ELSEVIER
Theoretical Computer Science 290 (2003) 2009–2020
Theoretical Computer Science
www.elsevier.com/locate/tcs

Note

Deterministic broadcasting time with partial knowledge of the network ☆

Gianluca De Marco[a,1], Andrzej Pelc[b,*,2]

[a] *Istituto per le Applicazioni Telematiche, Consiglio Nazionale delle Ricerche, 56124 Pisa, Italy*
[b] *Département d'Informatique, Université du Québec à Hull, C.P. 1250, Succ. B, Hull, Qué. Canada J8X 3X7*

Received 14 October 2000; received in revised form 7 November 2001; accepted 9 January 2002
Communicated by D. Peleg

Abstract

We consider the time of deterministic broadcasting in networks whose nodes have limited knowledge of network topology. Each node v knows only the part of the network within *knowledge radius* r from it, i.e., it knows the graph induced by all nodes at distance at most r from v. Apart from that, each node knows the maximum degree Δ of the network. One node of the network, called the *source*, has a message which has to reach all other nodes. We adopt the widely studied communication model called the *one-way* model in which, in every round, each node can communicate with at most one neighbor, and in each pair of nodes communicating in a given round, one can only send a message while the other can only receive it. This is the weakest of all store-and-forward models for point-to-point networks, and hence our algorithms work for other models as well, in at most the same time.

We show trade-offs between knowledge radius and time of deterministic broadcasting, when the knowledge radius is small, i.e., when nodes are only aware of their close vicinity. While for knowledge radius 0, minimum broadcasting time is $\Theta(e)$, where e is the number of edges in the network, broadcasting can be usually completed faster for positive knowledge radius. Our main results concern knowledge radius 1. We develop fast broadcasting algorithms and analyze their execution time. We also prove lower bounds on broadcasting time, showing that our algorithms are close to optimal.

☆ A preliminary version of this paper will appear in Proceedings of the 11th Annual International Symposium on Algorithms and Computation (ISAAC'00), December 2000, Taipei, Taiwan.

* Corresponding author.
E-mail addresses: demarco@imc.pi.cnr.it (G. De Marco), pelc@uqah.uquebec.ca (A. Pelc).

[1] This research was done during the stay of Gianluca De Marco at the Université du Québec à Hull.

[2] Supported in part by NSERC Grant OGP 0008136.

PII: S0304-3975(02)00073-7

1. Introduction

1.1. The problem

Broadcasting is one of the fundamental tasks in network communication. One node of the network, called the *source*, has a message which has to reach all other nodes. In *synchronous* communication, messages are sent in *rounds* controlled by a global clock. In this case the number of rounds used by a broadcasting algorithm, called its *execution time*, is an important measure of performance. Broadcasting time has been extensively studied in many communication models (cf. surveys [16,20,21]) and fast broadcasting algorithms have been developed.

If network communication is to be performed in a distributed way, i.e., message scheduling has to be decided locally by nodes of the network, without the intervention of a central monitor, the efficiency of the communication process is influenced by the amount of knowledge concerning the network, a priori available to nodes. It is often the case that nodes know their close vicinity (for example they know their neighbors) but do not know the topology of remote parts of the network.

The aim of this paper is to study the impact of the amount of local information available to nodes on the time of broadcasting. Each node v knows only the part of the network within *knowledge radius* r from it, i.e., it knows the graph induced by all nodes at distance at most r from v. Apart from that, each node knows only the maximum degree Δ of the network. In Section 4, we assume additionally that each node knows the total number n of nodes. We concentrate on the case when knowledge radius is small, i.e., when nodes are only aware of their close vicinity. We develop fast broadcasting algorithms and analyze their execution time. We also prove lower bounds on broadcasting time, showing that our algorithms are close to optimal, for a given knowledge radius.

1.2. Related work

Network communication with partial knowledge of the network has been studied by many researchers. This topic has been extensively investigated, e.g., in the context of radio networks. In [4] a lower bound $\Omega(n)$ on deterministic broadcasting time was shown under the assumption that nodes of a radio network know a priori their neighborhood. In [5,6,7,9,28], an even more restrictive assumption has been adopted, namely that every node knows only its own label (knowledge radius zero in our terminology). A series of increasingly faster deterministic broadcasting algorithms was proposed in these papers, culminating with the currently fastest one from [7], working in time $O(n \log^2 n)$. In [11] a restricted class of radio networks was considered, and partial knowledge available to nodes concerned the range of their transmitters.

