Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 28, no. 1, pp. 385–388 (2024)
DOI: 10.7155/jgaa.v28i1.2981

# A Linear-Time Optimal Broadcasting Algorithm in Stars of Cliques

*Peter Damaschke*

Department of Computer Science and Engineering,
Chalmers University of Technology and University of Gothenburg, SE-41296 Gothenburg, Sweden

Submitted: July 2024 Accepted: September 2024 Published: October 2024

Article type: Concise paper Communicated by: A. Anagnostopoulos

**Abstract.** For the telephone broadcast model, an $O(n \log n)$-time algorithm for constructing an optimal broadcasting scheme in a star of cliques with a total of $n$ vertices was recently presented by Ambashankar and Harutyunyan at IWOCA 2024. In the present note we give a considerably shorter and purified algorithm description and correctness proof. Moreover, we improve the time complexity to $O(n)$.

## 1 Introduction and Contribution

The Telephone Broadcast problem can be described as follows. Time is divided into discrete slots. We are given an undirected graph with informed and uninformed vertices. In every slot, every informed vertex can choose at most one uninformed neighbor and inform it. The goal is to inform all vertices after a minimum number of slots. Initially, only one vertex called the originator is informed.

The problem is NP-complete [11], even in 3-regular planar graphs, as first shown in [10], and for graphs with a feedback vertex set of size 1, that is, forests plus one vertex [12]. For polynomial algorithms in several special graph classes, see e.g. [7] and some later papers including [9, 8, 2] and more recent works such as [5]. Besides [7], further surveys can be found in [4, 6]. Further complexity results deal with the parameterized complexity and with lower bounds. For instance, the problem is FPT when parameterized by the size of a feedback edge set or a vertex cover [3], and in general graphs, computing an optimal broadcast needs double exponential time in the number of rounds under the ETH [12]. Amazingly, this bound means that a trivial algorithm is already the best one.

An $O(n \log n)$-time algorithm for constructing an optimal broadcasting strategy in a star of cliques was given in [1], however, with an algorithm description and proof of ca. eight technical pages of text. In the present note we streamline the derivation of this result, and as a byproduct we improve the time bound. The simplification is achieved by realizing that the binary representations

*E-mail address:* ptr@chalmers.se (Peter Damaschke)

of the clique sizes are "all what matters" for the problem, and by thoroughly working with them right from the beginning, without using other notations and taking detours. Reducing the proof to its essentials makes it more transparent, and this should also make it easier to generalize the result, e.g., to appropriately defined trees of cliques. In the light of the mentioned FPT results it might also be interesting to study the complexity of TELEPHONE BROADCAST parameterized by the vertex deletion distance to cluster graphs. However, this note only aims at improving the result from [1].

A star of cliques is a graph with $n = n_1 + \ldots + n_k + 1$ vertices, consisting of a disjoint union of $k$ cliques $Q_1, \ldots, Q_k$ on $n_1, \ldots, n_k$ vertices, respectively, and a center vertex $c$ which is adjacent to all other vertices. Although stars of cliques form a seemingly simple graphs class, optimal broadcasting turned out to be subtle [1]. We will show:

**Theorem 1.** An optimal strategy for TELEPHONE BROADCAST in a star of cliques with a total of $n$ vertices can be computed in $O(n)$ time.

## 2 The Proof

If the originator $u$ is not $c$, then let $S$ be any broadcasting strategy where $u$ informs some vertex $v \neq c$ in the first slot. We construct a modified strategy $S'$: First, $S'$ uses $u$ to inform $c$ rather than $v$. Then, as long as $S$ leaves $c$ uninformed, $S'$ behaves as $S$, and when $S$ uses $v$ to inform any vertex, $S'$ uses $c$ to inform the same vertex. As soon as $S$ uses some vertex $w$ to inform $c$, we use $w$ in $S'$ to inform $v$ instead. In fact, $wv$ is an edge, since all vertices informed by $S$ so far are in one clique. From now on, $S'$ behaves as $S$ again. Hence $S'$ does not work longer than $S$. Thus, we can pretend that $c$ is the originator but must inform some vertex in some distinguished clique first.

We first provide an algorithm ALG that, for given sizes $n_1, \ldots, n_k$ and given $b$, constructs a broadcasting strategy using at most $b$ slots, or reports that $b$ slots are not enough. We index the slots from right to left, in reverse temporal order: $\ldots, 3, 2, 1, 0$. Let $n_i = \sum_t n(i,t) \cdot 2^t$, where the $n(i,t)$ are the uniquely determined binary digits 1 or 0.

