# Broadcasting and Multicasting in Trees*

Johanne Cohen† Pierre Fraigniaud‡

Technical Report LRI-1265
Laboratoire de Recherche en Informatique
Univ. Paris-Sud, 91405 Orsay cedex, France

**Abstract**

The *multicast* problem is an information dissemination problem which consists, for one node of a network, to broadcast a data to a specified subset of nodes. The *broadcast* problem is the particular case of the multicast problem in which every node of the network is a destination node. Motivated by the common use of trees as underlying topology to support group-communications in telecommunication systems, this paper solves the multicast and broadcast problems in trees. More precisely, this paper considers these two problems under several variants of the *line model*. This communication model allows long-distance calls to be performed in a single round. The aim of the line model is to cover communication modes such as circuit-switching or wormhole, and certain aspects of ATM systems (virtual path), or WDM optical systems (single-hop routing). The difficulty of the broadcast and multicast problems is sensitive to slight variations of the model such as directed trees vs. undirected trees, single-port constraint vs. all-port constraint, edge-disjoint calls vs. vertex-disjoint calls, etc. For each of these variants, this paper gives a polynomial-time algorithm which returns either an optimal protocol, or a protocol which is optimal up to a constant multiplicative factor.

**Keywords:** Broadcast, multicast, group application.

*Preliminary versions of this paper have been presented in the 10th ACM Symposium on Parallel Algorithms and Architectures (SPAA '98) [5], in the 10th ACM-SIAM Symp. on Discrete Algorithms (SODA '99) [8], and in the DIMACS workshop on Robust Communication Networks [9].

†LORIA, Campus Scientifique, BP239, 54506 Vandœuvre les Nancy, France (`Johanne.Cohen@loria.fr`).

‡Laboratoire de Recherche en Informatique, Bât. 490, Université Paris-Sud, 91405 Orsay cedex, France (`Pierre.Fraigniaud@lri.fr`). Supported by the CNRS, and by the RNRT program R.O.M. The final version of this paper was partially written while the second author was visiting Carleton University at Ottawa, Canada. Additional support by the NATO and by Carleton University.

# Contents

# 1 Introduction

## 1.1 Motivations

Recent advances in telecommunication systems enhanced standard point-to-point communication protocols to multi-point protocols. These latter protocols are of particular interest for group applications. Those groups involve more than two users (some may even involve thousands of users) sharing a common application, e.g., video-conferences, distributed data-bases, media-spaces, games, etc. Several protocols have been proposed to handle and to control a large group of users. We refer to [11, 25] for surveys on multi-point applications and protocols. Solutions differ, e.g., according to the type of traffic which is induced by the shared application, and according to the quality of service required by the users. However, a common feature of these solutions is to propose multi-point architectures based on trees, either a single tree connecting all the group members (e.g., the protocol CBT [1]), or several trees, one for every source (e.g., the protocol PIM [10]). The traffic between the users is then routed along the edges of the tree(s).

The major communication problem related to multi-point applications consists to broadcast a message from one user to all the users of the application. This operation is called *broadcast* at the application level, though it is actually a *multicast* at the network level. The repetition of point-to-point connections between the source and the several destinations would significantly increase the traffic in the network, and it makes this solution not applicable in practice [11]. Therefore, the source must require the help of other nodes to relay messages. A message will then reach its destinations after having been relayed by several intermediate nodes (each intermediate node may possibly get one copy of the message if it belongs to the group). In order to preserve the broadcast application from transmission errors, and to bound the interval between successive receptions of consecutive packets, the number of hops between the source and each destination must be as small as possible. The aim of this paper is to provide polynomial algorithms which for any tree $T = (V, E)$, any source $u \in V$, and any set of destinations $D \subseteq V$, return a multicast protocol from $u$ to $D$ which minimizes the number of hops. The algorithms depend on the communication model for which the protocol is designed.

Several communication models have been considered in the literature. They decompose in mainly two classes: *local* models and *line* models. Local models are motivated by switching techniques such as store-and-forward or deflection routing, whereas line models are motivated by switching techniques such as circuit-switching or wormhole, and certain aspects of ATM (virtual path) or WDM optical systems (single-hop routing). In local models, node-to-node calls must be placed between neighbors, whereas, in line models, calls can be placed between nodes at distance greater than one. As a consequence, the diameter of the network (more precisely, the eccentricity of the source) is a lower bound on the number of hops required to solve the broadcast problem under local models, whereas this is no more a constraint under line models. The multicast and broadcast problems have been both solved in trees under several variants of the local model, but they are still open under most variants of the line model. This paper solves both multicast and broadcast problems in trees under all standard variants of the line model.

## 1.2 Line models

Communications proceed by sequence of simultaneous *calls*. During a given call, the source of the call sends its information to the destination of the call. A *round* is defined as a set of simultaneous calls. The efficiency of a communication protocol is estimated in terms of number of rounds. A multicast protocol performing in $k$ rounds is *optimal* if there is no protocol performing in less than $k$ rounds that achieves the same multicast. The construction of optimal protocols is strongly dependent of the communication constraints. The *single-port constraint* states that every node can give a call to at most one other node at a time, whereas the *all-port constraint* states that every node can call as many nodes at a time, though not more than its out-degree.

In the single-port local model, the set of calls at any given round forms a matching in the network. The structure of the set of calls performed at the same round in the line model is more complex, and it depends on additional hypotheses. In every variant of the line model, a call is a path connecting a source to one or more destination(s) according to whether the model is single-port or all-port. The *edge-disjoint constraint* states that simultaneous calls must be pairwise edge-disjoint. This constraint is motivated by the wish to avoid link-contention in the network. A stronger constraint aims to avoid node-contention: the *vertex-disjoint constraint* states that the calls performed at the same round must be pairwise vertex-disjoint. We will consider both *directed* and *undirected* trees. In the directed case, the links are supposed to be oriented from the source of the multicast (the root of the tree) toward the leaves. Thus, in the directed case, the edge-disjoint constraint states that two simultaneous calls must not share any *arc*.

Let $D \subseteq V$ be the destination-nodes of a multicast from $u \in V$. Every node can be involved to *transmit* calls. The *restricted regimen* states that only nodes in $D$ can be used to *relay* messages. More precisely, under the restricted regimen, a node $v \notin D$ can be crossed by calls, i.e., by paths, placed between nodes in $D$, but $v$ cannot be used to receive a message at a given round in order to forward it to other nodes later. To clarify this restriction, consider a directed tree $T$ in which the source (i.e., the root) $u$ has a single child $v$, and $v$ has $k > 2$ children $w_1, \dots, w_k$. Let $D = \{w_1, \dots, w_k\}$. In the unrestricted regimen all-port line model, multicast from $u$ to $D$ can be performed in 2 rounds: $u$ calls $v$ at the first round, and $v$ simultaneously calls $w_1, \dots, w_k$ at the second round. However, still in the all-port line model, multicast from $u$ cannot be performed in less than $k$ rounds under the restricted regimen because, for every $i$, $1 \le i \le k$, $w_i$ can be called only by $u$ since $v$ cannot be used to relay messages, but only to transmit calls. The restricted regimen is motivated by the wish to limit the use of nodes that are not directly involved in the group-communication because (1) not all routers in current telecommunication systems are able to run multicast protocols, and (2) nodes not involved in a group application may not accept to be bothered by the traffic of this application. Therefore, although it looks a bit artificial at a first glance, the restricted regimen is of major importance. Note that the restricted regimen does not apply for $D = V$, that is as far as the broadcast problem is concerned.

Therefore, we are considering four alternatives: single-port constraint vs. all-port constraint, edge-disjoint calls vs. vertex-disjoint calls, directed trees vs. undirected trees, and restricted regimen vs. unrestricted regimen. The all-port vertex-disjoint line model is not considered since the authors are not aware of systems in which nodes are able to initiate several calls simultaneously, but are unable to be traversed by more than a single call at a time.

### 1.3 Previous works

**Local model.** The local model has been investigated in the literature since the early 50's (see in particular the survey [17]). With the growing interest in both parallel systems and telecommunication systems, a huge literature from the late 80's to nowadays has been devoted to specific group-communication problems (see the surveys [14, 18]). In particular, the decision problem related to broadcasting under the single-port local model was proved to be NP-complete for arbitrary networks [31] (see also [24]). It gave rise to several approximation algorithms [2, 22, 29] and heuristics [15, 30]. Trees deserved a specific interest as it was shown that computing an optimal broadcast protocol in a tree under the single-port local model is polynomial [28]. A polynomial-time algorithm to find the nodes having minimal broadcast-time among all nodes of a tree, has been derived in [31]. The broadcast problem in undirected trees has been studied in [13] under the hypothesis that, at the beginning of the process, many nodes know the information. Finally, it was shown that, for any $n$, there exists an undirected tree whose broadcast-time from *any* source is at most $\log_\varrho n$ where $\varrho = \frac{1+\sqrt{5}}{2}$ (see [16, 23]), and that it is the best that can be achieved. From all these results, one can say that the broadcast problem is solved in trees under the local model.

The situation is different when long-distance calls are allowed, that is under the line model.

**Single-port edge-disjoint line model.** In this model, it was shown that every undirected $n$-node graph has a broadcast-time $\lceil \log_2 n \rceil$ (see [12], and also [21]). Interestingly, this result can be extended to the case in which the routes along which are performed the calls are chosen according to a shortest path routing function (see [7]). However, these results do not hold in directed graphs: take as a counter example the digraph $G$ in which a node $u$ has a unique outgoing arc to a node $v$, which has in turn $n-2$ outgoing arcs to $n-2$ vertices $w_1, \ldots, w_{n-2}$, each connected by an outgoing arc to node $u$. It takes $\lceil \frac{n}{2} \rceil$ rounds to broadcast from $u$ in $G$ under the single-port edge-disjoint line model. Actually, the broadcast problem is NP-complete for arbitrary directed networks in this model [6].

**Single-port vertex-disjoint line model.** In this model, the broadcast problem is also NP-complete for arbitrary networks [6]. It was therefore studied for specific architectures such as cycles or toruses [19, 20]. More interestingly, an $O(\frac{\log n}{\log \log n})$-approximation algorithm has been derived [22].

**All-port (edge-disjoint) line model.** In this model, the broadcast problem is NP-complete [6]. Some results have been however derived for specific architectures such as toruses or hypercubes [26, 27]).

**Tree networks.** For trees, very little is known under line models, apart in two cases.

- Under the single-port edge-disjoint line model, the result in [12] applies, that is the broadcast time of any $n$-node undirected tree is $\lceil \log_2 n \rceil$.
- Under the single-port vertex-disjoint line model, an $O(n^3)$-time algorithm which returns an optimal broadcast protocol in any directed tree is known (see [3, 4]). It was also shown in [3] that the duration of an optimal broadcast protocol in directed tree under the single-port vertex-disjoint line model is at most twice as long as an optimal broadcast protocol under the single-port edge-disjoint line model. However, it is pointed out in [4] that the complexity of broadcasting in directed trees under the single-port edge-disjoint line model remains unsolved.

Again, the main purpose of this paper is to solve the broadcast and multicast problems for trees in all variants of the line model.

### 1.4 Our results.

All our results are summarized in Table 1. More precisely:

| | | | edge-disjoint single-port | edge-disjoint all-port | vertex-disjoint single-port |
|---|---|---|---|---|---|
| **Undirected trees** | *Broadcast* | | Opt. [7, 12] | Opt. [Th. 3.2] | 3-apx [Th. 5.2] |
| | *Multicast* | *Unrest. reg.*<br>*Rest. reg.* | Opt. [7]<br>Opt. [7] | Opt. [Th. 6.1]<br>Open pb. | 3-apx [Cor. 6.2]<br>Open pb. |
| **Directed trees** | *Broadcast* | | 2-apx [4] & [Th. 5.3] | Opt. [Th. 3.1] | Opt. [4] & [Th. 5.1] |
| | *Multicast* | *Unrest. reg.*<br>*Rest. reg.* | 2-apx [Cor. 6.1]<br>2-apx [3, 4] & [Cor. 6.4] | Opt. [Th. 6.1]<br>Opt. [Th. 6.3] | Opt. [Th. 6.2]<br>Opt. [3, 4] & [Cor. 6.3] |

Table 1: Broadcast and multicast problems in trees under the variants of the line model.

- We have almost completely solved the multicast problem in *directed and undirected trees* under the *all-port edge-disjoint* line model. More precisely, for all but one of the variants of the model, we have given a polynomial-time algorithm which returns an optimal multicast protocol under the constraints of that variant. The multicast problem in directed trees under the restricted regimen is let open. The worst-case time-complexity of our slowest algorithm is $O(n \log n)$ for the broadcast problem, and $O(n^2)$ for the multicast problem, where $n$ denotes the number of nodes of the tree.

- For every variant of the *single-port vertex-disjoint* line model, we have given a polynomial-time algorithm which returns an optimal multicast protocol in *directed trees*. The worst-case time-complexity of our slowest algorithm is $O(n^2)$. This improves the worst case complexity of the algorithm in [4] by a factor of $\Theta(n)$.

- For every variant of the *single-port edge-disjoint* line model, we have given an $O(n^2)$-time 2-approximation algorithm for the multicast problem in *directed trees*. This latter result improves the worst case complexity of the 2-approximation algorithm in [3, 4] by a factor of $\Theta(n)$.

- For all variants of the *single-port vertex-disjoint* line model, but one, we have given an $O(n \log n)$-time 3-approximation algorithm for the multicast problem in *undirected trees*. The multicast problem under the restricted regimen is let open.