In [18] time of broadcasting and of two other communication tasks was studied in point-to-point networks assuming that each node knows only its own degree. However, the communication model was different from the one assumed in this paper: every node could simultaneously receive messages from all of its neighbors.

In [3] broadcasting was studied assuming a given knowledge radius, as we do in this paper. However, the adopted efficiency measure was different: the authors studied the number of messages used by a broadcasting algorithm, and not its execution time, as we do.

A topic related to communication in an unknown network is that of graph exploration [1,10,27]: a robot has to traverse all edges of an unknown graph in order to draw a map of it. In this context the complexity measure is the number of edge traversals which is proportional to execution time, as only one edge can be traversed at a time.

In the above papers communication algorithms were deterministic. If randomization is allowed, very efficient broadcasting is possible without knowing the topology of the network, cf., e.g., [4,15]. In fact, in [4] the differences of broadcasting time in radio networks between the deterministic and the randomized scenarios were the main topic of investigation.

Among numerous other graph problems whose distributed solutions with local knowledge available to nodes have been studied, we mention graph coloring [8,26], fault mending [25], label assignment [17], and frequency assignment [22].

1.3. The model and terminology

The communication network is modeled by a simple undirected connected graph with a distinguished node called the *source*. n denotes the number of nodes, e denotes the number of edges, Δ denotes the maximum degree, and D denotes the diameter of the graph. All nodes have distinct labels which are integers between 1 and n, but our algorithms and arguments are easy to modify when labels are in the range 1 to M, where $M \in O(n)$.

Communication is deterministic and proceeds in synchronous rounds controlled by a global clock. We say that a broadcasting algorithm works in time t for n-node networks of diameter D and maximum degree Δ if, for any such network, all nodes get the source message within t rounds, and no messages are sent after round t. Only nodes that already got the source message can transmit, hence, broadcasting can be viewed as a wake-up process. We adopt the widely used *one-way* model (cf. [21]), also called the 1-*port half-duplex model* [16]. In every round, each node can communicate with at most one neighbor, and in each pair of nodes communicating in a given round, one can only send an (arbitrary) message, while the other can only receive it. This model has been used, e.g., in [13,14,19,23,24]. It has the advantage of being the weakest of all store-and-forward models for point-to-point networks (cf. [16]), and hence our algorithms work also for other models (allowing more freedom in sending and/or receiving), in at most the same time.

For a natural number r we say that r is the *knowledge radius* of the network if every node v knows the graph induced by all nodes at distance at most r from v. Apart from that partial topological information, each node knows only the maximum degree Δ of the network. In Section 4, we assume additionally that each node knows the total number n of nodes. For example, if knowledge radius is 1, each node knows its own label, labels of all neighbors, knows which of its adjacent edges joins it with which neighbor, and knows which neighbors are adjacent between them. The latter assumption is where our definition of knowledge radius differs from that in [3], where knowledge radius r meant that a node v knows the graph induced by all nodes at distance at most r from v *with the exception of* adjacencies between nodes at distance exactly r from v. However, all our results hold for this weaker definition as well. In fact, we show that our lower bounds are valid even under the stronger notion and we construct the algorithms using only the weaker version from [3], thus obtaining all results under both definitions of knowledge radius.

1.4. Overview of results

We show trade-offs between knowledge radius and time of deterministic broadcasting, when knowledge radius is small, i.e., when nodes are only aware of their close vicinity. While for knowledge radius 0, minimum broadcasting time is $\Theta(e)$, where e is the number of edges in the network, broadcasting can be usually completed faster for positive knowledge radius. Our main results concern knowledge radius 1. We develop a broadcasting algorithm working in time $O(\min(n, D^2\Delta))$, and we show that for bounded maximum degree Δ this algorithm is asymptotically optimal. For any knowledge radius $r \leqslant D$, we show a broadcasting algorithm working in time $O(D^2\Delta/r)$, and for knowledge radius exceeding $c\log^4 n$, for some constant c, we show how to broadcast in time $O(D\Delta\log n)$. On the other hand, for any knowledge radius $r \leqslant D$, we prove a lower bound $\Omega(D\Delta/r)$ on broadcasting time, whenever $D\Delta \in O(n)$.