In every slot, the number of informed vertices in every clique is doubled. We set $h(i,t) = 1$ if $c$ informs another vertex of $Q_i$ in slot $t$, and $h(i,t) = 0$ otherwise. Let $h_i = \sum_t h(i,t) \cdot 2^t$. The number of informed vertices in $Q_i$ in the end is $\min\{h_i, n_i\}$. Hence, the strategy specified by the $h_i$ succeeds if and only if $\forall i : h_i \geq n_i$.

It remains to construct the $h_i$ from the $n_i$. We first suppose that no clique is distinguished, that is, some freely chosen vertex may be informed in the first slot. The ordering (2),(1),(0) of the following cases is intended.

(2) $n(i, b-1) = 1$ holds for two or more indices $i$. Then, no matter which $j$ we choose to set $h(j, b-1) = 1$, we violate some constraint $h_i \geq n_i$. Hence no solution exists.

(1) $n(i, b-1) = 1$ holds for exactly one index $i$. Then we must set $h(i, b-1) = 1$ to satisfy $h_i \geq n_i$. Now we can simply delete the leading 1 of $n_i$ to obtain a problem instance with one slot less. Note that $n_i$ becomes $n_i$ modulo $2^{b-1}$.

(0) $n(i, b-1) = 1$ holds nowhere. If we set any $h(i, b-1) = 1$, we get already $h_i > n_i$. Now we can delete $n_i$ to obtain a problem instance with one slot and one clique less. It is safe to choose some $i$ with maximum $n_i$. To see the correctness, assume by re-indexing that $n_1 \geq \ldots \geq n_k$, and assume that some successful strategy $S$ deletes $n_i$ for some $i > 1$. Then $S$ retains $n_1 \ldots, n_{i-1}, n_{i+1}, \ldots n_k$ whereas we retain $n_2 \ldots, n_i, n_{i+1}, \ldots n_k$. The remainder of $S$ still succeeds on our residual instance, since no clique is larger than in $S$.

Suppose that some distinguished clique $C_i$ must be chosen first. If $n(j, b-1) = 1$ for some $j \neq i$, we behave as in case (2). If $n(j, b-1) = 1$ holds for $j = i$ only, we behave as in case (1). If $n(j, b-1) = 1$ holds nowhere, we behave as in case (0), but for the prescribed $i$ (rather than a maximum $n_i$). Correctness is evident.

Now we can proceed inductively with the smaller instance in the same way. Moreover, the special case of a distinguished clique disappears after the first slot. If we never get case (2), the $h_i$ specify a successful strategy. This concludes the description of ALG.

An important remark is that, in every slot, ALG deletes at most one leading 1: In case (1), only one digit 1 is deleted, and in case (0), one complete number is deleted.

Next, we determine the smallest $b$ for which ALG succeeds. First suppose that no clique is distinguished. Then we index the cliques such that $n_1 \geq \ldots \geq n_k$.

Let $l_i$ be the largest $t$ with $n(i, t) = 1$. Note that $l_1 \geq \ldots \geq l_k$. Let $b$ be minimal with the property $b - i > l_i$ for all $i$. If we run ALG with slots $b-1, \ldots, 0$, then we are always in case (0), hence this instance has a solution. By the minimality of $b$, there exists some $i$ with $b-i-1 = l_i$. Let $i$ be the smallest such index. By monotonicity of the $l_i$, the numbers $n_1, \ldots, n_i$ have their leading 1s in the slots columns $b-2, \ldots, b-i-1$. By the minimality of $i$, we even have: Either $i = 1$ and $l_1 = b-2$, or $i > 1$ and the $i$ mentioned leading 1s are in the $i-1$ columns $b-3, \ldots, b-i-1$.

Now, let us run ALG with columns $b-3, \ldots, 0$. In the former case with $l_1 = b-2$ we violate $h_1 \geq n_1$. In the latter case, $i$ leading 1s occur in the first $i-1$ slots, but ALG deletes at most one in every slot. By the pigeonhole principle, it runs into case (2).

If some clique is distinguished, we call it $Q_1$ and index the other cliques such that $n_2 \geq \ldots \geq n_k$. We define $b$ as above. ALG starting in slot $b-1$ succeeds by the same reasoning as above. ALG starting in slot $b-3$ must first inform some vertex of $Q_1$. If $l_2 = b-3$, then it violates $h_2 \geq n_2$. If $b-3 > l_2$, then the $i-2$ slots $b-4, \ldots, b-i-1$ contain $i-1$ leading 1s, hence ALG runs into case (2).

