Optimized Broadcasting and Multicasting Protocols in Cut-through Routed Networks*

Johanne Cohen[†‡] Pierre Fraigniaud[‡] Jean-Claude Konig[§] André Raspaud[¶]

Abstract

This paper addresses the one-to-all broadcasting problem, and the one-to-many broadcasting problem, usually simply called *broadcasting* and *multicasting*, respectively. Broadcasting is the information dissemination problem in which a node of a network sends the same piece of information to all the other nodes. Multicasting is a partial broadcasting in the sense that only a subset of nodes forms the destination set. Both operations have many applications in parallel and distributed computing. In this paper, we study these problems in both *line* model, and *cut-through* model. The former assumes long distance calls between non-neighboring processors. The latter strengthens the line model by taking into account the use of a *routing function*. Long distance calls are possible in circuit-switched and wormhole routed networks, and also in many networks supporting optical facilities.

In the line model, it is well-known that one can compute in polynomial time a $\lceil \log_2 n \rceil$-round broadcast or multicast protocol for any arbitrary network. Unfortunately, such a protocol is often inefficient from a practical point of view because it does not use the resources of the network in a balanced way. In this paper, we present a new algorithm to compute broadcast or multicast protocols. This algorithm applies under both line and cut-through models. Moreover, it returns protocols that efficiently use the bandwidth of the network. From a complexity point of view, we also show that most of the optimization problems relative to the maximization of the efficiency of broadcast or multicast protocols in term of switching time or vertex load are NP-complete. We have derived polynomial efficient solutions for tree-networks however.

---

*All authors are supported by the research program PRS of the CNRS.

†Additional support by the DRET of the DGA.

‡On leave from the Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, France. Current address is Laboratoire de Recherches en Informatique, Univ. Paris-Sud, 91405 Orsay cedex, France.

§Département d'informatique, Université d'Evry, 91025 Evry Cedex, France

¶Laboratoire Bordelais de Recherche en Informatique, Université de Bordeaux 1, 33405 Talence Cedex, France

1 Introduction

Given any point-to-point interconnection network [24], *broadcasting* is the information dissemination problem in which a source node sends the same piece of information to all the other nodes of the network. Such a communication scheme typically appears in many parallel or distributed applications [17, 18, 29, 30]. It is one of the kernels of the Collective Communication Routines of the Message Passing Interface (MPI) library [8, 11]. Broadcasting is actually a particular case of *multicasting* in which a source node sends a unique message to an arbitrary subset of nodes. For instance, a broadcast at the application level in a multi-user parallel machine is actually a multicast at the system level. Multicasting has many applications for the control of parallel systems as it is involved in barrier synchronization or cache coherence; it is also a basic tool for the implementation of parallel data-bases [33].

In most of the modern distributed memory parallel computers, the store-and-forward routing mode has been replaced by various types of *cut-through* routing modes, including circuit-switching, wormhole routing [27] and single-hop Wavelength Division Multiplexing (WDM) [1, 26]. In the circuit-switching mode, when a node x sends a message to a non-neighboring node y, a path that directly connects these two nodes is created between them. The message from x is then transmitted towards y along the path. On each node, a router is in charge of transmitting messages, but intermediate processors do not necessarily receive the message that passes through them. Actually, a router sends a message to the local memory only when the local processor is one of the destinations specified in the header of the message. Wormhole routing differs from circuit-switching routing in the way messages are transmitted along the path from the source to the destination. In wormhole routing, a message is decomposed in small units called flits. The first flit is used to determine the route followed by the message at each intermediate node, and the remaining flits follow in a pipeline fashion (the last flit releases the intermediate connections). Wormhole routing does not require a whole path to be reserved between the source and the destination, it makes use of a number of links proportional to the length of the message. In single-hop WDM routing, upon reception of a communication-request (source,destination), the system is in charge of allocating a wavelength λ to a path P from the source to the destination so that no other path sharing a link with P has the same wavelength. When the communication-request has been satisfied, the message is encoded using λ, and it is routed along P. Like any cut-through routing, circuit-switching, wormhole and WDM routing are not very sensitive to the path length.

In this paper, we are interested in the communication complexity of broadcasting and multicasting in cut-through routed networks. For this purpose, we will make use of the so-called *line model* [4] which supposes that **(1)** a call involves exactly two nodes (these two nodes might be at distance greater than one), **(2)** a node can take part in at most one call at a time, and **(3)** any two paths corresponding to two simultaneous calls must be edge-disjoint[1]. As opposed to the so-called telephone model [12] which allows neighbor-to-neighbor communications only (and indeed has nothing to do with telephone networks), the line model allows long distance calls between non-neighboring nodes in order to reflect the cut-through ability[2]. However, the line model suffers from a major drawback. Indeed, in most of the systems, the paths followed by messages are determined by the use of a *routing function*. To take into account this fact, we will consider the following simple additional hypothesis to the line model : **(4)** paths followed by messages are constructed by application of a routing function. To simplify the analysis, such a routing function is modeled here as a function $R : V \times V \mapsto E$ where V denotes the set of vertices, and E denotes the set of edges of an undirected graph $G = (V, E)$. That is, when a message of destination y is currently at node x, it is routed through the edge $R(x, y)$. (Such an edge is always supposed to be incident with node x.) R is *adaptive* if the routing function returns several possible solutions to route a given message, that is $R : V \times V \mapsto \mathcal{P}(E)$. In this case, a selecting function is in charge of choosing a free channel among the selected links. A routing function is said to be *minimal* if, for any source-destination pair, any path generated by the routing function is of length the distance in the network between these two nodes. In this paper, the line model plus hypothesis 4 is called the *cut-through model* (the routing function can be adaptive or not).

Most of the known results about broadcasting under these hypotheses deal with particular network architectures as trees [4, 21, 22], cycles [19], meshes or tori [5, 10]. Many results have been also derived when hypothesis 2 is replaced by the *all-ports* hypothesis, that is, when a node can simultaneously communicate with as many nodes as its number of ports (see for instance the

[1] As in [4], but as opposed to [16], a sender or receiver node can be an inner node of another path.

[2] We will assume a two-way mode in this paper, although our results also apply to the one-way mode since broadcast and multicast protocols involve one-way calls as far as the number of rounds is concerned.

references in [13, 28]). Similarly, broadcasting has also been investigated when hypothesis 3 is replaced by a *vertex-disjoint* constraint [20], or by a similar constraint which assumes that an inner node of a path cannot be a sender or receiver [14, 15]. Unfortunately, most of these variations yield NP-complete problems when looking for the minimization of the number of rounds for broadcasting or multicasting, whereas we will see that it is not the case for the line model. Similarly, multicasting has been intensively studied in the literature. Results concern either store-and-forward routing (see for instance [23, 25]) or cut-through routing (see for instance the references in [6, 7]). As for the broadcasting problem, the network is usually fixed (generally a mesh). Moreover, many papers dealing with the multicasting problem make use of the *path-based* hypothesis that will not be considered in this paper. This hypothesis assumes that a message header can contain multiple destination addresses, and the flow control allows the intermediate destinations to get a copy of the message traversing their routers.

In this paper, we will consider the line model applied to *arbitrary* network topologies. Since cut-through routing is not very sensitive to the length of the paths, a primary approach to estimate the complexity of a broadcast or a multicast protocol in the line model is to count its number of rounds (a round been defined by the set of calls performed at the same time). Note that it does not mean that the network must be synchronous, it is just an estimation of the time required by the protocol *assuming* that the network is synchronous. Hypotheses 1 and 2 imply that the number of informed nodes can at most double at each round, and therefore $\lceil \log_2 n \rceil$ is a lower bound on the number of rounds necessary to perform broadcast from any node of any network of n nodes. Farley [4] showed that this bound is tight, that is, for any network G of n nodes, the number of rounds required to perform broadcast from any node of G is actually equal to $\lceil \log_2 n \rceil$. As we will see, Farley's theorem can be extended to the multicast problem: for any network $G = (V, E)$, for any source node $u \in V$, and for any destinations set $D \subseteq V$, $u \in D$, the number of rounds necessary to complete multicast from u in D is exactly $\lceil \log_2 |D| \rceil$.

The discussion is not closed however. This paper addresses two fundamental questions that arise in this field.

Question 1. Is it possible to adapt Farley's theorem in the cut-through model?

Question 2. What is the limit of the "furtiveness" of a broadcast or multicast protocol? (This will be quantified using several different measures.)

The former question is quite natural since, in the line model, paths are constructed somewhat off-line whereas the cut-through model constructs paths on-line by application of the routing function. The latter question is even more natural. Indeed, it is required that the traffic generated by a broadcast or a multicast protocol does not interfere significantly with other possible traffic. For instance, any broadcast initiated by a user-process of a multi-user parallel machine must not slow down the communications of other users. More generally, multicasts are often initiated by system processes which must not reduce the ability of user processes to exchange messages at their maximum rate. In other words, one requires that the number of resources used by a multicast or a broadcast be minimal, or, at least, be small. Unfortunately, Farley's protocol is not appropriate for that purpose because it is based on transmitting messages along edges of a spanning tree of the network. This induces lot of contentions, and high latency. Our main goal in this paper is to figure out whether it is possible to derive $\lceil \log_2 n \rceil$-round broadcast protocols that achieve a better use of the resources of the network for both line model and cut-through models.

Among the several possible parameters which measure the "furtiveness", and the efficiency of a multicast or a broadcast protocol, we will be interested in minimizing the total number of communication links that are used at each round of the protocol, or during the whole protocol. The less number of links is used, the most "furtive" is the protocol. Defining a *transmitter* as a router that is explicitly used to forward a message during a broadcast of a multicast protocol, we will be interested in minimizing the number of such transmitters. It allows to decrease the number of nodes which will be disturbed by the multicast protocols. Again, we will consider either a given round, or the whole protocol. Alternatively, we will be interested in minimizing the maximum number of communication paths that a given transmitter handles simultaneously during a multicast or a broadcast protocol. Indeed, we must not overload intermediate routers, so that they keep their ability to route other messages at their maximum speed. Of course, there is a tradeoff between the number of transmitters and the load of these transmitters. And last but not least, we will be interested in minimizing the maximum length of the paths used during a broadcast or a multicast protocol. This parameter will allow us to estimate the possible degradation of the time complexity of the protocol when the switching time of the routers cannot be neglected.