2. Preliminary results: knowledge radius 0

For knowledge radius 0 tight bounds on broadcasting time can be established: the minimum broadcasting time in this case is $\Theta(e)$, where e is the number of edges in the network.

We first make the following observation (cf. [18]).

Proposition 2.1. *In every broadcasting algorithm with knowledge radius* 0 *the source message must traverse every edge at least once.*

Proof. Consider a broadcasting algorithm A working correctly on every network. Suppose that G is a network such that the source message does not traverse edge $l = \{x, y\}$ during the execution of A. Let G' be the network resulting from G by removing edge l and adding a new node z and edges $\{x, z\}$ and $\{y, z\}$. Let t be the execution time of A on G. Consider the first t rounds of the run of algorithm A on the network G'. The actions of A in these rounds and the local states of all nodes except z are identical when A runs on G or on G'. Hence, no messages will be sent after round t when A is run on G', and consequently node z will not be informed. This is a contradiction. □

The following result establishes a natural lower bound on broadcasting time. Its proof is similar to that of Theorem 4.5 from [17].

Theorem 2.1. *Every broadcasting algorithm with knowledge radius* 0 *requires time at least e for networks with e edges.*

Proof. Consider a broadcasting algorithm A that works correctly on every network. Suppose, for the purpose of contradiction, that there exists a network $G=(V,E)$ and an execution ξ of the algorithm A on G working in fewer than $|E|$ rounds. By Proposition 2.1 the source message must traverse each edge of G at least once during execution ξ. Hence, there exist a round t and two different edges (u_1,w_1) and (u_2,w_2) such that the source message is sent on each of them for the first time in round t.

Suppose (without loss of generality) that u_i has sent the source message to w_i over the edge (u_i,w_i) in round t, for $i=1,2$. By the definition of the one-way model, all nodes u_1, u_2, w_1 and w_2, are distinct. Consider the network G_2 obtained from G by eliminating the edges (u_1,w_1) and (u_2,w_2), and replacing them by a new node v', and four new edges (u_1,v'), (v',w_1), (u_2,v') and (v',w_2). If algorithm A is invoked from the same source s on G_2, then its execution ξ_2 on G_2 will be identical to ξ up to round $t-1$, and moreover, in round t a message will be sent by the node u_i over the edge (u_i,v') , for $i=1,2$. This violates the constraints of the one-way model, as it causes the node v' to receive two messages in the same round, leading to contradiction. □

In the classic depth first search algorithm a token (the source message) visits all nodes and traverses every edge twice. In this algorithm only one message is sent in each round and hence the specifications of the one-way model are respected. This is a broadcasting algorithm working in $2e$ rounds and hence its execution time has optimal order of magnitude. It should be noted that depth first search does not require the knowledge of Δ or n. (It follows from [27] that, using a more subtle algorithm, broadcasting can be done in time $e+O(n)$ in any n-node network with e edges.) In view of Theorem 2.1, we have the following result.

Theorem 2.2. *The minimum broadcasting time with knowledge radius* 0 *is* $\Theta(e)$, *where e is the number of edges in the network.*

3. Knowledge radius 1

In order to present our first algorithm we need the notion of a layer of a network. For a natural number k, the kth *layer* of network G is the set of nodes at distance k from the source. The idea of Algorithm Conquest-and-Feedback is to inform nodes of the network layer by layer. After the $(k-1)$th layer is informed (*conquered*) every node of this layer transmits to any other node of this layer information about its neighborhood. This information travels through a partial tree constructed on nodes of the previous layers and consumes most of the total execution time of the algorithm. As soon as this information is exchanged among nodes of the $(k-1)$th layer (*feedback*), they proceed to relay the source message to nodes of layer k. The knowledge of all adjacencies between nodes from layer $k-1$ and nodes from layer k enables transmitting the source message without collisions.

We now present a detailed description of the algorithm.

Algorithm Conquest-and-Feedback

All rounds are divided into consecutive *segments* of length Δ. Rounds in each segment are numbered 1 to Δ. The set of segments is in turn partioned into *phases*. We preserve the invariant that after the kth phase all nodes of the kth layer know the source message.

