



Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Parameterized complexity of broadcasting in graphs ☆☆☆

Fedor V. Fomin^a, Pierre Fraignaud^b, Petr A. Golovach^{a,*}^a Department of Informatics, University of Bergen, PB 7803, 5020 Bergen, Norway^b Institut de Recherche en Informatique Fondamentale, Université Paris Cité and CNRS, 8 Pl. Aurélie Nemours, 75013 Paris, France

ARTICLE INFO

Communicated by S. Saurabh

Keywords:

Broadcasting

Telephone model

Parameterized complexity

ABSTRACT

The task of the broadcast problem is, given a graph G and a source vertex s , to compute the minimum number of rounds required to disseminate a piece of information from s to all vertices in the graph. It is assumed that, at each round, an informed vertex can transmit the information to at most one of its neighbors. The broadcast problem is known to NP-hard. We show that the problem is FPT when parameterized by the size k of a feedback edge set, or by the size k of a vertex cover, or by $k = n - t$, where t is the input deadline for the broadcast protocol to complete.

1. Introduction

The aim of *broadcasting* in a network is to transmit a message from a given source node of the network to all the other nodes. Let $G = (V, E)$ be a connected simple graph modeling the network, and let $s \in V$ be the source of a message M . The standard *telephone model* [19] assumes that the communication proceeds in synchronous rounds. At any given round, any node $u \in V$ aware of M can forward M to at most one neighbor v of u . The minimum number of rounds for broadcasting a message from s in G to all other vertices is denoted by $b(G, s)$, and we let $b(G) = \max_{s \in V} b(G, s)$ be the broadcast time of graph G . As the number of informed nodes (i.e., nodes aware of the message) can at most double at each round, $b(G, s) \geq \lceil \log_2 n \rceil$ in n -node networks. On the other hand, since G is connected, at least one uninformed node receives the message at any given round, and thus $b(G) \leq n - 1$. Both bounds are tight, as witnessed by the complete graph K_n and the path P_n , respectively. The problem of computing the broadcast time $b(G, s)$ for a given graph G and a given source $s \in V$ is NP-hard [31] and, moreover, the problem remains hard when restricted to chordal or grid graphs [22]. Also, the results of [27] imply that it is NP-complete to decide whether $b(G, s) \leq t$ for graphs with $n = 2^t$ vertices.

Three lines of research have emerged since the early days of studying broadcasting in the telephone model. One line is devoted to determining the broadcast time of specific classes of graphs judged important for their desirable properties as interconnection networks (e.g., hypercubes, de Bruijn graphs, Cube Connected Cycles, etc.). We refer to the surveys [14,21] for this matter. Another line of research takes inspiration from extremal graph theory. It aims at constructing n -node graphs G with optimal broadcast time $b(G) = \lceil \log_2 n \rceil$ and minimizing the number of edges sufficient to guarantee this property. Let $B(n)$ be the minimum number of edges of n -node graphs with broadcast time $\lceil \log_2 n \rceil$. It is known [16] that $B(n) = \Theta(n L(n))$ where $L(n)$ denotes the number of consecutive leading 1s in the binary representation of $n - 1$. On the other hand, it is still not known whether $B(\cdot)$ is non-decreasing for $2^t \leq n < 2^{t+1}$, for every $t \in \mathbb{N}$. We are interested in a third, more recent line of research, namely the design of algorithms computing

☆ This article belongs to Section A: Algorithms, automata, complexity and games, Edited by Paul Spirakis.

☆☆ A preliminary version of these results appeared as an extended abstract in the proceedings of WG 2023. The research leading to these results has received funding from the Research Council of Norway via the project BWCA (grant no. 314528) and the ANR project DUCAT (ref. ANR-20-CE48-0006).

* Corresponding author.

E-mail addresses: fedor.fomin@uib.no (F.V. Fomin), pierre.fraignaud@irif.fr (P. Fraignaud), petr.golovach@uib.no (P.A. Golovach).

<https://doi.org/10.1016/j.tcs.2024.114508>

Received 6 July 2023; Received in revised form 23 January 2024; Accepted 13 March 2024

Available online 18 March 2024

0304-3975/© 2024 The Author(s).

Published by Elsevier B.V. This is an open access article under the CC BY license

(<http://creativecommons.org/licenses/by/4.0/>).

efficient broadcast protocols. Note that a protocol for broadcasting from a source s in a graph G can merely be represented as a spanning tree T rooted at s , with an ordering of all the children of each node in the tree.

Polynomial-time algorithms are known for trees [31] and some classes of tree-like graphs [4,15,18]. Several (polynomial-time) approximation algorithms have been designed for the broadcast problem. In particular, the algorithm in [24] computes, for every graph G and every source s , a broadcast protocol from s performing in $2b(G, s) + O(\sqrt{n})$ rounds, hence this algorithm has approximation ratio $2 + o(1)$ for graphs with broadcast time $\gg \sqrt{n}$, but $\tilde{\Theta}(\sqrt{n})$ in general. Later, a series of papers tighten the approximation ratio, from $O(\log^2 n / \log \log n)$ [29], to $O(\log n)$ [1], and eventually $O(\log n / \log \log n)$ [10], which is, up to our knowledge the current best approximation ratio for the broadcast problem. Better approximation ratios can be obtained for specific classes of graphs [2,3,17].

Despite all the achievements obtained on the broadcast problem, it has not yet been approached from the parameterized complexity viewpoint [8]. There might be a good reason for that. Since at most 2^t vertices can have received the message after t communication rounds, an instance of the broadcast problem with time-bound t in an n -vertex graph is a no-instance whenever $n > 2^t$. It follows that the broadcast problem has a trivial kernel when parameterized by the broadcast time. This makes the natural parameterization by the broadcast time not very significant. Nevertheless, as we shall show in this paper, there is a parameterization below the natural upper bound for the number of rounds that leads to interesting conclusions.

1.1. Our results

Let TELEPHONE BROADCAST be the following problem: given a connected graph $G = (V, E)$, a source vertex $s \in V$, and a nonnegative integer t , decide whether there is a broadcast protocol from s in G that ensures that all the vertices of G get the message in at most t rounds. We first show that TELEPHONE BROADCAST can be solved in a single-exponential time by an exact algorithm.

Theorem 1. TELEPHONE BROADCAST can be solved in $3^n \cdot n^{O(1)}$ time for n -vertex graphs.

Motivated by the fact that the complexity of TELEPHONE BROADCAST remains open in pretty simple tree-like graphs (e.g., cactus graphs, and therefore outerplanar graphs), we first consider the *cyclomatic* number as a parameter, i.e., the minimum size of a feedback edge set, that is, the size of the smallest set of edges whose deletion results to an acyclic graph. We show that TELEPHONE BROADCAST is FPT when parameterized by this parameter.

Theorem 2. TELEPHONE BROADCAST can be solved in $2^{O(k \log k)} \cdot n^{O(1)}$ time for n -vertex graphs with cyclomatic number at most k .

As far as we know, no NP-hardness result is known on graphs of treewidth at most $k \geq 2$. While we did not progress in that direction, we provide an interesting result for a stronger parameterization, namely the *vertex cover* number of a graph. (Note that, for all graphs, the treewidth never exceeds the vertex cover number.) As for the cyclomatic number, we do not only show that, for every fixed k , the broadcast time can be found in polynomial time on graphs with vertex cover at most k , but we prove a stronger result: the problem is FPT.

Theorem 3. TELEPHONE BROADCAST can be solved in $2^{O(k^3)} \cdot n^{O(1)}$ time for n -vertex graphs with a vertex cover of size at most k .

Finally, we focus on graphs with very large broadcast time, for which the algorithm in [24] provides hope to derive very efficient broadcast protocols as this algorithm constructs a broadcast protocol performing in $2b(G) + O(\sqrt{n})$ rounds. While we were not able to address the problem over the whole range $\sqrt{n} \ll t \leq n - 1$, we were able to provide answers for the range $n - O(1) \leq t \leq n - 1$. More specifically, we consider the parameter $k = n - t$, and study the kernelization for the problem under such parametrization.

Theorem 4. TELEPHONE BROADCAST admits a kernel with $O(k)$ vertices in n -vertex graphs when parameterized by $k = n - t$.

As a direct consequence of Theorem 4, TELEPHONE BROADCAST is FPT for the parameterization by $k = n - t$. Specifically the problem can be solved in $2^{O(k)} \cdot n^{O(1)}$ time.

1.2. Related work

A classical generalization of the broadcast problem is the *multicast* problem, in which the message should only reach a given subset of target vertices in the input graph. Many of the previously mentioned approximation algorithms for the broadcast problem extend to the multicast problem, and, in particular, the algorithm in [10] is an $O(\log k / \log \log k)$ -approximation algorithm for the multicast problem with k target nodes.

Many variants of the telephone model have been considered in the literature, motivated by different network technologies. One typical example is the *line* model [11], in which a call between a vertex u and a vertex v is implemented by a path between u and v in the graph, with the constraint that all calls performed at the same round must be performed along edge-disjoint paths. (The intermediate nodes along the path do *not* receive the message, which “cut through” them.) Interestingly, the broadcast time of

every n -node graph is exactly $\lceil \log_2 n \rceil$. The result extends to networks in which the paths are constructed by an underlying routing function [6]. The vertex-disjoint variant of the line model, i.e., the line model in which the calls performed at the same round must take place along vertex-disjoint paths, is significantly more complex. There is an $O(\log n / \log \log n)$ -approximation algorithm for the vertex-disjoint line model [24], which naturally extend to an $O(\log n / \log \text{OPT})$ -approximation algorithm — see also [13] where an explicit $O(\log n / \log \text{OPT})$ -approximation algorithm is provided. It is also worth mentioning that the broadcast model has been also extensively studied in models aiming at capturing any type of node- or link-latencies, e.g., the message takes λ_e units of time to traverse edge e , and the algorithm in [1] also handles such constraints. Other variants take into account the size of the message, e.g., a message of L bits takes time $\alpha + \beta \cdot L$ to traverse an edge (see [23]). Under such a model, it might be efficient to split the original message into smaller packets and pipeline the broadcast of these packets through disjoint spanning trees [23,32].

2. Preliminaries

Parameterized complexity We refer to the book of Cygan et al. [8] for a detailed introduction to Parameterized Complexity and give here only crucial definitions.

Formally, a *parameterized problem* is language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ^* is the set of strings over a finite alphabet Σ . Hence, an instance of a parameterized problem is a pair (x, k) , where $x \in \Sigma^*$ is a string encoding the input and $k \in \mathbb{N}$ is a *parameter*. A parameterized problem is said to be *fixed-parameter tractable* (or FPT) if it can be solved in $f(k) \cdot |x|^{O(1)}$ time for some computable function $f(\cdot)$.

A *kernelization* algorithm (or simply a *kernel*) for a parameterized problem L is a polynomial algorithm that maps each instance (x, k) of L into an instance (x', k') of the same problem such that

- (i) $(x, k) \in L$ if and only if $(x', k') \in L$, that is, the instances (x, k) and (x', k') are equivalent, and
- (ii) $|x'| + k' \leq g(k)$ for a computable function $g(\cdot)$.