About Question 1, we will show that the answer is "yes" for *minimal* (possibly adaptive) routing functions. About Question 2, we will give polynomial algorithms minimizing the values of the parameters listed before, or we will alternatively show that the corresponding decision problems are NP-complete.

More precisely, in Section 3, we will derive a new polynomial algorithm which returns, for any networks G, a time-optimal multicast protocol (and, as a particular case, a time-optimal broadcast protocol) in G in the line model. This protocol minimizes the total number of edges used at each round. A major point is that this new protocol applies to the cut-through model also, under the simple condition that the routing function generates shortest paths only. This is an improvement compared to all time-optimal protocols previously described in the literature. Indeed, in order to be applied to the cut-through model, all these protocols require a routing function using the edges of a spanning tree (such a routing function cannot be used in practice since it would create a lot of contentions, in particular at the root of the tree, and the lengths of the routes would be much too large compared to the shortest paths between the sources and the destinations). In Section 3, we will also show that the decision problem corresponding to the minimization of the sum, taken over all rounds, of the total number of edges used at every round is NP-complete. However, we will show that our protocol is optimal up to a logarithmic multiplicative factor. In section 4, we will derive specific lower and upper bounds for tree-networks. These bounds approximate in a much better way the optimal total number of used edges.

Sections 5 and 6 are dedicated to the NP-completeness of minimizing the number of transmitters, or minimizing the load of the transmitters, or minimizing the total switching-time (i.e., the maximum length of the routing paths used to broadcast or to multicast). These results imply that routers must be sophisticated enough to support simultaneous routing of many paths without degradation of their switching-time. Similarly, even if it is not a major issue to minimize the length of the routes in cut-through networks, these results show that broadcasting or multicasting messages of small size is difficult to optimize when the switching-time cannot be neglected. Section 7 contains some concluding remarks.

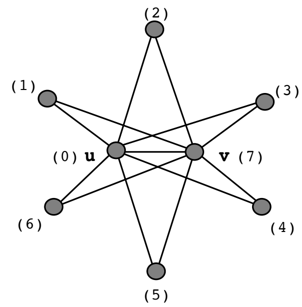

Figure 1: A double star of 8 vertices.

2 Line model versus cut-through model

The aim of this section is to point out the difference between the line model and the cut-through model. Let us consider the network on Figure 1. It is a *double-star* $S_n^2 = (V, E)$ with two centers u and v, and $n-2$ rays. Let us show how to broadcast in $\lceil \log_2 n \rceil$ rounds from any source of S_n^2 in the line model (Farley's theorem insures that such a broadcast protocol exists). Nodes at the extremities of the rays are labeled from 1 to $n-2$, node u is labeled 0, and node v is labeled $n-1$. Note that nodes u and v play the same role. Thus, there are only two cases to be considered: either the source of the broadcast is a center (say node 0), or it is one of the extremities of the $n-2$ rays (say node 1). In both cases, one can always proceed so that both nodes 0 and 1 are aware of the broadcasted message after one round. Now, if one considers the subgraph of the double star obtained by removing all the edges between v and the other nodes but u, we obtain a usual star with center u. On this star, the next $\lceil \log_2 n \rceil - 1$ rounds are given by: at round i, $i > 1$, if the node labeled k is aware of the message, then it informs the node labeled $k + 2^{i-1}$. There is no edge contention, and, after round $\lceil \log_2 n \rceil$, all nodes are aware of the message.

Let us now consider the non-adaptive routing function R that routes the messages on the double star as follows:

$$\begin{cases} \forall w \in V, w \neq u, w \neq v, \forall x \in V, R(w,x) = (w,u); \\ \forall x \in V, R(u,x) = (u,v); \\ \forall x \in V, R(v,x) = (v,x). \end{cases}$$

Assume $n = 8$. It is not possible to broadcast in 3 rounds from any source in the double star of 8 vertices under the cut-through model with this routing function. Indeed, to broadcast in 3 rounds, the last round would consists of 4 calls performed at the same time. Therefore, at least two of these calls would use the arc from u to v, which is impossible. (Recall that both line model and cut-through model require edge-disjoint paths.) This example shows that the cut-through model is more restrictive than the line model. In particular, it is not true that, for any network G, and for any routing function R on G, there exists a broadcast protocol from any node of G that performs in $\lceil \log_2 n \rceil$ rounds. However, one can state the following result:

Property 1 (cut-through version of Farley's theorem).
In the cut-through model, for any network G of n nodes, there exists a non-adaptive, and non-minimal routing function R such that the broadcasting time from any node of G is $\lceil \log_2 n \rceil$.

Proof. There exist many proofs of Farley's theorem and similar results (see [4, 16, 21, 22]). All of them use an arbitrary spanning tree of the considered network, and all the calls are performed along the edges of this spanning tree. Any spanning tree induces a routing function R since there is a unique path between any two nodes in a tree. Therefore, all paths used by the broadcast protocols derived in [4, 9, 16, 21, 22] can be generated by a routing function.

□

The proof of Property 1 is based on a routing function which follows the edges of a spanning tree of the network. Such a routing function induces a lot of contentions and hot-spots when it is used for other communication problems. Indeed, the traffic is not balanced, and the root of the tree is clearly overloaded. Moreover, though the use of shortest paths is not necessarily important in networks using circuit-switching or wormhole routing, such a routing function may route messages exchanged between neighbors along a path of length twice the diameter of the network! In the next section, we will show that it is possible to do much better.

3 A new multicast protocol for minimal routing functions

In this section, we will first focus on the total number of links that are used by a broadcasting or a multicasting protocol at each round, or during the whole protocol. By total number of links used during the whole protocol, we mean the sum, over all the rounds, of the number of links used at each round (an edge can therefore contribute more than once). As we saw, this parameter can be considered as a measure of the "furtiveness" of the protocol. At each round of a broadcasting protocol, nodes aware of the message are "matched" with other nodes that have not received the message yet. Let us formalize this fact:

Definition 1 *Let U be a subset of vertices of a graph $G = (V, E)$. A* pseudo-matching *in U is a set $\mathcal{P}$ of $\lfloor \frac{|U|}{2} \rfloor$ pairwise edge-disjoint paths in G such that every vertex of U (but one if $|U|$ is odd) is an extremity of a path in $\mathcal{P}$.*

The following result shows that a pseudo-matching in U exists for any choice of $U \subseteq V$. We denote by $d(x, y)$ the distance between two nodes x and y of a graph G.

Lemma 1 *Let U be a subset of vertices of a graph $G = (V, E)$. One can group the vertices of U in pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ where $k = \lfloor \frac{|U|}{2} \rfloor$, and, for all $i, j \in \{1, \ldots, k\}$, $x_i \in U$, $y_j \in U$, $x_i \neq x_j$, $y_i \neq y_j$, and $x_i \neq y_j$, such that any shortest path between x_i and y_i is edge-disjoint with any shortest path between x_j and y_j, $i \neq j$, and such that $\sum_{i=1}^{k} d(x_i, y_i)$ is minimum among all the possible choices of the pairs (x_i, y_i), $i = 1, \ldots, k$.*

Proof. Let $m = |U|$, and let us consider the complete graph K_m of m vertices where each vertex of K_m is identified with a vertex of U. We add weights on the edges of K_m: the edge $\{x, y\}$ has weight $d(x, y)$, the distance between x and y in G. Consider a perfect matching of minimum weight in K_m (or a matching that leaves just one node unmatched if m is odd). This matching induces a set $\mathcal{P}$ of $\lfloor \frac{|U|}{2} \rfloor$ shortest paths in G such that every vertex of U (but one if $|U|$ is odd) is an extremity of a path in $\mathcal{P}$. (In case of multiple shortest paths between two matched vertices, choose one arbitrarily.)

We claim that the paths of $\mathcal{P}$ are pairwise edge-disjoint. Indeed, assume that two paths $P_{\{x,y\}}$ and $P_{\{x',y'\}}$ of $\mathcal{P}$ are not edge-disjoint. It means that there exist two vertices z and z' such that $P_{\{x,y\}} = P_{\{x,z\}} \cup P_{\{z,z'\}} \cup P_{\{z',y\}}$, and $P_{\{x',y'\}} = P_{\{x',z\}} \cup P'_{\{z,z'\}} \cup P_{\{z',y'\}}$ or $P_{\{x',y'\}} = P_{\{x',z'\}} \cup P'_{\{z',z\}} \cup P_{\{z,y'\}}$ where $P_{\{z,z'\}}$ has at least one edge in common with $P'_{\{z,z'\}}$. Assume without loss of generality that $P_{\{x',y'\}} = P_{\{x',z\}} \cup P'_{\{z,z'\}} \cup P_{\{z',y'\}}$. It implies that the two matchings $\{x, y\}$ and $\{x', y'\}$ can be replaced by two other matchings $\{x, x'\}$ and $\{y, y'\}$. The former matching has a weight of $d(x, z) + d(z', y) + d(x', z) + d(z', y') + 2d(z, z')$ whereas the latter has a weight of $d(x, x') + d(y, y')$ which is less than or equal to $d(x, z) + d(z', y) + d(x', z) + d(z', y')$. Since

$d(z, z') \neq 0$, we obtain a contradiction with the fact that the original matching is of minimum weight, and therefore the paths of $\mathcal{P}$ are pairwise edge-disjoint.

□

Let us define the *weight* of a pseudo-matching $\mathcal{P}$ in U as the sum of the lengths of all the paths in $\mathcal{P}$. We are interested in minimizing the weights of the several pseudo-matchings generated at any round of a multicasting or a broadcasting protocol. A pseudo-matching of low weight requires low use of the bandwidth to perform exchanges between the extremities of its paths. The following theorem shows that this minimization is possible in polynomial time. It improves Farley's theorem.