### 1.5 Organization of the paper.

Section 2 presents some preliminary results, and introduces several notions that are crucial for the purpose of this paper. Then, Section 3 shows how to construct optimal broadcast protocols in trees under the all-port variants of the line model. Section 4 solves a matrix problem, called the *contention-free matrix problem*, whose solution allows to construct optimal broadcast protocols in directed trees under the single-port vertex-disjoint line model. This construction is given in Section 5 which focuses on the broadcast problem in trees under the single-port variants of the line model. Section 6 shows how to generalize all the previous results to the multicast problem. Finally, Section 7 contains some concluding remarks.

**Notation.** Throughout all the paper, $T = (V, E)$ denotes a tree (directed or not) rooted at the source $u_0$ of the broadcast (or multicast) process. For any $v \in V$, $T_v$ denotes the subtree of $T$

rooted at $v$ (i.e., if $w$ is the first node reached from $v$ along the shortest path from $v$ to $u$ in $T$, then removing the edge $(v, w)$ decomposes $T$ in two subtrees, $T_v$, containing $v$, and $T \setminus T_v$, containing $u$ and $w$). The number of nodes of $T$ is denoted by $n$, and the number of nodes of $T_v$, $v \neq u_0$, is denoted by $|T_v|$.

## 2 Shadows, and lexical optimality

A broadcast protocol $\mathcal{B}$ from $u_0$ in $T$ can be described by the list of calls performed by $\mathcal{B}$ at every round. Our constructions of optimal broadcast protocols are based on the following definition.

**Definition 2.1** *Let $\mathcal{B}$ be a broadcast protocol from $u_0$ performing in $r$ rounds in a tree $T = (V, E)$ rooted at $u_0$. The* shadow *of $\mathcal{B}$ on an edge $e \in E$ is the array* shad$(\mathcal{B}, e) = (x_1, \ldots, x_r)$*, $x_i \in \{-1, 0, 1\}$, such that*

- $x_i = 1$ *if and only if there is a call passing* downward *through $e$ at round $i$ of $\mathcal{B}$;*
- $x_i = -1$ *if and only if there is a call passing* upward *through $e$ at round $i$ of $\mathcal{B}$; and*
- $x_i = 0$ *if and only if no call is passing through $e$ at round $i$ of $\mathcal{B}$.*

Let "$\prec$" be the lexicographic order on the words whose letters are in $\{-1, 0, 1\}$, and satisfying $-1 \prec 0 \prec 1$.

**Definition 2.2** *Let $T = (V, E)$ be any tree, let $\mathcal{B}$ be a broadcast protocol from the root $u_0$ of $T$, and let $e \in E$. $\mathcal{B}$ is said* lexicographically optimal in $e$ *if* shad$(\mathcal{B}, e) \preceq$ shad$(\mathcal{B}', e)$*) for any broadcast protocol $\mathcal{B}'$ from $u_0$ in $T$.*

Note that a broadcast protocol $\mathcal{B}$ that is lexicographically optimal in $e$ does not necessarily minimize the number of calls passing through $e$.

**Remark.** For directed trees, shad$(\mathcal{B}, e) \in \{0, 1\}^r$ and thus a shadow can be viewed as an integer: shad$(\mathcal{B}, e) = (x_1, \ldots, x_r) = \sum_{i=1}^{r} x_i 2^{r-i}$. In this context, the relation "$\prec$" is simply the natural order "$<$" on the integers.

Some variants of the line model allow broadcast protocols to be lexicographically optimal on every edge simultaneously, as stated in the following theorem whose proof is reported in Appendix A.

**Theorem 2.1** *Under the all-port edge-disjoint line model, for any directed or undirected tree $T$ rooted at $u_0$, there exists an optimal broadcast protocol from $u_0$ in $T$ which is lexicographically optimal on every edge of $T$.*

However, the global lexical optimality cannot be always achieved for every variant of the line model. As a counter example, consider the single-port constraint, and let $T$ be the directed tree of three vertices $u_0, x$ and $y$, and two arcs $e_x = (u_0, x)$ and $e_y = (u_0, y)$. There are two possible optimal broadcast protocols from $u_0$ in $T$: $\mathcal{B}_x$ consists for $u_0$ to call $x$ first, and then $y$; $\mathcal{B}_y$ consists for $u_0$ to call $y$ first, and then $x$. We have shad$(\mathcal{B}_x, e_x) =$ shad$(\mathcal{B}_y, e_y) = (0, 1)$, but there is no broadcast protocol $\mathcal{B}$ satisfying shad$(\mathcal{B}, e_x) =$ shad$(\mathcal{B}, e_y) = (0, 1)$. Nevertheless, the single-port vertex-disjoint line model satisfies the following property.

**Lemma 2.1** *Let $T$ be a directed tree rooted at $u$. Assume $u$ has a unique child $v$. Let $e = (u, v)$, and let $\mathcal{B}$ be a broadcast protocol from $u$ under the single-port vertex-disjoint line model. For any integer $t >$ shad$(\mathcal{B}, e)$, there is a broadcast protocol $\mathcal{B}_t$ from $u$ such that* shad$(\mathcal{B}_t, e) = t$. *Moreover, $\mathcal{B}_t$ can be constructed from $t$ and $\mathcal{B}$ in $O(\log t)$ time.*

**Proof.** Let $t = (t_1, \ldots, t_s) > \text{shad}(\mathcal{B}, e) = (x_1, \ldots, x_r)$. Let $x' = (x'_1, \ldots, x'_s) = (0, \ldots, 0, x_1, \ldots, x_r)$ with $s - r$ zeros inserted at the front. $\mathcal{B}_t$ performs as follows.

- If $s > r$ then let $k = s - r$. For every $i$, $1 \leq i \leq k$, if $t_i = 1$ then $u$ calls $v$ at round $i$ of $\mathcal{B}_t$.
- If $s = r$ then let $k$ be the smallest index such that $t_k \neq x'_k$. For every $i$, $1 \leq i < k$, $\mathcal{B}_t$ performs at round $i$ as $\mathcal{B}$ does at the same round. At round $k$, $u$ calls $v$ in $\mathcal{B}_t$ (whereas $u$ was idle in $\mathcal{B}$ at round $k$).

The fact that $v$ is informed at round $k$ in $\mathcal{B}_t$ allows to simulate the role of $u$ in $\mathcal{B}$ during the remaining rounds of $\mathcal{B}_t$. More precisely, let $i \in \{k+1, \ldots, s\}$.

- If $t_i = x'_i$ then the calls of $\mathcal{B}_t$ performed at round $i$ are the calls of $\mathcal{B}$ performed at round $i - (s - r)$.
- If $t_i = 0$ and $x'_i = 1$ then the calls of $\mathcal{B}_t$ performed at round $i$ are the calls of $\mathcal{B}$ performed at round $i - (s - r)$, excepted that, if $u$ calls $w$ in $\mathcal{B}$ then $v$ calls $w$ in $\mathcal{B}_t$, and $u$ stays idle in $\mathcal{B}_t$. (This transformation is valid since $v$ is necessarily idle in $\mathcal{B}$ because it is traversed by a call from $u$).
- If $t_i = 1$ and $x'_i = 0$ then the calls of $\mathcal{B}_t$ performed at round $i$ are the calls of $\mathcal{B}$ performed at round $i - (s - r)$, excepted that
  - if $v$ is idle in $\mathcal{B}$ then $u$ calls $v$ in $\mathcal{B}_t$;
  - if $v$ calls $w$ in $\mathcal{B}$ then $u$ calls $w$ in $\mathcal{B}_t$, and $v$ stays idle in $\mathcal{B}_t$.

By construction, shad$(\mathcal{B}_t, e) = t$. Given $\mathcal{B}$ and $t$, the construction of $\mathcal{B}_t$ can be done in $O(\log t)$ phases by sequentially considering the $\lceil \log_2 t \rceil$ bits of the binary decomposition of $t$. At every phase, $\mathcal{B}_t$ is obtained from $\mathcal{B}$ by a constant number of operations. ■

Note that Lemma 2.1 does not holds in the single-port *edge-disjoint* line model. As a counter example, let $T$ be the directed tree of six nodes: $u, v, w_1, w_2, w_3, w_4$, where $u$ has a unique child $v$, and $v$ has fours children $w_i$, $i = 1, \ldots, 4$. Here is the list of calls of a broadcast protocol $\mathcal{B}$ from $u$ in $T$ performing in three rounds:

- Round 1: $u \to v$;
- Round 2: $u \to w_1$ and $v \to w_2$;
- Round 3: $u \to w_3$ and $v \to w_4$.

Let $e = (u, v)$, and let $t = 8$. We have shad$(\mathcal{B}, e) = 7$, thus $t >$ shad$(\mathcal{B}, e)$. However, one cannot broadcast from $u$ in $T$ in four rounds by giving a single call through $e$. Indeed, this round would be used by $u$ to inform $v$, and $u$ would be idle the rest of the time. Node $v$ would then need four additional rounds to inform the $w_i$'s.

**Vanishing and advanced calls.** Let us conclude this section by some simple remarks. Let $T = (V, E)$ be a tree rooted at $u_0$. Let $\mathcal{B}$ be a broadcast protocol from $u_0$ in $T$ that is lexicographically optimal in $e \in E$, and let $\text{shad}(\mathcal{B}, e) = (x_1, \ldots, x_r)$. Let $\mathcal{B}'$ be another broadcast protocol from $u_0$ in $T$, and let $\text{shad}(\mathcal{B}', e) = (y_1, \ldots, y_s)$, $s \geq r$. Let $x' = (x'_1, \ldots, x'_s) = (0, \ldots, 0, x_1, \ldots, x_r)$ with $s - r$ zeros at the front. If $T$ is directed then, from Definition 2.2, if there exists $i$ such that $x'_i = 1$ and $y_i = 0$, then we say that $x'_i$ *vanishes* in $\mathcal{B}'$. In that case, since $\mathcal{B}$ is lexicographically optimal in $e$, there exists $j < i$ such that $x'_j = 0$ and $y_j = 1$. Let $k$ be the largest index smaller than $i$ such that $x'_k < y_k$. The call $x'_i$ of $\mathcal{B}$ is said *advanced* at round $k$ in $\mathcal{B}'$. Similarly, if $T$ is undirected then, from Definition 2.2, if there exists $i$ such that $x'_i > y_i$, then we say that $x'_i$ vanishes in $\mathcal{B}'$. In that case, there exists $j < i$ such that $x'_j < y_j$. Again, with the same definition for $k$, the downward call $x'_i$ or the idle time $x'_i$ of $\mathcal{B}$ is said *advanced* at round $k$ in $\mathcal{B}'$, either by performing a call downward through $e$ in $\mathcal{B}'$ at round $k$ (whereas either no call or an upward call is performed through $e$ in $\mathcal{B}$ at round $k$), or by staying idle in $\mathcal{B}'$ at round $k$ (whereas round $k$ is used for an upward call through $e$ in $\mathcal{B}$). We summarize this discussion for further references as follows:

**Lemma 2.2** *A call (resp., a downward call or an idle time) of a lexicographically optimal broadcast protocol $\mathcal{B}$ in a directed tree (resp., undirected tree) which vanishes in another broadcast protocol $\mathcal{B}'$ must have been advanced in $\mathcal{B}'$ compared to its original position in $\mathcal{B}$.*

# 3 All-port constraint

This section in concerned with the all-port model. We start by considering directed trees. Undirected trees are considered afterward.

## 3.1 Directed trees

**Lemma 3.1** *Let $T$ be a directed tree rooted at $u_0$. The broadcast time of $u_0$ in $T$ is at most $2\lceil \log_2 n \rceil$ under the all-port edge-disjoint line model.*

**Proof.** The proof is by induction on $k = \lceil \log_2 n \rceil$. The result holds for $k = 1$. Assume it holds for $n \leq 2^k$, $k \geq 1$, and let $n \in \{2^k + 1, \ldots, 2^{k+1}\}$. Let $x$ be the node of $T$ such that $|T_x| \geq n/2$, and $|T_y| < n/2$ for every child $y$ of $x$. (Such a property is satisfied by exactly one vertex $x$.) Now, let us consider the following broadcast protocol from $u_0$. At the first round $u_0$ calls $x$. At the second round, $u_0$ is idle, and $x$ calls simultaneously all its children. Then, by induction hypothesis, $u_0$ can broadcast in $T \setminus T_x$ in at most $2k$ rounds, and every child $y$ of $x$ can broadcast in its subtree $T_y$ in at most $2k$ rounds. Thus the whole protocol takes at most $2 + 2k = 2\lceil \log_2 n \rceil$ rounds. ■

Our algorithm for constructing an optimal broadcast protocol in a directed tree under the all-port constraint proceeds bottom-up, from the leaves toward the root, by merging lexicographically optimal broadcast protocols in subtrees.

**Lemma 3.2** *Let $T$ be a directed tree rooted at $u_0$, let $v \in V$, $v \neq u_0$, and let $u$ be the parent of $v$ in $T$. Let $T_1, \ldots, T_p$ be the $p$ subtrees of $T$ rooted at the $p$ children $w_1, \ldots, w_p$ of $v$. Assume that, for any $i = 1, \ldots, p$, we know a broadcast protocol $\mathcal{B}_i$ from $v$ in $T_i$ which is lexicographically optimal on $e_i = (v, w_i)$. There is an $O(p \log n)$-time algorithm which returns a broadcast protocol from $u$ in $T_v$ which is lexicographically optimal on $e = (u, v)$.*

**Proof.** We merge the protocols $\mathcal{B}_i$'s. Let $q_i$ be the number of rounds of $\mathcal{B}_i$, i.e., $q_i$ is the length of the boolean array $\text{shad}(\mathcal{B}_i, e_i)$. Let $q = \max_{i=1,\dots,p} q_i$, and let $M$ be the $(p+1) \times q$ boolean matrix whose first row (say row 0) has a single 1-entry located at column $q$. Row $i$ of $M$, $1 \le i \le p$, consists of $\text{shad}(\mathcal{B}_i, e_i)$ possibly complemented with heading 0-entries if $q_i < q$. From $M$, we construct a boolean matrix $M'$ as follows. If every column of $M$ contains at most one 1-entry, then $M' = M$. Otherwise, let $c$ be the index of the leftmost column of $M$ containing more than one 1-entry. (That is $c$ is the smallest index such that column $c$ contains more than one 1-entry.) If there is no zero-column (that is a column with 0-entries only) left to column $c$, then add a zero-column at the beginning of $M$, relabel the columns from 1 to $q+1$, and let $d = 1$. Otherwise, let $d < c$ be the index of the rightmost zero-column located left to column $c$. Then move the 1-entry of row 0 from the last column to column $d$. The remaining entries are not modified, and the construction of $M'$ is completed.