Altogether, this shows that $b-2$ slots are not enough. Hence we run ALG only twice, to find some strategy with $b$ slots for sure, and to check whether $b-1$ slots are enough.

The binary representation of $n_i$ is straightforwardly computed in $O(n_i)$ time, hence in $O(n)$ time for all $i$. Since $\log_2$ is a concave function, the total number of binary digits is $O(n)$. By radix sort we can produce a prefix tree in $O(n)$ time. Its edges are labeled 1 and 0, the root-leaf paths display the binary representations of the $n_i$, and the leaves from left to right correspond to the numbers $n_1 \geq \ldots \geq n_k$. Its depth is $l_1 = O(\log n)$. Notice that $b = \min_i(i + l_i + 1)$. Using $l_1 \geq \ldots \geq l_k$ and $\sum_i l_i = O(n)$, we can easily compute $b$ incrementally in $O(n)$ time. As observed above, ALG with $b$ slots is always in case (0), hence it just successively deletes $n_1, \ldots, n_k$. ALG with $b-1$ slots sometimes encounters also case (1), but only in the last $l_1$ slots. Whenever ALG is in case (1), we delete the affected number $n_i$ from the prefix tree and insert the remainder of $n_i$, to restore the sorted order. Each of the $O(l_1)$ deletions and insertions needs $O(l_1)$ time. Thus, case (1) causes only $O(\log^2 n) \subset O(n)$ extra time.

## Acknowledgements

The author would like to thank the reviewers for their additional hints to related literature.

## References

[1] A. Ambashankar and H. A. Harutyunyan. Broadcasting in stars of cliques. In A. A. Rescigno and U. Vaccaro, editors, *Combinatorial Algorithms - 35th International Workshop, IWOCA 2024, Ischia, Italy, July 1-3, 2024, Proceedings*, volume 14764 of *Lecture Notes in Computer Science*, pages 485–496. Springer, 2024. doi:10.1007/978-3-031-63021-7_37.

[2] M. Cevnik and J. Zerovnik. Broadcasting on cactus graphs. *J. Comb. Optim.*, 33(1):292–316, 2017. doi:10.1007/S10878-015-9957-8.

[3] F. V. Fomin, P. Fraigniaud, and P. A. Golovach. Parameterized complexity of broadcasting in graphs. *Theor. Comput. Sci.*, 997:114508, 2024. doi:10.1016/J.TCS.2024.114508.

[4] P. Fraigniaud and E. Lazard. Methods and problems of communication in usual networks. *Discret. Appl. Math.*, 53(1-3):79–133, 1994. doi:10.1016/0166-218X(94)90180-5.

[5] M. S. Gholami, H. A. Harutyunyan, and E. Maraachlian. Optimal broadcasting in fully connected trees. *J. Interconnect. Networks*, 23(1):2150037:1–2150037:20, 2023. doi:10.1142/S0219265921500377.

[6] H. A. Harutyunyan, A. L. Liestman, J. G. Peters, and D. S. Richards. *The Handbook of Graph Theory*, chapter Broadcasting and gossiping, pages 1477–1494. Chapman and Hall/CRC, 3rd edition, 2013.

[7] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988. doi:10.1002/NET.3230180406.

[8] J. Hromkovič, C. Jeschke, and B. Monien. Optimal algorithms for dissemination of information in some interconnection networks. *Algorithmica*, 10(1):24–40, 1993. doi:10.1007/BF01908630.

[9] A. L. Liestman and J. G. Peters. Broadcast networks of bounded degree. *SIAM J. Discret. Math.*, 1(4):531–540, 1988. doi:10.1137/0401049.

[10] M. Middendorf. Minimum broadcast time is np-complete for 3-regular planar graphs and deadline 2. *Inf. Process. Lett.*, 46(6):281–287, 1993. doi:10.1016/0020-0190(93)90066-I.

[11] P. J. Slater, E. J. Cockayne, and S. T. Hedetniemi. Information dissemination in trees. *SIAM J. Comput.*, 10(4):692–701, 1981. doi:10.1137/0210052.

[12] P. Tale. Double exponential lower bound for telephone broadcast. *CoRR*, abs/2403.03501, 2024. URL: https://doi.org/10.48550/arXiv.2403.03501, doi:10.48550/ARXIV.2403.03501.