Theorem 1 *For any network $G = (V, E)$ of n nodes, and for any node u of G, one can compute in polynomial time a multicast protocol from u to any set D in G, $u \in D$, which performs in $\lceil \log_2 |D| \rceil$ rounds in the line model, and such that:*

(i) *all the calls are performed along shortest paths;*

(ii) *at any of the $\lceil \log_2 |D| \rceil$ rounds, the weight of the corresponding pseudo-matching is minimum.*

Proof. The $\lceil \log_2 |D| \rceil$ pseudo-matchings are constructed backwards. Start with $U_1 = D$, and, by Lemma 1, compute a pseudo-matching $\mathcal{P}_1$ in U_1 of minimum weight, and containing shortest paths only. Then choose one of the two extremities of each path in $\mathcal{P}_1$. This choice can be random or arbitrary, that is, for any path of $\mathcal{P}_1$, one extremity is selected arbitrarily. There is one exception for the path containing the source u because the source u must be selected. Keep also the unique isolated unmatched vertex in U_1 if exists. All these nodes form a set U_2. Then compute a pseudo-matching $\mathcal{P}_2$ in U_2 of minimum weight and containing shortest paths only. Again, this is possible by Lemma 1. Then, we extract a set U_3 from U_2 as U_2 was extracted from U_1, and we repeat the process until a set U_i is reduced to u. Clearly i satisfies $i = \lceil \log_2 |D| \rceil$. This protocol can be computed in polynomial time because one can compute a perfect matching of minimum weight in the complete graph in polynomial time [3].

□

This result also holds in the cut-through model because the way the shortest paths are selected in the proof of Lemma 1 does not matter. In particular, they can be constructed by using any minimal (and possibly adaptive) routing function. Therefore, we get the following result which strongly improves Property 1:

Theorem 2 *For any network $G = (V, E)$ of n nodes, for any subset $D \subseteq V$, for any node u inD, and for any minimal (and possibly adaptive) routing function on G, one can compute in polynomial time a multicast protocol from u to D which performs in $\lceil \log_2 |D| \rceil$ rounds in the cut-through model, and such that, at any of the $\lceil \log_2 |D| \rceil$ rounds, the weight of the corresponding pseudo-matching is minimum.*

Theorem 2 says that, whatever are the network and the (possibly adaptive) routing function on this network, if this routing function routes messages along shortest paths, it is possible to broadcast and multicast optimally in terms of rounds. Most of the routing functions used in the usual topologies (meshes, multi-dimensional tori,...) routes on shortest paths (XY-routing, e-cube routing,...). If wormhole routing is used, it is interesting to notice that there is no contradiction between the search for a deadlock free routing function, and the search for a routing function that insures fast broadcasting and multicasting. Indeed, most of the classical deadlock free routing functions generate shortest paths only.

Theorems 1 and 2 both say that one can easily minimize the interference of each round of any multicast protocol (the number of involved communication channels can be minimized at each round). Unfortunately, they do not say that the sum of the lengths of all the paths used during the whole multicast is minimum. As a counter example, consider the path $P_4 = (x_0, x_1, x_2, x_3)$, and assume that x_0 is the source. In the protocol given in the proof of Theorem 1, $\mathcal{P}_1 = \{(x_0, x_1), (x_2, x_3)\}$. So, if x_0 and x_3 are then selected (recall that the selection is arbitrary but for the source), then $\mathcal{P}_1 = \{(x_0, x_1, x_2, x_3)\}$. At each round, the number of involved communication channels is minimum (given as inputs the sets $U_1 = \{x_0, x_1, x_2, x_3\}$, and $U_2 = \{x_0, x_3\}$). However, the sum of all the lengths of the paths is $1 + 1 + 3 = 5$ whereas the minimum is 4.

The following result shows that minimizing globally the sum of the path lengths is NP-complete.

Theorem 3 *The following problem is NP-complete:*

Minimum Total Path Length (MTPL):

Instance: A graph $G = (V, E)$, a vertex u of G, a subset $D \subseteq V$, $u \in D$, and an integer k.

Question: Does there exist a multicast protocol from u to D in G performing in $\lceil \log_2 |D| \rceil$ rounds in the line model, and such that the sum of all the lengths of all the communication paths generated by this protocol is at most k?

The proof of Theorem 3 is based on the following lemma (recall that the telephone model allows neighbor-to-neighbor communications only):

Lemma 2 *The following problem is NP-complete:*

Log Broadcast (LogB):

Instance: A graph $G = (V, E)$ *of* n *vertices, and a vertex* u *of* G.

Question: Does there exist a broadcast protocol from u *in* G *performing in at most* $\lceil \log_2 n \rceil$ *rounds in the telephone model?*

Proof. It is well known that the following problem has been shown to be NP-complete by Johnson (see [31] for his proof):

Broadcast:

Instance: A graph $G = (V, E)$*, a vertex* u *of* G*, and an integer* k.

Question: Does there exist a broadcast protocol from u *in* G *performing in at most* k *rounds in the telephone model?*

Lemma 2 can be proved using nearly the same technique as the one used in [31] to show that the problem LogB is NP-complete.

□

Proof of Theorem 3. MTPL is clearly in NP. MTPL is NP-complete by transformation from Problem LogB. Let G, u be an instance of the LogB problem. We transform this instance in an instance G, D, u, k of the MTPL problem by setting $D = V$ and $k = n - 1$. Clearly, if there exists a broadcast protocol from u in G performing in at most $\lceil \log_2 n \rceil$ rounds in the telephone model, then there exists a broadcast protocol from u in G in $\lceil \log_2 n \rceil$ rounds in the line model and such that the sum of all the lengths of the communication paths is at most $n - 1$. Reciprocally, if there exists a broadcast protocol from u in G performing in $\lceil \log_2 n \rceil$ rounds in the line model, and such that the sum of all the lengths of the communication paths is at most $n - 1$, then all calls are performed between neighboring vertices, and therefore there exists a broadcast protocol from u in G performing in at most $\lceil \log_2 n \rceil$ rounds in the telephone model. □

Theorem 3 implies that it is difficult to minimize the total load of the edges during a multicast or a broadcast protocol. However, one can approximate this value up to a logarithmic factor.

Notation. For any network $G = (V, E)$, any node $u \in V$, and any set $D \subseteq V$, $u \in D$, let us denote by $S(G, D, u)$ the minimum, taken over all the multicast protocols from u to D in G performing in $\lceil \log_2 |D| \rceil$ rounds, of the sum of all the lengths of all the paths generated by the protocol.

Theorem 3 shows that minimizing $S(G, D, u)$ is an NP-complete problem. Nevertheless, we have:

Theorem 4 *Let $G = (V, E)$ be any graph, let $D \subseteq V$, and let $u \in D$. The total sum of the lengths of all the paths generated by the multicast protocol of Theorem 1 is at most $\lceil \log_2 |D| \rceil S(G, D, u)$.*

Proof. Since every node of D must receive the message, $S(G, D, u)$ is at least the number of edges n_s of a Steiner tree T [32] spanning D. On the other hand, for every i, $1 \leq i \leq \lceil \log_2 |D| \rceil$, the number of edges of the pseudo-matching of U_i constructed in the algorithm given in the proof of Theorem 1 contains at most the number of edges of T. Indeed, one can find a pseudo-matching of U_i by using only edges of T, and thus the pseudo-matching of U_i in G uses less edges. Therefore, for every i, $1 \leq i \leq \lceil \log_2 |D| \rceil$, the number of edges of the pseudo-matching of U_i is at most the number of edges n_s of a Steiner tree spanning D. Thus the protocol of Theorem 1 generates paths of total length at most $\lceil \log_2 |D| \rceil n_s \leq \lceil \log_2 |D| \rceil S(G, V, u)$.

□

We conclude the section by a property of pseudo-matchings.

Property 2 *A pseudo-matching of minimum weight is a forest (i.e. a set of disjoint trees).*

Proof. Let $\mathcal{P}$ be a pseudo-matching of minimum weight. If $\mathcal{P}$ is not a forest, then let us show that we can construct another pseudo-matching $\mathcal{P}'$ with fewer edges. Assume that there is a cycle in $\mathcal{P}$, and let $C = \{x_0, x_1, \ldots, x_{k-1}\}$, $k \geq 3$, be this cycle. C is actually composed as a union of sub-paths of p paths of $\mathcal{P}$. We identify particular vertices of C called *doors*. A door is a vertex x_i of C such that $(x_{i-1 \bmod k}, x_i)$ and $(x_i, x_{i+1 \bmod k})$ do not belong to the same path of $\mathcal{P}$.

Clearly the number of doors is strictly larger than 1. If the number of doors is 2, then C is composed of parts of two paths:

$$P_1 = \{y_1, y_2, \ldots, y_r, x_0, x_1, \ldots, x_s, y_{r+1}, \ldots, y_\ell\},$$

and

$$P_2 = \{z_1, z_2, \ldots, z_{r'}, x_s, x_{s+1}, \ldots, x_{k-1}, x_0, z_{r'+1}, \ldots, z_{\ell'}\}.$$

These paths could be easily replaced by

$$P_1' = \{y_1, y_2, \ldots, y_r, x_0, z_{r'+1}, \ldots, z_{\ell'}\}, \text{ and } P_2' = \{z_1, z_2, \ldots, z_{r'}, x_s, y_{r+1}, \ldots, y_\ell\}.$$

This would yield a pseudo-matching of lower weight, that is a contradiction.

If the number of doors is strictly larger than 2, let P be a path of $\mathcal{P}$ using some edges of C. $P = \{y_1, y_2, \ldots, y_\ell\}$. Let i be the smallest index such that $(y_i, y_{i+1}) \in C$. There exists j,$0 \leq j \leq k-1$, such that $(y_i, y_{i+1}) = (x_j, x_{j+1 \bmod k})$, and x_j is a door. There is another path of $\mathcal{P}$, say $P' = \{z_1, z_2, \ldots, z_m\}$, such that $(x_{j-1 \bmod k}, x_j) \in P'$. Assume w.l.g., that $(x_{j-1 \bmod k}, x_j) = (z_r, z_{r+1})$, $r < m$. We replace the two pairs of matched vertices y_1, y_ℓ, and z_1, z_m by z_1, y_ℓ, and y_1, z_m. The number of edges used by the new pseudo-matching is the same, but the number of doors of C has been decreased by one. We iterate this process until the number of doors is 2, yielding a contradiction.

