_0 = 1$. Suppose t is informed at round $p \geq 1$. At each round i, only s, possibly t (if $i > p$), and the vertices informed at round $i-1$ can make new calls. Thus $t_{i+1} \leq t_i + 1$ for $i \leq p$, and $t_{i+1} \leq t_i + 2$ for $i \geq p$. This upper bound value is maximized when $p = 1$, i.e., when s calls t immediately. In that case, we have $t_0 = 1$, $t_1 = 1$, $t_2 = 2$, and for $i \geq 2$, $t_i = t_{i-1} + 2$. Therefore, the total number of informed vertices by round i is at most $1+1+2+4+6+\cdots+2(i-1) = i(i-1)+2$. For $i = m+1$, this reaches exactly $n = 2+m(m+1)$. Hence, the broadcast cannot finish in less than $m+1$ rounds, i.e., must finish in exactly $m+1$

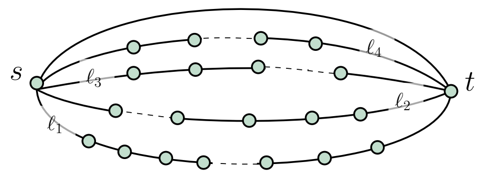


Figure 3: An instance $W = \{1, 3, 4, 4\}$ of RN3DM reduces to the above instance of telephone broadcasting in k-cylce graphs (with s as the originator and $br = 5$). The broadcast tree associated with with solution $\lambda = (4, 1, 3, 2)$ and $\mu = (3, 4, 1, 2)$ is highlighted.

rounds. This forces s to call t in round 1, and both s and t to call exactly one new neighbor in each subsequent round. Otherwise, the number of informed vertices at time t_i will be less than the above upper bound, and hence the broadcast cannot complete within $m + 1$ rounds.

Consequently, each path of length ℓ_i must be partitioned into two subpaths, one informed from s and the other from t, of lengths $\lambda(i)$ and $\mu(i)$ such that $\lambda(i) + \mu(i) = \ell_i$. Recall that $\ell_i = e - w_i$. Since both s and t inform exactly one new neighbor in each round, it follows that for any two distinct paths i and j, we must have $\lambda(i) \neq \lambda(j)$ and $\mu(i) \neq \mu(j)$. In other words, λ and μ are permutations of $[m]$. This corresponds exactly to a yes-instance of RN3DM. $\square$

4 Polynomial Time Approximation Schemes

4.1 PTAS for k-Cycle Graphs

In this section, we present a polynomial-time approximation scheme (PTAS) for Telephone Broadcasting in k-cycle graphs. Recall that a k-cycle graph is obtained by joining k cycles at a common central vertex r. We assume that the number of cycles k—and hence the optimal broadcast time—is asymptotically large; for constant k, the problem can be solved by a straightforward exhaustive search (see, e.g., [4]).

In what follows, we design a PTAS for the case where the broadcast originates at $s = r$. This restriction is without loss of generality: any c-approximation for the case that the center is the originator (when $s = r$) can be extended to a c-approximation in the general case ($s \neq r$). Suppose the originator s is a non-center vertex on some cycle C_m, at distance $d \geq 1$ from r. The message can first be forwarded along the shortest path from s to r. Once r is informed, it first sends the message to its other neighbor on C_m and then proceeds with the c-approximation algorithm on the rest of the graph. If the overall broadcast time is determined by a vertex in C_m, then the scheme is optimal. Otherwise, the process completes within $d + 1 + c \cdot \boldsymbol{br}^-$ rounds, where $\boldsymbol{br}^-$ is the optimal broadcast time excluding C_m. Since $\boldsymbol{br}^* \geq d + \boldsymbol{br}^-$, this yields the same approximation factor c.

We introduce the following problem, which is the key part of our PTAS algorithms.

Definition 4.1. *The input to the* Prefix Covering *problem is a* ground multiset $S = \{\ell_1, \ell_2, \ldots, \ell_n\}$ *of positive integers. The objective is to determine the minimum integer m such that the set $C = [m]$ can be partitioned into n subsets $C_0, C_1, \ldots, C_n$, such that for each $i \in [n]$, we have $|C_i| \leq 2$ and $\sum_{x \in C_i} x \geq \ell_i$. In this context, we say that ℓ_i is* covered *by the elements of C_i, and we refer to C as the* covering set *of S.*

Example 3. *When $S = \{13, 8, 7, 6\}$, the optimal solution is given by $C = [8]$, and we have $C_1 = \{8, 5\}, C_2 = \{7, 2\}, C_3 = \{4, 3\}, C_4 = \{6\}$ (and $C_0 = \{1\}$).*

We begin by establishing the equivalence between Telephone Broadcasting in k-cycle graphs and the Prefix Covering problem.

Lemma 4.2. *Let G be a k-cycle graph with central vertex s, where the i-th cycle contains c_i additional vertices besides s, and let $S = \{c_1, \ldots, c_k\}$. Then* Telephone Broadcasting *for instance (G, s) becomes equivalent to* Prefix Covering *for S.*

Proof Sketch: Let $\boldsymbol{br}$ denote the number of broadcast rounds. If the center s informs two neighbors of cycle i at rounds x and y, then the remaining c_i vertices are informed within $\boldsymbol{br}$ rounds if and only if we have $(\boldsymbol{br} - x + 1) + (\boldsymbol{br} - y + 1) \geq c_i$. This is equivalent to covering c_i by two elements of $[\,\boldsymbol{br}\,]$ whose sum is at least c_i. Thus, a broadcast schedule with $\boldsymbol{br}$ rounds corresponds to a feasible Prefix Covering solution, and conversely any such covering yields a valid broadcast schedule. □

Proof. We show that any feasible broadcasting scheme with $\boldsymbol{br}$ rounds, for instance (G, s), corresponds to a feasible covering of S using $[\boldsymbol{br}]$, and conversely any such covering yields a valid broadcasting scheme. This one-to-one correspondence ensures that the two problems have the same optimal solutions.

From Telephone Broadcasting *to* Prefix Covering*:* In any broadcast scheme with $\boldsymbol{br}$ rounds, $r (= s)$ must call at most two vertices on each cycle. If r informs neighbors in cycle i at rounds x and y, then the remaining $c_i - 1$ vertices of cycle i must be covered within the remaining rounds, which is feasible exactly when $(\boldsymbol{br} - x + 1) + (\boldsymbol{br} - y + 1) \geq c_i$. Thus, each c_i is covered by the pair $\{\boldsymbol{br} - x + 1, \boldsymbol{br} - y + 1\}$ in the set $[\boldsymbol{br}]$. Since r uses distinct rounds for distinct neighbors, the covering also respects the injectivity requirement. Hence, every broadcast scheme with $\boldsymbol{br}$ rounds yields a valid covering for S with $[\boldsymbol{br}]$.

From Prefix Covering *to* Telephone Broadcasting*:* Conversely, suppose S has a covering using $C = [\boldsymbol{br}]$. If c_i is covered by x and y, then in the broadcast scheme r informs two neighbors in the i'th cycle at rounds $\boldsymbol{br} - x + 1$ and $\boldsymbol{br} - y + 1$. Because $x + y \geq c_i$, the message traverses all of i'th cycle within $\boldsymbol{br}$ rounds. Injectivity of the covering ensures that r never attempts more than one call in a round. Thus, every covering for the Prefix Covering instance corresponds to a valid broadcast scheme with $\boldsymbol{br}$ rounds.