It is said that $g(\cdot)$ is the *kernel size*, and a kernel is *polynomial* if $g(\cdot)$ is polynomial. It is well-known that every decidable parameterized problem is FPT if and only if it has a kernel. However, there are FPT parameterized problems that do not admit polynomial kernels up to some reasonable complexity assumptions. It is common to present a kernelization algorithm as a series of *reduction rules*. A reduction rule for a parameterized problem is an algorithm that takes an instance of the problem and computes in polynomial time another instance that is more “simple” in a certain way. A reduction rule is *safe* (or *sound*) if the computed instance is equivalent to the input instance.

Graphs We use standard graph-theoretic notation and refer to the textbook of Diestel [9] for non-defined notions. We consider only finite simple undirected graphs. We use $V(G)$ and $E(G)$ to denote the sets of vertices and edges of a graph G . We denote by n and m the number of vertices and edges, respectively, if this does not create confusion. For a graph G and a subset $X \subseteq V(G)$ of vertices, we write $G[X]$ to denote the subgraph of G induced by X . We use $G - X$ to denote the graph obtained by deleting the vertices of X , that is, $G - X = G[V(G) \setminus X]$; we write $G - v$ instead of $G - \{v\}$ for a single element set. We use the same convention for edges and write $G - S$ and $G - e$ for the graph obtained from G by the removal of a set of edges S and a single edge e , respectively. For a vertex v , $N_G(v) = \{u \in V(G) \mid uv \in E(G)\}$ is the *open neighborhood* of v , and $d_G(v) = |N_G(v)|$ is the *degree* of v . For a set $S \subseteq V(G)$, $N_G(S) = \bigcup_{v \in S} N_G(v) \setminus S$. Two vertices u and v are (*false*) *twins* if $N_G(u) = N_G(v)$. It is said that $G' = G/e$ is obtained by the *contraction* of an edge $e = uv \in E(G)$ if G' is obtained by removing u and v and replacing them by a single vertex adjacent to all the vertices of $N_G(\{u, v\})$. Recall that in TELEPHONE BROADCAST we have a specific source vertex s . To keep such a vertex under contraction, we assume that the new vertex obtained by the contraction of an edge incident to s becomes the source and we use the same name s for it.

We write $P = v_1 \cdots v_k$ to denote a *path* with the vertices v_1, \dots, v_k and edges $v_1v_2, \dots, v_{k-1}v_k$; v_1 and v_k are the *end-vertices* of P and the vertices v_2, \dots, v_{k-1} are *internal*. We consider only simple paths, that is, the vertices v_1, \dots, v_k are distinct. We say that P is an (v_1, v_k) -*path*. The *length* of P is the number of edges in P , and the *distance* $\text{dist}_G(u, v)$ between two vertices u and v is the minimum length of a (u, v) -path. We remind that G is *connected* if for every two vertices u and v , G has a (u, v) -path. We always assume that the considered graphs are connected if it is not explicitly said to be otherwise. If G is disconnected, then the inclusion-wise maximal induced connected subgraphs of G are (*connected*) *components*. An edge $e = uv$ is a *bridge* of a connected graph G if $G - e$ is disconnected; note that $G - e$ has exactly two connected components and one of them contains u and the other v . We also remind that T is a *spanning tree* of G if T is a tree subgraph of G with the same set of vertices as G .

A *matching* M in a graph G is a set of edges with distinct endpoints. A vertex v is *saturated* in a matching M if v is incident to an edge of M .

A set of vertices S of a graph G is a *vertex cover* if each edge of G has at least one of its endpoints in S . The *vertex cover number* of G is the minimum size of a vertex cover. Note that for a vertex cover S , the set $I = V(G) \setminus S$ is an *independent set*, that is, any two distinct vertices of I are not adjacent. It is well-known (see [8]) that it is FPT to decide whether G has a vertex cover of size at most k when the problem is parameterized by k . The currently best algorithm, given by Chen, Kanj, and Xia [5], runs in $1.2738^k \cdot n^{O(1)}$ time.

A set of edges S of a graph G is a *feedback edge set* if $G - S$ has no cycle. The *cyclomatic number* of a graph G is the minimum size of a feedback edge set. It is well-known, that for a connected graph G , the cyclomatic number is $m - n + 1$ and a feedback edge set can be found in linear time by constructing a spanning tree (see, e.g., [7,9]).

Broadcasting Let G be a graph and let $s \in V(G)$ be a source vertex from which a message is broadcasted. In general, a broadcasting protocol is a mapping that for each round $i \geq 1$, assigns to each vertex $v \in V(G)$ that is either a source or has received the message in rounds $1, \dots, i - 1$, a neighbor u to which v sends the message in the i -th round. However, it is convenient to note that it can be assumed that each vertex v that got the message, in the next $d \leq d_G(v)$ rounds, transmits the message to some neighbors in a certain order in such a way that each vertex receives the message only once. This allows us to formally define a *broadcasting protocol* as a pair $(T, \{C(v) \mid v \in V(T)\})$, where T is a spanning tree of G rooted in s and for each $v \in V(T)$, $C(v)$ is an ordered set of children of v in T . As soon as v gets the message, v starts to send it to the children in T in the order defined by $C(v)$. For G and $s \in V(G)$, we use $b(G, s)$ to denote the minimum integer $t \geq 0$ such that there is a broadcasting protocol such that every vertex of G gets the message after t rounds. We say that a broadcasting protocol ensuring that every vertex gets a message in $b(s, G)$ rounds is *optimal*. We use the following straightforward observation.

Observation 1. *Let T be a spanning tree of G and let $s \in V(G)$. Then $b(G, s) \leq b(T, s)$.*

Observation 1 can be generalized as follows.

Observation 2. *Let T be a tree subgraph of G and let $s \in V(T)$. Then $b(G, s) \leq b(T, s) + |V(G) \setminus V(T)|$.*

Proof. Let $P = (T, \{C(v) \mid v \in V(T)\})$ an optimal broadcasting protocol for T . We extend it to a protocol for G as follows. First, we construct a spanning subtree T' of G by extending T , that is, T is a subtree of T' . For every $v \in V(T)$, we construct the ordered set of the children $C'(v)$ by appending to the end of $C(v)$ the children of v in T' that are not in T in arbitrary order. For every $v \in V(T') \setminus V(T)$, we define $C'(v)$ to be an arbitrary ordered set of the children of v . We have that $(T', \{C'(v) \mid v \in V(T')\})$ is a broadcasting protocol for G that takes at most $b(T, s) + |V(G) \setminus V(T)|$ rounds to transfer the message to the vertices of G because the vertices of T get the message in the first $b(T, s)$ rounds and in every subsequent round, at least one vertex of $V(G) \setminus V(T)$ receives the message unless all the vertices are already aware of the message. \square

As it was proved by Proskurowski [28] and Slater, Cockayne, and Hedetniemi [31], $b(G, s)$ can be computed in linear time for trees by dynamic programming.

Lemma 1 ([28,31]). *For an n -vertex tree T and $s \in V(T)$, $b(T, s)$ can be computed in $\mathcal{O}(n)$ time.*

3. Exact algorithm for the broadcast problem

We prove that TELEPHONE BROADCAST can be solved in a single-exponential time by an exact algorithm, hence establishing Theorem 1. We do so using dynamic programming over subsets (see the textbook [12] for an introduction to exact exponential algorithms).

3.1. Proof of Theorem 1

Let (G, s, t) be an instance of TELEPHONE BROADCAST. For each $i \in \{0, \dots, t\}$, we enumerate all subsets of vertices $X \subseteq V(G)$ containing s such that $G[X]$ is a connected graph and $b(G[X], s) \leq i$. We denote these families of sets \mathcal{L}_i for $i \in \{0, \dots, t\}$. Observe that (G, s, t) is a yes-instance if and only if $V(G) \in \mathcal{L}_t$. For $i = 0$, it is straightforward that \mathcal{L}_i contains the unique set $\{s\}$. We consecutively compute \mathcal{L}_i for $i = 1, 2, \dots, t$ using the following claim.

Claim 3.1. *Let $X \subseteq V(G)$ such that $s \in X$ and $G[X]$ is connected, and let $i \geq 1$. Then $b(G[X], s) \leq i$ if and only if there is $Y \subseteq X$ such that (i) $s \in Y$, $G[Y]$ is connected, $b(G[Y], s) \leq i - 1$, and (ii) either $X = Y$ or the bipartite graph H with the set of vertices X and the edges $E(H) = \{uv \in E(G) \mid u \in Y, v \in X \setminus Y\}$ has a matching saturating every vertex of $X \setminus Y$.*

Proof of Claim 3.1. Suppose that for $X \subseteq V(G)$, it holds that $s \in X$, $G[X]$ is connected, and $b(G[X], s) \leq i$. Consider an optimal broadcasting protocol for $G[X]$ and s . Let $Y \subseteq X$ be the set of vertices that receive the message in the first $i - 1$ rounds. We have that (i) holds for Y . Condition (ii) is trivial if $X = Y$. Assume that this is not the case and $X \setminus Y \neq \emptyset$. Then for every $v \in X \setminus Y$, there is $u_v \in Y$ such that v receives the message of the i -th round from u_v . We obtain that $M = \{vu_v \mid v \in X \setminus Y\}$ is a matching in H saturating every vertex of $X \setminus Y$.

Assume now that there is $Y \subseteq X$ satisfying (i) and (ii). Then there is a broadcasting protocol for $G[Y]$ and s such that every vertex of Y gets the message in $i - 1$ rounds. If $X = Y$, then $b(G[X], s) \leq i - 1$. Suppose that $X \setminus Y \neq \emptyset$. Then H has a matching M which saturates every vertex of $X \setminus Y$. We extend the protocol for Y and s by defining that each vertex $u \in Y$ which is saturated in M sends the message to the neighbor along the edge of M incident to v in the last available round. Thus each vertex of $X \setminus Y$ gets the message in i rounds. \square

Suppose that $i \geq 1$ and the family \mathcal{L}_{i-1} is already constructed. Initially, we set $\mathcal{L}_i := \mathcal{L}_{i-1}$. Then we consider all subsets $X \subseteq V(G)$ and proper $Y \subset X$ such that $s \in X$, $G[X]$ is connected, and $Y \in \mathcal{L}_{i-1}$. For each pair (X, Y) of such sets, we check whether the

bipartite graph H with the set of vertices X and the edges $E(H) = \{uv \in E(G) \mid u \in Y, v \in X \setminus Y\}$ has a matching saturating every vertex of $X \setminus Y$, and if this holds, then we set $\mathcal{L}_i := \mathcal{L}_i \cup \{X\}$.