□

In the next section, we will show how to improve Theorem 4 for trees.

4 Broadcasting and multicasting in tree-networks

We will derive upper and lower bounds on $S(G, D, u)$ for trees that are tighter than the bound of Theorem 4. Many applications of networking make use of trees as the underlying topology for exchanging data or control messages. This is why it is interesting to treat tree-networks as a special case.

Let us start with a few definitions.

Notation. Let α be any non-negative integer. If $2^{k-1} \leq \alpha < 2^k$, then the binary representation of α is denoted by $\alpha_k, \ldots, \alpha_2, \alpha_1$ such that $\alpha = \sum_{i=1}^{k} \alpha_i 2^{i-1}$. We denote by $g(\alpha)$ the number of 1's in the binary representation of α, that is $g(\alpha) = \sum_{i=1}^{k} \alpha_i$. We also define the function f by induction: $f(0) = 0$, $f(1) = 1$, and,

$$\text{for } \alpha > 1,\ f(\alpha) = \min\left(f(\lceil \frac{\alpha}{2} \rceil), f(\lfloor \frac{\alpha}{2} \rfloor)\right) + (\alpha \bmod 2).$$

In terms of binary expression, the reader can check that, for any non-empty binary string M, $f(M0) = f(M)$, $f(M01) = f(M) + 1$, and, for any string $N = 11\ldots11$ of length at least 2, $f(N) = 2$, and $f(M0N) = f(M1) + 1$. For instance,

$$f(110011101011) = f(1100111011) + 1 = f(11001111) + 2 = f(1101) + 3 = f(11) + 4 = 6.$$

Let $T = (V, E)$ be a tree with n vertices. Let u be the source of a multicast of destination set D, $u \in D$. We consider T as rooted in u. Any set U, $u \in U$, induces a weight function w_U on the edges of T as follows. For any edge $e = \{x, y\} \in E$, the removing of e from T decomposes T is two trees T_x and T_y. We denote by T_e the tree which does not contain u. We define $w_U(e) = |V(T_e) \cap U|$.

Consider what happen to an edge e of weight $w_D(e)$. Let $K_e = V(T_e) \cap U$. Intuitively, if $|K_e|$ is even, then e will not be used in a pseudo-matching of D. However, if $|K_e|$ is odd, then e will be used, and, depending on the choice of one of the two extremities of the path passing through e (see the proof of Theorem 1), we will be let with $\lceil \frac{|K_e|}{2} \rceil$ or $\lfloor \frac{|K_e|}{2} \rfloor$ destinations on the same side of u relatively to e. Thus, intuitively, $\sum_{e \in E} f(w_D(e))$ is a lower bound for $S(T, D, u)$.

To get the intuition of an upper bound, let $S_i = \sum_{e \in E} (w_D(e))_i$, $1 \leq i \leq \lceil \log_2 |D| \rceil$, that is the sum of the ith bits of the binary expressions of all the weights $w_D(e)$, $e \in E$. If the least significant bit of $w_D(e)$ is 1, then e will be used in the pseudo-matching of D. On the contrary, if this bit is 0, then the edge will not be used. However, this property does not hold at the second round, that is it is not necessarily true that only edges with the second least significant bit equal to 1, will be used. Indeed, this property is strongly related to the choice of one of the two extremities of the path passing through e (see the proof of Theorem 1). We will show further that there is a way of performing the choice so that at most $S_{i+1} + \frac{T_i}{2}$ edges are used at round $i+1$ where T_i denotes the number of edges used at round i. Since $\sum_i T_i \leq 2 \sum_i S_i$, and $\sum_{e \in E} g(w_D(e)) = \sum_{i=1}^{\lceil \log_2 n \rceil} S_i$, we will get an upper bound of $2 \sum_{e \in E} g(w_D(e))$ for $S(T, D, u)$.

More formally, we have the following result:

Theorem 5 *Let $T = (V, E)$ be a tree, let $D \subseteq V$ be any set of vertices, and let $u \in D$. We have $\sum_{e \in E} f(w_D(e)) \leq S(T, D, u) \leq 2 \sum_{e \in E} g(w_D(e))$.*

Proof. We construct the multicast protocol backward, providing an accumulation protocol (reversing the communication scheme for accumulation results in a communication scheme for multicasting). In the accumulation problem, each node of D has a piece of information that must be collected by u. An accumulation protocol will be described by a sequence $U_0, U_1, \ldots, U_{\lceil \log_2 |D| \rceil}$ of subsets of nodes with $U_0 = D$, $U_{\lceil \log_2 |D| \rceil} = u$, and $U_{t+1} \subseteq U_t$. For $0 \leq t \leq \lceil \log_2 |D| \rceil$, U_t actually denotes a set of nodes such that the union of all the pieces of information known by them at round t is exactly the union of all the pieces of information originally known by all the vertices of D. The set U_{t+1} is constructed by using the information of U_t only. U_t is called the *active* set at time t. A vertex belonging to an active set is called an active node.

In the considered tree T, given a set U of active vertices, an edge e is said to be *even* relatively to U if the removal of this edge decomposes the tree into two subtrees, each rooted at one extremity of e, which have both an even number of active vertices of U. The edge e is said to be *odd* relatively to U otherwise.

Upper bound. Let us introduce another notation. Given two vertices x and y of the tree, we denote by $s_t(x, y)$ the number of the penultimate zero-bits in the binary expressions of the weights $w_{U_t}(e)$ of the edges e belonging to the shortest path from x to y in T (the sets U_t are defined in the lower bound part of the proof). In other words, if there are q edges on the shortest path from x to y in T, and if the weights of these q edges are denoted by $w^{(1)}, \ldots, w^{(q)}$, then $s_t(x, y) = q - \sum_{i=1}^{q} w_2^{(i)}$. Finally, for the sake of simplicity, let us first assume that $|D| = 2^k$.

Our accumulation protocol proceeds as in the proof of Theorem 1 by a successive construction of pseudo-matchings of the active sets U_t, $t = 0, \ldots, \lceil \log_2 |D| \rceil$. The only thing that we have to specify carefully is the way to choose which extremity of each path of the pseudo-matching we keep for the next round. More precisely, round t of our protocol can be decomposed in two phases:

1. Construct a pseudo-matching in U_t using only odd edges relatively to U_t (see [16]).

2. For each pair of vertices x and y matched in the pseudo-matching in U_t, let z be the first common ancestor of x and y. If $s_t(z, x) = s_t(z, y)$, then choose the vertex which is the closest to the root u, and put it in the set U_{t+1} (choose arbitrarily any of these two vertices if $d(x, u) = d(y, u)$). If $s_t(z, x) < s_t(z, y)$ then put x in U_{t+1}, and if $s_t(z, x) > s_t(z, y)$ then put y in U_{t+1}.

We refer the reader to [16, 21, 22] to figure out how phase 1 can be performed in polynomial time.

Let us show that this $\lceil \log_2 |D| \rceil$-time multicast protocol satisfies that the sum over its $\lceil \log_2 |D| \rceil$ rounds of the number of edges used at each round is upper bounded by $2 \sum_{e \in E} g(w_D(e))$. For this purpose, let us prove the following:

Claim 1. If there are α odd edges before round t (that is relatively to U_t), then there are at most $\beta + \alpha/2$ odd edges after round t (that is relatively to U_{t+1}), where β is the sum of the penultimate bits of the weights $w_{U_t}(e)$ of all the edges $e \in E$.

Indeed, since $|D|$ is a power of 2, the number of active nodes is divided by 2 after each round. Clearly, for any even edge e at round t, $w_{U_{t+1}}(e) = \frac{w_{U_t}(e)}{2}$, and its parity at round $t+1$ depends on the penultimate bit of $w_{U_t}(e)$: an even edge turns odd if the penultimate bit of its previous weight was a 1. The case of odd edges is more complicated. Any odd edge e relatively to U_t will be used during round t, say for a communication between x and y. Let z be the first common ancestor of x and y, and let $w_{U_t}(e) = 2k+1$. Let us consider the several possibilities for the choice of the node $v \in \{x, y\}$ put in U_{t+1}.

– If $v \in V \setminus V(T_e)$, then $w_{U_{t+1}}(e) = k$, and hence e stays odd if and only if the penultimate bit of its previous weight was a 1. This case is therefore taken into account in the term β of Claim 1.

– If $v \in V(T_e)$, then $w_{U_{t+1}}(e) = k+1$, and hence e stays odd if and only if the penultimate bit of its previous weight was a 0. Since v minimizes the number of zeros at the penultimate position of the weights of the edges between x, y, and their common ancestor z, this case is therefore taken into account in the term $\alpha/2$ of Claim 1.

Therefore, as claimed, if there are α odd edges before the round t, then there are at most $\beta + \alpha/2$ odd edges after round t, where β is the sum of the penultimate bits of the weights $w_{U_t}(e)$ of all the edges $e \in E$. Now, let $(w_D(e))_i$ be the ith bit of the binary representation of $w_D(e)$, and let $S_i = \sum_{e \in E} (w_D(e))_i$, for $1 \le i \le \lceil \log_2 |D| \rceil$. By definition, we have

$$\sum_{e \in E} g(w_D(e)) = \sum_{i=1}^{\lceil \log_2 |D| \rceil} S_i. \qquad (1)$$

At the first round, at most S_1 edges are used since there are only S_1 odd edges. Following Claim 1, at most $S_2 + S_1/2$ edges are used at the second round. More generally, we have:

Claim 2. At any round t, $1 \le t \le \lceil \log_2 |D| \rceil$, at most $\sum_{i=1}^{t} \frac{S_i}{2^{t-i}}$ edges are used in the pseudo-matching.

We prove this claim using Claim 1 by showing that the sum of the penultimate bits of the weights $w_{U_t}(e)$ of all the edges $e \in E$ is equal to S_t. We have seen that if $w_{U_t}(e)$ is even, then $w_{U_{t+1}}(e) = \frac{w_{U_t}(e)}{2}$, and therefore the most significant bits are not modified. If $w_{U_t}(e) = 2k+1$, two cases can be considered: $w_{U_{t+1}}(e) = k$ or $w_{U_{t+1}}(e) = k+1$. In the former case, the most significant bits are not modified. The latter case occurs only when the penultimate bits of $w_{U_t}(e)$ is a 0, and therefore the most significant bits are not modified. This shows that the sum of the penultimate bits of the weights $w_{U_t}(e)$ of all the edges $e \in E$ is equal to S_t.