Therefore, Telephone Broadcasting in k-cycle graphs is equivalent to the Prefix Covering problem. □

4.1.1 PTAS for Prefix Covering

In this section, we present a PTAS for the Prefix Covering problem, that is, an algorithm with approximation factor $(1 + \epsilon)$ for a given $\epsilon > 1$. By Lemma 4.2, this would give a PTAS for Telephone Broadcasting in k-cycle graphs. As a high-level view of the algorithm, it addresses the covering problem through a two-stage approximation scheme that leverages the concept of rounded sets (see Definition 4.3) to reduce the problem complexity. The main idea is to transform the original ground multiset and the covering set into a simplified version with constant support (distinct elements), making it feasible to apply an exhaustive search which maintains provable approximation guarantees.

The algorithm operates as follows. Given an instance S of Prefix Covering, we first construct a *rounded set* $\mathfrak{R}_p(S)$, where each element of S is rounded up to a larger value in S so that the support of $\mathfrak{R}_p(S)$ has size p. Here p is a constant integer chosen as a function of ϵ. Since rounding increases item values, covering $\mathfrak{R}_p(S)$ may require a larger covering set. Nevertheless, we show that the optimal covering of $\mathfrak{R}_p(S)$ is within a factor $(1+2/p)$ of the optimal covering of S (Proposition 4.4). Thus, it suffices to focus on covering $\mathfrak{R}_p(S)$ rather than S. For that, we use a binary search over candidate values m, where m specifies the size of the covering set $C = [m]$. For each m, we define a rounded version $\mathfrak{R}_p(C)$, obtained by rounding up the elements of C so that its support also has size p. We prove that if $\mathfrak{R}_p(S)$ can be covered using $\mathfrak{R}_p(C)$, then it can also be covered using $C' = [m']$ with $m' \leq (1+1/p)m$ (Claim 4.1). Conversely, if $\mathfrak{R}_p(S)$ cannot be covered by $\mathfrak{R}_p(C)$, then it cannot be covered by $C(=[m])$. Hence, verifying whether $\mathfrak{R}_p(C)$ successfully covers $\mathfrak{R}_p(S)$ is sufficient for approximating the original instance. This verification can be performed by exhaustive search over $\mathfrak{R}_p(S)$ and $\mathfrak{R}_p(C)$. Since both sets have constant support (bounded by p), this step can be carried out in time independent of n (Lemma 4.5). In what follows, we provide the details of the algorithm.

Based on Definition 4.1, an instance of Prefix Covering can be defined as a tuple (S, C), where $S = \{\ell_1, \ell_2, \ldots, \ell_n\}$ and $C = [m]$ are the ground multiset and the covering set, respectively. Throughout this section, we assume that $\ell_1 \geq \ell_2 \geq \ldots \geq \ell_n$.

Definition 4.3. *Let S be a finite set and let p be a positive integer. The* rounded set $\mathfrak{R}_p(S)$ *of S with p parts is constructed as follows:*

(i). Order the elements of S in non-increasing sequence: $s_1 \geq s_2 \geq \cdots \geq s_{|S|}$.

(ii). Partition this ordered sequence into p consecutive parts $P_1, P_2, \ldots, P_p$, where:

- *For $i = 1, 2, \ldots, p-1$: P_i contains exactly $\lceil |S|/p \rceil$ elements,*
- *P_p contains the remaining $|S| - (p-1)\lceil |S|/p \rceil$ elements.*

(iii). For each part P_i, let m_i denote the maximum element in P_i. Then $\mathfrak{R}_p(S)$ is the multiset obtained by replacing each element in P_i with m_i for all $i \in \{1, 2, \ldots, p\}$.

Example 4. *For $S = \{123, 67, 65, 45, 43, 43, 43, 18, 12, 12, 10, 6, 4, 4, 1, 1\}$ and $p = 4$, we get $\mathfrak{R}_4(S) = \{123, 123, 123, 123, 43, 43, 43, 43, 12, 12, 12, 12, 4, 4, 4, 4\}$.*

Proposition 4.4. *$\boldsymbol{OPT}(\mathfrak{R}_p(S)) \leq \boldsymbol{OPT}(S) + 2\lceil |S|/p \rceil$, where $\boldsymbol{OPT}$ is the optimal cover for a specific instance of* Prefix Covering.

Proof. By definition, $\mathfrak{R}_p(S)$ consists of p parts. Given an optimal cover of S, form a cover of $\mathfrak{R}_p(S)$ by assigning the numbers given to the part i in S to the part $(i+1)$ in $\mathfrak{R}_p(S)$ for $i \in [p-1]$. As the size and value of part $i+1$ is not greater than the value of part i, the set of numbers that can cover part i can also cover part $i+1$.

Therefore, the first part is the only part that needs to be covered. We show that adding an extra $2\lceil |S|/p \rceil$ to the covering set is enough to cover the first part. This is because every $\ell_i \in S$ is less than 2 items of size $\boldsymbol{OPT}(S)$, as otherwise it would not be possible to cover ℓ_i with at most two numbers in $[\boldsymbol{OPT}(S)]$. Thus, by adding an extra $2\lceil |S|/p \rceil$ items to the covering set, we can cover $\mathfrak{R}_p(S)$. □

Example 5 (Based on Example 4)**.** *In Example 4, items in a covering set that were used to cover $\{123, 67, 65, 45\}$ in the optimal cover of S will be used to cover $\{43, 43, 43, 18\}$ in $\mathfrak{R}_p(S)$; those used to cover $\{43, 43, 43, 18\}$ will be used to cover $\{12, 12, 12, 12\}$ in $\mathfrak{R}_p(S)$, and so on. The only remaining uncovered part in $\mathfrak{R}_p(S)$ will be the first part ($(123, 123, 123, 123)$), which has size $|S|/p$. Moreover, every element in that part has size at most $2\boldsymbol{OPT}(S)$. Therefore, the first part in $\mathfrak{R}_p(S)$ can be covered by an extra $2|S|/p$ in the covering set. In the example, if $\boldsymbol{OPT}(S) = 73$ in the above example, we can cover $\{123, 123, 123, 123\}$ by adding $\{74, 75, 76, 77, 78, 79, 80, 81\}$ to the optimal cover set of S to cover $\mathfrak{R}_p(S)$ using $\boldsymbol{OPT}(S) + 2|S|/p$ rounds.*

Given that $\boldsymbol{OPT}(S) \geq |S|$, we can conclude that an optimal covering of $\mathfrak{R}_p(S)$ gives a $(1+2/p)$ approximation for covering S. Therefore, we can now focus on covering $\mathfrak{R}_p(S)$, which has constant support p.