The first phase consists of the first segment (i.e., it lasts Δ rounds). In consecutive rounds of this segment the source informs all of its neighbors, in increasing order of their labels. (If the degree of the source is smaller than Δ, the remaining rounds of the segment are idle.)

Any phase k, for $k>1$, consists of $2k-1$ segments (i.e., it lasts $\Delta(2k-1)$ rounds). Suppose by induction that after phase $k-1$ all nodes of layer $k-1$ have the source message. Moreover, suppose that a tree spanning all nodes of layers $j<k$ is distributedly maintained: every node v of layer j remembers from which node $P(v)$ of layer $j-1$ it received the source message for the first time, and remembers the round number $r(v)\leqslant\Delta$ in a segment in which this happened. We suppose by induction that $r(v)\neq r(v')$ for nodes v and v' for which $P(v)=P(v')$.

We now describe phase k of the algorithm. Its first $2(k-1)$ segments are devoted to exchanging information about neighborhood among nodes of layer $k-1$ (feedback). Every such node transmits a message containing its own label and labels of all its neighbors. During the first $k-1$ segments messages travel toward the source: one segment is devoted to get the message one step closer to the source. More precisely, a node v of layer $j<k-1$ which got feedback messages in a given segment transmits their concatenation to $P(v)$ in round $r(v)$ of the next segment. The definitions of $r(v)$ and $P(v)$, and the inductive assumption guarantee that collisions are avoided. After these $k-1$ segments the source gets all feedback messages. From the previous phase the source knows all labels of nodes in layer $k-2$. Since neighbors of a node in layer $k-1$ can only belong to one of the layers $k-2$, $k-1$ or k, the source can deduce from information available to it the entire bipartite graph B_k whose node sets are layers $k-1$ and k and edges are all graph edges between these layers. The next $k-1$ segments are devoted to broadcasting the message describing graph B_k to all nodes of layer $k-1$. Every node of layer $j<k-1$ which already got this message relays it to nodes of layer $j+1$ during the next segment, using precisely the same schedule as it used to broadcast the source message in phase $j+1$. By the inductive assumption collisions are avoided.

Hence, after $2(k-1)$ segments of phase k the graph B_k is known to all nodes of layer $k-1$. The last segment of the phase is devoted to relaying the source message to all nodes of layer k. This is done as follows. Every node v of layer $k-1$ assigns consecutive *slots* $s=1,\dots,\delta$, $\delta\leqslant\Delta$ to each of its neighbors in layer k, in increasing order of their labels. Since B_k is known to all nodes of layer $k-1$, all slot assignments are also known to all of them. Now transmissions are scheduled as follows. For any node v of layer $k-1$ and any round r of the last segment, node v looks at its neighbor w in layer k to which it assigned slot r. It looks at all neighbors of w in layer $k-1$ and defines the set $A(w)$ of those among them which assigned slot r to w. If the label of v is the smallest among all labels of nodes in $A(w)$, node v transmits the source message to w in round r of the last segment, otherwise v remains silent in this round. This schedule avoids collisions and guarantees that all nodes of layer k get the source message by the end of the kth phase. Moreover, the condition stating that $r(v) \neq r(v')$ for nodes v and v' for which $P(v) = P(v')$, is preserved. Hence, the invariant is maintained, which implies that broadcasting is completed after at most D phases, where D is the diameter of the network. □

Phase 1 lasts Δ rounds, and each phase k, for $k > 1$, lasts $\Delta(2k - 1)$ rounds. Since broadcasting is completed after D phases, its execution time is at most

$$\Delta\left(1 + \sum_{k=2}^{D}(2k-1)\right) \in \mathrm{O}(D^2\Delta).$$

Hence we get

Theorem 3.1. *Algorithm Conquest-and-Feedback completes broadcasting in any network of diameter D and maximum degree Δ in time* $\mathrm{O}(D^2\Delta)$.