Let $\mathcal{B}$ be the broadcast protocol obtained from $M'$ as follows. Let $d$ be the position of the 1-entry on row 0 of $M'$. For every $i$, $i = 1, \dots, d-1$, $u$ calls $x$ at round $i$ of $\mathcal{B}$ if and only if $v$ calls $x$ at round $i$ of some $\mathcal{B}_j$. At round $d$, $u$ calls $v$. Then $u$ stays idle for all the remaining rounds. All the other calls of $\mathcal{B}$ are those of the $\mathcal{B}_j$'s. Note that $v$ can be involved in more than a single call simultaneously since two or more $\mathcal{B}_j$'s may simultaneously place a call from $v$ after round $d$. Nevertheless, this is allowed by the all-port constraint.

The time required to construct $\mathcal{B}$ is the time it takes to construct $M'$, that is $O(pq)$. From Lemma 3.1, $q_i \le 2\lceil \log_2 n \rceil$ for every $i$. Therefore, $\mathcal{B}$ is constructed in $O(p \log n)$ time.

We claim that $\mathcal{B}$ is lexicographically optimal in $e$. The rest of the proof is dedicated to the proof of that claim. If there was no added zero-column, then $\text{shad}(\mathcal{B}, e) = (x_1, \dots, x_{d-1}, 1, 0, \dots, 0)$ where, for $i < d$, $x_i = \sum_{j=1}^{p} \text{shad}(\mathcal{B}_j, e_j)_i \in \{0,1\}$. If a zero-column has been added to $M$, then $\text{shad}(\mathcal{B}, e)$ has $q+1$ entries, and $\text{shad}(\mathcal{B}, e) = (1, 0, \dots, 0)$. Assume for the purpose of contradiction that there exists a broadcast protocol $\mathcal{B}'$ from $u$ in $T_v$ such that $\text{shad}(\mathcal{B}', e) < \text{shad}(\mathcal{B}, e)$. Let $k$ be the round at which $u$ calls $v$ in $\mathcal{B}'$. We consider two cases.

**Case 1.** $\text{shad}(\mathcal{B}, e)$ has $q+1$ entries. Two sub-cases:

- **(a)** $k < c$. Since there is no zero-column left to column $c$ in $M$, some $\mathcal{B}_i$ is giving a call through $e_i$ at round $k$. Therefore, from Lemma 2.2, this call has been advanced in $\mathcal{B}'$, say at round $k' < k$. In turn, since there is no zero-column left to column $c$ in $M$, some $\mathcal{B}_j$ is giving a call through $e_j$ at round $k'$. Therefore, from Lemma 2.2, this call has been advanced in $\mathcal{B}'$, say at round $k'' < k' < k$. We construct in this way a decreasing sequence of indices. Since this sequence is bounded from below, it necessarily ends, yielding a contradiction.
- **(b)** $k \ge c$. The contending 1's of column $c$ imply that one of these calls must be advanced in $\mathcal{B}'$ at round $k' < c \le k$. Applying the reasoning of Case 1(a), we construct again a decreasing sequence of indices yielding a contradiction.

**Case 2.** $\text{shad}(\mathcal{B}, e)$ has $q$ entries. Let $\text{shad}(\mathcal{B}', e) = (y_1, \dots, y_q)$, and let $\ell$ be the smallest index such that $y_\ell < x_\ell$. Note that $\ell \le d$ because $x_i = 0$ for every $i > d$. Moreover, $\ell \ne k$ because $y_k = 1$. Two sub-cases:

- **(a)** $k < \ell$. Since $y_i = x_i$ for every $i < \ell$, in particular $x_k = y_k = 1$ and thus some $\mathcal{B}_i$ is giving a call through $e_i$ at round $k$. From Lemma 2.2, this call must be advanced in $\mathcal{B}'$, say at round $k' < k$. By repeating for $k'$ what we did for $k$, we construct a decreasing sequence of indices. Since this sequence is bounded from below, it necessarily ends, yielding a contradiction.

**(b)** $\ell < k$. Two sub-cases:

**(*i*)** $k \leq d$. Hence we have $\ell < d$, and thus, since $x_\ell = 1$, some $\mathcal{B}_i$ is giving a call through $e_i$ at round $\ell$. Since $y_\ell = 0$ and $v$ is not yet informed at the $\ell$-th round of $\mathcal{B}'$, from Lemma 2.2, this call must be advanced in $\mathcal{B}'$, say at round $k' < \ell < k$. Applying the reasoning of Case 2(a), we construct again a decreasing sequence of indices yielding a contradiction.

**(*ii*)** $k > d$. Two sub-cases:

- If $d < k < c$, then since there are exactly one 1-entry on every column of $M$ between column $d$ and column $c$, some $\mathcal{B}_i$ is giving a call through $e_i$ at round $k$. From Lemma 2.2, this call must be advanced in $\mathcal{B}'$ at round $k' < k$. We construct again a decreasing sequence of indices yielding a contradiction.
- If $c \leq k$, then the contending 1's of column $c$ imply that one of these calls must be advanced in $\mathcal{B}'$ at round $k' < c \leq k$. We construct again a decreasing sequence of indices yielding a contradiction.

All cases lead to a contradiction. Therefore $\mathcal{B}$ is lexicographically optimal, which completes the proof. ■

**Theorem 3.1** *There is an $O(n \log n)$-time algorithm which, given any directed tree $T$ rooted at $u_0$, returns an optimal broadcast protocol from $u_0$ in $T$ under the all-port edge-disjoint line model.*

**Proof.** The algorithm proceeds bottom-up, from the leaves toward the root. For each arc $e = (u, v)$ incoming to a leaf $v$, the optimal broadcast protocol $\mathcal{B}_u$ from $u$ in $T_v$ consists of a unique call from $u$ to $v$. We get $\text{shad}(\mathcal{B}_u, e) = (1)$. Let $v \in V$, $v \neq u_0$, be an internal node, and let $u$ be the father of $v$ in $T$. Let $T_1, \dots, T_p$ be the $p$ subtrees of $T$ rooted at the $p$ children $w_1, \dots, w_p$ of $v$. Let $e_i = (v, w_i)$, $i = 1, \dots, p$. Assume that, for every $i = 1, \dots, p$, we know a broadcast protocol $\mathcal{B}_i$ that is lexicographically optimal in $e_i$. From Lemma 3.2, a broadcast protocol $\mathcal{B}_u$ that is lexicographically optimal in $e = (u, v)$ can be constructed in $O(p \log n)$ time. At the root, let $S_1, \dots, S_p$ be the $p$ subtrees of $T$ rooted at the $p$ children $v_1, \dots, v_p$ of $u_0$. Let $e_i = (u_0, v_i)$, $i = 1, \dots, p$. Given the $p$ broadcast protocols $\mathcal{B}_i$ that are lexicographically optimal in $e_i$, respectively, merging these protocols at $u_0$ takes a constant time since $u_0$ can perform several calls simultaneously. The resulting protocol is an optimal broadcast protocol from $u_0$ in $T$ because it is lexicographically optimal in all incident edges of $u_0$.

The total time required by this bottom-up construction is $O(\sum_{v \in V} \deg^+(v) \log n)$ where $\deg^+(v)$ is the number of children of $v$ in $T$. Therefore, the whole construction takes $O(n \log n)$ time. ■

## 3.2 Undirected trees

We proceed for undirected trees in a similar way as we did for directed trees. However, merging broadcast protocols is a bit more tricky in undirected trees than in directed trees because of the upward calls. (Recall that we are dealing with the all-port model here.)

**Lemma 3.3** *Let $T$ be a tree rooted at $u_0$, let $v \in V$, $v \neq u_0$, and let $u$ be the parent of $v$ in $T$. Let $T_1, \dots, T_p$ be the $p$ subtrees of $T$ rooted at the $p$ children $w_1, \dots, w_p$ of $v$. Assume that, for any $i = 1, \dots, p$, we know a broadcast protocol $\mathcal{B}_i$ from $v$ in $T_i$ that is lexicographically optimal on $e_i = (v, w_i)$. There is an $O(p \log n)$-time algorithm which returns a broadcast protocol from $u$ in $T_v$ that is lexicographically optimal on $e = (u, v)$.*

**Proof.** We use the same notation as in the proof of Lemma 3.2. Therefore, we start with a matrix $M$ of $p+1$ rows and $q$ columns labeled from left to right. The first row (row 0) contains only zeros, but a single 1-entry at column $q$. The $p$ other rows are the $p$ shadows of the protocols $\mathcal{B}_i$'s. Note that $M$ has its entries in $\{-1, 0, 1\}$. From $M$, we derive a matrix $M'$ as follows. If every column of $M$ satisfies that the sum of its entries is in $\{-1, 0, 1\}$, then $M' = M$. Otherwise, let $c$ be the smallest index such that column $c$ has the sum $S$ of its entries not in $\{-1, 0, 1\}$. If $S < -1$ then the 1-entry of the first row is moved to the column $c$. If $S > 1$, then let $d < c$ be the largest index smaller than $c$ such that the sum of the entries of column $d$ is in $\{-1, 0\}$. If there is no such column, then a zero-column is added to the left of the first column of $M$. The 1-entry of the first row is moved to the column $d$. The remaining entries are not modified. The construction of $M'$ is completed.

Let $\mathcal{B}$ be the broadcast protocol from $u$ in $T_v$ obtained from $M'$ as follows. Let $d$ be the column-index of the 1-entry of the row 0 of $M'$. For every $j$, $j = 1, \dots, d-1$, let $M'_{i,j}$, $i = 1, \dots, p$, be the $p$ entries of rows 1 to $p$ of the column $j$ of $M'$ (row 0 contains a 0-entry). Since $\sum_{i=1}^{p} M'_{i,j} \in \{-1, 0, 1\}$, we pair every 1 with a $-1$, such that at most one 1-entry is not paired, or at most one $-1$-entry is not paired. If $M'_{i_1,j}$ is paired with $M'_{i_2,j}$, say $M'_{i_1,j} = -1$ and $M'_{i_2,j} = 1$, then the source of the upward call of $\mathcal{B}_{i_1}$ calls the destination of the downward call of $\mathcal{B}_{i_2}$. If it remains a 1 that is not paired, say $M'_{i_0,j} = 1$ is not paired, then $u$ calls the destination of that call of $\mathcal{B}_{i_0}$. If it remains a $-1$ that is not paired, say $M'_{i_0,j} = -1$ is not paired, then the source of that call of $\mathcal{B}_{i_0}$ calls $u$ (this call is actually useless for the purpose of transmitting information, but it allows to minimize the shadow on $e$). At round $d$, there are three cases according to the sum $\Sigma$ of the entries of column $d$, distinct from the 1-entry of row 0.

- If $\Sigma < -1$, then the 1's are paired with the $-1$'s. It remains at least two $-1$ that are unpaired, say $M'_{i_1,d}$ and $M'_{i_2,d}$. In $\mathcal{B}$, the upward call of $\mathcal{B}_{i_1}$ has destination $v$, and the upward call of $\mathcal{B}_{i_2}$ has destination $u$. The possible other upward calls are removed.
- If $\Sigma = -1$, then the 1's and the $-1$'s are paired as before. The remaining $-1$, say $M'_{i_0,d}$, corresponds to an upward call of $\mathcal{B}_{i_0}$. In $\mathcal{B}$, this call has destination $v$.
- If $\Sigma = 0$, then the 1's and the $-1$'s are paired as before, and a call from $u$ is added in $\mathcal{B}$ to inform $v$.

After round $d$, $u$ stays idle in $\mathcal{B}$ and node $v$ takes in charge the calls inside $T_v$ corresponding to the 1-entries in the shadows of the $\mathcal{B}_i$'s. Moreover, $v$ provides upward calls through $e$ toward $u$ at every round.

As for Lemma 3.2, the complexity of computing $\mathcal{B}$ from the $\mathcal{B}_i$'s is $O(p \log n)$. Note that Lemma 3.1 holds for the undirected case as well.

The rest of the proof is dedicated to proving that $\mathcal{B}$ is lexicographically optimal on $e$. If there is no added column, then $\text{shad}(\mathcal{B}, e) = (x_1, \dots, x_{d-1}, x_d, -1, \dots, -1)$ where, for $i < d$,

$x_i = \sum_{j=1}^{p} \text{shad}(\mathcal{B}_j, e_j)_i$, and $x_d = sign(1 + \sum_{j=1}^{p} \text{shad}(\mathcal{B}_j, e_j)_d)$. If a column has been added, then $\text{shad}(\mathcal{B}, e)$ has $q+1$ entries, and $\text{shad}(\mathcal{B}, e) = (1, -1, \ldots, -1)$. Assume, for the purpose of contradiction, that there exists a broadcast protocol $\mathcal{B}'$ such that $\text{shad}(\mathcal{B}', e) < \text{shad}(\mathcal{B}, e)$. Let $k$ be the round at which $v$ is informed in $\mathcal{B}'$. Note that, as opposed to Lemma 3.2, $v$ is not necessarily informed by $u$.