Let m be a given value and let $C = [m]$. We present a procedure that either constructs a covering of $\mathfrak{R}_p(S)$ using the set $C' = [m']$, where $m' = m + \lceil m/p \rceil$, or certifies that $\mathfrak{R}_p(S)$ cannot be covered using $[m]$. For this purpose, we use the rounded set $\mathfrak{R}_p(C)$ to cover $\mathfrak{R}_p(S)$. We define the *stretched covering* $\mathfrak{R}_p(C)$ as the set obtained by partitioning the C and rounding up its elements (as in Definition 4.3), ensuring that the covering set has constant support p. This enables a binary-search strategy to find the smallest feasible value of m.

Example 6. *Suppose* $\mathfrak{R}_p(S) = \{123, 123, 123, 123, 43, 43, 43, 43, 12, 12, 12, 12, 4, 4, 4, 4\}$ *and* $p = 3$*. For* $m = 12$*, we have* $C = \{12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$*, and we aim to cover* $\mathfrak{R}_p(S)$ *with* $\mathfrak{R}_p(C) = \{12, 12, 12, 12, 8, 8, 8, 8, 4, 4, 4, 4\}$.

Claim 4.1. *Given that we can cover* S *with* $\mathfrak{R}_p(C)$*, where* $C = [m]$*, we can cover* S *using* $C' = [m']$*, where* $m' = m + \lceil m/p \rceil$*.*

Proof. Based on definition, $\mathfrak{R}_p(C)$ contains p parts. Suppose we have a covering with $\mathfrak{R}_p(C)$ for S. Associate value $m + \lceil m/p \rceil - (i-1)$ to the i'th largest number in C for $i \in [m]$. This transformation only increases the value as the first block with original value m matches to $m+1, \ldots, m + \lceil m/p \rceil$, the second block matches to $m - \lceil m/p \rceil + 1, \ldots, m$, and so on. More specifically, the i'th block matches to $m - (i-1)\lceil m/p \rceil + 1, \ldots, m - (i-2)\lceil m/p \rceil$, which is greater than the original value of the block. Now, as the new set is a subset of C' and every value in C is larger than its original value, the same covering can cover S with the corresponding values in C'. □

Example 7 (Based on Example 6)**.** *If we can cover* $\mathfrak{R}_p(S)$ *using* $\mathfrak{R}_p(C) = \{12, 12, 12, 12, 8, 8, 8, 8, 4, 4, 4, 4\}$*, we can cover* $\mathfrak{R}_p(S)$ *using* $C' = \{16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$*: items of* $\mathfrak{R}_p(S)$ *that were covered by* $(12, 12, 12, 12)$ *in* $\mathfrak{R}_p(C)$ *can be covered with* $(16, 15, 14, 13)$ *in* C'*, whatever was covered by* $(8, 8, 8, 8)$ *in* $\mathfrak{R}_p(C)$*, can be covered by* $(12, 11, 10, 9)$ *in* C'*, etc. Hence, if we can cover* $\mathfrak{R}_p(S)$ *with* $\mathfrak{R}_p(C)$*, we can cover* $\mathfrak{R}_p(S)$ *using* C'*; which means we can cover* $\mathfrak{R}_p(S)$ *using* $[m']$*.*

Lemma 4.5. *Given two sets* $\mathfrak{R}_p(S)$ *and* $\mathfrak{R}_p(C)$*, there exists an algorithm that, in time* $\mathcal{O}(2^{p^3} |S| (|S| + |C|))$*, either outputs a covering of* $\mathfrak{R}_p(S)$ *using the covering set* $\mathfrak{R}_p(C)$ *(if one exists) or correctly reports that no such covering is possible.*

Proof. Both the ground set $\mathfrak{R}_p(S)$ and the covering set $\mathfrak{R}_p(C)$ have constant support p. Fix a distinct value $v \in \mathfrak{R}_p(S)$. For each pair (a, b) of elements from $\mathfrak{R}_p(C)$, we can encode how many times (a, b) is used to cover elements of $\mathfrak{R}_p(S)$ equal to v. Since there are at most $|S|$ such elements, this count can be represented in $\log |S|$ bits. As there are at most p^2 pairs from $\mathfrak{R}_p(C)$ and p distinct values in $\mathfrak{R}_p(S)$, any candidate covering can be encoded using at most $p^3 \log |S|$ bits. Hence, the number of candidate solutions is bounded by $\mathcal{O}(2^{p^3}|S|)$. Checking whether a candidate encoding yields a valid covering takes $\mathcal{O}(|S| + |C|)$ time. Therefore, the total running time is $\mathcal{O}(2^{p^3}|S|(|S| + |C|))$. Moreover, whenever a valid covering exists, the algorithm outputs it by interpreting the corresponding encoding; otherwise, it correctly reports that no covering is possible. □

To summarize, given a covering guess size m and ground set $\mathfrak{R}_p(S)$, we form $\mathfrak{R}_p(C)$, and check whether we can cover $\mathfrak{R}_p(S)$ using $\mathfrak{R}_p(C)$ with an exhaustive approach (Lemma 4.5). If we cannot find a covering, we return bad-guess; we cannot cover $\mathfrak{R}_p(S)$ using $[m]$. If we find a solution, use the covering given by $\mathfrak{R}_p(C)$ to cover $\mathfrak{R}_p(S)$ using $[m']$, where $m' = m + \lceil m/p \rceil$. This gives a $(1+1/p)$-approximation for covering $\mathfrak{R}_p(S)$. Applying the Proposition 4.4, we can get an approximating factor of $(1+2/p)(1+1/p)$, which can be as close as we want to 1 by setting p large enough. Using binary search adds a multiplicative factor of $\log \boldsymbol{OPT}$ to the runtime we used for the exhaustive search (see Lemma 4.5). Hence, the overall running time is $\mathcal{O}(2^{p^3}|S|(|S| + \boldsymbol{OPT}) \log \boldsymbol{OPT})$. We can conclude the following:

Theorem 4.6. *There is a PTAS for the* PREFIX COVERING *problem that terminates in* $\mathcal{O}(2^{p^3}|S|(|S| + \boldsymbol{OPT}) \log \boldsymbol{OPT})$*, where* $p = 3/\epsilon^2$*.*

Proof. Given an instance S of PREFIX COVERING with $\epsilon > 0$, set $p = \lceil 3/\epsilon^2 \rceil$ to ensure $(1 + 2/p)(1 + 1/p) \leq 1 + \epsilon$. Construct $\mathfrak{R}_p(S)$ by rounding up elements in S. We use binary search over m to find $\boldsymbol{OPT}_p(\mathfrak{R}_p(S))$, where $\boldsymbol{OPT}_p(S)$ is the minimum m such that S can be covered using $\mathfrak{R}_p([m])$. For each candidate set $[m]$:

(i). Construct $\mathfrak{R}_p(C)$ where $C = [m]$ as per Definition 4.3.

(ii). Use exhaustive search (Lemma 4.5) to determine if $\mathfrak{R}_p(C)$ can cover $\mathfrak{R}_p(S)$ in time $\mathcal{O}(2^{p^3}|S|(|S| + m))$.