Claim 3.1 guarantees correctness of the construction of \mathcal{L}_i from \mathcal{L}_{i-1} . To evaluate the running time, note that we consider at most 3^n pairs of sets (X, Y) . Then for each pair, H can be constructed in polynomial time and then the existence of a matching saturating the vertices of $X \setminus Y$ can be verified in polynomial time in the standard way (see, e.g., [7]). Thus \mathcal{L}_i can be constructed in $3^n \cdot n^{O(1)}$ time. Taking into account that we iterate for $i = 1, 2, \dots, t$ and we can assume without loss of generality that $t \leq n - 1$, we have that the total running time is $3^n \cdot n^{O(1)}$.

4. TELEPHONE BROADCAST parameterized by the cyclomatic number

In this section, we prove Theorem 2. We need some auxiliary results.

Let T be a tree and let x and y be distinct leaves, that is, vertices of degree one in T . For an integer $h \geq 0$, we use $b_h(T, x, y)$ to denote the minimum number of rounds needed to broadcast the message from the source x to y in such a way that every vertex of T gets the message in at most h rounds. We assume that $b_h(T, x, y) = +\infty$ if $b(T, x) > h$. We prove that $b_h(T, x, y)$ can be computed in linear time similarly to $b(T, s)$ (see [28,31]). The difference is that it is more convenient to use recursion instead of dynamic programming.

Lemma 2. *For an n -vertex tree T with given distinct leaves x and y of T and an integer $h \geq 0$, $b_h(T, x, y)$ can be computed in $\mathcal{O}(n)$ time.*

Proof. Let T be a tree and let x and y be leaves of T . Let also $h \geq 0$ be an integer. We use Lemma 1 to check whether $b(T, x) \leq h$ and set $b_h(T, x, y) = +\infty$ if $b(T, x) > h$ because in this case the message cannot be transmitted to every vertex of T in h rounds. From now we assume that $b(T, x) \leq h$. Notice that this implies that $b_h(T, x, y) < +\infty$.

Let $P = v_1 \dots v_\ell$ be the (x, y) -path in T , $x = v_1$ and $y = v_\ell$. Note that if $\ell = 2$, then, trivially, $b_h(T, x, y) = 1$. Suppose that $\ell \geq 3$. Denote by T' the inclusion-wise maximal subtree of T containing v_2 and y such that v_2 is a leaf of T' . We construct recurrences that allow to compute $b_h(T, x, y)$ from $b_{h'}(T', v_2, y)$ for a certain $h' < h$.

If $d_T(v_2) = 2$, then $b_h(T, x, y) = 1 + b_{h-1}(T', v_2, y)$ by definition.

Suppose that $d_T(v_2) \geq 3$. Denote by u_1, \dots, u_k the neighbors of v_2 distinct from v_1 and v_3 , and let T_1, \dots, T_k be the connected components of $T - v_2$ containing u_1, \dots, u_k , respectively. Notice that $k \leq h - 2$, because at least $k + 2$ rounds are needed to transmit the message to u_1, \dots, u_k and v_3 . Observe also that the message is broadcasted from v_2 to every u_i and to broadcast the message from each u_i to other vertices of T_i , we can use an optimal protocol for T_i with the source u_i . We compute $b(T_i, u_i)$ for $i \in \{1, \dots, k\}$ and assume that $b(T_1, u_1) \geq \dots \geq b(T_k, u_k)$. Observe that $\max\{b(T_i, u_i) + i \mid i \in \{1, \dots, k\}\} \leq h - 1$, because at least $\max\{b(T_i, u_i) + i \mid i \in \{1, \dots, k\}\} + 1$ rounds are necessary to transmit the message to all the vertices of T_1, \dots, T_k .

Because $\max\{b(T_i, u_i) + i \mid i \in \{1, \dots, k\}\} \leq h - 1$, there is $j \in \{1, \dots, k + 1\}$ such that either $j \leq k$ and $\max\{b(T_i, u_i) + i + 1 \mid i \in \{j, \dots, k\}\} \leq h - 1$ or $j = k + 1$. We find the minimum value of j satisfying this condition. Observe that $j + 1$ is the minimum number of a round such that the message can be transmitted from v_2 to v_3 in such a way that every vertex of T_1, \dots, T_k gets the message in at most h rounds. Because we aim to minimize the transmission time to $y = v_k$, we set $b_h(T, x, y) = j + b_{h-j}(T', v_2, y)$.

The correctness of the algorithm follows from its description. To evaluate the running time, note that to compute $b_h(T, x, y)$, we have to find $P = v_1 \dots v_\ell$ and then for each $i \in \{2, \dots, \ell\}$, compute the neighbors u'_1, \dots, u'_{k_i} of v_i distinct from v_{i-1} and v_{i+1} and the components T'_1, \dots, T'_{k_i} of $T - V(P)$ containing u'_1, \dots, u'_{k_i} . All these computations can be done in the total linear time. Also in $\mathcal{O}(n)$ time, we can order T'_1, \dots, T'_{k_i} by the decrease $b(T'_p, u'_p)$ for $p \in \{1, \dots, k_i\}$. Then computing $b_h(T, x, y)$ can be done in linear time following the recurrences. \square

We also need a subroutine computing the minimum number of rounds for broadcasting from two sources with the additional constraint that the second source starts sending the message with a delay. Let T be a tree and let x and y be distinct leaves of T . Let also $h \geq 0$ be an integer. We use $d_h(T, x, y)$ to denote the minimum rounds needed to broadcast the message from x and y to every vertex of $T - y$ in such a way that y can send the message starting from the $(h + 1)$ -th round (that is, we assume that y gets the message from outside in the h -th round).

Lemma 3. *For an n -vertex tree T with given distinct leaves x and y of T and an integer $h \geq 0$, $d_h(T, x, y)$ can be computed in $\mathcal{O}(n^2)$ time.*

Proof. Let T be a tree and let x and y be leaves of T . Let also $h \geq 0$ be an integer. Consider the (x, y) -path $P = v_1 \dots v_\ell$ in T , where $x = v_1$ and $y = v_\ell$. Then we have that $d_h(T, x, y) = b(T - y, x)$ if $b(T - y, x) \leq h$. If $b(T - y, x) > h$, then

$$d_h(T, x, y) = \min_{i \in \{2, \dots, \ell\}} \max\{b(T_1^i, x), b(T_2^i, y) + h\}, \tag{1}$$

where T_1^i is the connected component of $T - v_{i-1}v_i$ containing x and T_2^i is the connected component of $T - v_{i-1}v_i$ containing y . The equality (1) follows from the observation that the sets of vertices receiving the message from x and y , respectively, should form a partition of $V(T)$, where each part induces a subtree. The path P can be found in linear time. Then computing $d_h(T, x, y)$ can be done in $\mathcal{O}(n^2)$ time using Lemma 1. \square

Now we are ready to prove Theorem 2.

4.1. Proof of Theorem 2

Let (G, s, t) be an instance of TELEPHONE BROADCAST. If G is a tree, then we can compute $b(G, s)$ in linear time using Lemma 1. Assume that this is not the case, and let $k = m - n + 1 \geq 1$ be the cyclomatic number of G . We find in linear time a feedback edge set S of size k by finding an arbitrary spanning tree F of G and setting $S = E(G) \setminus E(F)$.

We iteratively construct the set U as follows. Initially, we set

$$U := W = \{s\} \cup \{v \in V(G) \mid v \text{ is an endpoint of an edge of } S\}.$$

Then while $G - U$ has a vertex v such that G has three internally vertex-disjoint paths joining v and U , we set $U := U \cup \{v\}$. The properties of U are summarized in the following claims.

Claim 4.1. $|U| \leq 4k$.

Proof of Claim 4.1. The claim is trivial if $U = W$ because $|U| \leq 2k + 1 \leq 4k$ in this case. Assume that $|U| > |W|$, that is, $G - W$ has a vertex v such that G has three internally vertex-disjoint paths joining v and W . Consider the graph G' obtained from G by the iterative deletion of all the vertices of degree one that are not in W . Observe that $F = G' - S$ is a tree whose leaves are in W , and each vertex $v \in V(G') \setminus W$ can be joined by three vertex-disjoint paths with W if and only if v is a vertex of degree at least three in F . It is folklore knowledge that if ℓ is the number of leaves and p is the number of vertices of degree at least three in a tree, then $p \leq \ell - 2$. This implies that U is constructed from W by adding at most $|W| - 2 \leq 2k - 1$ vertices. Hence, $|U| \leq 4k$. \square

Claim 4.2. For each connected component F of $G - U$, F is a tree such that each vertex $x \in U$ has at most one neighbor in F and

- (i) either U has a unique vertex x that has a neighbor in F ,
- (ii) or U contain exactly two vertices x and y having neighbors in F .

Proof of Claim 4.2. We have that each connected component F of $G - U$ is a tree because $G - S$ is a tree and the endpoints of the edges of S are in U . For the same reasons, each vertex $x \in U$ is adjacent to at most one vertex of each F . Consider arbitrary F . If only one vertex $x \in U$ has a neighbor in F , then (i) is fulfilled. Suppose that at least two vertices of U are adjacent to some vertices of F . Then exactly two vertices $x, y \in U$ have neighbors in F because, otherwise, F would have a vertex v which could be joined with U by three vertex-disjoint paths but such a vertex would be included in U . This implies that (ii) holds and concludes the proof. \square

If F is a connected component of $G - U$ satisfying (i) of Claim 4.2, then we say that F is a x -tree and x is its *anchor*. For a connected component F of $G - U$ satisfying (ii), we say that F is an (x, y) -tree and call x and y *anchors* of F . We also say that F is *anchored* in x (x and y , respectively). Because $G - S$ is a tree and $|U| \leq 4k$ by Claim 4.1, we immediately obtain the next property.

Claim 4.3. For every distinct $x, y \in U$, $G - U$ has at most one (x, y) -tree. Furthermore, the graph H with $V(H) = U$ such that $xy \in E(H)$ if and only if $G - U$ has an (x, y) -tree is a forest. In particular, the total number of (x, y) -trees is at most $4k - 1$.

To prove the theorem, we have to verify the existence of a broadcasting protocol $P = (T, \{C(v) \mid v \in V(T)\})$ that ensures that every vertex receives the message after at most t rounds. To do it, we guess the *scheme* of P restricted to U . Namely, we consider the graph G' obtained from G by the deletion of the vertices of x -trees for all $x \in U$ and for each vertex $x \in U$, we guess how the message is broadcasted to x and from x to the neighbors of x in G' . Notice that $T' = T[V(G')]$ is a tree by the definition of G' . Observe also that for each x -tree F for $x \in U$, the message is broadcasted to the vertices of F from x , because $s \in U$. In particular, this means that the parents of the vertices of U in T' are in G' . For each $v \in U$ distinct from s , we guess its parent $p(v) \in V(G')$ in T' and assume that $p(s) = s$. Then for each $v \in U$, we guess the ordered subset $R(v)$ of vertices of $N_{G'}(v) \setminus \{p(v)\}$ such that $R(v) = C(v) \cap N_{G'}(v)$. We guess $p(v)$ and $R(v)$ for $v \in U$ by considering all possible choices. To guess $R(v)$ for each $v \in U$, we first guess the (unordered) set $S(v)$ and then consider all possible orderings of the elements of $S(v)$. The selection of $p(v)$ and $S(v)$ is done by brute force. However, we are only interested in choices, where the selection of the neighbors $p(v)$ and $S(v)$ of v for $v \in U$ can be extended to a spanning tree T' of G' .