**Claim 3.1** *No $\mathcal{B}_i$ has a downward call through $e_i$ or an idle time of $e_i$ advanced to some round $k' < \min\{k, c\}$ in $\mathcal{B}'$.*

*Proof.* Assume, for the purpose of contradiction, that there exists such a $\mathcal{B}_i$. Let $S^+$ (resp. $S^-$) be the number of $\mathcal{B}_j$'s giving a call upward (resp. downward) through the $e_j$'s at round $k'$. Since $k' < c$, $|S^+ - S^-| \in \{0, 1\}$. Since $k' < k$, $v$ cannot be involved in a call in $\mathcal{B}'$ because it is not yet informed. Therefore, from Lemma 2.2, a downward call or an idle time of some $\mathcal{B}_{i'}$ is advanced from round $k'$ to some round $k'' < k'$. One can repeat for $k''$ what we did for $k'$. We construct in this way a decreasing sequence of indices. Since this sequence is bounded from below, it necessarily ends, and gives rise to $\text{shad}(\mathcal{B}', e) \geq \text{shad}(\mathcal{B}, e)$, a contradiction. ◇

Let $\text{shad}(\mathcal{B}', e) = (y_1, \ldots, y_q)$, and let $\ell$ be the smallest index such that $y_\ell < x_\ell$. We consider several cases according to the value of the sum $S$ of the entries of column $c$.

**Case 1.** $S < -1$, that is there are at least two more $-1$-entries than 1-entries in column $c$ of $M$. In that case, $\ell < d = c$ because $x_d = -1$. Two sub-cases:

- **a)** $\ell \leq k$. Let us consider round $\ell$. If $x_\ell = 1$ and $y_\ell = 0$ or $-1$, the pairing of the $-1$ and 1-entries of column $\ell$ of $M$ either lets unsatisfied in $\mathcal{B}'$ at least one $\mathcal{B}_i$ giving a call downward through $e_i$ at round $\ell$, or one idle $\mathcal{B}_j$ must give a call upward through $e_j$. From Lemma 2.2, this downward call or this idle time must be advanced in $\mathcal{B}'$ at a round $k' < \ell$. If $x_\ell = 0$ and $y_\ell = -1$, then $\mathcal{B}'$ afford the ability of giving a call upward through $e$. Thus, from Lemma 2.2, either a call downward through $e_i$ of some $\mathcal{B}_i$, or an idle time of some $\mathcal{B}_i$, has been advanced in $\mathcal{B}'$, at round $k' < \ell$. Since $\ell < \min\{k, c\}$, we get a contradiction with Claim 3.1.
- **b)** $\ell > k$. Let us consider round $k$. In order to inform $v$ in $\mathcal{B}'$, either $u$ calls $v$, or some $\mathcal{B}_i$ that is able to place a call upward does it. Let $S'$ be the sum of the entries of the $k$-th column of $M$. Since $k < \ell < d = c$, we have $S' \geq -1$ and $x_k = y_k$. Thus either a downward call, or an idle time of some $\mathcal{B}_i$, has been advanced at round $k' < k$ in $\mathcal{B}'$. Since $k = \min\{k, c\}$, we get a contradiction with Claim 3.1.

**Case 2.** $S > 1$, that is there are at least two more 1-entries than $-1$-entries on column $c$ of $M$. In that case, $\ell < c$ because $\ell \leq d$ and $d < c$. If $k < \ell$, then we use the same arguments as Case 1(b). If $\ell < d$, then we use the same arguments as in Case 1(a). Otherwise, three sub-cases assuming $\ell = d \leq k$:

- **a)** $\ell = d = k < c$. Since $y_\ell < x_\ell$, a downward call or an idle time of some $\mathcal{B}_i$ has been advanced from round $\ell$ to some round $k' < \ell = \min\{k, c\}$ in $\mathcal{B}'$. This is a contradiction with Claim 3.1.
- **b)** $\ell = d < k < c$. Every column $\gamma$ of $M$, $d < \gamma < c$, has the sum of its entries equal to 1. Therefore, for $v$ to be informed at round $k$, a downward call or an idle time of some $\mathcal{B}_i$ has been advanced at some round $k' < k = \min\{k, c\}$. This is a contradiction with Claim 3.1.

**c)** $\ell = d < c \leq k$. Then at least two contending 1's in column $c$ forces one of these calls to be advanced at some round $k' < c = \min\{c, k\}$. This is a contradiction with Claim 3.1.

**Case 3.** The sum of the entries of every column of $M$ is in $\{-1, 0, 1\}$. Therefore $M' = M$ and $d = q$. Consider round $k$ and get a contradiction with claim 3.1 (with $c = +\infty$).

All cases lead to a contradiction. Therefore $\mathcal{B}$ is lexicographically optimal in $e$. ■

**Theorem 3.2** *There is an $O(n \log n)$-time algorithm which, given any tree $T$ rooted at $u_0$, returns an optimal broadcast protocol from $u_0$ in $T$ under the all-port edge-disjoint line model.*

**Proof.** Same as the proof of Theorem 3.1 by applying Lemma 3.3 instead of Lemma 3.2. ■

## 4 The contention-free matrix problem

This section describes a polynomial-time algorithm solving a matrix problem similar to the matrix problems solved in the proofs of Lemma 3.2 and Lemma 3.3. The interest of this problem will appear clearly in the forthcoming sections.

**Definition 4.1** *Given a $p \times q$ boolean matrix $M$, a* contention-free version *of $M$ is a $p \times q'$ boolean matrix $M'$, $q' \geq q$, such that:*

- *$M'$ has at most one 1-entry per column, and*
- *every row $r$ of $M'$, viewed as the binary representation of an integer, is larger than the corresponding row $r$ of $M$, $1 \leq r \leq p$.*

For instance, for any $p \times q$ boolean matrix $M$, the $p \times (p + q)$ boolean matrix $M'$ whose $p$ first columns form the $p \times p$ identity matrix, and the $q$ last columns form the zero-matrix, is a contention-free version of $M$. Such a solution $M'$ may not be "minimum" even in term of number of columns. The following definition makes explicit the parameter that we want to optimize.

**Definition 4.2** *Let $M$ be a $p \times q$ boolean matrix, the* shadow *of $M$, denoted by* $\text{shad}(M)$, *is the boolean array of size $q$ such that the $i$-th entry of* $\text{shad}(M)$ *is* 1 *if and only if there is at least one* 1*-entry in the $i$-th column of $M$.*

We look for contention-free version of $M$ that are optimal in the following sense.

**Definition 4.3** *Given a $p \times q$ boolean matrix $M$, a contention-free version $M'$ of $M$ is* minimal *if* $\text{shad}(M')$*, viewed as the binary representation of an integer, is minimum among the shadows of all the contention-free versions of $M$.*

Note that there can be several minimal contention-free versions of a matrix, even up to a permutation of the rows. On the other hand, the shadow of the possibly many minimal contention-free versions of a matrix is unique. As an example, let us consider the matrix

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

The reader can check that two minimal contention-free versions of $M$ are

$$M_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

$M_1$ and $M_2$ are two different matrices, even up to a permutation of their rows. However, $M_1$ and $M_2$ have both a shadow equal to $62 = (111110)_2$. The following result, by Cohen, Fraigniaud, and Mitjana, was announced in [8]:

**Theorem 4.1** *There is an $O(q(p+q))$-time algorithm which computes a minimal contention-free version of any $p \times q$ boolean matrix.*

A sketch of the proof of Theorem 4.1 can be found in [9]. The complete proof is reported thereafter.

Let $M$ be a $p \times q$ boolean matrix. We describe an algorithm which transforms $M$ into a minimal contention-free version of $M$. This is achieved via a sequence of elementary matrix operations of two types (columns are labeled from left to right):

- insertion of a zero-column at position 0 of the current matrix, and
- shifting of an existing zero-column from position $t-1$ to position $t$, that is an exchange between columns $t-1$ and $t$.

The shift operation has an important consequence on the 1-entries of the matrix. When a zero-column is shifted one position to the right, from position $t-1$ to position $t$, the entries of the matrix are modified according to the following rule:

**Rule 1.** For every $i$, $1 \leq i \leq p$, if there is a 1-entry originally at column $t$ and row $i$, then, after the exchange of a zero-column at position $t-1$ with the column $t$, all 1-entries of row $i$ at position $t' > t$ are switched to 0.

Rule 1 is motivated by the fact that, for any $k$, $2^{k+1} > \sum_{i=0}^{k} a_i 2^i$ for any $a_i \in \{0,1\}$, $i = 0,\ldots,k$. Therefore, any row modified according to Rule 1 is larger than the original, whatever are its entries left to position $t$.

Our algorithm is formally described in Algorithm 1. An example of the run of Algorithm 1 is provided on Figure 1. Informally, Algorithm 1 performs as follows. The $q$ columns of $M$ are considered from left to right (Instruction 1). Problems occur when there are two or more 1-entries in the current column (Instruction 6). On Figure 1(a), this occurs at column 4 since there is a single 1-entry in each of the three leftmost columns of $M$. The goal of Algorithm 1 is to produce enough zero-columns on the left of the current column to spread out the contending 1's over these zero-columns. Therefore, Algorithm 1 tries to increase the number of zero-columns on the left of the current column by shifting existing zero-columns from their current position to their right, and by applying Rule 1 (Instruction 14). Possibly, a zero-column is inserted at position 0 (Instruction 20), and additional zero-columns are provided (Instruction 26). These latter zero-columns are then inserted directly to the left of the current column for the purpose of speeding up the execution of the algorithm.

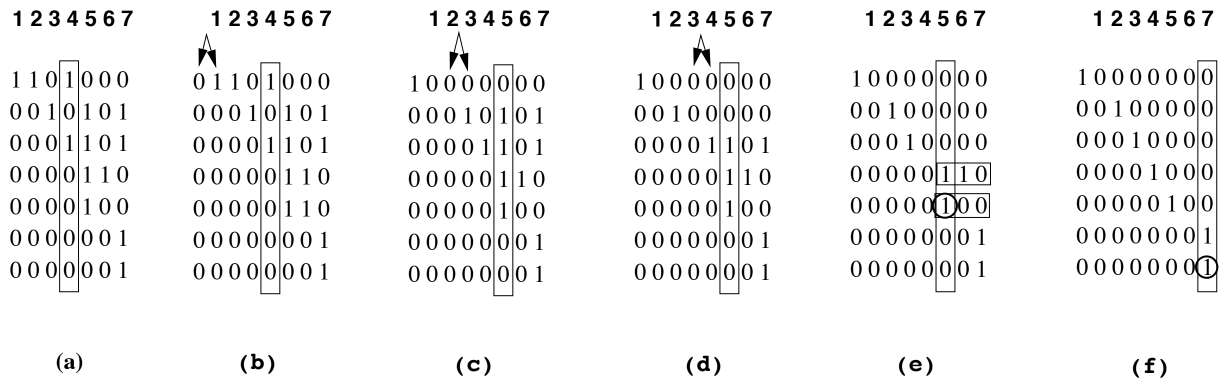


Figure 1: An example of the execution of Algorithm 1.

On Figure 1(a), there is no zero-column at the current phase of the algorithm, and thus a zero-column is inserted at position 0, as shown on Figure 1(b). Then the two first columns are exchanged and Rule 1 is applied. This application has a major consequence on the matrix: all 1-entries of the first row, but the leading 1-entry, are switched to 0. This creates a new zero-column, and one of the two contending 1's of column 4 vanishes (see Figure 1(c)).

Algorithm 1 then considers position 5 (now the 6th column from the left). Four 1-entries are contending at position 5 of the matrix. The rightmost zero-column is then shifted to the right. It is worth to notice that it is the rightmost zero-column on the left of the current column which is considered. Choosing this column instead of any zero-column has an important effect of the shadow of the resulting matrix. The effect of the shift in the example of Figure 1 is to delete one contending 1-entry (see Figure 1(d)). The zero-column is then shifted once more to the right. Again, it deletes one contending 1-entry (see Figure 1(e)). Once there are enough zero-columns to solve all conflicts between 1-entries in the current column, the contending 1's are spread out over these zero-columns. Note that if after all possible shifting there is still not enough zero-columns to absorb the contenting 1's, then some additional zero-columns are inserted (Instruction 26). In our example, there is one zero-column and there are two contending 1's, so there is no need to insert a new zero-column (see Figure 1(e)). The choice of the unique 1-entry of column 5 which is *not* moved to a zero-column matters. Algorithm 1 keeps in place the 1-entry which corresponds to the row with the *minimum lexicographic order*, starting from the current column (Instruction 28). In our example, it means that the 1-entry of row 5 is let in place, while the 1-entry of row 4 is moved to the zero-column. Indeed, from the current position, row 4 is 110 whereas row 5 is 100, and $110 > 100$ in lexicographic order.

At this point of the running of Algorithm 1 on the example of Figure 1, we are let with the matrix on Figure 1(f) in which the last 1-entry of row 4 has been switched to 0. Letting in place the 1-entry of the smallest row in lexicographic order has the effect to postpone other conflicts with this row as far as possible. In the example of Figure 1, letting in place the 1-entry of row 5 transforms the penultimate column of the matrix into a zero-column. Therefore, the conflict appearing at position 7 can be solved easily by moving one of the two 1-entries at position 6.

The shadow of the resulting matrix is $(10111111)_2$, and we claim that it is minimum among all contention-free versions of the matrix of Figure 1(a).