Therefore, the binary search routine gives the smallest m such that $\mathfrak{R}_p(S)$ can be covered using $\mathfrak{R}_p([m])$. We construct a covering of $\mathfrak{R}_p(S)$ using $C' = [m']$, where $m' \leq (1 + 1/p)m = (1 + 1/p)\boldsymbol{OPT}_p(\mathfrak{R}_p(S)) \leq (1 + 1/p)\boldsymbol{OPT}(\mathfrak{R}_p(S))$ by Claim 4.1. Given that $C' = [m']$ covers $\mathfrak{R}_p(S)$, it covers S as well. Using Proposition 4.4, we have $m' \leq (1 + 1/p)\boldsymbol{OPT}(\mathfrak{R}_p(S)) \leq (1 + 1/p)(1 + 2/p)\boldsymbol{OPT}(S)$. The running time is $\mathcal{O}(\log \boldsymbol{OPT})$ binary search iterations, each taking $\mathcal{O}(2^{p^3}|S|(|S| + \boldsymbol{OPT}))$ time, giving total complexity $\mathcal{O}(2^{p^3}|S|(|S| + \boldsymbol{OPT}) \log \boldsymbol{OPT})$ where $p^3 = \mathcal{O}(1/\epsilon^6)$. □

Lemma 4.2 and Theorem 4.6 imply the following Corollary.

Corollary 4.7. *There exists a PTAS for* TELEPHONE BROADCASTING *in k-cycle graphs. In particular, for a k-cycle graph on n vertices and any $\epsilon > 0$, the algorithm computes a $(1 + \epsilon)$-approximate broadcast schedule in time* $O(2^{1/\epsilon^6} k(k + n) \log n)$*.*

4.2 PTAS for k-Path Graphs

In this section, we present a polynomial-time approximation scheme (PTAS) for TELEPHONE BROADCASTING in k-path graphs. Consider a k-path graph G with endpoints s and t.

In what follows, we assume that s is the designated source, holding the message at time 0, and that t acts as a *late source*, receiving the message at time α, with $\alpha \leq d(s,t)$, where $d(s,t)$ denotes the distance between s and t in G. We show that any instance of the (single-source) problem with an arbitrary originator can be reduced to this double-source setting (with s as the source and t as the late source) without changing the approximation factor (Lemma 4.8).

In Lemma 4.10, we establish an equivalence between double-source broadcasting and the integer covering problem that we call DOUBLE PREFIX COVERING (Definition 4.9).

Finally, by applying a rounding-and-covering approach similar to the one used for PREFIX COVERING in k-cycle graphs, we obtain a PTAS for DOUBLE PREFIX COVERING, and consequently for TELEPHONE BROADCASTING in k-path graphs (Section 4.2.1). In what follows, we describe these steps in detail. First, we prove the following lemma, which implies we can restrict our attention to the double-source broadcasting.

Lemma 4.8. *Any c-approximation algorithm for* Telephone Broadcasting *in k-path graphs in the double-source setting—where s is the primary source and t is a late source receiving the message at time α—can be extended to a c-approximation algorithm for the (single-source) general setting, where the originator v_o may be any vertex of the graph.*

Proof Sketch: Without loss of generality, assume v_o is closer to s than to t. At round 1, v_o sends the message along its path toward s, and at round 2, it sends the message along the path toward t. Once s receives the message, it immediately forwards it along its shortest path toward t. Finally, when t receives the message, it first transmits it back along its path toward v (to ensure that the neighbor is informed if it has not already been). These enforced decisions affect the broadcast time only by a constant number of rounds (the approximation remains the same). If s is informed at time τ, then t is informed at time $\tau' = \tau + \alpha$. $\square$

Proof. Let v_o be the originator vertex lying in path P_i at distances $d(v_o, s)$ and $d(v_o, t)$ from endpoints s and t, respectively. Without loss of generality, assume $d(v_o, s) \leq d(v_o, t)$. Let $d(v_o, s)$. The message is first forwarded along the shortest path from v_o to s, which takes $d(v_o, s)$ rounds. At round $d(v_o, s) + 1$, vertex s is informed and can begin broadcasting. Vertex t is informed either directly through v_o (after $d(v_o, t)$ rounds) or through s via some other path P_j (whichever occurs first). Let α denote the delay with which t receives the message relative to s. Once both s and t are informed, they proceed with the c-approximation algorithm on the remaining graph.

We consider two cases depending on how t receives the message:

Case I: t is informed directly through v_o. Here $\alpha = d(v_o, t) - d(v_o, s) + 1$. If the global broadcast time is determined by a vertex in P_i, then the scheme is optimal. Otherwise, the process completes within $d(v_o, s) + c \cdot \boldsymbol{br}^{-}(\alpha)$ rounds, where $\boldsymbol{br}^{-}(\alpha)$ is the optimal broadcast time for the subproblem excluding P_i with sources s and t where t has delay α. We have $\boldsymbol{br} \geq d(v_o, s) + \boldsymbol{br}^{-}(\alpha - 1) \geq d(v_o, s) - 1 + \boldsymbol{br}^{-}(\alpha)$ because v_o can inform its neighbor toward t in its first round, which results in t getting informed one round sooner. Therefore, this yields the same approximation factor c.

Case II: t is informed through s via another path P_j. Let ℓ_j be the number of vertices of path P_j (excluding s and t). Then t is informed at round $d(v_o, s) + 1 + \ell_j$, giving delay $\alpha = \ell_j$. If the global broadcast time is determined by a vertex in P_i or P_j, then the scheme is optimal. Otherwise, the process completes within $d(v_o, s) + 1 + c \cdot \boldsymbol{br}^{-}(\alpha)$ rounds, where $\boldsymbol{br}^{-}(\alpha)$ is the optimal broadcast time for the subproblem excluding P_i and P_j with sources s and t where t has delay α. Since $\boldsymbol{br} \geq d(v_o, s) + 1 + \boldsymbol{br}^{-}(\alpha)$, this yields the same approximation factor c. $\square$

We introduce the following problem, which is the key part of our PTAS algorithms.

Definition 4.9. *The input to the* Double Prefix Covering *problem is a* ground multiset *$S = \{\ell_1, \ell_2, \ldots, \ell_n\}$ of positive integers together with a non-negative integer β. The objective is to determine the minimum integer m such that there exist functions $c : S \to \{0\} \cup [m]$ and $d : S \to \{0\} \cup [m - \beta]$ satisfying $c(i) + d(i) \geq \ell_i$ for all $i \in [n]$. Moreover, injectivity is required in the following sense: for any $i \neq j$, if both $c(i) > 0$ (respectively $d(i) > 0$), then $c(i) \neq c(j)$ (respectively $d(i) \neq d(j)$). In this setting, we say that each ℓ_i is* covered *by the pair $\{c(i), d(i)\}$, and we refer to $C = [m]$ and $D = [m - \beta]$ as the* covering sets *of S.*

Example 8. *When $S = \{8, 4, 4, 2\}$, the optimal solution is given by $C = [5]$ and $\beta = 2$, and we have $c(1) = 5, d(1) = 3$; $c(2) = 4, d(2) = 0$, $c(3) = 2, d(3) = 2$, $c(4) = 1, d(4) = 1$.*

In what follows, we establish the equivalence between double-source broadcasting in k-path graphs and Double Prefix Covering (Definition 4.9).