Now, it is easy to check that

$$\sum_{t=1}^{\lceil \log_2 |D| \rceil} \sum_{i=1}^{t} \frac{S_i}{2^{t-i}} \le 2 \sum_{i=1}^{\lceil \log_2 |D| \rceil} S_i,$$

that is, using Equation 1,

$$\sum_{t=1}^{\lceil \log_2 |D| \rceil} \sum_{i=1}^{t} \frac{S_i}{2^{t-i}} \le 2 \sum_{e \in E} g(w_D(e)).$$

This completes the poof of the upper bound when $|D| = 2^k$.

If $|D|$ is not a power of two, that is $|D| = 2^k - a$ where $0 < a < 2^k$, one can consider multicasting in a tree T' of $n+a$ vertices consisting of T plus a "new" vertices directly connected to the root u, and forming a stable set S. (Recall that a stable set is a set of nodes with no edge between them.) Let $D' = D \cup S$. D' has a power of two number of vertices. Thus, we can apply the previously described algorithm to T' and D', so that any vertex of S communicates either with another vertex in S, or with u. Indeed, any pseudo-matching which does not satisfy that any vertex of S communicates either with another vertex in S, or with u can be transformed into another one which does satisfy this property without increasing the number of edges. Assume for instance that $s \in S$ is matched with $v \notin S \cup \{u\}$. Two cases must be considered: either there exists another vertex $s' \in S$ which is matched with $v' \notin S$, or not. In the former case, we can just replace the matchings $\{s, v\}$ and $\{s', v'\}$ by $\{s, s'\}$ and $\{v, v'\}$. In the latter case, we can replace the matchings $\{s, v\}$ and $\{u, w\}$ by $\{s, u\}$ and $\{v, w\}$ where w denotes the vertex originally matched with u. Let us call $\mathcal{A}'$ this algorithm.

If one does not consider communications involving vertices in S, $\mathcal{A}'$ induces an algorithm $\mathcal{A}$

which accumulates the information of D in the root u of the tree T. Note that u is the only possible idle vertex of any active set during the execution of $\mathcal{A}$ by construction of the algorithm $\mathcal{A}'$. Let us denote by $S_{\mathcal{A}}$ (respectively $S_{\mathcal{A}'}$) the global sum of the lengths of the paths generated by the execution of $\mathcal{A}$ (respectively $\mathcal{A}'$). Since $|D'|$ is a power of 2, we have $S_{\mathcal{A}'} \leq 2\sum_{e\in E(T')} g(w_{D'}(e))$. Actually, we can strengthen a bit this upper bound: $S_{\mathcal{A}'} \leq 2\sum_{e\in E(T)} g(w_{D'}(e)) + \hat{S}(K_{1,a}, K_{1,a}, u)$ where $K_{1,a}$ is the star of $a+1$ vertices rooted in u, and where the function $\hat{S}$ is defined as S except that the minimization is taken over all the multicast protocols in which the only possible unmatched active vertex is the source of the multicast. Now, since, in $\mathcal{A}'$, no vertex in S communicates with a vertex in T different from the root, we have $S_{\mathcal{A}} \leq S_{\mathcal{A}'} - \hat{S}(K_{1,a}, K_{1,a}, u)$. Therefore $S(T, D, u) \leq 2\sum_{e\in E(T)} g(w_{D'}(e))$. Finally, since $g(w_{D'}(e)) = g(w_D(e))$, we get $S(T, D, u) \leq 2\sum_{e\in E(T)} g(w_D(e))$ which concludes the proof of the upper bound.

Lower bound. Assume first that $|D| = 2^k$. We prove, by induction on k, that any edge e is used at least $f(w_D(e))$ times during any multicasting protocol from u to D. This result clearly holds for $k = 0$. Let $k > 0$, and assume that any edge e is used at least $f(w_D(e))$ times during any multicast protocol from u to any set D such that $|D| = 2^i$, $0 \leq i < k$.

At any round of any multicast protocol performing in an optimal number of rounds, all the active nodes take part in the communications, and, as proved in [16, 21, 22], only the odd edges can be used in the pseudo-matching. Let D' be the set of active nodes after the first round of an arbitrary accumulation protocol. $|D'| = 2^{k-1}$.

– If the edge e is even before the first round, then no call uses this edge, and the weight of e is simply divided by 2. By induction hypothesis, the edge e will be used at least $f(w_{D'}(e))$ times during any multicast protocol, that is at least $f(w_D(e))$ times by definition of f since $w_D(e)$ is even.

– If the edge e is odd before the first round, then a call between two vertices uses this edge at the first round of any accumulation protocol because there is an odd number of destinations on each side of e. According to the choice of the active node for the next round, the weight of edge e will be equal to $\lceil w_D(e)/2 \rceil$ or $\lfloor w_D(e)/2 \rfloor$. Therefore the edge e is used at least $1+\min(f(\lceil w_D(e)/2 \rceil), f(\lfloor w_D(e)/2 \rfloor))$, that is at least $f((w_D(e))$ by definition of the function f.

The case where $|D|$ is not a power of 2 can be solved using the same argument as in the proof of the upper bound. This concludes the proof.

□

Remark. Although we think that the two bounds of Theorem 5 differ on the average by a small factor (possibly a constant), we did not succeed to prove this fact. It was confirmed by many experiments. There are trees however for which the two bounds differ by more than a constant. Nevertheless, in many examples like the path, the star, the complete binary tree, and the binomial tree, these two bounds differ by at most a factor of 3. Also, it can be easily shown that the expected value of the ratio $\frac{g(\alpha)}{f(\alpha)}$ is bounded by a constant when α is uniformly randomly chosen.

5 Minimization of the load and of the number of involved routers

In this section, we address the ability of a multicasting protocol not to disturb other applications by traversing nodes which are not directly concerned by the multicasted message. Moreover, we also study the load of the routers during a multicasting or a broadcasting protocol. Indeed, a router is usually able to route any permutation of its input ports to its output ports. However, the output channel allocation process is sequential in general, and thus could be time-consuming when many messages traverse the same router at the same time. Therefore, it could be an issue to minimize the load of the routers. Note that proofs of theorems are rather technical, and can be omitted at a first reading of this section.

5.1 Minimization of the number of transmitters

We show below that minimizing the number of routers involved in a time-optimal multicast protocol is NP-complete. This result hold in both cases: either when we consider the protocol round after round, or globally. We will see that the problem is polynomial for trees however.

Definition 2 *Let $\mathcal{P}$ be a pseudo-matching in a subset of vertices U of a graph G. A* transmitter *is a vertex of G which is traversed by a path of $\mathcal{P}$ (that is an inner node of a path of $\mathcal{P}$).*

Theorem 6 *The following problem is NP-complete:*

Minimum Number Of Transmitters (MNOT):

Instance: A graph $G = (V, E)$, a subset $U \subseteq V$ of vertices, and an integer k.

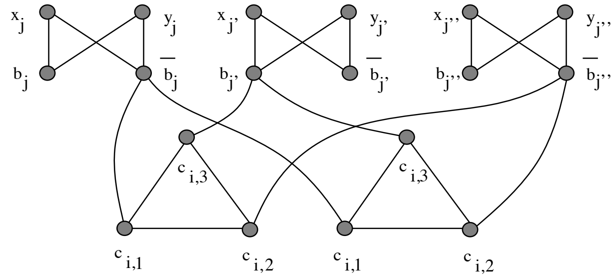

Figure 2: Construction in the proof of Theorem 6. The clause c_i satisfies $c_i[1] = \bar{b}_j$, $c_i[2] = \bar{b}_{j''}$, and $c_i[3] = b_{j'}$. Thus nodes $c_{i,1}$ and $c'_{i,1}$ are both connected to node $\bar{b}_j$, whereas nodes $c_{i,2}$ and $c'_{i,2}$ are both connected to node $b_{j'}$, and nodes $c_{i,3}$ and $c'_{i,3}$ are both connected to node $\bar{b}_{j''}$.

Question: Does there exist a pseudo-matching in U which has at most k transmitters?

Proof. MNOT is clearly in NP. MNOT is NP-complete by transformation from 3SAT. Let $C = \{c_1, c_2, \ldots, c_m\}$ be an instance of 3SAT (where each clause c_i has exactly three literals $c_i[1], c_i[2]$, and $c_i[3]$) on the boolean variables $B = \{b_1, b_2, \ldots, b_n\}$. We construct an instance of MNOT as follows. We associate two copies of K_3 to each clause c_i of C. In each K_3, there is a vertex for each literal. They are denoted by $c_{i,1}, c_{i,2}$, and $c_{i,3}$ in one K_3, and $c'_{i,1}, c'_{i,2}$, and $c'_{i,3}$ in the other K_3. We also associate one copy of $K_{2,2}$ to each boolean variable b_i in B: the first partition of the bipartite graph $K_{2,2}$ is composed of two vertices denoted b_i and $\bar{b}_i$, and the second partition is composed of two vertices denoted x_i and y_i. We connect the literals of each clause to the corresponding boolean vertices as follows (see Figure 2): if $c_i[\ell] = b_j$ (resp. $c_i[\ell] = \bar{b}_j$) then we add the edges $\{c_{i,j}, b_j\}$ and $\{c'_{i,j}, b_j\}$ (resp. $\{c_{i,j}, \bar{b}_j\}$ and $\{c'_{i,j}, \bar{b}_j\}$). We obtain a graph G with $6m + 4n$ vertices. Let U be the subset of vertices composed of the $6m$ vertices of the complete graphs K_3, plus the $2n$ vertices x_i, y_i, $i = 1, \ldots, n$. Set $k = n$ (note that there are at least n transmitters in any pseudo-matching in U because of the vertices x_i and y_i in each $K_{2,2}$). We claim that C is satisfiable if and only if there exists a pseudo-matching in U with at most n transmitters (i.e. with exactly n transmitters).