**Algorithm 1** This algorithm computes a contention-free version of a $p \times q$ boolean matrix $M$.

**For** $i$:=1 **to** $q$ **do**
  /* *We sparse the columns from column 1 to column q* */
  $\mathcal{C}_i$ := current column;
  **If** $\mathcal{C}_i$ is a zero-column **then**
    $Z := Z \cup \{\mathcal{C}_i\}$;
    /* *Z currently denotes the set of zero-columns left to the current column* */
  **Else**
    **If** there is more than a single 1-entry in $\mathcal{C}_i$ **then**
      $nb_1$ := # 1's in $\mathcal{C}_i$;
      $W$ := set of consecutive zero-columns immediately to the left of $\mathcal{C}_i$;
      not_yet_inserted := true;
      **While** ($nb_1 > |W| + 1$) **and** ($Z \neq W$ **or** not_yet_inserted) **do**
      /* *while there is not enough zero-column immediately to the left of* $\mathcal{C}_i$ */
        **If** $Z \neq W$ **then**
        /* *A zero-column can be moved rightward* */
          $Z' := Z \setminus W$;
          $c$ := rightmost zero-column in $Z'$;
          Shift $c$ one column to the right, and apply rule 1;
          $Z$ := set of zero-columns left to $\mathcal{C}_i$;
          $W$ := set of consecutive zero-columns immediately to the left of $\mathcal{C}_i$;
          $nb_1$ := # 1's in $\mathcal{C}_i$;
        **EndIf**
        **If** ($nb_1 > |W| + 1$) **and** ($W = Z$) **and** (not_yet_inserted) **then**
        /* *One needs to insert a zero-column at position 0 of the matrix* */
          Insert a zero-column at position 0;
          not_yet_inserted := false;
          $Z$ := set of zero-columns left to $\mathcal{C}_i$;
          $W$ := set of consecutive zero-columns immediately to the left of $\mathcal{C}_i$;
        **EndIf**
      **EndWhile**
      **If** $nb_1 > |W| + 1$ **then** insert $nb_1 - |W| - 1$ zero-columns left to $\mathcal{C}_i$;
      /* *If there is not enough zero-columns to solve all contentions,* */
      /* *then additional zero-columns are inserted immediately to the left of* $\mathcal{C}_i$ */
      /* *The* $nb_1$ *1's are now spread out over the zero-columns of* $W$ */
      Truncate each row with a 1 in $\mathcal{C}_i$ in order to keep only entries to the right of $\mathcal{C}_i$;
      $\ell$ := index of the row of minimum lexicographic order among the truncated rows;
      $W' := nb_1 - 1$ rightmost columns of $W$
      Let the 1-entry of row $\ell$ in place in $\mathcal{C}_i$, and spread out the $nb_1$ other 1's of $\mathcal{C}_i$ over $W'$;
      $Z$ := set of zero-columns left to the current column;
    **EndIf**
  **EndIf**
**EndFor**

The fact that Algorithm 1 computes a minimal contention-free version of any $p \times q$ boolean matrix $M$ is based on the following three lemmas. For any matrix $M$, $\text{shad}^*(M)$ denotes the shadow of any minimal contention-free version of $M$. Note that $\text{shad}^*(M) \geq \text{shad}(M)$, and that $\text{shad}^*(M) = \text{shad}(M)$ if and only if $M$ has at most one 1-entry in each of its columns.

**Lemma 4.1** *Let $A_i$ and $B_i$ be the $i$-th row of two matrices $A$ and $B$, respectively. If $A_i \leq B_i$ for every $i$, then* $\text{shad}^*(A) \leq \text{shad}^*(B)$.

**Proof.** Let $B^*$ be a minimal contention-free version of $B$. For every $i$, $B_i^* \geq B_i \geq A_i$. Therefore, $B^*$ is a contention-free version of $A$. Thus $\text{shad}^*(A) \leq \text{shad}(B^*) = \text{shad}^*(B)$. ∎

**Lemma 4.2** *Let $A$ and $B$ be two matrices of $p$ rows, $p \geq 0$, and $q_A$ and $q_B$ columns, respectively. Let $X_i, Y_i$, $i = 1, \ldots, k$, be $2k$ 1-dimensional arrays, $k \geq 2$, where the $X_i$'s are of size $q_A$ and the $Y_i$'s are of size $q_B$. Let*

$$M = \begin{pmatrix} X_1 & 0 & \ldots & 0 & 1 & Y_1 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ X_k & 0 & \ldots & 0 & 1 & Y_k \\ A & 0 & \ldots & 0 & 0 & B \end{pmatrix}$$

*be a matrix of $p+k$ rows, and $q_A + k + q_B$ columns. Assume that there is at most one 1-entry in each of the $q_A$ leftmost columns of $M$. For $i = 1, \ldots, k$, let*

$$M^{(i)} = \begin{pmatrix} X_1 & * & \ldots & * & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ X_{i-1} & * & \ldots & * & 0 & 0 \\ X_i & 0 & \ldots & 0 & 1 & Y_i \\ X_{i+1} & * & \ldots & * & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ X_k & * & \ldots & * & 0 & 0 \\ A & 0 & \ldots & 0 & 0 & B \end{pmatrix}$$

*where the $*$'s forms the $(k-1) \times (k-1)$ identity matrix. Let $i_0$ be such that $Y_{i_0} \leq Y_i$ for every $i \neq i_0$. Then* $\text{shad}^*(M) = \text{shad}^*(M^{(i_0)})$.

**Proof.** From Lemma 4.1, $\text{shad}^*(M) \leq \text{shad}^*(M^{(i_0)})$. The reverse inequality is a bit more delicate to establish. Let $M^*$ be a minimal contention-free version of $M$.

**Claim 4.1** *There is no zero-column in $M^*$ between columns $q_A + 1$ and $q_A + k$.*

*Proof.* For the purpose of contradiction, assume the reverse, and let $c$ be the index of this zero-column. Let $R$ be the set of rows of $M$ whose some 1-entry has been "moved" left to column $q_A + 1$ in $M^*$, that is rows of $M^*$ of the form $r^* = (x_1, \ldots, x_{q_A}, 0, \ldots, 0)$, $x_i \in \{0, 1\}$, satisfying $r^* > r$ where $r$ is the original row in $M$. Since there is a zero-column between columns $q_A + 1$ and $q_A + k$, and since the contention between the $k$ 1-entries of column $q_A + k$ must be solved, $R$ is not empty. Let $j$ be the index of the row of $R$ whose rightmost 1-entry is the most to the right among all rows in $R$. Let $c'$ be the column index of the rightmost 1-entry of row $j$. By moving the 1-entry of row $j$ from position $c'$ to position $c$, and by restoring the original entries of this row between position $c'$ and $q_A$, we get a contention-free version $M'$ of $M$. Indeed, there is at most one 1-entry on each of the $q_A$ leftmost columns of $M$, so $M'$ has still at most one 1-entry on each column by the choice of row $j$. Moreover, row $j$ of $M'$ is still larger than the original row $j$ of $M$. Since $\text{shad}(M') < \text{shad}(M^*)$, we get a contradiction. $\diamond$

Based on this claim, we proceed in two cases:

**Case 1.** All the $k$ 1-entries of column $q_A + k$ of $M$ have been moved to the left. Then one can reorder entries of $M^*$ between columns $q_A + 1$ and $q_A + k$ so that the resulting matrix $N$ is a contention-free version of $M^{(i_0)}$, and $\text{shad}(N) = \text{shad}(M^*)$. Indeed, the fact that there is no zero-column in $M^*$ between columns $q_A + 1$ and $q_A + k$ make exchanges between the 1-entries on these columns possible. (Recall that, in the statement of the theorem, the $*$'s form the $(k-1) \times (k-1)$ identity matrix, this is why the rearrangement is required.)

**Case 2.** One of the $k$ 1-entries of column $q_A + k$ of $M$ stands at the same position in $M^*$. Then let $i$ be the row index of this entry.

– If $i = i_0$, then, again, by reordering the positions of the $k - 1$ other 1-entries of $M^*$ between columns $q_A + 1$ and $q_A + k$, solving the contentions at column $q_A + k$ of $M$, $M^*$ can be transformed in a contention-free version $N$ of $M^{(i_0)}$ such that $\text{shad}(N) = \text{shad}(M^*)$.

– If $i \neq i_0$, then let $Y_i^*$ be the $q_B$ rightmost entries of row $i$ in $M^*$, and let $c$ be the column index of the rightmost 1-entry of row $i_0$ in $M^*$. We consider two sub-cases.

**a)** If $c > q_A$, then let $N$ be the matrix obtained from $M^*$ by:

1. Exchanging all the entries of row $i$ with those of row $i_0$, from position $c$.
2. Possibly reordering the positions of some 1-entries of rows $1, \ldots, k$ of $M^*$ between columns $q_A + 1$ and $q_A + k$ to make each of these rows larger than the corresponding row of $M^{(i_0)}$.

We have $\text{shad}(N) = \text{shad}(M^*)$, and $N$ is a contention-free version of $M^{(i_0)}$ because $Y_i^* \geq Y_i \geq Y_{i_0}$.

**b)** If $c \leq q_A$, then at least one of the 1-entries of the $p \times (q_A + k + q_B)$ matrix $A0B$ has been moved left to column $q_A + k$ since, otherwise, there would be a zero-column in $M^*$ between columns $q_A + 1$ and $q_A + k$, which is impossible by Claim 4.1. Therefore, let $R$ be the set of rows of the $p \times (q_A + k + q_B)$ matrix $A0B$ whose some 1-entry has been moved left to column $q_A + k$. Let $j$ be the index of the row of $R$ whose rightmost 1-entry is the most to the right among rows in $R$. Let $c'$ be the column index of the rightmost 1-entry of row $j$. We have $c' > q_A$. Indeed, otherwise, there would be a zero-column between position $q_A$ and $q_A + k$, in contradiction with Claim 4.1. Then let $N$ be the matrix obtained from $M^*$ by:

1. Exchanging all the entries of row $i$ with those of row $j$, from position $c'$.
2. Possibly reordering the positions of some of the 1-entries of rows $1, \ldots, k$ of $M^*$ between columns $q_A + 1$ and $q_A + k$ to make each of these rows larger than the corresponding row of $M^{(i_0)}$.

$N$ is a contention-free version of $M^{(i_0)}$ because $1Y_i^* \geq 0B_j$. Moreover, $\text{shad}(N) = \text{shad}(M^*)$.

In each case, we can construct a contention-free version $N$ of $M^{(i_0)}$ such that $\mathrm{shad}(N) = \mathrm{shad}(M^*) = \mathrm{shad}^*(M)$. Therefore $\mathrm{shad}^*(M^{(i_0)}) \leq \mathrm{shad}^*(M)$. ■

Given two matrices $A$ and $B$ of the same number of rows $p$, and of $q$ and $q'$ columns respectively, $A|B$ denotes the $p \times (q+q')$ matrix obtained by putting $A$ and $B$ next to each other, $A$ on the left, and $B$ on the right.

**Lemma 4.3** *Let $M = A|x|B|y|C$ where $A$ is a matrix of $q_A$ columns, $q_A \geq 0$, with at most one 1-entry per column, $x$ is a zero-column, $B$ is a matrix of $q_B$ columns, $q_B \geq 1$, with exactly one 1-entry per column, $y$ is a column with at least two contending 1-entries, and $C$ is an unspecified boolean matrix of $q_C$ columns, $q_C \geq 0$. Let $M'$ be the matrix resulting from $M$ after an exchange between $x$ and the leftmost column of $B$. We have* $\mathrm{shad}^*(M) = \mathrm{shad}^*(M')$.

**Proof.** From Lemma 4.1, $\mathrm{shad}^*(M) \leq \mathrm{shad}^*(M')$. The remaining of the proof is dedicated to the proof of the reverse inequality. Let $M^*$ be a minimal contention-free version of $M$. Assume, w.l.o.g., that the 1-entry of the leftmost column of $B$ stands on row 1. Let $M^*_i$ be the $i$th row of $M^*$. We consider two cases.

Case 1. $M^*_1 \geq (A_1, 1, 0, \ldots, 0)$ where $A_1$ is the first row of $A$, and the 1-dimensional array of the right hand side is of size $q_A+q_B+q_C+2$. Then $M^*$ is a contention-free version of $M'$, and therefore $\mathrm{shad}^*(M) \geq \mathrm{shad}^*(M')$.

Case 2. $(A_1, 1, 0, \ldots, 0) > M^*_1 \geq (A_1, 0, B_1, 0, C_1)$. Since at least one 1-entry of $B|y$ must be moved to the left of column $q_A + 2$ of $M$, let $i$ be a row index of $M^*$ such that (a) $M^*_i \geq (A_i, 1, 0, \ldots, 0)$, and (b) the rightmost 1-entry of $M^*_i$ is the most to the right among rows satisfying inequality (a). Let $c$ be the column-index of the rightmost 1-entry of $M^*_i$. Let $N$ be the matrix obtained from $M^*$ as follows:

1. The $q_B + 1 + q_C$ rightmost entries of row $i$ are set to be equal to the $q_B + 1 + q_C$ rightmost entries of row 1;
2. The $q_A + 1$ leftmost entries of row $i$ are restored as they were originally in $M$;
3. The entry at column $c$ of row 1 is set to 1; and
4. The entries of row 1 left to column $c$ are set to 0.

Since $M^*$ and $A$ are matrices with at most one 1-entry per column, the resulting matrix $N$ satisfies the same property by the choice of row $i$. Moreover, we have $(B_1, y_1, C_1) > (B_i, y_i, C_i)$ because, by hypothesis, the 1-entry of the leftmost column of $B$ stands on row 1. Therefore, $N$ is a contention-free version of $M'$, and $\mathrm{shad}(N) = \mathrm{shad}(M^*)$. Therefore $\mathrm{shad}^*(M) \geq \mathrm{shad}^*(M')$. ■

Now, we know enough to prove Theorem 4.1.