Lemma 4.10. *Let G be a k-path graph formed by k paths with common endpoints s and t, where the ith path contains $\ell_i \geq 0$ internal vertices (in addition to s and t). Consider a double-source broadcasting instance on G, where s receives the message at time 0 and t at time α. Now, define an instance I of* Double Prefix Covering *as $I = (S = \{\ell_1, \ldots, \ell_k\},\ \beta = \alpha - 1)$. Then computing the minimum broadcast time for this double-source* Telephone Broadcasting *instance is equivalent to solving the* Double Prefix Covering *instance I.*

Proof. Consider the covering of the instance I with parameter m. We claim that there exists a valid broadcasting scheme for G within $\boldsymbol{br} = m$ rounds. For each $i \in [k]$, let $c(i)$ and $d(i)$ denote the assigned values in the covering. Then s informs the ith path at round $m - c(i) + 1$ and t at round $m - d(i) + 1$. Thus, s (respectively t) uses disjoint rounds drawn from $\{\, m - c + 1 \mid c \in C \,\} = [m]$ (respectively from $\{\, m - d + 1 \mid d \in [m - \alpha + 1] \,\} = [\alpha, m]$). Hence, both s and t follow a valid broadcast scheme.

It remains to argue coverage. Since $c(i) + d(i) \geq \ell_i$, the remaining time after the calls at rounds $m - c(i) + 1$ and $m - d(i) + 1$ is sufficient to cover all ℓ_i internal vertices of path i. Thus, the covering implies a feasible broadcasting scheme.

Conversely, suppose we have a broadcasting scheme using $\boldsymbol{br}$ rounds. For each $\ell_i \in S$, if s and t call vertices in path i at rounds y_s^i and y_t^i, define $c(i) = \boldsymbol{br} - y_s^i + 1$ and $d(i) = \boldsymbol{br} - y_t^i + 1$. Since calls at different paths use distinct rounds, the values $c(i)$ (resp. $d(i)$) are all distinct. Moreover, the chosen values lie in $\{\boldsymbol{br} - x + 1 \mid x \in [\boldsymbol{br}]\} = [\boldsymbol{br}]$ and $\{\boldsymbol{br} - x + 1 \mid x \in [\alpha, \boldsymbol{br}]\} = [\boldsymbol{br} - \beta]$, respectively. Finally, feasibility of the broadcasting scheme ensures $c(i) + d(i) = (\boldsymbol{br} - y_s^i + 1) + (\boldsymbol{br} - y_t^i + 1) \geq \ell_i$, so the functions c and d indeed form a valid covering of S. This establishes the equivalence. □

4.2.1 PTAS for Double Prefix Covering

In the light of Lemmas 4.8 and 4.10, presenting a PTAS for broadcasting in k-path graphs reduces to presenting a PTAS for the Double Prefix Covering problem. This is the focus of the present section. Specifically, we prove Theorem 4.13, which establishes the existence of a $(1 + \epsilon)$-approximation algorithm for Double Prefix Covering for any $\epsilon > 0$. Our approach closely parallels the rounding and covering techniques used in Section 4.1.1 for k-cycle graphs. In particular, it also addresses the covering problem through a two-stage approximation scheme that leverages the concept of rounded sets (see Definition 4.3) to reduce the problem complexity. The main idea is to transform the original ground multiset and the two covering sets into simplified versions with constant support (i.e., bounded numbers of distinct elements), so that exhaustive search becomes computationally feasible while still preserving provable approximation guarantees.

Our PTAS for Double Prefix Covering proceeds by reducing the original instance to one with bounded support. Given an input multiset S, we first form its *rounded version* $\mathfrak{R}_p(S)$, where each element is rounded up to a larger element of S so that the number of distinct values becomes p. Here p depends only on ϵ. Although rounding inflates item sizes, we show that the optimal covering cost for $\mathfrak{R}_p(S)$ is at most a $(1 + 1/p)$ factor larger than that of S (Proposition 4.11). Thus, solving the rounded instance suffices for approximation.

To search for the optimal covering, we use binary search over candidate values m, where m determines the covering sets $C = [m]$ and $D = [m - \beta]$. For each candidate m, we construct rounded sets $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$, each with support size p. We prove that if $\mathfrak{R}_p(S)$ can be covered using $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$, then the original instance can be covered using $C' = [m']$ and $D' = [m' - \beta]$ with $m' \leq (1 + 1/p)m$ (Claim 4.2). Conversely, if we cannot cover $\mathfrak{R}_p(S)$ with the rounded sets, then it is not possible to cover it using $[m]$ and $[m - \beta]$. It therefore suffices to test whether $\mathfrak{R}_p(S)$ is coverable using $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$. Since all three sets have constant support (bounded by p), this check can be carried out by exhaustive search in constant time (Lemma 4.12). The remainder of this section provides details of the algorithm.

Proposition 4.11. *$\boldsymbol{OPT}(\mathfrak{R}_p(S)) \leq \boldsymbol{OPT}(S) + \lceil |S|/p \rceil$, where $\boldsymbol{OPT}$ is the optimal cover for a specific instance of* Double Prefix Covering.

Proof. By definition, $\mathfrak{R}_p(S)$ consists of p parts. Given an optimal cover of S, form a cover of $\mathfrak{R}_p(S)$ by assigning the numbers given to the part i in S to the part $(i+1)$ in $\mathfrak{R}_p(S)$ for $i \in [p-1]$. As the size and value of part $i+1$ is not greater than the value of part i, the set of numbers that can cover part i can also cover part $i+1$.

Therefore, the first part is the only part that needs to be covered. We show that adding an extra $\lceil |S|/p \rceil$ to the covering set is enough to cover the first part. This is because every $\ell_i \in S$ is less than $2\boldsymbol{OPT}(S) - \beta$, as otherwise it would not be possible to cover v_i with a number in C and a number in D for the optimal value of m. Thus, by adding an extra $\lceil |S|/p \rceil$ items to the m, we can cover $\mathfrak{R}_p(S)$. ☐

Since $\boldsymbol{OPT}(S) \geq |S|$, it follows that any covering of $\mathfrak{R}_p(S)$ yields a $(1 + 1/p)$-approximation for covering S. This allows us to restrict attention to $\mathfrak{R}_p(S)$, which has constant support p.

To cover $\mathfrak{R}_p(S)$, we apply a binary strategy. For a candidate value m, we design a procedure that either produces a covering using $[m + \lceil m/p \rceil]$, or certifies that no covering exists with $[m]$. By iteratively refining m, we identify the smallest value for which $\mathfrak{R}_p(S)$ can be covered using the rounded covering sets $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$, where $C = [m]$ and $D = [m - \beta]$.