Assume C is satisfiable. We construct a pseudo-matching in U with exactly n transmitters as follows. For $i = 1, \ldots, n$, connect x_i to y_i with a path traversing the vertex b_i if the solution assigns b_i to $true$, or traversing the vertex $\bar{b}_i$ if the solution assigns b_i to $false$. For $i = 1, \ldots, m$, at least one of the 3 literals in c_i is true, say $c_i[\ell]$. if $c_i[\ell] = b_j$ (resp. $c_i[\ell] = \bar{b}_j$) then connect the two vertices $c_{i,\ell}$ and $c'_{i,\ell}$ by a path traversing the vertex b_j (resp. $\bar{b}_j$). Then, in each copy of K_3 corresponding to the clause c_i, connect $c_{i,\ell'}$ with $c_{i,\ell''}$, and $c'_{i,\ell'}$ with $c'_{i,\ell''}$, $\ell' \neq \ell \neq \ell''$, by a path of length 1. We get a pseudo-matching in U with exactly n transmitters.

Conversely, assume there exists a pseudo-matching in U with exactly n transmitters. Due to the structure of U and G, there is a transmitter in each copy of $K_{2,2}$ because the x's and the y's must be matched. In the $K_{2,2}$ corresponding to the jth variable, $j = 1, \ldots, n$, if the transmitter is node b_j, then we assign $b_j = true$, else we assign $b_j = false$. In each K_3, at least one vertex must be connected by a path of length at least 2 to a vertex of another K_3. This path traverses one of the transmitters. It means that, in each K_3, there is an edge connected to a node corresponding to a literal which is true. Therefore there is at least a true literal in each clause, and C is satisfiable.

□

Remark. From the proof of Theorem 5, one can check that the problem MNOT is solvable in polynomial time for trees. Indeed, in the terminology of this proof, only odd edges take part in the pseudo-matching, and, therefore, one can check in polynomial time whether a vertex is or is not a transmitter according to the number of its incident odd edges.

The following result shows that minimizing the number of involved routers is not only difficult when one considers a single round, but also when one considers the whole protocol.

Property 3 *The following problem is NP-complete:*

MINIMUM TOTAL NUMBER OF TRANSMITTERS (MTNOT):

Instance: A graph G, a vertex u of G, and an integer k.

Question: Does there exist a time-optimal broadcast protocol from u in the line model which involves at most k transmitters during its whole execution?

Proof. The problem MTNOT with $k = 0$ is equivalent to the NP-complete problem LOGB. Indeed, if there exists a $\lceil \log_2 n \rceil$-time broadcast protocol $\mathcal{B}$ such that no vertex is a transmitter, then all the communications are neighbor-to-neighbor, and therefore only use one edge. So the protocol $\mathcal{B}$ satisfies the telephone model constraints. Conversely, if $\mathcal{B}$ is a $\lceil \log_2 n \rceil$-time broadcast protocol in the telephone model, then the corresponding number of transmitters is reduced to zero.

□

5.2 Matching index, and load of the routers

Let $\mathcal{P}$ be a pseudo-matching of a subset of vertices U of a graph G. Let us define the *load* of a given transmitter x to be the number of paths which are traversing x (x is not an end vertex). This parameter is denoted by $\mu(U, \mathcal{P}, x)$. The load of the pseudo-matching $\mathcal{P}$ in U is denoted by $\mu(U, \mathcal{P})$, and is defined by $\mu(U, \mathcal{P}) = \max_x \mu(U, \mathcal{P}, x)$.

Definition 3 *The* matching index *of a subset of vertices U of a graph $G = (V, E)$ is the minimum of $\mu(U, \mathcal{P})$ over all the pseudo-matchings $\mathcal{P}$ in U. It is denoted by $\mu(U)$. The* matching index *of a graph G is defined as $\mu(V)$. When the minimization is restricted to pseudo-matching containing shortest paths only, these parameters are denoted by $\mu_m(U)$, and $\mu_m(V)$ respectively.*

Clearly, if we want to minimize the load of the routers, we have to find pseudo matchings $\mathcal{P}$ for which $\mu(U, \mathcal{P}) = \mu(U)$ at every round of the multicast or the broadcast protocol. Unfortunately, we have the following result:

Theorem 7 *The following problem is NP-complete:*

MATCHING INDEX OF SHORTEST PATHS (MISP):

Instance: A graph $G = (V, E)$, a subset $U \subseteq V$, and an integer k.

Question: $\mu_m(U) \leq k$?

Proof. Clearly, $MISP \in NP$ because one can check in polynomial time whether a pseudo matching $\mathcal{P}$ in U contains shortest path only, and satisfies $\mu(U, \mathcal{P}) \leq k$. To show that $MISP$ is NP-complete, let consider the following problem:

MATCHING INDEX OF SHORTEST PATH(1) (MISP(1)):

Instance: A graph G, and a subset $U \subseteq V$.

Question: $\mu_m(U) \leq 1$?

Let us prove that $MISP(1)$ is NP-complete by transformation from the Vertex Cover problem:

VERTEX COVER (VC):

Instance: A graph $G = (V, E)$, and a positive integer b.

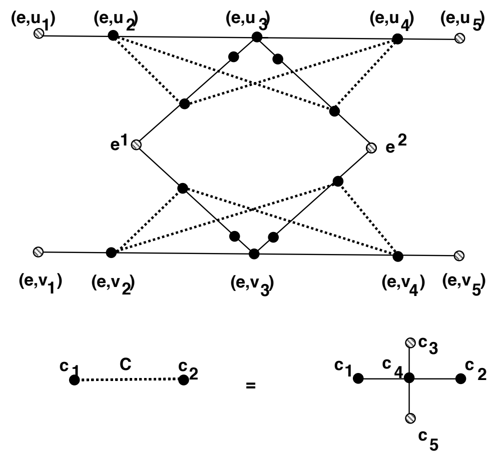

Figure 3: The gadget of an edge $e = (u, v)$, as described in the proof of Theorem 7 (dotted lines represent a cross of five vertices). Vertices in U are marked in grey.

Question: Is there a vertex cover of size b or less, i.e, a subset $S \subseteq V$ such that $|S| \leq b$ and, for each edge $(u, v) \in E$, at least one of u and v belongs to S?

Let an arbitrary instance of VC be given by the graph $G = (V, E)$, and the positive integer b. We construct a graph $G' = (V', E')$, and a subset of vertices U such that, in the graph G', the subset U satisfies $\mu_m(U) \leq 1$ if and only if G has a vertex cover of size b or less.

The graph G' consists of $2b$ paths P_i, $i = 1, \ldots, 2b$, of three vertices (x_i, y_i, z_i) plus other vertices. Connections between these other vertices are functions of the connections of the vertices of G: each edge e of G corresponds to a subgraph $G_e = (V_e, E_e)$ of G'. Such a subgraph is called the gadget of e. The gadget of an edge $e = (u, v) \in E$ is illustrated on Figure 3. It is composed of 44 vertices. Only 20 of them are explicitly drawn in Figure 3. The 24 others are virtually presented in the form of dotted lines: each dotted line $C = c_1 - c_2$ represents a cross of five vertices. The five vertices at the top (resp. bottom) of the gadget of the edge $e = (u, v)$ drawn on Figure 3 are denoted by (e, u_i), $i = 1, \ldots, 5$ (resp. (e, v_i), $i = 1, \ldots, 5$). We also identify two particular vertices denoted by $e^{(1)}$, and $e^{(2)}$ on Figure 3. For each vertex $v \in V$, let $e_1, \ldots, e_d$ be its d incident edges (d is the degree of v). For $i = 1, \ldots, d-1$, the gadget of the edge e_i is linked to the gadget of edge e_{i+1} by an edge $((e_i, v_5), (e_{i+1}, v_1))$ (see Figure 4). All these connections form a path

$$Q_v = \{(e_1, v_1), (e_1, v_2) \ldots, (e_1, v_5), (e_2, v_1), \ldots, (e_2, v_5), \ldots, (e_d, v_1), \ldots, (e_d, v_5)\}.$$

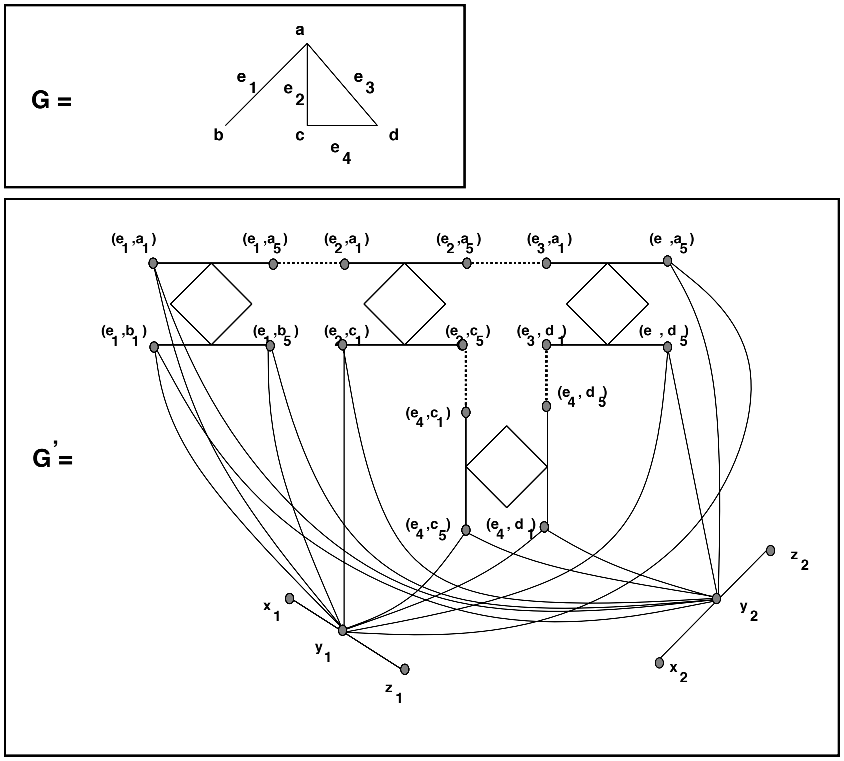

Figure 4: An example of the construction described in the proof of Theorem 7 when $b = 1$. Diamonds are symbols of the gadgets drawn on Figure 3.