For large values of D and Δ the following simple Algorithm Fast-DFS may be more efficient than Algorithm Conquest-and-Feedback. It is a DFS-based algorithm using an idea from [2]. The source message is considered as a token which visits all nodes of the graph. In every round only one message is transmitted, hence collisions are avoided. The token carries the list of previously visited nodes. At each node v the neighborhood of v is compared to this list. If there are yet non-visited neighbors, the token passes to the lowest labeled of them. Otherwise, the token backtracks to the node from which v was visited for the first time. If there is no such node, i.e., if v is the source, the process terminates. In this way all nodes are visited, and the token traverses only edges of an implicitly defined DFS tree, rooted at the source, each of these edges exactly twice. Avoiding sending the token on non-tree edges speeds up the process from time $\Theta(e)$ to $\Theta(n)$. Hence we get

Proposition 3.1. *Algorithm Fast-DFS completes broadcasting in any n-node network in time* $\mathrm{O}(n)$.

Since the diameter D may be unknown to nodes, it is impossible to predict which of the two above algorithms is faster for an unknown network. However, simple interleaving of the two algorithms guarantees broadcasting time of the order of the better of them in each case. Define Algorithm Interleave which, for any network G executes steps of Algorithm Conquest-and-Feedback in even rounds and steps of Algorithm Fast-DFS in odd rounds. As soon as the source learns that the faster of the two algorithms completed broadcasting (which happens in time $\mathrm{O}(\min(n, D^2\Delta))$), it broadcasts a message *stop* which prevents the slower algorithm from continuing. This additional broadcast does not change time complexity. Hence we have

Theorem 3.2. *Algorithm Interleave completes broadcasting in any n-node network of diameter D and maximum degree Δ in time* $\mathrm{O}(\min(n, D^2\Delta))$, *assuming that nodes know the parameter Δ.*

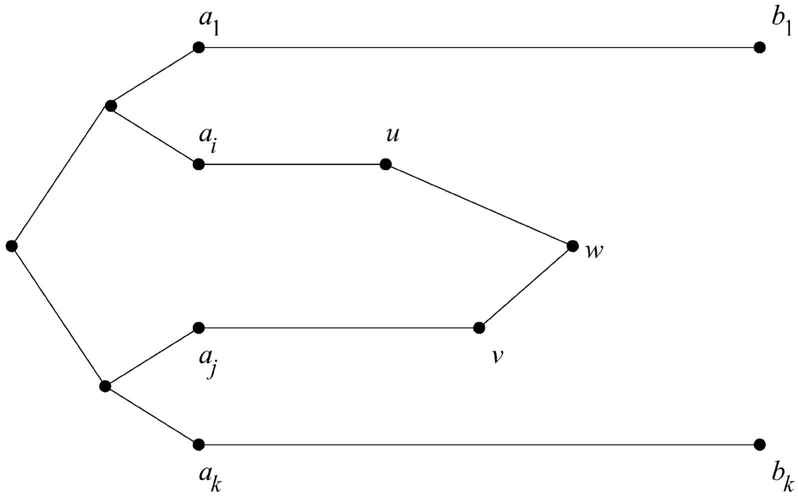

Fig. 1. Graph $T(u, v)$.

The following lower bound shows that, for constant maximum degree Δ, the execution time of Algorithm Interleave is asymptotically optimal.

Theorem 3.3. *Any broadcasting algorithm with knowledge radius 1 requires time* $\Omega(\min(n, D^2))$ *in some constant degree n-node networks of diameter D.*

Proof. Fix parameters n and $D < n$. Since D is the diameter of a constant degree network with n nodes, we must have $D \in \Omega(\log n)$. Consider a complete binary tree rooted at the source, of height $h \leqslant D/3$ and with k leaves $a_1, \ldots, a_k$, where $k \in \Omega(n/D)$. It has $2k - 1$ nodes. Assume for simplicity that D is even and let $L = D/2 - h$. Thus $L \in \Omega(D)$. Attach disjoint paths of length L (called *threads*) to all leaves. Denote by b_i the other end of the thread attached to a_i, and call this thread the ith thread. Again assume for simplicity that $2k - 1 + kL = n$, and thus the resulting tree T has n nodes and diameter D. (It is easy to modify the construction in the general case.)

Next consider any nodes u and v belonging to distinct threads, respectively, ith and jth, of T. Define the graph $T(u, v)$ as follows (see Fig. 1): remove the part of the ith thread between u and b_i (including b_i), and the part of the jth thread between v and b_j (including b_j), and add a new node w joining it to u and to v. Arrange the remaining nodes in a constant degree tree attached to the source, so as to create an n-node graph of constant degree and diameter D.