Claim 4.2. *Given that we can cover S with $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$, we can cover S using $C' = [m']$ and $D' = [m' - \beta]$, where $m' = m + \lceil m/p \rceil$.*

Proof. Based on definition, $\mathfrak{R}_p(C)$ contains p parts. Suppose that we have a covering with $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$ for S. Associate value $m + \lceil m/p \rceil - (i-1)$ to the i'th largest number in C for $i \in [m]$. This transformation only increases the value as the first block with original value m matches to $m+1, \ldots, m + \lceil m/p \rceil$, the second block matches to $m - \lceil m/p \rceil + 1, \ldots, m$, and so on. More specifically, the i'th block matches to $m - (i-1)\lceil m/p \rceil + 1, \ldots, m - (i-2)\lceil m/p \rceil$, which is greater than the original value of the block. Similarly, associate value $m + \lceil m/p \rceil - \beta - (i-1)$ to the i'th largest number in D for $i \in [m - \beta]$. As the new sets are subsets of $C' = [m']$ and $D' = [m' - \beta]$, and every value in $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$ is now larger than its original value, the same covering can cover S with the corresponding values in C' and D'. ☐

Lemma 4.12. *Given three sets $\mathfrak{R}_p(S)$, $\mathfrak{R}_p(C)$, and $\mathfrak{R}_p(D)$, there exists an algorithm that, in time $\mathcal{O}(2^{p^3} |S| (|S| + |C| + |D|))$, either outputs a covering of $\mathfrak{R}_p(S)$ using the covering sets $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$ or correctly reports that no such covering is possible.*

Proof. The ground set $\mathfrak{R}_p(S)$ and the covering sets have constant support p. For any fixed distinct member of $\mathfrak{R}_p(S)$ like v and each combination pair of members of $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$, we can encode the number of times the pair is used to cover the members of $\mathfrak{R}_p(S)$ with value v in $\log |S|$ bits. This is because we have at most $|S|$ members with value equal to v.

Given that there are p^2 distinct pairs from the covering set and p types of value in $\mathfrak{R}_p(S)$, any potential solution for the problem can be encoded in $p^2.p.\log |S| = p^3 \log |S|$ bits; we can exhaustively try all and answer whether $\mathfrak{R}_p(C)$ can cover $\mathfrak{R}_p(S)$ by trying all potential solutions.

Thus, the algorithm tries $\mathcal{O}(2^{p^3}|S|)$ solutions. And, each solution takes linear time to check. Therefore, overall it takes $(2^{p^3}|S|(|S| + |C| + |D|))$ to decide whether it is possible to cover $\mathfrak{R}_p(S)$ using $\mathfrak{R}_p(C)$. ☐

To summarize, for a candidate covering size m and ground set $\mathfrak{R}_p(S)$, we construct $\mathfrak{R}_p(C)$ and $\mathfrak{R}_p(D)$ and test, via exhaustive search (Lemma 4.12), whether $\mathfrak{R}_p(S)$ can be covered. If no covering is found, we conclude that $\mathfrak{R}_p(S)$ cannot be covered with $[m]$ (bad guess). If a covering is found, we extend it to a covering with $m' = m + \lceil m/p \rceil$, thereby obtaining a $(1 + 1/p)$-approximation for $\mathfrak{R}_p(S)$. Combining this with Proposition 4.11, we achieve an overall approximation factor of $(1 + 1/p)^2$, which approaches 1 as p grows.

Theorem 4.13. *There is a PTAS for the* Double Prefix Covering *problem that terminates in* $\mathcal{O}(2^{p^3}|S|(|S| + \boldsymbol{OPT}) \log \boldsymbol{OPT})$*, where* $p = 3/\epsilon^2$.

Proof. Given an instance S of Prefix Covering with $\epsilon > 0$, set $p = \lceil 3/\epsilon^2 \rceil$ to ensure $(1+1/p)^2 \leq 1 + \epsilon$. Construct $\mathfrak{R}_p(S)$ by rounding up elements in S. We use binary search over m to find $\boldsymbol{OPT}_p(\mathfrak{R}_p(S))$, where $\boldsymbol{OPT}_p(S)$ is the minimum m such that S can be covered using $\mathfrak{R}_p([m])$ and $\mathfrak{R}_p([m - \beta])$. For each candidate set $[m]$:

(i). Construct $\mathfrak{R}_p(C)$ where $C = [m]$ as per Definition 4.3.

(ii). Use exhaustive search (Lemma 4.5) to determine if $\mathfrak{R}_p(C)$ can cover $\mathfrak{R}_p(S)$ in time $\mathcal{O}(2^{p^3}|S|(|S| + m))$.

Therefore, the binary search routine gives the smallest m such that $\mathfrak{R}_p(S)$ can be covered using $\mathfrak{R}_p([m])$ and $\mathfrak{R}_p([m - \beta])$. We construct a covering of $\mathfrak{R}_p(S)$ using $C' = [m']$ and $D' = [m' - \beta]$, where $m' \leq (1 + 1/p)m = (1 + 1/p)\boldsymbol{OPT}_p(\mathfrak{R}_p(S)) \leq (1 + 1/p)\boldsymbol{OPT}(\mathfrak{R}_p(S))$ by Claim 4.2. Given that $C' = [m']$ and $D' = [m' - \beta]$ cover $\mathfrak{R}_p(S)$, they cover S as well. Using Proposition 4.11, we have $m' \leq (1 + 1/p)\boldsymbol{OPT}(\mathfrak{R}_p(S)) \leq (1 + 1/p)^2\boldsymbol{OPT}(S)$. The running time is $\mathcal{O}(\log \boldsymbol{OPT})$ binary search iterations, each taking $\mathcal{O}(2^{p^3}|S|(|S| + \boldsymbol{OPT}))$ time, giving total complexity $\mathcal{O}(2^{p^3}|S|(|S| + \boldsymbol{OPT}) \log \boldsymbol{OPT})$ where $p^3 = \mathcal{O}(1/\epsilon^6)$. ☐

Combining Lemma 4.10 and Theorem 4.13 implies the following Corollary.

Corollary 4.14. *There exists a PTAS for* Telephone Broadcasting *in k-path graphs. In particular, for a k-path graph on n vertices and any $\epsilon > 0$, the algorithm computes a $(1 + \epsilon)$-approximate broadcast schedule in time* $O(2^{1/\epsilon^6} k(k + n) \log n)$.

5 Graphs of Bounded Bandwidth

In this section, we present a polynomial algorithm that computes the optimal broadcasting time for graphs of bounded bandwidth. Intuitively, the bandwidth constraint limits the degree of G (in contrast to other families such as k-cycle or k-path graphs), and a bandwidth ordering exposes a small "active window" of vertices at each step. This property enables a dynamic programming (DP) approach. For the remainder of this section, we assume that a bandwidth-k ordering $(v_1, \ldots, v_n)$ of the vertices is given. For clarity of exposition, we also assume that the source of the broadcast is v_1; this restriction will be relaxed later.

DP Table. We define a DP table with entries of the form $dp[i][s]$, where $i \in [0, n]$ and s is a *state*. The value $dp[i][s]$ denotes the minimum broadcast time for informing the first i vertices in the ordering while ending in configuration s. The state s encodes all necessary information about the *boundary vertices* $B_i = \{v_{i-k+1}, \ldots, v_i\}$, namely, how these vertices are informed and how they connect to the rest of a potential broadcast tree. Intuitively, a state describes the partial structure of a broadcast forest spanning boundary vertices together with their timing information.

We fill the table row by row, starting from $i = k$ up to $i = n$. The last row corresponds to the entire graph; the minimum value among its states equals the optimal broadcast time.

Definition of states. Fix i, and let $u_1, \ldots, u_k$ denote the boundary vertices. Let $Bf = \{x_1, \ldots, x_k\}$ be the k vertices immediately before B_i and $Af = \{y_1, \ldots, y_k\}$ be the k vertices immediately after B_i. Together, the vertices $x_1, \ldots, x_k,\ u_1, \ldots, u_k,\ y_1, \ldots, y_k$ form the *active window*. Every neighbor of a boundary vertex lies within this window.