**Proof of Theorem 4.1.** Algorithm 1 constructs a finite sequence of matrices $M_0 = M, M_1, \ldots, M_k$, such that $M_i$ is obtained from $M_{i-1}$ either by shifting a zero-column to the right, or by distributing 1-entries over zero-columns. Lemma 4.2 shows that the distribution of the 1's over the zero-column preserves $\mathrm{shad}^*(M)$. Similarly, Lemma 4.3 shows that the shift of a zero-column also preserves $\mathrm{shad}^*(M)$. Therefore, $\mathrm{shad}^*(M_i) = \mathrm{shad}^*(M_{i-1})$, that is $\mathrm{shad}^*(M) = \mathrm{shad}^*(M_k)$. Since $M_k$ is a contention-free version of $M$, we get that $\mathrm{shad}^*(M) = \mathrm{shad}(M_k)$.

It just remains to compute the time-complexity of Algorithm 1. The *for*-loop is executed $q$ times. However, Instruction 5 is not performed more than $p$ times because there are $p$ rows, and solving a contention between 1-entries creates at least one row whose all entries are null after the current position. Let $i$ be an index of the *for*-loop for which there is a contention. From what was said before, there are at most $p$ such indices. Let $k_i$ be the number of contending 1-entries. All instructions before the *while*-loop do not require more than $O(p+q)$ time units. The *while*-loop is executed at most $q.k_i$ times because each execution of the loop corresponds to a right-shift of a zero-column, and one cannot move a zero-column more than $q$ times to the right, this for each of the $k_i$ 1-entries. Actually, one can slightly modify the algorithm so that there are no more than $q$ right-shifts in total. Indeed, when shifting the zero-columns to the right, one can jump columns that were previously exchanged with a zero-column because rule 1 was already applied for these columns. Therefore, rule 1 is not applied more than $q$ times. Application of rule 1 has a cost of $O(q)$ since at most one row is updated after a right-shift. All other instructions inside the *while*-loop have a cost of $O(p+q)$. Instruction 28 has a cost of $O(q.k_i)$, same as Instruction 30. Therefore, the total time-complexity of Algorithm 1 is $O(q(q+p)+q^2+\sum_i q.k_i)$. We have $\sum_i k_i \leq 2p$ because solving contending 1-entries at a column $c$ removes the 1-entries on the right of column $c$ in all but one of the contending rows. Therefore, we get that the time-complexity of Algorithm 1 is $O(q(q+p))$. ■

# 5 Single-port constraint

In this section, we will show how to apply Theorem 4.1 to derive optimal broadcast protocols in trees under the single-port constraint. We start by the vertex-disjoint constraint.

## 5.1 Vertex-disjoint constraint

Let us first study the case of directed trees. The case of undirected trees comes afterward.

### 5.1.1 Directed trees

Again, we apply a bottom-up construction consisting of merging protocols of intermediate levels.

**Lemma 5.1** *Let $T$ be a directed tree rooted at $u_0$, let $v \in V$, $v \neq u_0$, and let $u$ be the parent of $v$. Let $T_1, \ldots, T_p$ be the $p$ subtrees of $T$ rooted at the $p$ children $w_1, \ldots, w_p$ of $v$. Assume that, for every $i = 1, \ldots, p$, we know a broadcast protocol $\mathcal{B}_i$ from $v$ in $T_i$ which is lexicographically optimal on $e_i = (v, w_i)$. There exists an $O(pn)$-time algorithm which returns a broadcast protocol from $u$ in $T_v$ that is lexicographically optimal on $e = (u, v)$.*

**Proof.** Let $M$ be the $p \times q$ boolean matrix whose $p$ rows are the $p$ shadows $\text{shad}(\mathcal{B}_i, e_i)$, possibly complemented with heading zero-entries if $q$ is larger than the number $q_i$ of entries in $\text{shad}(\mathcal{B}_i, e_i)$.

Let $M^*$ be the minimal contention-free version of $M$ obtained by application of Algorithm 1 on $M$.

Let $\overline{M}$ be the $(p+1) \times q$ boolean matrix obtained from $M^*$ by adding a row at position 0. This row has all its entries set to 0, but one. The single 1-entry of row 0 of $\overline{M}$ is placed at the column-index of the rightmost zero-column of $M^*$ (if any). If $M^*$ has no zero-column, then a zero-column is added at the left of $M^*$, and the 1-entry of row 0 of $\overline{M}$ is placed on this column.

Let $\mathcal{B}$ be the broadcast protocol from $u$ in $T_v$ derived from $\overline{M}$ as follows. Let $c$ be the index of the column containing the 1-entry on row 0 of $\overline{M}$. Let $i < c$. If there is a 1-entry on column $i$, say on row $j \geq 1$, then $u$ gives a call inside $T_j$ at round $i$, otherwise $u$ stays idle. At round $c$, $u$ calls $v$. During the remaining calls, $u$ stays idle. Let $i > c$. If there is a 1-entry on column $i$, say on row $j \geq 1$, then $v$ gives a call inside $T_j$ at round $i$. The destinations of the calls performed from $u$ or $v$ at rounds $i \neq c$ are not specified here. However, since each row of $\overline{M}$ is larger than the corresponding row of $M$, we get from Lemma 2.1 that every row $j \geq 1$ of $\overline{M}$ describes a broadcast protocol $\mathcal{B}'_j$ in $T_j$. $\mathcal{B}'_j$ can be obtained from $\mathcal{B}_j$ and the $j$-th row of $\overline{M}$ in $O(q)$ time. Thus the construction of $\mathcal{B}$ requires $O(pq)$ time, once the matrix $\overline{M}$ has been computed. From Theorem 4.1, it takes $O(p(p+q))$ time to compute $M^*$, and $\overline{M}$ just requires $O(pq)$ additional time units to be constructed. So the whole construction of $\mathcal{B}$ takes $O(p(p+q)) = O(pn)$ time since $q \leq n$.

It remains to show that $\mathcal{B}$ is lexicographically optimal on $e$. Let

$$\text{shad}(M^*) = (x_1, \ldots, x_{c-1}, 0, 1, \ldots, 1),$$

where $x_i \in \{0,1\}$ for every $i = 1, \ldots, c-1$, and there are $k \geq 0$ 1's left to the 0-entry. Then

$$\text{shad}(\mathcal{B}, e) = (x_1, \ldots, x_{c-1}, 1, 0, \ldots, 0).$$

Assume for the purpose of contradiction that there is a broadcast protocol $\mathcal{B}'$ such that $\text{shad}(\mathcal{B}', e) < \text{shad}(\mathcal{B}, e)$. Then let $\overline{M}'$ be the boolean matrix of $p+1$ rows such that there is a 1-entry at row $i \in \{1, \ldots, p\}$ and column $j$ of $\overline{M}'$ if and only if $u$ gives a call to a node of $T_i$ at round $j$ of $\mathcal{B}'$. Moreover, there is a 1-entry on row 0 and column $j$ of $\overline{M}'$ if and only if $u$ calls $v$ at round $j$ of $\mathcal{B}'$. Let $M'$ be the boolean matrix obtained from $\overline{M}'$ by removing row 0. $M'$ is a contention-free version of $M$ because (1) $\text{shad}(\mathcal{B}', e_i) \geq \text{shad}(\mathcal{B}_i, e_i)$ for every $i$ (since the $\mathcal{B}_i$'s are lexicographically optimal), and (2) there is at most one 1-entry per column of $M'$ (since $\mathcal{B}'$ satisfies the single-port constraint). Let

$$\text{shad}(M') = (y_1, \ldots, y_{d-1}, 0, z_1, \ldots, z_r)$$

where $r \geq 0$, $y_i \in \{0,1\}$ and $z_i \in \{0,1\}$ for all $i$, and $d$ is the round at which $u$ calls $v$ in $\mathcal{B}'$. Let

$$\text{shad}(\mathcal{B}', e) = (y_1, \ldots, y_{d-1}, 1, 0, \ldots, 0).$$

Let us show that $\text{shad}(\mathcal{B}') < \text{shad}(\mathcal{B})$ and $\text{shad}(M^*) \leq \text{shad}(M')$ are in contradiction. For that purpose, let us consider three cases according to the relative values of $c$ and $d$.

- If $c > d$ then $\text{shad}(\mathcal{B}') < \text{shad}(\mathcal{B})$ and $\text{shad}(M^*) \leq \text{shad}(M')$ imply $(x_1, \ldots, x_{d-1}) = (y_1, \ldots, y_{d-1})$. Then $\text{shad}(M^*) \leq \text{shad}(M')$ implies $x_d = 0$. On the other hand, $\text{shad}(\mathcal{B}') < \text{shad}(\mathcal{B})$ implies $x_d = 1$, a contradiction.

- If $c = d$ then $\text{shad}(\mathcal{B}') < \text{shad}(\mathcal{B})$ implies that $(y_1, \ldots, y_{d-1}) < (x_1, \ldots, x_{d-1})$. On the other hand, $\text{shad}(M^*) \leq \text{shad}(M')$ implies that $(x_1, \ldots, x_{d-1}) \leq (y_1, \ldots, y_{d-1})$, a contradiction.

- If $c < d$ then $\text{shad}(\mathcal{B}') < \text{shad}(\mathcal{B})$ and $\text{shad}(M^*) \leq \text{shad}(M')$ imply $(x_1, \ldots, x_{c-1}) = (y_1, \ldots, y_{c-1})$. Then $\text{shad}(\mathcal{B}') < \text{shad}(\mathcal{B})$ implies $y_c = 0$. On the other hand, $\text{shad}(M^*) > \text{shad}(M')$ implies $y_c = 1$, a contradiction.

Every case yields a contradiction. Therefore, $\mathcal{B}$ is lexicographically optimal on $e$. ■

**Theorem 5.1** *There is an $O(n^2)$-time algorithm which, given any directed tree $T$ rooted at $u_0$, returns an optimal broadcast protocol from $u_0$ in $T$ under the single-port vertex-disjoint line model.*

**Proof.** Same as the proof of Theorem 3.1. Only the time complexity changes, since merging protocols using Lemma 5.1 is more costly than using Lemma 3.2. The total time required by the bottom-up construction is $O(\sum_{v\in V} \deg^+(v)n) = O(n^2)$. ■

### 5.1.2 Undirected trees

**Theorem 5.2** *There exists an $O(n \log n)$-time* 3*-approximation algorithm for the broadcast problem in trees under the single-port vertex-disjoint line model. More precisely, this approximation algorithm returns a broadcast protocol which is optimal up to an* additive *factor of* $2\lceil \log_2 n\rceil$.

**Proof.** Let $T = (V,E)$ be a tree rooted at $u_0$. We construct a broadcast protocol $\mathcal{B}$ from $u_0$ in $T$ by induction on $k = \lceil \log_2 n\rceil$. If $k = 1$, then $\mathcal{B}$ consists of one call from $u_0$ to the leaf. If $k > 1$ then $T$ is a tree of at least three nodes. Analogously to the proof of Lemma 3.1, let $x$ be the node of $T$ such that $|T_x| > n/2$, and $|T_y| \leq n/2$ for every child $y$ of $x$. Let $y_1,\dots,y_p$ be the $p$ children of $x$. By induction, compute a broadcast protocol $\mathcal{B}_i$ from $y_i$ in $T_{y_i}$, for every $i$. Let $T' = T \setminus T_x$. $T'$ has at most $\lceil \frac{n}{2}\rceil$ nodes. By induction, compute a broadcast protocol $\mathcal{B}'$ from $u_0$ in $T'$. The protocol $\mathcal{B}$ is then the following. At the first round, $u_0$ calls $x$. During the remaining rounds, two broadcast protocols, one from $u_0$ in $T'$, and one from $x$ in $T_x$, are performed in parallel. $\mathcal{B}'$ is applied in $T'$. The broadcast protocol inside $T_x$ performs as follows. Node $x$ calls its $p$ children successively in the following order. Let $\delta_i$ be the number of rounds of $\mathcal{B}_i$, and let $\sigma$ be the permutation of $p$ elements such that $\delta_{\sigma(1)} \geq \ldots \geq \delta_{\sigma(p)}$. Then $x$ calls the $y_i$'s in the order $y_{\sigma(1)},\dots,y_{\sigma(p)}$. Once $y_i$ gets the information, it broadcasts this information in $T_{y_i}$ according to $\mathcal{B}_i$. The construction of $\mathcal{B}$ is completed. This construction takes $O(n\log n)$ time because sorting the $\delta_i$'s takes $O(p \log p)$, and $\sum_{v\in V} O(\deg^+(v)\log(\deg^+(v))) = O(n \log n)$.

We claim that the broadcast protocol $\mathcal{B}$ completes in at most $t + \lceil \log_2 n\rceil$ rounds where $t$ is the number of rounds of an optimal broadcast protocol from $u_0$ in $T$.