We now consider the class of graphs consisting of the tree T and of all graphs $T(u, v)$ defined above. We will show that any broadcasting algorithm with knowledge radius 1 which works correctly on this class requires time $\Omega(\min(n, D^2))$ in the tree T. Fix a broadcasting algorithm A.

Since the algorithm must work correctly on T, the source message must reach all nodes b_i, and consequently it must traverse all threads. For each thread define the *front* as the farthest node from the source in this thread, that knows the source message. Call each move of a front a *unit of progress*. Consider only the second half of each thread, the one farther from the source. Thus, $kL/2$ units of progress must be made to traverse those parts of threads.

Consider fronts u and v in second halves of two distinct threads and suppose that these fronts move in the same round t. Observe that before this is done, information which u has about its neighborhood must be transmitted to v or vice versa. Otherwise, the local states of u and v in round t are the same when the algorithm is run on T and on $T(u,v)$. However, simultaneous transmission from u and v in $T(u,v)$ results in a collision in their common neighbor w and thus the assumptions of the model are violated. Since u and v are in second halves of their respective threads, the distance between them is at least L, hence transmission of information from u to v requires at least L rounds after u becomes a front.

Units of progress are *charged* to rounds in which they are made in the following way. If at least two units of progress are made in a round, all of them are charged to this round. We call this the *first way of charge*. If only one unit of progress is made in a round we charge this unit to this round and call it the *second way of charge*.

Partition all rounds into disjoint segments, each consisting of L consecutive rounds. Fix such a segment of rounds, and let $t_1 < \cdots < t_s$ be rounds of this segment in which at least two units of progress are made. Let A_{t_i}, for $i = 1,\ldots,s$, be the set of thread numbers in which progress is made in round t_i. Notice that, for any $i \leqslant s$, the set $(A_{t_1} \cup \cdots \cup A_{t_{i-1}}) \cap A_{t_i}$ can have at most one element. Indeed, if $a, b \in (A_{t_1} \cup \cdots \cup A_{t_{i-1}}) \cap A_{t_i}$, for $a \neq b$, then fronts u in thread a and v in thread b move simultaneously in round t_i but neither information about neighborhood of u could reach v nor information about neighborhood of v could reach u because this information could only be sent less than L rounds before t_i.

Since $|(A_{t_1} \cup \cdots \cup A_{t_{i-1}}) \cap A_{t_i}| \leqslant 1$ for any $i \leqslant s$, it follows that $|A_{t_1}| + \cdots + |A_{t_s}| \leqslant k + s \leqslant k + L$. Hence at most $k+L$ units of progress can be charged to rounds of a segment in the first way. Clearly at most L units of progress can be charged to rounds of a segment in the second way. Hence, a total of at most $k + 2L$ units of progress can be charged to rounds of each segment. Since $kL/2$ units of progress must be made to traverse second halves of all threads, broadcasting requires at least $kL/(2(k+2L))$ segments and thus at least $kL^2/(2(k+2L))$ rounds. If $k \leqslant L$ we have $kL^2/(2(k+2L)) \geqslant kL/6 \in \Omega(n)$, and if $k \geqslant L$ we have $kL^2/(2(k+2L)) \geqslant L^2/6 \in \Omega(D^2)$. Hence, we have always the lower bound $\Omega(\min(n, D^2))$ on broadcasting time in the tree T. □

4. Larger knowledge radius

In this section we present two upper bounds and one lower bound on broadcasting time with larger knowledge radius r.

The first upper bound is given by a straightforward modification of Algorithm Conquest-and-Feedback, described in Section 3. Instead of “conquering” layers one by one giving “feedback” after each layer, the source message is broadcast to *seg-*

ments of r consecutive layers, using a predetermined tree spanning nodes of these layers. Then all nodes of the last layer of the segment send back to the source the information about the part of the network at distance r from them, using the same spanning tree and schedules similar as in the original algorithm. The source extends the current tree to a tree spanning the next segment of layers and transmits this entire information along this tree. Thus, the next segment of r layers can be "conquered". A phase informing a segment of r layers and giving feedback on the next segment takes at most $O(D\Delta)$ rounds, and $O(D/r)$ such phases are needed. This proves the following.

Theorem 4.1. *For any positive integer* $r \leqslant D$, *there exists a broadcasting algorithm with knowledge radius* r *which completes broadcasting in any network of diameter* D *and maximum degree* Δ *in time* $O(D^2\Delta/r)$.