A state s encodes a potential partial broadcast tree consistent with the active window by storing the following information:

- **Parents.** For each u_i, its parent $p(u_i)$ (unless it is the source v_1). Each $p(u_i)$ is chosen from at most $2k$ active vertices or a special "no parent" symbol, yielding $(2k+1)^k$ possibilities.
- **Ranks.** For each u_i, its rank among the children of $p(u_i)$, ordered by the rounds in which $p(u_i)$ calls them. Since $p(u_i)$ has at most $2k$ children, there are at most $2k$ options per vertex, and $(2k)^k$ total options.
- **Outgoing calls.** For each u_i, the active vertex that it calls in each round after being informed. For every round slot, there are $2k$ options (one of up to $2k-1$ children or a null option if unused), giving at most $(2k)^{2k+1}$ possibilities per u_i.
- **Inform times.** The time at which the root of each subtree is informed. Since there are at most k subtrees (each containing a boundary vertex), and each root may be at a round in $[n]$, this yields n^k possibilities.

Proposition 5.1. *If S denotes the space of all states, then $|S| \le (2k+1)^k \cdot (2k)^k \cdot (2k)^{k(2k+1)} \cdot n^k$, which is $\mathcal{O}(n^k)$ for constant k.*

For a state s, let $\ell(s)$ denote the latest time at which any active node in the broadcast subtrees of s is informed. Note that the global broadcast time of a scheme realizing s may exceed $\ell(s)$ if the last informed vertex lies outside the active window.

Filling the DP table. A state s is *valid* for step i if its parent–child assignments among nodes in its broadcast forest respect the edges of G. Also, the outgoing calls in the state must match the rank and parent fields. For example, if u_1 and u_2 are not adjacent in G but u_1 is the parent of u_2 in s, then s is invalid at i. For any invalid state at step i, we set $dp[i][s] = +\infty$.

Base case. When $i = k$, any valid state represents a single broadcast tree rooted at v_1 (informed at time 0). For such states, we set $dp[k][s] = \ell(s)$.

Transitions. To compute $dp[i+1][s]$ for a valid state s, we look for states s' that are valid at step i and *compatible* with s. Compatibility means that information concerning the overlap between active states at steps i and $i+1$ matches in both states: the inform times of shared vertices, their parent–child relations, and the directions and timings of edges incident to them must all agree. Further, indices shift in the natural way: for example, u_1 at step i corresponds to x_k at step $i+1$, and u_2 at step i corresponds to u_1 at step $i+1$. Vertices x_1 and y_k enter or leave the active window accordingly. See Figure 4 for an illustration.

If s^* is a compatible state with s with minimum $dp[i][s^*]$, we set $dp[i+1][s] = \max\{dp[i][s^*], \ell(s)\}$. We also record previous$(s) = s^*$, which allows reconstruction of the optimal broadcast tree. If no compatible state exists, then $dp[i+1][s] = +\infty$. Intuitively, realizing state s at $i+1$ requires extending from some compatible predecessor state at i, and the broadcast time is determined by whichever is larger: the time already incurred or the latest inform time among the active vertices.

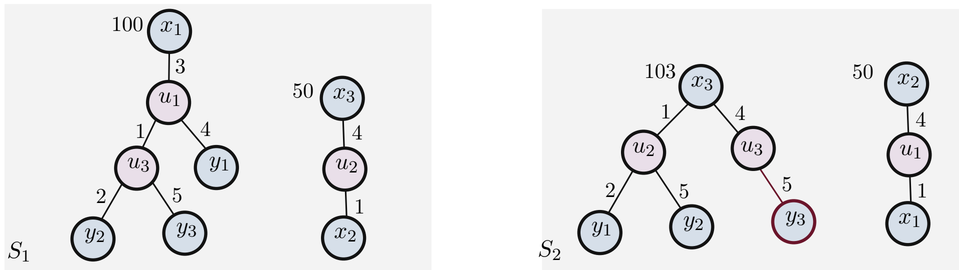


Figure 4: An illustration of states and their compatibility for a graph of bandwidth $k = 3$. The bandwidth ordering is $(\ldots, x_1, x_2, x_3, u_1, u_2, u_3, y_1, y_2, y_3, \ldots)$. The boundary vertices u_i are highlighted in red. When computing DP values at step $i+1$, state S_2 is compatible with state S_1. Note that the shift in the active window implies, for example, that u_1 in S_1 at step i becomes x_3 in S_2 at step $i+1$, u_2 becomes u_1, x_2 becomes x_1, and so on.

Example 9. *In Figure 4, suppose that at step $i+1$, the best compatible predecessor of state S_2 is S_1 with $dp[i][S_1] = 120$. Since $\ell(S_2) = 109$, we obtain $dp[i+1][S_2] = \max\{120, 109\}$.*

Sources other than v_1. If the broadcast source is not v_1, then we consider all $O((2k)!)$ possible neighbor orderings in which the source s informs its neighbors. For a fixed ordering, the problem splits into two independent subproblems: one for vertices preceding s in the bandwidth ordering and one for vertices following it. The DP applies symmetrically to both sides. The only modification is in the base case: a state is valid only if it is consistent with the fixed ordering of s's initial calls. Since k is constant, this does not affect the asymptotic time complexity.

Using an inductive argument, one can establish the correctness of the above approach.

Theorem 5.2. Telephone Broadcasting *can be solved optimally on graphs of bandwidth k in time $f(k) \cdot n^{k^2+O(k)}$. In particular, for a constant k, there is a polynomial-time algorithm for computing an optimal broadcast schedule.*

Proof.

Setup. Recall that we have a fixed bandwidth-k ordering $(v_1, \ldots, v_n)$. As discussed, at step i the DP keeps states s that summarize the broadcast restricted to the *active window* $W_i := \{x_1, \ldots, x_k, u_1, \ldots, u_k, y_1, \ldots, y_k\}$, where $B_i = \{u_1, \ldots, u_k\}$ is the set of boundary nodes. Recall that, for a state s, $\ell(s)$ is the latest inform time among active vertices encoded by s.

Correctness. We use induction to prove the following invariant:

Inductive invariant. For every $i \geq k$ and every valid state s at step i, the entry $dp[i][s]$ of the DP table is the minimum broadcast time of any feasible broadcast scheme for informing $v_1, \ldots, v_i$ which ends up at state s.

Base $i = k$. In the base case, the round in which each vertex of $v_1, \ldots, v_k$ is informed is embedded into the state s. This is because they are all boundary vertices. Therefore, the minimum broadcast time over schedules realizing s is exactly $\ell(s)$, so $dp[k][s] = \ell(s)$ as set by the algorithm.

Step $i \to i+1$. Let s be any valid state at step $i+1$. Choose a compatible predecessor s^* at step i that minimizes $dp[i][s^*]$.