Let T' be an arbitrary spanning tree of G' rooted in s . Let T'' be the tree obtained from T' by the iterative deletion of leaves not included in U . Observe that T'' is a tree such that $U \subseteq V(T'')$ and each leaf of T'' is a vertex of U . By Claim 4.2, each edge of T'' is either an edge of $G[U]$ or is an edge of an (x, y) -path Q for distinct $x, y \in U$ such that the internal vertices of Q are the vertices of the (x, y) -tree F ; in the second case, each edge of Q is in T'' . Notice also that $s \in V(T'')$ and for each $v \in U$ distinct from s , the parent of v in T is the parent of v in T'' with respect to the source vertex s . Hence, our first step in constructing $p(v)$ and $S(v)$, is to consider all possible choices of T'' . Observe that $G[U]$ has at most $\binom{4k}{2}$ edges by Claim 4.1 and the total number of (x, y) -trees is at most $4k - 1$ by Claim 4.3. Because T'' is a tree, it contains $|U| - 1$ edges of $G[U]$ and (x, y) -paths Q in total. We obtain that we have $k^{\Theta(k)}$ possibilities to choose T'' . From now on, we assume that T'' is fixed.

The choice of T'' defines $p(v)$ for $v \in U \setminus \{s\}$. For each $v \in U$, we initiate the construction of $S(v)$ by including in the set the neighbors of v in T'' distinct from $p(v)$. We proceed with guessing of $S(v)$ by considering (x, y) -trees F for $x, y \in U$ such that the (x, y) -path with the internal vertices in F is not included in T'' . Clearly, the vertices of every F of such a type should receive the message either via x , or via y , or via both x and y . Let F be an (x, y) -tree of this type. Denote by x' and y' the neighbors of x and y , respectively. We have that either $x' \in S(x)$ and $y' \notin S(y)$, or $x \notin S(x)$ and $y \in S(y)$, or $x' \in S(x)$, $y' \in S(y)$ and $x' \neq y'$. Thus, we have three choices for F . By Claim 4.3, the total number of choices is $2^{O(k)}$. We go over all the choices and include the vertices to the sets $S(v)$ for $v \in U$ with respect to them. By Claim 4.2, this concludes the construction of the sets $S(v)$. From now on, we assume that $S(v)$ for $v \in U$ are fixed.

We construct the ordered sets $R(v)$ by considering all possible orderings of the elements of $S(v)$. The number of these orderings is $\prod_{v \in U} (|S(v)|!)$. Recall that the sets $S(v)$ are sets of neighbors of v in a spanning tree of G' . This and Claims 4.2 and 4.3 imply that $\sum_{v \in U} |S(v)| \leq 2(|U| - 1) + 2(4k - 1) \leq 16k$. Therefore, the total number of orderings is $\prod_{v \in U} (|S(v)|!) = k^{O(k)}$. This completes the construction of $R(v)$. Now we can assume that $p(v)$ for each $v \in U \setminus \{s\}$ and $R(v)$ for each $v \in U$ are given.

The final part of our algorithm is checking whether the guessed scheme for a broadcasting protocol can be extended to the protocol itself. This is done in two stages.

In the first stage, we compute for each $v \in U$, the minimum number $r(v)$ of a round in which v can receive the message and the ordered set $C(v)$. Initially, we set $r(s) = 0$ and set $X := \{s\}$. Then we iteratively either compute $C(v)$ for $v \in X$ or extend X by including a new vertex $v \in U \setminus X$ and computing $r(v)$. We proceed until we get $X = U$ and compute $C(v)$ for every $v \in U$. We also stop and discard the current choice of the scheme if we conclude that the choice cannot be extended to a broadcasting protocol terminating in at most t steps.

Suppose that there is $v \in X$ such that $C(v)$ is not constructed yet. Notice that $r(v)$ is already computed. To construct $C(v)$, we observe that for each v -tree F anchored in v , the vertices of F should receive the message via v . Hence, to construct $C(v)$, we extend $R(v)$ by inserting the neighbors of v in the v -trees. If there is no v -tree anchored in v , then we simply set $C(v) = R(v)$. Assume that this is not the case and let F_1, \dots, F_k be the v -trees anchored in v . Denote by u_1, \dots, u_k the neighbors of v in T_1, \dots, T_k , respectively. Because the message is broadcasted from v to each u_i , we can assume that to broadcast the message from u_i to the other vertices of T_i , an optimal protocol requiring $b(T_i, u_i)$ rounds is used. We compute the values $b(T_i, u_i)$ for all $i \in \{1, \dots, k\}$ and assume that $b(T_1, u_1) \geq \dots \geq b(T_k, u_k)$.

If $r(v) + |R(v)| + k > t$, we discard the current choice of the scheme, because we cannot transmit the message to the neighbors of v in t rounds. Notice also that $r(v) + \max\{b(T_i, u_i) + i \mid i \in \{1, \dots, k\}\}$ rounds are needed to transmit the message to the vertices of all v -trees. Hence, if $r(v) + \max\{b(T_i, u_i) + i \mid i \in \{1, \dots, k\}\} > t$, we discard the considered scheme. From now on, we assume that $r(v) + |R(v)| + k \leq t$ and $r(v) + \max\{b(T_i, u_i) + i \mid i \in \{1, \dots, k\}\} \leq t$.

The main idea for constructing $C(v)$ is to ensure that the message is sent to the vertices of $R(v)$ as early as possible. To achieve this, we put u_1, \dots, u_k in $C(v)$ in such a way, that the message is sent to each u_i as late as possible. Since $|C(v)| = |R(v)| + k$, we represent $C(v)$ as an $|R(v)| + k$ -element array whose elements are indexed $1, 2, \dots, |R(v)| + k$. Because $b(T_1, u_1) \geq \dots \geq b(T_k, u_k)$, we can assume that the ordering of the vertices u_1, \dots, u_k in $C(v)$ is (u_1, \dots, u_k) . Therefore, we insert u_i in $C(v)$ consecutively for $i = k, k - 1, \dots, 1$.

Suppose that $i \in \{1, \dots, k\}$ and u_{i+1}, \dots, u_k are in $C(v)$. Denote by h_{i+1} the index of u_{i+1} assuming that $h_{k+1} = |R(v)| + k + 1$. We find maximum positive integer $h < h_{i+1}$ such that $r(v) + h + b(T_i, u_i) \leq t$ and set the index $h_i = h$ for u_i . In other words, we find the maximum index that is prior to the index of u_{i+1} such that if we transmit the message from v to u_i in the h -th round after v got aware of the message, then the vertices of T_i still may get the message in t rounds. After placing u_1, \dots, u_k into the array, we place the vertices of $R(v)$ in the remaining $|R(v)|$ places following the order in $R(v)$. This completes the construction of $C(v)$.

Suppose that $U \setminus X \neq \emptyset$ and for each $v \in X$, $C(v)$ is given. We assume that for each $v \in X$, the elements of $C(v)$ are indexed $1, \dots, |C(v)|$ according to the order. By the construction of the schemes, there is $y \in U \setminus X$ such that y receives the message from some vertex $x \in X$ either directly or via some (x, y) -tree F anchored in x and y . We find such a vertex y , compute $r(y)$, and include y in X .

If there is $y \in U \setminus X$ such that $p(y) = x \in X$, then we set $r(y) = r(x) + h$, where h is the index of y in $C(x)$ and set $X := X \cup \{y\}$. Since $r(x)$ is the minimum number of a round when x gets the message, $r(y)$ is the minimum number of a round in which y gets the message. Suppose that such a vertex y does not exist. Then by the construction of the considered scheme, there are $x \in X$ and $y \in U \setminus X$ such that the tree T'' which was used to construct the scheme contains an (x, y) -path whose internal vertices are in the (x, y) -tree F anchored in x and y . This means that the neighbor x' of x in F is included in $C(x)$. Let h be the index of x' in $C(x)$. We also have that $y' = p(y)$ is the unique neighbor of y in F . In other words, we have to transmit the message from x to y according to the scheme. To compute $r(y)$, we have to transmit the message as fast as possible. For this, we use Lemma 2. Notice that the vertices of F should receive the message in at most $t' = t - r(x) - h + 1$ rounds because x receives the message in the round $r(x)$ and $h - 1$ vertices of $C(x)$ get the message before x' . Let F' be the tree obtained from F by adding the vertices x, y and the edges xx', yy' . We compute $b_{r'}(F', x, y)$ using the algorithm from Lemma 2. If $b_{r'}(F', x, y) = +\infty$, we discard the considered scheme because y cannot receive the message in t rounds. Otherwise, we set $r(y) = r(x) + (h - 1) + b_{r'}(F', x, y)$ and set $X := X \cup \{y\}$.

This completes the first stage where we compute $r(v)$ and $C(v)$ for $v \in U$. Observe that if we completed this stage without discarding the considered choice of the scheme, we already have a partially constructed broadcasting protocol that ensures that (i) the vertices of v -trees for $v \in U$ get the message in at most t rounds, (ii) the vertices of (x, y) -trees that are assigned by the scheme to transmit the message from x to y receive the message in at most t rounds, and (iii) for each $v \in U$, $r(v)$ is the minimum number of a round when v can receive the message according to the scheme. By Claim 4.2, it remains to check whether the vertices of (x, y) -trees

F that are not assigned by the scheme to transmit the message from x to y or vice versa can receive the message in at most t rounds. We do it using Lemmas 1 and 3.

Suppose that F is a (x, y) -tree anchored in $x, y \in U$ such that $p(x), p(y) \notin V(F)$, that is, F is not assigned by the scheme to transmit the message from x to y or vice versa. Let x' and y' be the neighbors in F of x and y , respectively. By the construction of the schemes, we have three cases: (i) $x' \in C(x)$ and $y' \notin C(y)$, (ii) $x' \notin C(x)$, $y' \in C(y)$, and (iii) $x' \in C(x)$, $y' \in C(y)$, and $x' \neq y'$.

In case (i), the vertices of F should receive the message via x . Clearly, we can use an optimal protocol for F with the source x' to transmit the message from x' . Let h be the index of x' in $C(x)$. We use Lemma 1, to verify whether $t - r(x) - h \geq b(F, x)$. If the inequality holds, we conclude that the message can be transmitted to the vertices of F in at most t rounds. Otherwise, we conclude that this is impossible and discard the scheme. Case (ii) is symmetric and the arguments are the same.