Finally, the first and last vertices of Q_v, (e_1, v_1) and (e_d, v_5) are connected to the vertices $y_1, ..., y_{2b}$ by a complete bipartite graph $K_{2,2b}$ (see Figure 4). The set U is then defined by

$$\begin{aligned} U &= \{(e, u_1), (e, u_5), (e, v_1), (e, v_5) \ / \ e^{(1)}, e^{(2)}, \ e \in E\} \\ &\cup \ \{c_3, c_5, \text{ for each cross } C\} \\ &\cup \ \{x_i, y_i, z_i, \ i = 1, \ldots, 2b\}. \end{aligned}$$

All these nodes are marked in grey on Figures 3 and 4. Of course, G' and U can be constructed in polynomial time from G and b. Let us list some simple properties which are satisfied by the gadget of an edge $e = (u, v)$. For any shortest path pseudo-matching $\mathcal{P}$ in U satisfying $\mu(U, \mathcal{P}) \leq 1$, we have the following properties:

1. In any cross C, vertices c_3 and c_5 must be connected together, otherwise $\mu(U, \mathcal{P}, c_4) > 1$. Moreover, no other matching can pass through cross C, otherwise the load of c_4 would increase.

2. Vertices $e^{(1)}$, and $e^{(2)}$ must be connected together. Indeed, connecting $e^{(1)}$ by a shortest path

to any node of U distinct from $e^{(2)}$ requires to pass through a cross, and we have seen that this is not possible.

3. Vertex (e, u_1) can only be connected to vertex (e, u_5), or to a vertex which does not belong to the same gadget, otherwise the matching would require to pass through a cross.

4. If vertex (e, u_1) is matched to vertex (e, u_5), then vertex (e, v_1) is not matched to vertex (e, v_5), otherwise the load of (e, u_3) or (e, v_3) would be greater than 1 because of the matching $e^{(1)} - e^{(2)}$. In other words, at least two of the four vertices of the form (e, u_1), (e, u_5), (e, v_1) or (e, v_5) must be matched to vertices which do not belong to the same gadget.

Let us show that there exists a pseudo-matching in U of maximum load 1 and with shortest paths only, if and only if G has a vertex cover of size b or less.

$\diamond$ Suppose that there exists a shortest path pseudo-matching $\mathcal{P}$ in U such that $\mu(U, \mathcal{P}) \leq 1$. From properties 1 to 4 stated before, we already know what is the form of matchings involving nodes inside a gadget. For each path P_i, only one vertex of P_i can be matched to a vertex not in P_i, otherwise the load of y_i would be strictly greater than 1. For the same reason, no path connecting two vertices not belonging to P_i can pass through the vertex y_i. Without loss of generality, we can assume that vertex y_i is the vertex which must be matched to a vertex not in the path P_i. Let u be a vertex of G, and let $e_1, e_2, \ldots, e_d$ be its incident edges. If the vertex (e_1, u_1) is matched to a vertex y_i, then, from Property 3, vertex (e_1, u_5) will be matched to vertex (e_2, u_1). Then (e_2, u_5) will be matched to (e_3, u_1), and so on until vertex (e_d, u_5) is matched to some vertex y_j.

Let S be the set of vertices of G such that $u \in S$ if and only if (e_1, u_1) and (e_d, u_5) are both matched to vertices of type y_i, $i \in \{1, \ldots, 2b\}$, in the pseudo-matching. Note that $|S| \leq b$. Assume that S is not a vertex cover of G. Then let e be an edge of G such that none of its two endpoints belong to S. From Property 4, at least two vertices of the gadget corresponding to e must be connected to a vertex not in this gadget. Let u be an extremity of e such that (e, u_1) and (e, u_5) are both matched to vertices not in the gadget of e. Vertex (e, u_5) is matched to a vertex (e', u_1), vertex (e', u_5) is matched to a vertex (e'', u_1), and so on. Finally, if the degree of u is d, and if $e_1, e_2, \ldots, e_d$ denote the d incident edges of u, then vertex (e_{d-1}, u_5) is matched to vertex (e_d, u_1) of path Q_u, and vertex (e_d, u_5) must be matched to a vertex belonging to $\{y_i, i = 1, ..., 2b\}$ (otherwise, the load of a vertex in $\{y_i, i = 1, ..., 2b\}$ would be larger than 1). For the same reason, $(e_1, 1)$ must be matched to a vertex belonging to $\{y_i, i = 1, \ldots, 2b\}$. Thus $u \in S$, and this is a contradiction. Therefore, S is a vertex cover of G size b or less.

$\diamond$ Conversely, suppose that $S \subseteq V$ is a vertex cover of G of size b or less. If $|S| < b$, then we can add some vertices to S in order to obtain a vertex cover S' of exactly b vertices. So, we can assume without loss of generality that $|S| = b$. For any vertex s of G, let d be its degree, and let $e_1, e_2, \ldots, e_d$ be its d incident edges. Vertices of S are labeled from 1 to b. We construct the pseudo-matching $\mathcal{P}$ in U as follows:

- if $s \in S$, then let i be its label in S. We set the following: y_{2i} is matched to (e_1, s_1), (e_1, s_5) is matched to (e_2, s_1), $\ldots$, (e_d, s_5) is matched to y_{2i+1};
- if $s \notin S$, then for any $i \in \{1, \ldots, d\}$, we set the following: vertex (e_i, s_1) is matched to (e_i, s_5), and the path from $e_i^{(1)}$ to $e_i^{(2)}$ passes through the vertex (e_i, v_3) where $e_i = (s, v)$.

One can check that this pseudo-matching $\mathcal{P}$ is such that $\mu(U, \mathcal{P}) \leq 1$, and that all paths in $\mathcal{P}$ are shortest paths.

□

Theorem 7 shows that it is difficult to minimize the load of the routers at each round. Unfortunately, this is also true for the whole protocol. Given a graph G, and a vertex u of G, we denote by $M(G, u)$ the minimum of $\max_{i=1,...,\lceil \log_2 n \rceil} \mu(U_i, \mathcal{P}_i)$ over all the broadcast protocols $\mathcal{B}$ from u in G which perform in $\lceil \log_2 n \rceil$ rounds in the line model, and where $\mathcal{P}_i$ is the set of paths corresponding to the pseudo-matching in U_i generated at round i of the broadcasting protocol $\mathcal{B}$. We have:

Property 4 *The following problem is NP-complete:*

MINIMUM BROADCAST LOAD (MBL):

Instance: A graph G, a vertex u of G, and an integer k.

Question: $M(G, u) \leq k$?

Proof. If there exists a broadcasting protocol $\mathcal{B}$ such that $M(G, u) = 0$, then, no vertex is a transmitter at any round. So, protocol $\mathcal{B}$ is a broadcasting protocol from u in $\lceil \log_2 n \rceil$ rounds in the telephone model. Clearly, the reciprocal holds. Therefore, the problem MBL with $k = 0$ is equivalent to the NP-complete problem LogB.

□

6 Minimization of the switching time

In this section, we focus on the switching times induced by long paths between two vertices. Even if cut-through routed networks are not too much sensitive to path length, one may require that a call does not exceed a reasonable distance between the source and the destination. Given a pseudo-matching $\mathcal{P}$ of a subset U of vertices of a graph G, we denote by $L(U,\mathcal{P})$ the maximum length of the paths of $\mathcal{P}$. Let $L(U) = \min_{\mathcal{P}} L(U,\mathcal{P})$, where $\mathcal{P}$ is a pseudo-matching of U. If the switching time of the routers cannot be neglected, $L(U)$ is an important parameter to minimize at each round of a multicast protocol. Unfortunately, we have:

Theorem 8 *The following problem is NP-complete:*

Maximum Length of a Pseudo Matching (MLPM):

Instance: A graph $G = (V, E)$, a subset $U \subseteq V$, and an integer k.

Question: $L(U) \leq k$?

Proof. *MLPM* is clearly in NP. To prove that it is NP-complete, we transform the Vertex Cover problem to *MLPM* with $k = 7$. Let an arbitrary instance of VC be given by a graph $G = (V, E)$, and a positive integer b. We construct an instance of *MLPM*, that is a graph $G' = (V', E')$ and a subset U of vertices of G' such that U has a pseudo-matching $\mathcal{P}$ satisfying $L(U,\mathcal{P})) \leq 7$ if and only if G has a vertex cover of size b or less. The construction of G' is quite similar to the one of the proof of Theorem 7. The only differences are:

1. each of the $2b$ paths P_i of length 3 is transformed into a single vertex y_i, $i = 1, \ldots, 2b$;
2. the gadgets of the edges of G are as shown on Figure 5;
3. each bipartite graph $K_{2,2b}$ connecting the two endpoints of a path Q_u to the vertices y_i, $i = 1, \ldots, 2b$, is slightly transformed by replacing each edge by a path of length 7.

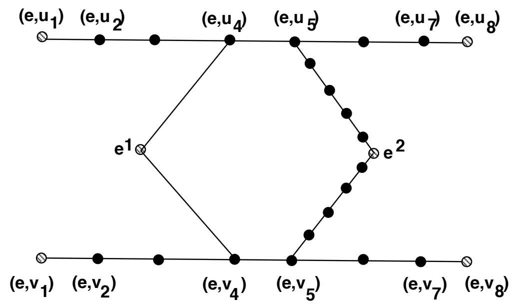

Figure 5: The gadget of an edge $e = (u, v)$ used in the proof of the Theorem 8. Vertices in U are marked in grey.

The set U is defined exactly as in the proof of Theorem 7. The reader can check that the gadgets presently defined have the same properties as the gadgets defined in the proof of Theorem 7. Actually, the remainder of the proof is similar to the proof of Theorem 7.