By the inductive hypothesis there is broadcast scheme $\mathcal{S}^*$ that realizes s^* and informs $v_1, \ldots, v_i$ in $dp[i][s^*]$ rounds. Extend $\mathcal{S}^*$ inside the active window to realize s at step $i+1$ (this is possible by compatibility and validity). The broadcast time of the extended scheme is $\max(dp[i][s^*], \ell(s))$, which is exactly the value assigned by the recurrence. Hence, $dp[i{+}1][s]$ upper-bounds the true optimum among schedules realizing s.

On the other hand, let $\mathcal{S}$ be any feasible schedule on G_{i+1} that realizes s with minimum broadcast time τ. Let s' be the state of $\mathcal{S}$ at step i; then s' is valid at i and compatible with s. By the inductive hypothesis, $dp[i][s']$ is no more than the broadcast time of $\mathcal{S}$ for informing $v_1, \ldots, v_i$. Since realizing s from s' cannot finish later than $\max\{dp[i][s'], \ell(s)\}$, we get $dp[i{+}1][s] \leq \max\{dp[i][s'], \ell(s)\} \leq \tau$. Thus, $dp[i{+}1][s]$ lower-bounds the optimum as well. This proves the inductive invariant.

We conclude that taking the minimum $dp[n][s]$ over valid states s yields the optimal broadcast time on G.

Sources other than v_1. If the source is $s \neq v_1$, we enumerate the $O((2k)!)$ possible orders of s's calls (only neighbors within distance k in the bandwidth ordering matter). For each fixed order, the same inductive argument applies independently to the two sides of s, and we take the best over the constant many orders. This preserves optimality.

Running time. Let S be the set of all states. By Proposition 5.1, we have $|S| \leq (2k{+}1)^k \cdot (2k)^k \cdot (2k)^{k(2k+1)} \cdot n^k \;=\; g(k)\, n^k$. That is, the size of the DP table would be at most $g(k)n^{k+1}$. For every i and s, comparing every state s' of the previous iteration (i.e. $i-1$) to find out whether they are compatible would take $O(|S|.n)$ time. Therefore, the cost of computing dp for all states is $(g(k))^2\, n^{2k+2}$. Overall, this yields a total time $f(k) \cdot n^{O(k)}$, for some function $f(k)$ independent of n. Note that the extra $O((2k)!)$ factor for non-v_1 sources is absorbed into $f(k)$ since k is constant. This matches the claimed bound. □

6 Concluding Remarks

In this work, we advanced the study of Telephone Broadcasting by establishing new hardness results and developing approximation schemes for specific graph families. On the hardness side, we resolved long-standing questions by proving NP-hardness for k-cycle and k-path graphs. On the algorithmic side, we presented polynomial-time approximation schemes (PTAS) for Telephone Broadcasting in both k-cycle and k-path graphs. We also introduced an exact algorithm for graphs of bounded bandwidth, thereby extending the tractable frontier of the problem.

As future work, it would be interesting to adapt the integer covering formulations to other broadcasting models or broader sparse graph classes. In particular, for series–parallel graphs, our hardness result via k-path graphs rules out tractability, yet, to the best of our knowledge, no constant-factor approximation is currently known.

References

[1] Aida Aminian, Shahin Kamali, Seyed-Mohammad Seyed-Javadi, et al. On the complexity of telephone broadcasting from cacti to bounded pathwidth graphs. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, pages 10–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025.

[2] Puspal Bhabak and Hovhannes A. Harutyunyan. Constant approximation for broadcasting in k-cycle graph. In *Algorithms and Discrete Applied Mathematics*, pages 21–32. Springer International Publishing, 2015.

[3] Puspal Bhabak and Hovhannes A Harutyunyan. Approximation algorithm for the broadcast time in k-path graph. *Journal of Interconnection Networks*, 19(04):1950006, 2019.

[4] Maja Čevnik and Janez Žerovnik. Broadcasting on cactus graphs. *Journal of Combinatorial optimization*, 33:292–316, 2017.

[5] Peter Damaschke. A linear-time optimal broadcasting algorithm in stars of cliques. *Journal of Graph Algorithms and Applications*, 28(1):385–388, 2024.

[6] Yudai Egami, Tatsuya Gima, Tesshu Hanaka, Yasuaki Kobayashi, Michael Lampis, Valia Mitsou, Edouard Nemery, Yota Otachi, Manolis Vasilakis, and Daniel Vaz. Broadcasting under structural restrictions. *arXiv preprint arXiv:2504.13669*, 2025.

[7] Michael Elkin and Guy Kortsarz. A combinatorial logarithmic approximation algorithm for the directed telephone broadcast problem. *SIAM Journal on Computing*, 35(3):672, 2005.

[8] Michael Elkin and Guy Kortsarz. Sublogarithmic approximation for telephone multicast. *Journal of Computer and System Sciences*, 72(4):648–659, 2006.

[9] Fedor V Fomin, Pierre Fraigniaud, and Petr A Golovach. Parameterized complexity of broadcasting in graphs. *Theoretical Computer Science*, 997:114508, 2024.

[10] Fraigniaud and Mitjana. Polynomial-time algorithms for minimum-time broadcast in trees. *Theory of Computing Systems*, 35(6):641–665, 2002.

[11] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[12] Saber Gholami, Hovhannes A Harutyunyan, and Edward Maraachlian. Optimal broadcasting in fully connected trees. *Journal of Interconnection Networks*, 23(01):2150037, 2023.

[13] Hovhannes Harutyunyan, George Laza, and Edward Maraachlian. Broadcasting in necklace graphs. In *Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering*, pages 253–256, 2009.

[14] Hovhannes A. Harutyunyan and Narek A. Hovhannisyan. Improved approximation for broadcasting in k-path graphs. In *Combinatorial Optimization and Applications - 16th International Conference, COCOA 2023, Hawaii, HI, USA, December 15-17, 2023, Proceedings, Part II*, volume 14462, pages 111–122. Springer.

[15] Hovhannes A Harutyunyan, Narek Hovhannisyan, and Edward Maraachlian. Broadcasting in chains of rings. In *Proceedings of the Fourteenth International Conference on Ubiquitous and Future Networks*, pages 506–511, 2023.

[16] Sandra M Hedetniemi, Stephen T Hedetniemi, and Arthur L Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.

[17] Arthur L. Liestman and Joseph G. Peters. Broadcast networks of bounded degree. *SIAM Journal on Discrete Mathematics*, 1(4):531–540, 1988.

[18] Weiping Shang, Pengjun Wan, and Xiaodong Hu. Approximation algorithms for minimum broadcast schedule problem in wireless sensor networks. *Frontiers of Mathematics in China*, 5:75–87, 2010.

[19] Peter J. Slater, Ernest J. Cockayne, and Stephen T. Hedetniemi. Information dissemination in trees. *SIAM Journal on Computing*, 10(4):692–701, 1981.

[20] Elena Stöhr. Broadcasting in the butterfly network. *Information Processing Letters*, 39(1): 41–43, 1991.

[21] Prafullkumar Tale. Telephone broadcast on graphs of treewidth two. *Theoretical Computer Science*, page 115282, 2025.

[22] Wenci Yu, Han Hoogeveen, and Jan Karel Lenstra. Minimizing makespan in a two-machine flow shop with delays and unit-time operations is NP-hard. *Journal of Scheduling*, 7(5):333–348, 2004.