The proof is by induction on $k = \lceil \log_2 n\rceil$. The result holds for $k = 1$. Assume that it holds for every $n \leq 2^k$. Let $n \in \{2^k+1,\dots,2^{k+1}\}$, let $T$ be a tree of $n$ nodes, and let $x$ and the $y_i$'s be as defined in the construction of $\mathcal{B}$. By induction, every $\mathcal{B}_i$ completes in

$$\delta_i \leq t_i + 2k$$

rounds, where $t_i$ is the number of rounds of an optimal broadcast protocol from $y_i$ in $T_{y_i}$. Let $\theta$ be the permutation of $p$ elements such that $t_{\theta(1)} \geq \ldots \geq t_{\theta(p)}$. Let $\tau$ be the number of rounds of an optimal broadcast protocol from $x$ in $T_x$. We have

$$\max_{i=1,\dots,p}\{i + t_{\theta(i)}\} - 1 \leq \tau \leq \max_{i=1,\dots,p}\{i + t_{\theta(i)}\}.$$

The upper bound is obtained by calling $y_i$ at round $i$, $i = 1,\dots,p$. The lower bound comes from the single-port constraint. Indeed, this constraint implies that, if a call is given downward through some $e_i = (x, y_i)$, then no downward call can be given through $e_j = (x, y_j)$, $j \neq i$. Therefore, since

$$\max_{i=1,\dots,p}\{(i-1) + t_{\mu(i)}\} \geq \max_{i=1,\dots,p}\{(i-1) + t_{\theta(i)}\}, \text{ for any permutation } \mu \text{ of } p \text{ symbols,}$$

we get that the order $\theta$ is the best possible order in which the calls through the $e_i$'s can be given. In other words, broadcast in $T_{y_{\theta(i)}}$ can't start before round $i-1$ without delaying the broadcast process in some $T_{y_{\theta(j)}}$, $j < i$, which may cause a possible increase of the time to complete the $p$ broadcasts from $x$.

Now, let $\delta$ be the number of rounds of our broadcast protocol from $x$ in $T_x$. We have

$$\delta = \max_{i=1,\dots,p} \{i + \delta_{\sigma(i)}\},$$

where $\delta_{\sigma(1)} \geq \dots \geq \delta_{\sigma(p)}$. We get $\delta \leq \max_{i=1,\dots,p}\{i + \delta_{\theta(i)}\}$, and thus

$$\delta \leq \max_{i=1,\dots,p} \{i + t_{\theta(i)}\} + 2k.$$

Therefore,

$$\delta \leq (\tau + 1) + 2k.$$

In addition, let $\delta'$ be the number of rounds of our broadcast protocol $\mathcal{B}'$, and let $\tau'$ be the number of rounds of an optimal broadcast protocol from $u_0$ in $T'$. From the induction hypothesis, we have

$$\delta' \leq \tau' + 2k.$$

Since the number of rounds of $\mathcal{B}$ is $\max\{1 + \delta, 1 + \delta'\} \leq \max\{\tau + 2k + 2, \tau' + 2k + 1\}$, and since $t \geq \max\{\tau, \tau'\}$, we get that $\mathcal{B}$ completes in at most $t + 2k + 2$ rounds, that is in at most $t + 2\lceil \log_2 n \rceil \leq 3t$ rounds. ■

## 5.2 Edge-disjoint constraint

The broadcast problem has been already solved in undirected trees under the single-port edge-disjoint line model (see [7, 12]). Therefore, this section deals with directed trees only.

**Theorem 5.3** *There exists an $O(n^2)$-time 2-approximation algorithm for the broadcast problem in directed trees under the single-port edge-disjoint line model.*

**Proof.** Let $T$ be a directed tree rooted at $u_0$. Let $t_e$ and $t_v$ be respectively the broadcast time from $u_0$ in $T$ under the single-port edge-disjoint line model, and the broadcast time from $u_0$ in $T$ under the single-port vertex-disjoint line model. We have

$$t_e \leq t_v \leq 2\, t_e. \tag{1}$$

The first inequality $t_e \leq t_v$ is straightforward since pairwise vertex-disjoint paths are pairwise edge-disjoint. The second inequality comes from the fact that, in a directed tree, any set $S$ of pairwise edge-disjoint paths can be decomposed into two subsets $S'$ and $S''$ such that any two paths belonging to the same subset are vertex-disjoint. Indeed, let $G$ be the graph whose vertices are the paths in $S$, and there is an edge between two paths if they share any vertex of $T$. Since $T$ is a tree, there is no cycle in $G$, and thus $G$ is a forest. Therefore, $G$ is bipartite, and its vertices can be colored with two colors. This completes the proof of Equation 1. Since Theorem 5.1 says that an optimal broadcast algorithm under the single-port vertex-disjoint line model can be derived in $O(n^2)$-time, the result follows. ■

# 6 Multicasting

Let $T = (V, E)$ be a tree, let $u_0 \in V$, and let $D \subseteq V$. As stated in the introduction, we have considered two variants of the multicast problem from $u_0$ in $D$. The restricted regimen states that only nodes in $D$ can be used to relay messages. The restricted regimen requires protocols that differ significantly from those derived for the unrestricted regimen. Therefore, this section is divided into two sections focusing on the restricted and the unrestricted regimen respectively.

## 6.1 Unrestricted regimen

**Theorem 6.1** *There is an $O(n \log n)$-time algorithm which, given any tree $T = (V, E)$ (directed or not) rooted at $u_0$, and any $D \subseteq V$, returns an optimal multicast protocol from $u_0$ to $D$ under the all-port edge-disjoint line model.*

**Proof.** We modify the proof of Lemma 3.2 and Lemma 3.3 to take into account that, with the same notations as in the statements of these lemmas, possibly $v \notin D$.

If $v \in D$ then the protocols are merged as described in the proof of Lemmas 3.2 and 3.3.

If $v \notin D$, then the matrix $M$ has $p$ rows instead of $p + 1$. The construction of $M'$ works the same apart that the operation of moving the 1-entry of row 0 from column $q$ to some column $d < q$ (motivated by a conflict in $M$) is transformed into: (1) add a new row in $M'$, (2) place a 1-entry at position $d$ of this row, and (3) set all the remaining entries of this row to 0.

The broadcast protocol $\mathcal{B}$ is then constructed according to the rules stated in the proofs of Lemmas 3.2 and 3.3 (if no additional row is created in $M'$, then $d$ is set to $q + 1$). The lexical optimality of $\mathcal{B}$ is obtained by the same arguments as those of the proofs of Lemma 3.2 and Lemma 3.3, in particular by setting the time $k$ at which $u$ calls $v$ to $k = q+2$ if $v$ is not informed in $\mathcal{B}'$. The whole protocol is then obtained according to the construction in the proof of Theorem 3.1. ■

**Theorem 6.2** *There is an $O(n^2)$-time algorithm which, given any directed tree $T = (V, E)$ rooted at $u_0$, and any $D \subseteq V$, returns an optimal multicast protocol from $u_0$ to $D$ under the single-port vertex-disjoint line model.*

**Proof.** We revisit the construction given in the proofs of Lemma 5.1 and Theorem 5.1. We use the same notation as in these proofs.

Let us start by Lemma 5.1. We are given $u, v$, and the $p$ children $w_i$ of $v$, $i = 1, \ldots, p$. We are also given $p$ multicast protocols $\mathcal{B}_i$ in $D_i \subseteq V(T_i)$ that are respectively lexicographically optimal in $e_i = (v, w_i)$, $i = 1, \ldots, p$. We are looking for a multicast protocol, either from $u$ to $D = \{v\} \cup (\cup_{i=1}^{p} D_i)$, or from $u$ to $D = \cup_{i=1}^{p} D_i$, depending on whether or not $v \in D$. For that purpose, we construct $M$ and $M'$ as in the proof of Lemma 5.1. If $D = \{v\} \cup (\cup_{i=1}^{p} D_i)$, then the resulting protocol $\mathcal{B}$ given in the proof of Lemma 5.1 is lexicographically optimal in $e = (u, v)$. The situation is a little bit more complex if $v \notin D$. In this latter case, we also consider the $p \times q$ boolean matrix $N$ whose $p$ rows are the $p$ shadows $\text{shad}(\mathcal{B}_i, e_i)$. Let $N'$ be a $p \times q'$ minimal contention-free version of $N$.

- If $\text{shad}(M') \leq \text{shad}(N')$, then the multicast protocol $\mathcal{B}$ is the one described in the proof of Lemma 5.1 in the sense that it is worth to use $v$ as relay.

- Otherwise, $\mathcal{B}$ is the following: for $i = 1$ to $q'$, if there is a 1-entry on column $i$ of $N'$, then $u$ gives a call at round $i$, otherwise $u$ stays idle. If the 1-entry of column $i$ is located on row $j$, then the destination of the call is in $T_j$, and it can be computed in $O(q')$ time using Lemma 2.1.

$\mathcal{B}$ is lexicographically optimal in $e = (u, v)$ because $M'$ yields a multicast protocol that has the smallest shadow on $e$ among the protocols using $v$ to relay messages, and $N'$ yields a multicast protocol that has the smallest shadow on $e$ among the protocols that are *not* using $v$ to relay messages.

The bottom-up construction of the proof of Theorem 5.1 then works the same, although it is not applied directly on $T$ but on the tree $T_D$ obtained from $T$ and $D$ by pruning $T$ so that every leave belongs to $D$. ■

For the same reasons as those given in the proof of Theorem 5.3, it is a direct consequence of Theorem 6.2 that an optimal solution for the multicast problem in directed trees under the single-port edge-disjoint line model can be approximated up to a factor of two.

**Corollary 6.1** *There exists an $O(n^2)$-time 2-approximation algorithm for the multicast problem in directed trees under the single-port edge-disjoint line model.*

Moreover, we have:

**Corollary 6.2** *There exists an $O(n \log n)$-time 3-approximation algorithm for the multicast problem in trees under the single-port vertex-disjoint line model. More precisely, this approximation algorithm returns a multicast protocol that is optimal up to an* additive *factor of $2\lceil \log_2 |D| \rceil$ where $D$ is the set of destinations.*

**Proof.** The proof is based on the same protocol as the one described in the proof of Theorem 5.2. The only difference is that the induction is now based on the number of destination nodes. In other words, $x$ is the node such that (1) $x$ has more than $\frac{|D|}{2}$ destination nodes in its subtree, and (2) every child of $x$ has at most $\frac{|D|}{2}$ destination nodes in its subtree. Note that $x$ may or may not be a destination node. However, both cases are allowed under the unrestricted regimen. Note also that the same analysis as in the proof of Theorem 5.2 works because the multicast time from $u_0$ to $D$ is at least the multicast time from $x$ to $D \cap T_x$, and it is at least the multicast time from $x$ to $(T \setminus (T_x \setminus \{x\})) \cap D$. ■

## 6.2 Restricted regimen

In this section, we assume that only nodes in the destination set $D$ can be used to relay messages.

**Corollary 6.3** *There is an $O(n^2)$-time algorithm which, given any directed tree $T = (V, E)$ rooted at $u_0$, and any $D \subseteq V$, returns an optimal multicast protocol from $u_0$ in $D$ under the restricted-regimen single-port vertex-disjoint line model.*

**Proof.** Let $T = (V, E)$ be a tree, and let $D \subset V$ be a set of destination nodes. Let $T_D$ be the tree obtained from $T$ and $D$ as follows. Let $x \notin D$, and let $y$ be the parent of $x$. Node $x$ is removed and every child of $x$ (if any) becomes a new child of $y$. This operation is repeated until all nodes belong to $D$.

Let $P$ be the shortest path from $a \in D$ to $b \in D$ in $T$, and let $P'$ be the shortest path from $a' \in D$ to $b' \in D$ in $T$. Each of these two paths corresponds to a shortest path in $T_D$ with same extremities. Let $P_D$ and $P'_D$ be these two paths. $P$ and $P'$ are vertex-disjoint in $T$ if and only if $P_D$ and $P'_D$ are vertex-disjoint in $T_D$. Therefore the broadcast time from $u_0$ in $T_D$ is exactly the multicast time from $u_0$ in $D \subseteq V$. An optimal broadcast protocol from $u_0$ in $T_D$ can be computed in $O(n^2)$ time by application of Theorem 5.1. This protocol can be transformed into a multicast protocol from $u_0$ to $D$ in $O(1)$ time. ■

**Theorem 6.3** *There is an $O(n^2)$-time algorithm which, given any directed tree $T = (V, E)$ rooted at $u_0$, and any $D \subseteq V$, returns an optimal multicast protocol from $u_0$ in $D$ under the restricted-regimen all-port edge-disjoint line model.*

**Proof.** We transform $T$ into a tree $T_D$ obtained from $T$ and $D$ as follows. $T$ is first pruned so that every leaf is in $D$. Then, let $x \in D$, $x \neq u_0$, let $y$ be a child of $x$, and let $z$ be a child of $y$. If $y \notin D$ and $z \notin D$ then $z$ is removed from $T$, and the children of $z$ (if any) are directly connected to $y$. Repeat this operation until no branch of $T$ from $u_0$ to any leaf contains two consecutive nodes both distinct from $u_0$, and both not in $D$. $T_D$ requires $O(n^2)$ time to be constructed. Because of the restricted regimen, the multicast time from $u_0$ to $D$ in $T$ is the same as the multicast time from $u_0$ to $D$ in $T_D$. Moreover, an optimal multicast protocol from $u_0$ to $D$ in $T_D$ can be directly obtained from an optimal multicast protocol from $u_0$ to $D$ in $T$ by applying the same set of calls. Thus, for the remaining of the proof, we assume that $D$ is such that every leaf is in $D$ and, for every branch from $u_0$ to a leaf, there is no two consecutive nodes both distinct from $u_0$ and both not in $D$.

Let $v$ be an internal node of $T$, let $u$ be $v$'s parent, and let $w_1, \ldots, w_p$ be the $p$ children of $v$. Let $D_i = D \cap V(T_{w_i})$, and assume that, for every $i$, we know a multicast protocol $\mathcal{B}_i$ from $v$ to $D_i$, which is lexicographically optimal in $e_i = (v, w_i)$. Let us compute the shadow of a multicast protocol $\mathcal{B}_u$ from $u$ to $D \cap V(T_v)$ which is lexicographically optimal in $e = (u, v)$.

If $v \in D$, then $\mathcal{B}_u$ can be obtained in $O(p \log n)$ time by a direct application of Lemma 3.2.