□

Concerning the whole protocol, we have a similar bad news for both the maximum and the sum of the maximum lengths of the paths. Given a graph G and a vertex u of G, we denote by $L_{sum}(G, u)$ the minimum of the sum of all the maximum lengths of the $\lceil \log_2 n \rceil$ pseudo-matchings generated by a broadcast protocol, over all the broadcast protocols from u performing in $\lceil \log_2 n \rceil$ rounds in the line model. More formally: $L_{sum}(G, u) = \min_{\mathcal{B}} \sum_{i=1}^{\lceil \log_2 n \rceil} L(U_i, \mathcal{P}_i)$ where $\mathcal{P}_i$ is the pseudo-matching in U_i at round i of the broadcast protocol, $i = 1, \ldots, \lceil \log_2 n \rceil$. We have:

Property 5 *The following problem is NP-complete:*

Sum Max:

Instance: A graph G, a vertex u of G, and an integer k.

Question: $L_{sum}(G, u) \leq k$?

Proof. If there exists a broadcasting protocol $\mathcal{B}$ such that $L_{sum}(G, u) \leq \lceil \log_2 n \rceil$, then every call satisfies the telephone model constraints. Thus protocol $\mathcal{B}$ is a broadcasting protocol from u in $\lceil \log_2 n \rceil$ rounds in telephone model. The reciprocal clearly holds. So the Sum-Max problem with $k = \lceil \log_2 n \rceil$ is equivalent to the NP-complete problem LogB.

□

Given a graph G and a vertex u of G, we denote by $L_{max}(G,u)$ the minimum of the maximum length of all the paths of the $\lceil \log_2 n \rceil$ pseudo-matchings generated at each round by a broadcast protocol over all the broadcast protocols from u performing in $\lceil \log_2 n \rceil$ rounds in the line model. More formally: $L_{max}(G,u) = \min_{\mathcal{B}} \max_{i=1 \ldots \lceil \log_2 n \rceil} L(U_i, \mathcal{P}_i)$ where $\mathcal{P}_i$ is the pseudo matching in U_i at round i of the broadcast protocol, $i = 1, \ldots, \lceil \log_2 n \rceil$. We have:

Property 6 *The following problem is NP-complete:*

MAX MAX:

Instance: A graph G, a vertex u of G, and an integer k.

Question: $L_{max}(G,u) \leq k$?

Proof. Similar to Property 5: the MAX-MAX problem with $k = 1$ is equivalent to the NP-complete problem LOGB.

□

7 Conclusion and open problems

Table 1 summarizes the complexities of the minimization problems related to the multicast and the broadcast problems in both line model, and cut-through model.

	Broadcasting/multicasting	Pseudo-matching
Sum of the paths lengths	NP-complete (Theorem 3)	**P** (Lemma 1)
Number of transmitters	NP-complete (Property 3)	NP-complete (Theorem 6)
Load of the transmitters	NP-complete (Property 4)	NP-complete (Theorem 7)
Maximum path length	NP-complete (Property 5)	NP-complete (Theorem 8)
Sum of the maximum paths lengths	NP-complete (Property 6)	

Table 1: Complexities of minimization problems related to the pseudo-matching problem, and to the broadcast and multicast problems

Even if the situation seems "despairing" at the first glance (most of the problems are NP-complete), the reader must keep in mind that the main parameter to optimize in cut-through networks is the use of the bandwidth. Indeed, the first property that a communication protocol in line or cut-though models must satisfy is to be free of link contention. From these points of view, we have derived important results:

1. We have derived a polynomial algorithm that returns, for any network, and any minimal possibly adaptive routing function, a broadcast or a multicast protocol that performs in the minimal number of rounds.

2. Moreover, such broadcast or multicast protocols satisfy that the sum of the path-lengths at any round is minimum. In other words, the total bandwidth required at each round is minimum.

3. Even if minimizing the use of the bandwidth for the whole protocol is NP-complete, our algorithm approaches the lower bound up to a logarithmic multiplicative factor.

4. We have derived a specific strategy for tree-networks which approaches the lower bound up to a small multiplicative factor that is conjectured to be a constant on the average.

The minimization of several second order parameters as the number of transmitters, the load of the transmitters, the maximum length of the paths, and the sum of the maximum length of the paths used at each round yield NP-complete decision problems. Since these problems might be not so critical for cut-through routing, we did not try to derive approximated solutions. However, if minimizing these parameters turns to be a major issue for some specific technical reasons, such algorithms should be derived. This is therefore an important direction for further research on this topic.

Other problems seem to be more important to solve however. We indicate below two directions that we are currently investigating:

The all-port model. As we pointed out in the introduction, one can consider a model in which any node u of degree Δ_u is able to generate Δ_u messages that can be simultaneously sent by the router to different destinations. If the network is regular of degree Δ, we have $b(G) \geq \lceil \log_{\Delta+1} n \rceil$. It has been recently shown that knowing whether $b(G) \leq k$ for an arbitrary network G, and an arbitrary constant k, is NP-complete [2]. Thus this problem requires polynomial approximation algorithms.

The gossiping problem. This problem is also called all-to-all broadcasting. It consists of a simultaneous broadcasting from all the nodes of the network. In the 1-port line model, we have $b(G) \geq g(G) \geq 2\min_{u\in V} b(G,u) - 1$ by performing first an accumulation of all the messages at a given vertex, and then performing a broadcasting from this vertex. Although such an protocol is quite efficient in term of rounds, it does not balance the traffic and create contention at the accumulation node [22]. Therefore, more efficient protocols are required.

Acknowledgment. The authors want to thank the anonymous referees whose valuable comments helped a lot to improve the quality of the original manuscript.

References

[1] J-C. Bermond, L. Gargano, S. Perennes, A. Rescigno, and U. Vaccaro. Efficient collective communication in optical networks. Technical Report 95-65, I3S, Sophia-Antipolis, France, 1995.

[2] J. Cohen. NP-complete communication problems. Draft.

[3] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the Natianal Bureau of Standards-B*, 69B:125–130, 1965.

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

[5] R. Feldmann, J. Hromkovič, S. Madhavapeddy, B. Monien, and P. Mysliwietz. Optimal algorithms for dissemination of information in generalized communication modes. *Discrete Applied Mathematics*, 53(1-3):55–78, 1994.

[6] E. Fleury and P. Fraigniaud. Strategies for multicasting in meshes. In *23rd International Conference on Parallel Processing (ICPP'94)*, 1994.

[7] E. Fleury and P. Fraigniaud. Analysis of deadlock-free path-based wormhole multicasting in meshes in case of contentions. In *6th Symposium on the Frontiers of Massively Parallel Computing (Frontiers '96)*, Annapolis, Maryland, October 1996. IEEE Computer Society Press.

[8] Message Passing Interface Forum. MPI: A message-passing interface standard. Technical Report CS-94-230, Computer Science Department, University of Tennessee, Knoxville, TN 37996, 1994.

[9] P. Fraigniaud. Broadcasting in trees. Research report 95-26, Laboratoire de l'Informatique du Parallélisme, ENS-Lyon, France, 1995.

[10] P. Fraigniaud and J. Peters. Structured Communication in torus networks. In IEEE, editor, *28th Annual Hawaii International conference on system sciences*, pages 584–593, 1995.

[11] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message Passing Interface.* The MIT Press, 1994.

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

[13] C-T. Ho and M-Y. Kao. Optimal broadcast in all-port wormhole-routed hypercubes. *IEEE Transactions on Parallel and Distributed Systems*, 6(2):200–204, 1995.

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

[15] 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.

[16] J. Hromković, R. Klasing, W. Unger, and H. Wagerer. Optimal algorithm for broadcast and gossip in the edge-disjoint path mode. In Springer-Verlag, editor, *4th Scandinavian Workshop on Algorithm Theory (SWAT'94)*, volume 824 of *LNCS*, pages 219–230, 1994.

[17] S.L. Johnsson. Communication efficient basic linear algebra computations on hypercube architectures. *Journal of Parallel and Distributed Computing*, 4:133–172, 1987.

[18] S.L. Johnsson and Ching-Tien Ho. Optimum broadcasting and personalized communication in hypercubes. *IEEE Transactions on Computers*, 38(9):1249–1268, 1989.

[19] J.O. Kane and J. Peters. Line broadcasting in cycles. *Discrete Applied Mathematics*, to appear.

[20] G. Kortsarz and D. Peleg. Approximation algorithms for minimum time broadcast. In *Proceedings of the 1992 Israel Symposium on Theory of Computer Science*, number 601 in Lecture Notes in Computer Science. Springer-Verlag, 1992.

[21] C. Laforest. Broadcast and gossip in line-communication mode. Technical Report 1005, Laboratoire de Recherche en Informatique, Univ. Paris Sud, 91405 Orsay, France, 1995.

[22] C. Laforest. Gossip in trees under line-communication mode. In L. Bougé, P. Fraigniaud, A. mignotte, and Y. Robert, editors, *Euro-Par '96: Parallel Processing*, volume 1123 of *LNCS*, pages 333–340. Springer, 1996.

[23] Y. Lan, A-H. Esfahanian, and L. Ni. Multicast in hypercube multiprocessor. *Journal of Parallel and Distributed Computing*, 8:30–41, 1990.

[24] T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann, 1992.

[25] X. Lin and L. M. Ni. Multicast communication in multicomputer networks. *IEEE Transaction on Parallel and Distributed Systems*, 4(10):1104–1117, 1993.

[26] M. Marsan, A. Bianco, E. Leonardi, and F. Neri. Topologies for wavelength-routing all-optical networks. *IEEE/ACM transactions on networking*, 1(5):534–546, 1993.

[27] L. Ni and P. McKinley. A survey of wormhole routing techniques in direct networks. *Computers*, 26(2):62–76, feb 1993.

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

[29] Y. Saad and M.H. Schultz. Data communication in hypercubes. *Journal of Parallel and Distributed Computing*, 6:115–135, 1989.

[30] Y. Saad and M.H. Schultz. Data communication in parallel architectures. *Parallel Computing*, 11:131–150, 1989.

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

[32] P. Winter. Steiner problem in networks: a survey. *IEEE Networks*, 17(2):129–167, 1987.

[33] H. Xu, P. McKinley, and L. Ni. Efficient implementation of barrier synchronization in wormhole-routed hypercube multicomputers. *Journal of Parallel and Distributed Computing*, 16:172–184, 1992.