Suppose that $x' \in C(x)$, $y' \in C(y)$, and $x' \neq y'$. Then the vertices of F receive the message from both x and y . Denote by i and j the indexes of x' and y' in $C(x)$ and $C(y)$, respectively. By symmetry, we assume without loss of generality that $r(x) + i \leq r(y) + j$ and let $h = (r(y) + j) - (r(x) + i)$. Notice that the vertices of F start to get the message after the round $r(v) + i - 1$. Denote by F' the tree obtained from F by adding the vertices x, y and the edges xx', yy' . We use Lemma 3 and compute $d_h(F', x, y)$. If $d_h(F', x, y) \leq t - r(v) - i + 1$, then we obtain that the message can be transmitted to the vertices of F in at most t rounds. Otherwise, we cannot do it and discard the scheme.

This completes the description of the second stage of the verification of whether the considered scheme can be extended to a broadcasting protocol terminating in at most t rounds.

If we find a scheme that allows us to conclude that the message can be broadcasted in at most t rounds, we conclude that (G, s, t) is a yes-instance. Otherwise, if every scheme gets discarded, we return that (G, s, t) is a no-instance of TELEPHONE BROADCAST. This concludes the description of the algorithm.

To argue that the algorithm is correct, note that if there is a scheme for which we report that (G, s, t) is a yes-instance, then the description of the verification procedure of the extendability of the scheme implies the existence of a broadcasting protocol terminating in at most t rounds. For the opposite direction, assume that (G, s, t) is a yes-instance of TELEPHONE BROADCAST. Then there is a broadcasting protocol P terminating in at most t rounds. For $v \in V(G)$, denote by $r_P(v)$ the number of the round on which v gets the message. Among all $P = (T, \{C'(x) \mid x \in V(T)\})$ terminating in t rounds, we select P to be protocol such that $\sum_{v \in U} r_P(v)$ is minimum. We construct G' as described in the algorithm and consider $T' = T[V(G')]$. Then we construct T'' from T' by the iterative deletion of leaves that are not included in U . Then we define the scheme by setting $p(v)$ to be parent of each $v \in U$ in T'' with respect to the root s ($p(s) = s$) and defining $R(v) = C'(v) \cap N_{T''}(v)$ for $v \in U$. Observe that the obtained scheme is one of the schemes constructed by the algorithm. Then the description of the algorithm implies that the verification step of the algorithm returns the yes-answer for the scheme. In particular, we get that for every $v \in U$, $r(v) = r_P(v)$ and the indices of the elements of $R(v)$ in the set $C(v)$ obtained by the algorithm are the same as in $C'(v)$.

To evaluate the running time, observe that the feedback edge set S and the set of vertices U are constructed in polynomial time. Given U , we can list all x -trees and (x, y) -trees for $x, y \in U$ in polynomial time. Then the schemes can be constructed in $k^{\Theta(k)}$ time. For each scheme, the verification step requires polynomial time as we use the polynomial algorithms from Lemmas 1, 2, and 3. We conclude that the total running time is $2^{\Theta(k \log k)} \cdot n^{\Theta(1)}$. This completes the proof.

5. TELEPHONE BROADCAST parameterized by the vertex cover number

In this section, we prove Theorem 3. Recall that we aim to show that TELEPHONE BROADCAST is FPT on graphs with the vertex cover number at most k when the problem is parameterized by k . We start with some auxiliary claims about the broadcasting on a graph with a given vertex cover S .

Lemma 4. *Let G be a graph with at least one edge and $s \in V(G)$. Let also S be a vertex cover of G . Then there is an optimal broadcasting protocol for G with the source s such that the vertices of S receive the message in at most $2|S| - 1$ rounds.*

Proof. Let $(T, \{C(v) \mid v \in V(T)\})$ be an optimal broadcasting protocol for G with the source s . Observe that we can assume without loss of generality that the protocol is *greedy*, meaning that for each round i , if it holds that v is a vertex that is aware of the message after i rounds, it would always send the message in the next round to some neighbor unless all the neighbors either got the message in the first i rounds or will receive the message in the $(i + 1)$ -th round from vertices that are distinct from v . In other words, a vertex does not stay idle if it can communicate the message to a new recipient. Furthermore, we can assume that for each $v \in V(T)$, the children of v of degree one, i.e., leaves of T , are at the end of the ordered set $C(v)$. Otherwise, we can rearrange each $C(v)$ to achieve this property and the obtained protocol would be optimal. Also, we can include the leaf children of each v in $C(v)$ in arbitrary order because any rearrangement of these vertices does not change the number of rounds. Hence, we assume that $(T, \{C(v) \mid v \in V(T)\})$ is a greedy optimal broadcasting protocol such that for each $v \in V(T)$, (i) the children of v of degree one are at the end of the ordered set $C(v)$ and (ii) the children of degree one that are in S are before the children of degree one in $I = V(G) \setminus S$. We claim that every vertex of S gets the message in the first $2|S| - 1$ round.

For the sake of contradiction, assume that this is not the case. Then there is a nonnegative integer $h \leq 2|S| - 2$ such that neither in round h nor in round $h + 1$, there is a vertex of S that receives the message. Let $S' \subseteq S$ be the set of vertices of S that got the message after $h - 1$ rounds and let $S'' = S \setminus S'$. Let also $I' \subseteq I$ be the set of vertices of I that got the message after $h - 1$ rounds and denote by $I'' \subseteq I$ the set of vertices of I that get the message in the h -th round. Because no vertex of S gets the message in the h -th round, we have that $I'' \neq \emptyset$. Observe that no vertex of $I' \cup I''$ is adjacent to a vertex of S'' as, otherwise, at least one vertex

of S' would get the message in the $(h + 1)$ -th round. In particular, this means that the vertices of I'' are leaves of T . Because G is connected and T is a spanning tree, we have that there is a vertex $v \in S'$ which is adjacent to a vertex $u \in S''$ in T . Since the protocol is greedy, v sent the message to a vertex of I'' in the h -th round. However, this contradicts that the protocol satisfies (i) and (ii) for v . If u is not a leaf of T , then v should send the message to u instead of a vertex of I'' by (i), and if u is a leaf, then u is a leaf in S and v should send the message to u in the h -th round by (ii). The obtained contradiction proves the lemma. \square

We also use the bound for the number of vertices of $I = V(G) \setminus S$ getting the message in the first p rounds.

Lemma 5. *Let G be a graph with at least one edge and $s \in V(G)$. Let also S be a vertex cover of G and $p \geq 1$ be an integer. Then for any broadcasting protocol for G with the source s , at most $p|S|$ vertices of $I = V(G) \setminus S$ receive the message in the first p rounds.*

Proof. Note that each vertex $v \neq s$ in I can receive the message only from its neighbor in S . If $s \in S$, then each vertex of S can send the message to at most p neighbors in p round. Therefore, at most $p|S|$ can get the message in p rounds. If $s \in I$, then each vertex of S can send the message to at most $p - 1$ neighbors. Hence, at most $(p - 1)|S| + 1 \leq p|S|$ vertices of I can receive the message in p rounds. \square

Now we are ready to prove Theorem 3.

5.1. Proof of Theorem 3

Let (G, s, t) be an instance of TELEPHONE BROADCAST and let $k \geq 0$ be an integer. TELEPHONE BROADCAST is trivial for $k = 0$ as G has the empty vertex cover if and only if G has a single vertex (recall that G is connected by our assumption). Also, the problem is trivial if G has no edges or $t \geq n$. Hence, we assume that G has at least two vertices, $t < n$, and $k \geq 1$.

We use the algorithm of Chen, Kanj, and Xia [5] to find in $1.2738^k \cdot n^{\mathcal{O}(1)}$ time a vertex cover S of G of size at most k . If the algorithm fails to find such a set, then we stop and return the answer that G has no vertex cover of size at most k . From now on, we assume that S is given and $I = V(G) \setminus S$. If $|I| \leq 2k^2$, then $|V(G)| \leq 2k^2 + k$ and we solve the problem in $2^{\mathcal{O}(k^2)} \cdot k^{\mathcal{O}(1)}$ time using Theorem 1. So we can assume that $|I| > 2k^2$.

We compute the partition of $\{L_1, \dots, L_p\}$ of I into classes of false twins, that is, two vertices u and v are in the same set L_i if and only if they have exactly the same neighbors in S . Since $|S| \leq k$, we have that $p \leq 2^k$. Observe also that the partition can be constructed in linear time (see [26]). As it is common for the parameterization by the vertex cover number, we exploit the property that two vertices $u, v \in L_i$ for some $i \in \{1, \dots, p\}$ that are distinct from s are *indistinguishable*. In particular, given a broadcasting protocol $(T, \{C(v) \mid v \in V(T)\})$, we can exchange u and v , that is, replace the former u by v and former v by u in T and the ordered sets, and this would keep the number of rounds the same.

By Lemma 5, at most $2k^2$ vertices of I can receive the message in the first $2k$ rounds and we have that $|I| > 2k^2$. Hence, if $t \leq 2k$, we conclude that (G, s, t) is a no-instance. From now on we can assume that $t > 2k$.

By Lemma 4, there is an optimal broadcasting protocol such that all the vertices of S receive the message in the first $2k - 1$ rounds. Combining this with Lemma 5, we obtain that there is an optimal broadcasting protocol such that the set of vertices X that receive the message in the first $2k$ rounds satisfies the properties (i) $S \subseteq X$ and (ii) for $Y = X \cap I$, $|Y| \leq 2k^2$.

We guess Y using the fact that the vertices of the same set L_i are indistinguishable. We consider all possibilities to choose p nonnegative integers ℓ_1, \dots, ℓ_p such that (a) $\ell_i \leq |L_i|$ for each $i \in \{1, \dots, p\}$, (b) if $s \in L_i$ for some $i \in \{1, \dots, p\}$, then $\ell_i \geq 1$, and (c) $\ell_1 + \dots + \ell_p \leq 2k^2$. Notice that because $p \leq 2^k$, there are $2^{\mathcal{O}(k^3)}$ possibilities to choose ℓ_1, \dots, ℓ_p . Given ℓ_1, \dots, ℓ_p , for each $i \in \{1, \dots, p\}$, we can select ℓ_i vertices from L_i in such a way that if $s \in L_i$, then s is selected, and the other vertices are taken arbitrarily, and then denote the set of selected vertices by Y_i . We have that if there is an optimal broadcasting protocol such that the set of vertices X that receive the message in the first $2k$ rounds satisfying (i) and (ii), then there is an optimal broadcasting protocol such that the set of vertices X that receive the message in the first $2k$ rounds is $S \cup (Y_1 \cup \dots \cup Y_p)$. We verify whether the choice of X is *feasible*, that is, the vertices of X can get messages in $2k$ rounds, using Theorem 1. For this, we check whether $G[X]$ is connected and if this holds, then run the algorithm from the theorem for $G[X]$, s , and $t = 2k$. We discard the choice of X if either $G[X]$ is disconnected or the algorithm reports that the considered instance is a no-instance. Observe that the running time of the algorithm is $2^{\mathcal{O}(k^2)}$ in this case.