If $v \notin D$ then let $M$ be the $p \times q$ boolean matrix whose $p$ rows are the $p$ shadows $\mathrm{shad}(\mathcal{B}_i, e_i)$. Let $M'$ be a $p \times q'$ minimal contention-free version of $M$. From $M'$, we derive a multicast protocol $\mathcal{B}_u$ from $u$ in $D$ by the usual technique: for $i = 1, \ldots, q'$, if there is a 1-entry on column $i$ of $M'$, say at row $j$, then $u$ calls a node of $T_{w_j}$ at round $i$ of $\mathcal{B}_u$. The destination node of this call can be computed from Lemma 2.1. Note that this lemma does apply to $T_D$ under the restricted regimen because, since $v \notin D$, every $w_i$ is in $D$. This construction requires $O(pn)$ time.

The whole multicast protocol is constructed in $O(n^2)$ time, bottom-up from the leaves in the same way as done in the proof of Theorem 3.1. ■

Finally, as a direct consequence of Corollary 6.3, and by the same arguments as in the proof of Theorem 5.3, we have:

**Corollary 6.4** *There exists an $O(n^2)$-time 2-approximation algorithm for the multicast problem in directed trees under the restricted-regimen single-port edge-disjoint line model.*

## 7 Conclusion

As a brief conclusion, let us just point out directions for further researches.

First, there are two (over eighteen) problems that remains unsolved in Table 1, namely the multicast problem in undirected trees under the restricted regimen, for both the all-port edge-disjoint model and the single-port vertex-disjoint model. The techniques derived in this paper does not seem to apply. Apart these two cases, the broadcast and multicast problems are almost completely solved in trees, for all variants of the line model.

More generally, deriving approximation algorithms for the broadcast (or multicast) problem in arbitrary graphs or digraphs remains a challenging problem in all models. Recall that the best result for the local model is an $O(\log n)$-approximation algorithm [2], and the best known result for the single-port vertex-disjoint line model is an $O(\frac{\log n}{\log \log n})$-approximation algorithm [22]. There is room for improvements.

## References


[1] T. Ballardie, P. Francis, and J. Crowcroft. Core based tree (CBT): an architecture for scalable inter-domain multicast routing. In *proceedings of ACM SIGCOM*, pages 85–95. ACM press, 1993.

[2] A. Bar-Noy, S. Guha, J. Naor, and B. Schieber. Multicasting in heterogeneous networks. In *30th ACM Symposium on Theory of Computing (STOC '98)*, pages 448–453, 1998.

[3] B. Birchler, A.-H. Esfahanian, and E. Torng. Toward a general theory of unicast-based multicast communication. In *21st Workshop on Graph-Theoretic Concepts in Computer Science (WG '95)*, volume 1017 of *LNCS*, pages 237–251, 1995.

[4] B. Birchler, A.-H. Esfahanian, and E. Torng. Information dissemination in restricted routing networks. In *International Symposium on Combinatorics and Applications*, pages 33–44, 1996.

[5] J. Cohen. Broadcasting, multicasting and gossiping in trees under the all-port line model. In *10th ACM Symposium on Parallel Algorithms and Architectures (SPAA '98)*, pages 164–171, 1998.

[6] J. Cohen. *Communications multipoints dans le modèle commuté*. PhD thesis, LRI, Université Paris-Sud, 91405 Orsay, France, 1999.

[7] J. Cohen, P. Fraigniaud, J.-C. Konig, and A. Raspaud. Optimized broadcasting and multicasting protocols in cut-through routed networks. *IEEE Transaction on Parallel and Distributed Systems*, 9(8):788–802, 1998.

[8] J. Cohen, P. Fraigniaud, and M. Mitjana. Scheduling calls for multicasting in tree-networks. In *10th ACM-SIAM Symp. on Discrete Algorithms (SODA '99)*, pages 881–882, 1999.

[9] J. Cohen, P. Fraigniaud, and M. Mitjana. Minimal contention-free matrices with application to multicasting. In *DIMACS Workshop on Robust Communication Networks*, To appear.

[10] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C-G. Liu, and L. Wei. An architecture for wide-area multicast routing. In *SIGCOM*, pages 126–135. ACM press, 1994.

[11] C. Diot, W. Dabbous, and J. Crowcroft. Multipoint communication: a survey of protocols, functions, and mechanisms. *IEEE Journal on Selected Areas in Communications*, 15(3):277–290, 1997.

[12] A. Farley. Minimum-time line broadcast networks. *Networks*, 10:59–70, 1980.

[13] A. Farley and A. Proskurowski. Broadcasting in trees with multiple originators. *SIAM Journal Alg. Disc. Meth.*, 2(4):381–386, 1981.

[14] P. Fraigniaud and E. Lazard. Methods and Problems of Communication in Usual Networks. *Discrete Applied Mathematics*, 53:79–133, 1994.

[15] P. Fraigniaud and S. Vial. Approximation algorithms for broadcasting and gossiping. *Journal of Parallel and Distributed Computing*, 43:47–55, 1997.

[16] H. Harutyunyan. Doctoral Thesis (in Russian), Armenia.

[17] S.M. Hedetniemi, S.T. Hedetniemi, and A. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18:319–349, 1986.

[18] J. Hromković, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting and gossiping). In Ding-Zhu Du and D. Frank Hsu, editors, *Combinatorial Network Theory*, pages 125–212. Kluwer Academic, 1995.

[19] J. Hromković, R. Klasing, and E. Stohr. Dissemination of information in vertex-disjoint paths mode. *Computer and Artificial Intelligence*, 15(4):295–318, 1996.

[20] J. Hromković, R. Klasing, E. Stohr, and Wagener H. Gossiping in vertex-disjoint paths mode in $d$-dimensional grids and planar graphs. *Information and Computation*, 123(1):17–28, 1995.

[21] J. Hromković, R. Klasing, W. Unger, and H. Wagener. Optimal algorithms for broadcast and gossip in the edge-disjoint paths mode. *Information and Computation*, 133(1):1–33, 1997.

[22] G. Kortsarz and D. Peleg. Approximation algorithms for minimum time broadcast. In *Proceedings of the Israel Symposium on Theory of Computer Science*, volume 601 of *LNCS*, pages 67–78, 1992.

[23] R. Labahn. Extremal broadcasting problems. *Discrete Applied Mathematics*, 23:139–155, 1989.

[24] M. Middendorf. Minimum broadcast time is NP-complete for 3-regular planar graphs and deadline 2. *Information Processing Letters*, 46:281–287, 1993.

[25] W. Mostafa and M. Singhal. A taxonomy of multicast protocols for internet applications. *Computer Communications*, 20:1448–1457, 1998.

[26] S. Perennes. Broadcasting in hypercubes under the circuit-switched model. In *IEEE Int. Parallel and Distributed Processing Symp. (IPDPS)*, 2000.

[27] J. Peters and M. Syska. Circuit-switched broadcasting in torus networks. *IEEE Transactions on Parallel and Distributed Systems*, 7(3):246–255, 1996.

[28] A. Proskurowski. Minimum broadcast trees. *IEEE Transactions on Computers*, c-30:363–366, 1981.

[29] R. Ravi. Rapid rumor ramification: approximating the minimum broadcast time. In *35-th IEEE Symposium on Foundations of Computer Science*, pages 202–213, 1994.

[30] P. Scheuermann and G. Wu. Heuristic algorithms for broadcasting in point-to-point computer networks. *IEEE Transactions on Computers*, C-33(9):804–811, 1984.

[31] P. Slater, E Cockayne, and S. Hedetniemi. Information dissemination in trees. *SIAM Journal on Computing*, 10(4):692–701, 1981.

# A Proof of Theorem 2.1

Let us first assume that $T$ is directed. Let $\mathcal{B}$ be an optimal broadcast protocol from $u_0$ in $T = (V, E)$. Assume that $\mathcal{B}$ is not lexicographically optimal on every arc. The arcs for which $\mathcal{B}$ is not lexicographically optimal are called "bad" arcs. Let us consider the bad arc $e = (u, v) \in E$ that is closest to the root (if there are more than one such arc, pick one arbitrarily). We have $\text{shad}(\mathcal{B}, e) > \text{shad}(\mathcal{B}_e, e)$ where $\mathcal{B}_e$ is a broadcast protocol from $u_0$ in $T$ that is lexicographically optimal in $e$. Let us show how to transform $\mathcal{B}$ in $\mathcal{B}'$ such that $\text{shad}(\mathcal{B}', e) = \text{shad}(\mathcal{B}_e, e)$, and such that the calls of $\mathcal{B}'$ are performed according to $\mathcal{B}_e$ inside $T_v$, and according to $\mathcal{B}$ outside $T_v$. The aim of the following is to show how these two protocols can be linked by calls through $e$. In both $\mathcal{B}$ and $\mathcal{B}_e$, there are at most $r$ calls passing through $e$, where $r$ is the minimal broadcast-time from $u_0$ in $T$. Let $\text{shad}(\mathcal{B}_e, e) = (x_1, \ldots, x_r) \in \{0,1\}^r$, and let $\text{shad}(\mathcal{B}, e) = (y_1, \ldots, y_r) \in \{0,1\}^r$. Let $k$ be the smallest index such that $x_k < y_k$.

- At every round $i < k$, if $x_i = 1$, then $y_i = 1$. That is if there is a call in $\mathcal{B}_e$, say from $a$ to $b$, traversing $e$, then there is a call in $\mathcal{B}$, say from $a'$ to $b'$, traversing $e$. This latter call is transformed into a call from $a'$ to $b$ in $\mathcal{B}'$.
- At round $k$, there is no call traversing $e$ in $B_e$, but there is a call traversing $e$ in $\mathcal{B}$ from some node $w$ to some node in $T_v$. This call is transformed into a call from $w$ to $u$ in $\mathcal{B}'$.
- At every round $i > k$, the call traversing $e$ in $\mathcal{B}$ (if any) is removed. On the other hand, if there is a call traversing $e$ in $\mathcal{B}_e$ from some node $w$ to some node $w'$, then $u$ calls $w'$ in $\mathcal{B}'$. Note that $u$ may be involved in other calls in $\mathcal{B}$ but the all-port constraint allows to add another call from $u$.

The broadcast protocol $\mathcal{B}'$ satisfies $\text{shad}(\mathcal{B}', e) = \text{shad}(\mathcal{B}_e, e)$. For every arc $e'$ of $T \setminus T_v$, $\text{shad}(\mathcal{B}', e') \leq \text{shad}(\mathcal{B}, e')$ because some calls of have been removed in $T \setminus T_v$ and no call has been added. Therefore, the set of bad arcs may have increased, but the additional bad arcs are included in $T_v$, that is are below the considered bad arc. Therefore, one can repeat for $\mathcal{B}'$ what we did for $\mathcal{B}$ while preserving the optimal shadow of $e$. After at most $|E|$ operations of that type, there is no more bad arcs.

Now, let us assume that $T$ is undirected. Again, let $\mathcal{B}$ be an optimal broadcast protocol from $u_0$ in $T = (V, E)$, and assume that $\mathcal{B}$ is not lexicographically optimal on every arc. Let us consider any branch $B$ of $T$ from $u_0$ to some leaf, and let $e = (u, v) \in E$ be the arc of $B$ that is closest to the root and such that $\text{shad}(\mathcal{B}, e) > \text{shad}(\mathcal{B}_e, e)$ where $\mathcal{B}_e$ is a broadcast protocol that is lexicographically optimal in $e$. We transform $\mathcal{B}$ into $\mathcal{B}'$ such that $\text{shad}(\mathcal{B}', e) = \text{shad}(\mathcal{B}_e, e)$ and such that the calls of $\mathcal{B}'$ are performed according to $\mathcal{B}_e$ inside $T_v$, and according to $\mathcal{B}$ outside $T_v$.

Let $\text{shad}(\mathcal{B}_e, e) = (x_1, \ldots, x_r) \in \{-1, 0, 1\}^r$, and let $\text{shad}(\mathcal{B}, e) = (y_1, \ldots, y_r) \in \{-1, 0, 1\}^r$. Let $k$ be the smallest index such that $x_k < y_k$.

- At every round $i < k$, if $x_i = 1$, then $y_i = 1$. That is if there is a call traversing $e$ downward in $\mathcal{B}_e$, say from $a$ to $b$, then there is a call traversing $e$ downward in $\mathcal{B}$, say from $a'$ to $b'$. This latter call is transformed into a call from $a'$ to $b$ in $\mathcal{B}'$. Similarly, if $x_i = -1$, then $y_i = -1$. That is if there is a call traversing $e$ upward in $\mathcal{B}_e$, say from $a$ to $b$, then there is a call traversing $e$ upward in $\mathcal{B}$, say from $a'$ to $b'$. This latter call is transformed into a call from $a$ to $b'$ in $\mathcal{B}'$.

- At round $k$, let us consider two cases.

  - $x_k = 0$ and $y_k = 1$. Then there is a call traversing $e$ downward in $\mathcal{B}$ from some node $w$ to some node $w'$. This call is transformed into a call from $w$ to $u$ in $\mathcal{B}'$.
  - $x_k = -1$, and $y_k = 1$ or $0$. Then there is a call traversing $e$ upward in $\mathcal{B}_e$ from some node $w$ to some node $w'$. This call is transformed into a call from $w$ to $u$ in $\mathcal{B}'$. If $y_k = 1$, the corresponding call of $\mathcal{B}$ is just removed.

- At every round $i > k$, if there is a call traversing $e$ downward in $\mathcal{B}_e$, say from $w$ to $w'$, then $u$ calls $w'$ in $\mathcal{B}'$. If there is a call traversing $e$ upward in $\mathcal{B}_e$, say from $w$ to $w'$, then $w$ calls $u$ in $\mathcal{B}'$.

For the same reason as in the directed case, there is no more bad arcs after at most $|E|$ operations of that type.