Since the vertices of S and Y are getting the message in the first $2k$ rounds, then the remaining vertices of $Z = I \setminus Y$ vertices should receive the message from their neighbors in S in the following $t' = t - 2k$ rounds. Observe that because each vertex of S is aware of the message after $2k$ rounds in a hypothetical solution, any vertex of S can send the message to any neighbor in I and the vertices of S do not send the message to S . This means that each vertex $v \in Z$ should be assigned to a vertex of S which is sending the message to v in such a way that for each $u \in S$, at most t' vertices are assigned to u . Finding such an assignment boils down to solving a special case of the b -MATCHING problem (see, e.g., [25,30]). However, in our case, it is easier to reduce the problem to finding a bipartite matching directly. We construct a set of vertices S' by replacing each vertex $u \in S$ by t' copies. Then we construct the auxiliary bipartite graph H with the set of vertices $S' \cup Z$ such that each vertex $v \in Z$ is adjacent to the copies of $u \in S$ if and only if v is adjacent to u in G . It is straightforward to see that H has a matching saturating every vertex of Z if and only if there is a required assignment of the vertices of Z to the vertices of S . Because H has at most $t'n \leq n^2$ vertices and at most m edges, we can find a matching saturating every vertex of H (if exists) in $\mathcal{O}(nm)$ time by the classical Hopcroft-Karp algorithm [20]. If the

algorithm finds a matching, we conclude that (G, s, t) is a yes-instance and stop. If the auxiliary matching problem has no solution for all choices of ℓ_1, \dots, ℓ_p and corresponding feasible sets X , we report that (G, s, t) is a no-instance.

To show correctness, we follow the description of the algorithm. We can assume that (G, s, t) is an instance of TELEPHONE BROADCAST with $k \geq 1$, $2k^2 < t < |V(G)|$, $|V(G)| \geq 2$, and $|I| > 2k^2$ for $I = V(G) \setminus S$, because otherwise correctness follows directly from the description.

Suppose that (G, s, t) is a yes-instance. By Lemmas 4 and 5, G and s have an optimal broadcasting protocol such that the set of vertices X that receive the message in the first $2k$ rounds satisfies the properties (i) $S \subseteq X$ and (ii) for $Y = S \cap I$, $|Y| \leq 2k^2$. For each $i \in \{1, \dots, p\}$, let $\ell_i = L_i \cap Y$. We have that ℓ_1, \dots, ℓ_p are nonnegative integers such that (a) $\ell_i \leq |L_i|$ for each $i \in \{1, \dots, p\}$, (b) if $s \in L_i$ for some $i \in \{1, \dots, p\}$, then $\ell_i \geq 1$, and (c) $\ell_1 + \dots + \ell_p \leq 2k^2$. Therefore, we consider the choice of ℓ_1, \dots, ℓ_p in one of the branches of our algorithm. Because the vertices of each L_i are indistinguishable, we obtain that for our choice of Y_1, \dots, Y_p , there is an optimal broadcasting protocol P such that the set of vertices X that receive the message in the first $2k$ rounds is exactly $X' = S \cup (Y_1 \cup \dots \cup Y_p)$. In particular, this means that X' is feasible. Hence, the choice of ℓ_1, \dots, ℓ_p and X' is not discarded by the algorithm. Then the auxiliary graph H has a matching saturating every vertex of $Z = I \setminus Y$ and our algorithm returns a correct yes-answer.

For the opposite direction, assume that there is a choice of nonnegative integers ℓ_1, \dots, ℓ_p satisfying (a)–(c) such that for the corresponding choice of Y_1, \dots, Y_p , $X = S \cup (Y_1 \cup \dots \cup Y_p)$ is feasible and H has a matching saturating every vertex of Z . We have that there is a broadcasting protocol P that ensures that the vertices of X receive the message in the first $2k$ rounds. We extend the protocol by defining that a vertex $v \in Z$ receives the message from the vertex $u \in S$ if v is matched to the copy of u in H . The order in which a vertex $u \in S$ sends the message to the vertices of Z is arbitrary. Because at most $t' = t - 2k$ vertices of Z are assigned to every $u \in S$, we obtain that all the vertices get the message in t rounds. Thus, (G, s, t) is a yes-instance. This completes the proof of correctness.

To evaluate the running time, notice that we consider $2^{\mathcal{O}(k^3)}$ choices of ℓ_1, \dots, ℓ_p . For each choice, we either discard it in $2^{\mathcal{O}(k^2)}$ time or solve the auxiliary matching problem in polynomial time. This implies that the total running time is $2^{\mathcal{O}(k^3)} \cdot n^{\mathcal{O}(1)}$. This completes the proof of Theorem 3.

6. Kernelization for the parameterization by $k = n - t$

In this section, we prove Theorem 4. Recall that we parameterize TELEPHONE BROADCAST by $k = n - t$. Hence, it is convenient for us to denote the considered instances as triples (G, s, k) throughout the section instead of $(G, s, n - k)$.

Let (G, s, k) be an instance of TELEPHONE BROADCAST. We exhaustively apply the following reduction rules in the order in which they are stated. The first rule is straightforward because $b(G, s) \leq n - 1$ and (G, s, k) is a yes-instance if $k \leq 1$. Also if $k > n$, then (G, s, k) is a no-instance.

Reduction Rule 6.1. If $k \leq 1$, then return a trivial yes-instance, e.g., the instance with $G = (\{s\}, \emptyset)$ and $k = 0$, and stop. If $k > n$, then return a trivial no-instance, e.g., the instance with $G = (\{s, v\}, \{sv\})$ and $k = 0$, and stop.

Observe that after applying Reduction Rule 6.1, $n \geq 2$, because if $n = 1$, then either $k \leq 1$ or $k > n$ and we would stop. Notice that if $d_G(s) = 1$, then the source s sends the message to its unique neighbor in the first round. This allows us to delete s and define a new source using the following rule whose safeness is straightforward.

Reduction Rule 6.2. If $d_G(s) = 1$, then let v be the neighbor of s , set $G := G - s$ and define $s := v$.

Now we can assume that $d_G(s) \geq 2$. By the next rule, we delete certain pendent vertices.

Reduction Rule 6.3. If there is a vertex $v \in V(G)$ such that for the set of vertices of degree one $W \subseteq N_G(v)$, it holds that $|W| \geq |V(G) \setminus W|$, then select an arbitrary $w \in W$ and set $G := G - w$.

Claim 6.1. Reduction Rule 6.3 is safe.

Proof of Claim 6.1. Let $v \in V(G)$ and assume the set of vertices W of degree one in the neighborhood of v satisfies the condition $|W| \geq |V(G) \setminus W|$. Let also $w \in W$ and $G' = G - w$. Notice that because $d_G(s) \geq 2$, $w \neq s$. Since $d_G(w) = 1$, w receives the message from v and w does not send the message anywhere. This implies that $b(G, s) - 1 \leq b(G', s) \leq b(G, s)$. To show the claim, it is sufficient to prove that $b(G', s) = b(G, s) - 1$.

Consider an optimal broadcasting protocol $(T, \{C(x) \mid x \in V(T)\})$ for G . Because the vertices of W are adjacent to v and have degree one, the vertices of W are children of v in T . Since each vertex $x \in W$ has no children in T , we can assume without loss of generality that the vertices of W are the last vertices in $C(v)$ and, furthermore, w is the last vertex in this ordered set. Then the vertices of $V(G) \setminus W$ get the message in at most $|V(G)| - |W| - 1$ rounds. Because $|V(G) \setminus W| \leq |W|$, we obtain that w gets the message in the last round. Moreover, w is a unique vertex that gets the message in the last round. This implies that $(T', \{C'(x) \mid x \in V(T')\})$, where $T' = T - w$, $C'(x) = C(x)$ for $x \in V(T') \setminus \{v\}$ and $C'(v) = C(v) \setminus \{w\}$, is a broadcasting protocol for G' that ensures that every vertex gets the message in $b(G, s) - 1$ rounds. Thus $b(G', s) = b(G, s) - 1$. \square

To state the following rule, we introduce an auxiliary notation. For a vertex v of a graph H , we define $\rho_H(v) = \max\{\text{dist}_H(v, u) \mid u \in V(H)\}$.

Reduction Rule 6.4. If G has a bridge $e = uv$ such that $G - e$ has two connected components G_1 and G_2 , where $s, u \in V(G_1)$, $v \in V(G_2)$, $d_G(u) = 2$, and $|V(G_1)| < \text{dist}_{G_1}(s, u) + \rho_{G_2}(v)$, then set $G := G/e$.

Claim 6.2. *Reduction Rule 6.4 is safe.*

Proof of Claim 6.2. Let $e = uv$ be a bridge of G such that $G - e$ has two connected components G_1 and G_2 , where $s, u \in V(G_1)$, $v \in V(G_2)$, and $|V(G_1)| < \text{dist}_{G_1}(s, u) + \rho_{G_2}(v)$. Let also $G' = G/e$. Because to reach the vertices of G_2 , u should send the message to v , we have that $b(G, s) - 1 \leq b(G', s) \leq b(G, s)$. To prove the claim, we show that $b(G', s) = b(G, s) - 1$.

Let $(T, \{C(x) \mid x \in V(T)\})$ be an optimum broadcasting protocol for G . Because e is a bridge, v is a child of u in T . Also for every $x \in V(G_2)$, $C(x)$ contains only vertices of G_2 . Because u is a unique vertex in $V(G_1)$ which sends the message outside $V(G_1)$ and does it only once, we have that every vertex of $V(G_1)$ receives the message in at most $|V(G_1)|$ rounds. On the other side, we can observe that $V(G_2)$ has a vertex that gets the message only in at least $\text{dist}_{G_1}(s, u) + \rho_{G_2}(v)$ rounds. Therefore, the vertices that receive the message in the last round are in G_2 . We define the protocol $(T', \{C'(x) \mid x \in V(T')\})$ for G' as follows. We set $T' = T/e$, and for every $x \in V(T) \setminus \{u, v\}$, we define $C'(x) = C(x)$. Let w be the vertex obtained from u and v by the contraction of e . To construct $C'(w)$, note that $C(u) = \{v\}$, because $d_G(u) = 2$. We define $C'(w) = C(v)$. Observe that by the protocol for G' , each vertex $x \in V(G_2)$ gets the message in one round earlier than in the protocol for G . We conclude that each vertex of G receives the message in at most $b(G, s) - 1$ rounds. This means that $b(G', s) = b(G, s) - 1$. \square

From now on, we can assume that Reduction Rules 6.1–6.4 are not applicable. We run the standard breadth-first search (BFS) algorithm on G from s (see, e.g., [7] for the description). The algorithm produces a spanning tree B of G of shortest paths and the partition of $V(G)$ into *BFS-levels* L_0, \dots, L_r , where L_i is the set of vertices at distance i from s for every $i \in \{1, \dots, r\}$.

We apply the following rule whose safeness immediately follows from Observation 1 and Lemma 1.

Reduction Rule 6.5. Compute $b(B, s)$ and if $b(B, s) \leq n - k$, then return a trivial yes-instance and stop.

Then we apply the final rule.

Reduction Rule 6.6. If there is $v \in L_i$ for some $i \in \{0, \dots, r - 1\}$ such that for $X = N_G(v) \cap L_{i+1}$ and for the (s, v) -path P in B , it holds that (i) $|X| \geq 2k + 1$ and (ii) the total number of vertices in nontrivial, i.e., having at least two vertices, connected components of $G - V(P)$ containing vertices of X is at least $4k - 2$, then return a trivial yes-instance and stop.

Claim 6.3. *Reduction Rule 6.6 is safe.*

Proof of Claim 6.3. Let $v \in L_i$ for some $i \in \{0, \dots, r - 1\}$ and let P be the (s, v) -path in B . Denote by G_1, \dots, G_ℓ the nontrivial connected components of $G - V(P)$ such that each of them contains at least one vertex of $X = N_G(v) \cap L_{i+1}$. We assume that $|V(G_1)| \geq \dots \geq |V(G_\ell)| \geq 2$. Suppose that $|V(G_1)| + \dots + |V(G_\ell)| \geq 4k - 2$. To show that the rule is safe, we have to prove that $b(G, s) \leq n - k$. By Observation 2, it suffices to show that there is a tree subgraph T of G containing s with $b(T, s) \leq |V(T)| - k$. We consider two cases.

Assume first that $|V(G_1)| \geq k + 1$. Let $u \in V(G_1) \cap X$. We find a tree subgraph F in G_1 with exactly $k + 1$ vertices such that $u \in V(F)$. Then to construct the tree T , we take P and F and make v adjacent to u . Because $|X| \geq 2k + 1$, there are k distinct vertices $u_1, \dots, u_k \in X \setminus V(F)$. We include them in T by making them adjacent to v . We define the broadcasting protocol $(T, \{C(x) \mid x \in V(T)\})$ as follows. For each $x \in V(P) \setminus \{v\}$, x has a unique child composing $C(x)$. We define $C(v) = (u, u_1, \dots, u_k)$ and for each $x \in V(F)$, $C(x)$ is an arbitrary ordering of the children of x in F . Because $|V(F)| = k + 1$, we have that $b(T, s) \leq |V(P)| + |V(F)| - 1 \leq |V(T)| - k$.

Suppose now that $k \geq |V(G_1)| \geq \dots \geq |V(G_\ell)|$. Let T_i be a spanning tree of G_i and let $u_i \in V(T_i) \cap X$ for each $i \in \{1, \dots, \ell\}$. We construct T from P and T_1, \dots, T_ℓ by making v adjacent to u_1, \dots, u_ℓ . We define the broadcasting protocol $(T, \{C(x) \mid x \in V(T)\})$ as follows. For each $x \in V(P) \setminus \{v\}$, x has a unique child composing $C(x)$. We define $C(v) = (u_1, \dots, u_\ell)$ and for each $i \in \{1, \dots, \ell\}$ and $x \in V(T_i)$, $C(x)$ is an arbitrary ordering of the children of x in T_i . Observe that

$$b(T, s) \leq |V(P)| - 1 + \max\{|V(T_i)| + i - 1 \mid i \in \{1, \dots, \ell\}\}.$$

Since $|V(T_i)| \leq k$ for $i \in \{1, \dots, k\}$, $b(T, s) \leq |V(P)| + k + \ell - 2$. If $\ell \geq 2k$, then $|V(G_1)| + \dots + |V(G_\ell)| \geq 2\ell$ and $|V(T)| = |V(P)| + |V(G_1)| + \dots + |V(G_\ell)| \geq |V(P)| + \ell + 2k$. Hence, $|V(T)| - b(T, s) \geq k$. If $\ell \leq 2k$, then $b(T, s) \leq |V(P)| + 3k - 2$ and $|V(T)| - b(T, s) \geq (4k - 2) - (3k - 2) \geq k$. This completes the proof. \square

The crucial property of the instance obtained by applying Reduction Rules 6.1–6.6 is given in the following lemma.

Lemma 6. *Suppose that Reduction Rules 6.1–6.6 are not applicable to (G, s, k) . Then $|V(G)| \leq 18k - 12$.*

Proof. Recall that B is a BFS-tree rooted in s and $b(B, s) > n - k$ because of Reduction Rule 6.5. Recall also that L_0, \dots, L_r are the BFS-levels. For a vertex $v \in V(B)$, we use B_v to denote the subtree of B induced by the descendants of v in B including v itself. If $|V(G)| \leq k$, then the claim holds. Assume that $|V(G)| \geq k + 1$. Because $|V(B)| = |V(G)| \geq k + 1$, there is $v \in V(B)$ such that $|V(B_v)| \geq k + 1$ but $|V(B_u)| \leq k$ for every child u of v in B . Let P be the (s, v) -path in B . We define $Y = V(G) \setminus (V(P) \cup V(B_v))$. Clearly, $|V(G)| = |V(B)| = |Y| + |V(P)| + |V(B_v)| - 1$. To prove the lemma, we show upper bounds for $|Y|$, $|V(P)|$, and $|V(B_v)|$. First, we show a bound for the size of Y .

Claim 6.4. $|Y| \leq k - 1$.

Proof of Claim 6.4. For the sake of contradiction, assume that $|Y| \geq k$. Let $P = v_1 \dots v_p$, where $s = v_1$ and $v = v_p$. We consider the following broadcasting protocol $(B, \{C(x) \mid x \in V(B)\})$ for B . For every $i \in \{1, \dots, p-1\}$, we construct $C(v_i)$ by making v_{i+1} the first element of the ordered set and then append the other children of v_i in B in arbitrary order. For $x \in V(B) \setminus \{v_1, \dots, v_{p-1}\}$, $C(x)$ is an arbitrary ordering of the children of x . Notice that the vertices of P are getting the message in the first $|V(P)| - 1$ rounds. Then because $|Y| \geq k$ and each vertex $y \in Y$ is reachable in $B - V(B_v)$ from $V(P) \setminus \{v\}$ by a (v_i, y) -path for some $i \in \{1, \dots, p\}$, at least k vertices of Y should receive the message in the first $|V(P)| + k - 1$ rounds. Similarly, because $|V(B_v)| \geq k + 1$, at least k vertices of $B_v - v$ should receive the message in the first $|V(P)| + k - 1$ rounds. Hence, at least $|V(P)| + 2k$ vertices of B get the message in the first $|V(P)| + k - 1$ rounds. This implies that the total number of rounds for the protocol $(B, \{C(x) \mid x \in V(B)\})$ is at most $|V(B)| - k$. This contradicts the assumption that Reduction Rule 6.5 is not applicable and proves the claim. \square

Next, we use Claim 6.4 to upper bound the number of vertices of P .

Claim 6.5. $|V(P)| \leq 4k - 2$.

Proof of Claim 6.5. The proof is by contradiction. Assume that $|V(P)| \geq 4k - 1$. Note that $v \in L_p$ for $p \geq 4k - 2$ and $r \geq p + 1 \geq 4k - 1$. By Claim 6.4, $|Y| \leq k - 1$. Notice that $d_G(s) \geq 2$, because Reduction Rule 6.2 is not applicable. By the pigeonhole principle, there is $q \leq 3k - 3$ such that $|L_{q-1}| = |L_q| = |L_{q+1}| = 1$, that is, $L_{q-1} = \{v_{q-1}\}$, $L_q = \{v_q\}$, and $L_{q+1} = \{v_{q+1}\}$. This implies that $d_G(v_q) = 2$ and $v_q v_{q+1}$ is a bridge of G . Consider the connected components G_1 and G_2 of $G - v_q v_{q+1}$ and assume that $s, v_q \in V(G_1)$ and $v_{q+1} \in V(G_2)$. We have that $|V(G_1)| \leq q + 1 + |Y| \leq 4k - 3$. It also holds that $\text{dist}_{G_1}(s, v_q) = q$ and $\rho_{G_2}(v_{q+1}) = r - q - 1$. Hence, $\text{dist}_{G_1}(s, v_q) + \rho_{G_2}(v_{q+1}) = r - 1 \geq 4k - 2$. Thus, $|V(G_1)| \leq 4k - 3 < \text{dist}_{G_1}(s, v_q) + \rho_{G_2}(v_{q+1})$ but this implies that Reduction Rule 6.4 would be applicable for the bridge $v_q v_{q+1}$. This gives a contradiction that proves the claim. \square

Finally, we upper bound the size of B_v .

Claim 6.6. $|V(B_v)| \leq 13k - 9$.

Proof. Assume that $v \in L_p$ for some $p \in \{1, \dots, r\}$ and denote by $Z \subseteq X = N_G(v) \cap L_{p+1}$ the set of vertices having of degree one. We consider two cases depending on the size of X .

Suppose that $|X| \geq 2k + 1$. Because of Reduction Rule 6.6, the total number of vertices in nontrivial, i.e., having at least two vertices, connected components of $G - V(P)$ containing vertices of X is at most $4k - 3$. Hence, $B_v - Z$ has at most $4k - 2$ vertices. Because $|Y| \leq k - 1$ and $|V(P)| \leq 4k - 2$ by Claims 6.4 and 6.5, respectively, $B - Z$ has at most $(k - 1) + (4k - 2) + (4k - 2) - 1 = 9k - 6$ vertices, that is, $|V(G) \setminus Z| \leq 9k - 6$. Because Reduction Rule 6.3 is not applicable, we have that $|Z| \leq |V(G) \setminus Z| - 1 \leq 9k - 7$. We obtain that $|V(B_v)| = |V(B_v) \setminus Z| + |Z| \leq 13k - 9$.

Assume now that $|X| \leq 2k$. Suppose that T_1, \dots, T_ℓ are the subtrees of B_v rooted in the children u_1, \dots, u_ℓ of v , respectively, such that $|V(T_i)| \geq 2$ for each $i \in \{1, \dots, \ell\}$. Notice that $\ell \leq |X| \leq 2k$ and $|V(T_i)| \leq k$ because of the choice of v . We claim that $|V(T_1)| + \dots + |V(T_\ell)| \leq 4k - 2$.

The proof is by contradiction. Let $|V(T_1)| + \dots + |V(T_\ell)| \geq 4k - 1$. Consider the tree T constructed from P and T_1, \dots, T_ℓ by making v adjacent to u_1, \dots, u_ℓ . We show that $b(T, s) \leq |V(T)| - k$. For this, we define the broadcasting protocol $(T, \{C(x) \mid x \in V(T)\})$ as follows. For each $x \in V(P) \setminus \{v\}$, x has a unique child composing $C(x)$. We define $C(v) = (u_1, \dots, u_\ell)$ and for each $i \in \{1, \dots, \ell\}$ and $x \in V(T_i)$, $C(x)$ is an arbitrary ordering of the children of x in T_i . We have that

$$b(T, s) \leq |V(P)| - 1 + \max\{|V(T_i)| + i - 1 \mid i \in \{1, \dots, \ell\}\}.$$

Since $|V(T_i)| \leq k$ for $i \in \{1, \dots, \ell\}$ and $\ell \leq 2k$, $b(T, s) \leq |V(P)| + k + \ell - 2 \leq |V(P)| + 3k - 1$. Because $|V(T)| = |V(P)| + |V(T_1)| + \dots + |V(T_\ell)| \geq |V(P)| + 4k - 1$, we obtain that $b(T, s) \leq |V(T)| - k$. However, T is a subgraph of the tree B and $s \in V(T)$. Then by Observation 2, $b(B, s) \leq b(T, s) + |V(B) \setminus V(T)| \leq |V(B)| - k = n - k$ contradicting that Reduction Rule 6.5 is not applicable. This proves that $|V(T_1)| + \dots + |V(T_\ell)| \leq 4k - 2$. Because $|V(B_v)| = |X| - \ell + |V(T_1)| + \dots + |V(T_\ell)| + 1$, we obtain that $|V(B_v)| \leq |X| + |V(T_1)| + \dots + |V(T_\ell)| \leq 2k + 4k - 2 \leq 6k - 2$. This completes the proof of the claim. \square

Summarizing the upper bounds from Claims 6.4–6.6, we derive that

$$|V(G)| = |Y| + |V(P)| + |V(B_v)| - 1 \leq (k - 1) + (4k - 2) + (13k - 9) = 18k - 12.$$

This concludes the proof. \square

By Lemma 6, if we do not stop during the exhaustive applications of Reduction Rules 6.1–6.6, then for the obtained instance (G, s, k) , $|V(G)| \leq 18k - 12$. Hence, to complete the kernelization algorithm, we return (G, s, k) .

It is straightforward to see that Reduction Rules 6.1–6.6 can be applied in polynomial time. In particular, BFS and finding bridges can be done in linear time by classical graph algorithms (see, e.g., the textbook [7]). Thus, the total running time of the kernelization algorithm is polynomial. This completes the proof of Theorem 4.

7. Conclusion

In our paper, we initiated the study of TELEPHONE BROADCAST from the parameterized complexity viewpoint. In this section, we discuss further directions of research.

We observed that TELEPHONE BROADCAST is trivially FPT when parameterized by t and Theorem 1 implies that the problem can be solved in $3^{2^t} \cdot n^{O(1)}$ time. Is it possible to get a better running time for the parameterization by t ?

In Theorem 4, we obtained a polynomial kernel for the parameterization by $k = n - t$, that is, for the parameterization below the trivial upper bound for $b(G, s)$. This naturally leads to the question about parameterization below some other bounds for this parameter. We note that the parameterization of TELEPHONE BROADCAST above the natural lower bound $b(G, s) \geq \log n$ leads to a para-NP-complete problem. To see this, observe that for graphs with $n = 2^t$ vertices, $b(G, s) \leq t$ if and only if G has a binomial spanning tree rooted in s , and it is NP-complete to decide whether G contains such a spanning tree [27].

In Theorems 2 and 3, we considered structural parameterizations of TELEPHONE BROADCAST by the cyclomatic and vertex cover numbers, respectively. It is interesting to consider other structural parameterizations. In particular, is TELEPHONE BROADCAST FPT when parameterized by the *feedback vertex number* and *treewidth* (we refer to [8] for the definitions)? For the parameterization by treewidth, the complexity status of TELEPHONE BROADCAST is open even for the case when the treewidth of the input graphs is at most two, that is, for series-parallel graphs.

CRedit authorship contribution statement

Fedor V. Fomin: Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing. **Pierre Fraigniaud:** Conceptualization, Investigation, Writing – original draft, Writing – review & editing. **Petr A. Golovach:** Conceptualization, Investigation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

We are grateful to the anonymous reviewer for the valuable comments and suggestions.

References

- [1] Amotz Bar-Noy, Sudipto Guha, Joseph Naor, Baruch Schieber, Message multicasting in heterogeneous networks, *SIAM J. Comput.* 30 (2) (2000) 347–358.
- [2] Puspall Bhabak, Hovhannes A. Harutyunyan, Constant approximation for broadcasting in k-cycle graph, in: 1st International Conference Algorithms and Discrete Applied Mathematics (CALDAM), in: *Lecture Notes in Computer Science*, vol. 8959, Springer, 2015, pp. 21–32.
- [3] Puspall Bhabak, Hovhannes A. Harutyunyan, Approximation algorithm for the broadcast time in k-path graph, *J. Interconnect. Netw.* 19 (4) (2019) 1950006:1–1950006:22, <https://doi.org/10.1142/S0219265919500063>.
- [4] Maja Cevnik, Janez Zerovnik, Broadcasting on cactus graphs, *J. Comb. Optim.* 33 (1) (2017) 292–316.
- [5] Jianer Chen, Iyad A. Kanj, Ge Xia, Improved upper bounds for vertex cover, *Theor. Comput. Sci.* 411 (40–42) (2010) 3736–3756, <https://doi.org/10.1016/j.tcs.2010.06.026>.
- [6] Johanne Cohen, Pierre Fraigniaud, Jean-Claude König, André Raspaud, Optimized broadcasting and multicasting protocols in cut-through routed networks, *IEEE Trans. Parallel Distrib. Syst.* 9 (8) (1998) 788–802.
- [7] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Introduction to Algorithms*, 3rd edition, MIT Press, ISBN 978-0-262-03384-8, 2009, <http://mitpress.mit.edu/books/introduction-algorithms>.
- [8] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, Saket Saurabh, *Parameterized Algorithms*, Springer, ISBN 978-3-319-21274-6, 2015.
- [9] Reinhard Diestel, *Graph Theory*, 4th edition, Graduate Texts in Mathematics, vol. 173, Springer, ISBN 978-3-642-14278-9, 2012.
- [10] Michael Elkin, Guy Kortsarz, Sublogarithmic approximation for telephone multicast, *J. Comput. Syst. Sci.* 72 (4) (2006) 648–659.
- [11] Arthur M. Farley, Minimum-time line broadcast networks, *Networks* 10 (1) (1980) 59–70.

- [12] Fedor V. Fomin, Dieter Kratsch, Exact Exponential Algorithms, Texts in Theoretical Computer Science. An EATCS Series, Springer, ISBN 978-3-642-16532-0, 2010.
- [13] Pierre Fraigniaud, Approximation algorithms for minimum-time broadcast under the vertex-disjoint paths mode, in: 9th European Symposium on Algorithms (ESA), in: LNCS, vol. 2161, Springer, 2001, pp. 440–451.
- [14] Pierre Fraigniaud, Emmanuel Lazard, Methods and problems of communication in usual networks, Discrete Appl. Math. 53 (1–3) (1994) 79–133.
- [15] Mohammad Saber Gholami, Hovhannes A. Harutyunyan, Edward Maraachlian, Optimal broadcasting in fully connected trees, J. Interconnect. Netw. 23 (1) (2023) 2150037:1–2150037:20, <https://doi.org/10.1142/S0219265921500377>.
- [16] Michelangelo Grigni, David Peleg, Tight bounds on minimum broadcast networks, SIAM J. Discrete Math. 4 (2) (1991) 207–222.
- [17] Hovhannes A. Harutyunyan, Narek Hovhannisyan, Broadcasting in split graphs, in: Marios Mavronicolas (Ed.), Algorithms and Complexity - 13th International Conference, CIAC 2023, Larnaca, Cyprus, June 13–16, 2023, Proceedings, in: Lecture Notes in Computer Science, vol. 13898, Springer, 2023, pp. 278–292.
- [18] Hovhannes A. Harutyunyan, Edward Maraachlian, On broadcasting in unicyclic graphs, J. Comb. Optim. 16 (3) (2008) 307–322, <https://doi.org/10.1007/s10878-008-9160-2>.
- [19] Sandra Mitchell Hedetniemi, Stephen T. Hedetniemi, Arthur L. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (4) (1988) 319–349, <https://doi.org/10.1002/net.3230180406>.
- [20] John E. Hopcroft, Richard M. Karp, An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, SIAM J. Comput. 2 (4) (1973) 225–231, <https://doi.org/10.1137/0202019>.
- [21] Juraj Hromkovic, Ralf Klasing, Andrzej Pelc, Peter Ruzicka, Walter Unger, Dissemination of Information in Communication Networks - Broadcasting, Gossiping, Leader Election, and Fault-Tolerance, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2005.
- [22] Klaus Jansen, Haiko Müller, The minimum broadcast time problem for several processor networks, Theor. Comput. Sci. 147 (1&2) (1995) 69–85, [https://doi.org/10.1016/0304-3975\(94\)00230-G](https://doi.org/10.1016/0304-3975(94)00230-G).
- [23] S. Lennart Johnsson, Ching-Tien Ho, Optimum broadcasting and personalized communication in hypercubes, IEEE Trans. Comput. 38 (9) (1989) 1249–1268.
- [24] Guy Kortsarz, David Peleg, Approximation algorithms for minimum-time broadcast, SIAM J. Discrete Math. 8 (3) (1995) 401–427.
- [25] László Lovász, Michael D. Plummer, Matching Theory, AMS Chelsea Publishing, Providence, RI, ISBN 978-0-8218-4759-6, 2009, Corrected reprint of the 1986 original [MR0859549].
- [26] Ross M. McConnell, Jeremy P. Spinrad, Modular decomposition and transitive orientation, Discrete Math. 201 (1–3) (1999) 189–241, [https://doi.org/10.1016/S0012-365X\(98\)00319-7](https://doi.org/10.1016/S0012-365X(98)00319-7).
- [27] Christos H. Papadimitriou, Mihalis Yannakakis, The complexity of restricted spanning tree problems, J. ACM 29 (2) (1982) 285–309, <https://doi.org/10.1145/322307.322309>.
- [28] Andrzej Proskurowski, Minimum broadcast trees, IEEE Trans. Comput. 30 (5) (1981) 363–366, <https://doi.org/10.1109/TC.1981.1675796>.
- [29] R. Ravi, Rapid rumor ramification: approximating the minimum broadcast time, in: 35th IEEE Symposium on Foundations of Computer Science (FOCS), 1994, pp. 202–213.
- [30] Alexander Schrijver, Combinatorial Optimization: Polyhedra and Efficiency. Vol. C, Algorithms and Combinatorics, vol. 24.C, Springer-Verlag, Berlin, 2003, Disjoint Paths, Hypergraphs, Chapters 70–83.
- [31] Peter J. Slater, Ernest J. Cockayne, Stephen T. Hedetniemi, Information dissemination in trees, SIAM J. Comput. 10 (4) (1981) 692–701, <https://doi.org/10.1137/0210052>.
- [32] Quentin F. Stout, Bruce Wagar, Intensive hypercube communication. Prearranged communication in link-bound machines, J. Parallel Distrib. Comput. 10 (2) (1990) 167–181.