From now on, we assume that apart from the portion of the graph within knowledge radius and apart from the maximum degree Δ, every node also knows the total number n of nodes. The upper bound established in Theorem 4.2 uses this assumption, and the lower bound given in Theorem 4.3 is valid in spite of this additional knowledge.

Our second algorithm working for larger knowledge radius is based on a result from [8] concerning distributed edge coloring in graphs. The authors give an algorithm that colors edges of a graph with $O(\Delta \log n)$ colors (incident edges have different colors) in time $O(\log^4 n)$. It is assumed that nodes know parameters n and Δ. The communication model used in [8] is more "liberal" than the one-way model: every node can communicate with all neighbors in a single round. However, since total running time of the algorithm from [8] is at most $c \log^4 n$, for some constant c, it follows that, for knowledge radius exceeding $c \log^4 n$, this coloring can be obtained with no communication at all. Hence we may assume that an $O(\Delta \log n)$ edge coloring is a priori known to nodes of the graph, for such knowledge radius.

Algorithm Color-and-Transmit

All nodes of the network have, as input, a fixed distributed k-coloring of edges, where $k \in O(\Delta \log n)$. More specifically, every node knows colors of its incident edges. The algorithm works in phases. The first phase lasts Δ rounds and each of the following phases lasts k rounds. In phase 1 the source sends the message to all of its neighbors. In round r of phase p, for $p \geqslant 2$, any node v that got the source message for the first time in the previous phase $p-1$, sends the message on its incident edge of color r. □

By definition of k-edge coloring, collisions are avoided. After D phases broadcast is completed. Hence we get the following result.

Theorem 4.2. *There exists a constant* c *such that if nodes have knowledge radius exceeding* $c \log^4 n$, *and know parameters* n *and* Δ, *then Algorithm Color-and-Transmit completes broadcasting in any* n*-node network of diameter* D *and maximum degree* Δ *in time* $O(D\Delta \log n)$.

We now present a lower bound on broadcasting time, for arbitrary knowledge radius r. This lower bound shows, for example, that if knowledge radius satisfies the assumption of Theorem 4.2 and is polylogarithmic in n then broadcasting time of Algorithm Color-and-Transmit exceeds the optimal time at most by a polylogarithmic factor.

Theorem 4.3. *Assume that $D\Delta \in O(n)$ and let $r \leqslant D$. Any broadcasting algorithm with knowledge radius r requires time $\Omega(D\Delta/r)$ on some n-node tree of maximum degree Δ and diameter D.*

Proof. Assume for simplicity that r divides D and that $1+(D-r)(\Delta-1)=n$. It is easy to modify the construction in the general case, using the assumption $D\Delta \in O(n)$. Let $d=\Delta-1$. Let T be a tree consisting of a root and d disjoint paths (branches) of length r attached to it. Let $k=D/r-1$ and consider k copies $T_1,\dots,T_k$ of T. Let the source be the root of T_1 and identify the root of T_{i+1} with some leaf of T_i, for any $i=1,\dots,k-1$. The resulting tree T^* has maximum degree $d+1=\Delta$, diameter $r+kr=D$, and $kdr+1=n$ nodes. We will show that any algorithm with knowledge radius r requires time $\Omega(D\Delta/r)$ on some labeled tree isomorphic to T^*.

Consider any broadcasting algorithm A. When the root v_i of T_i gets the source message in round t, its local state is the same regardless of the leaf of T_i to which T_{i+1} is attached. (This is due to the fact that knowledge radius is equal to the depth of T_i.) Hence, if T_{i+1} is attached to the leaf in the branch of T_i corresponding to the *last* child of v_i which gets the source message, then the root v_{i+1} of T_{i+1} gets the source message at least $d+r-1$ rounds after round t. Consequently, broadcasting in T^* takes time at least $k(d+r-1)\geqslant kd \in \Omega(D\Delta/r)$. □

5. Conclusion

Our results show a significant gap in time of broadcasting between networks with knowledge radius 0 and 1: from $\Theta(e)$ to $\Theta(\min(n, D^2\Delta))$. The comparison of the lower bound $\Omega(D^2)$ for knowledge radius 1 with the upper bound $O(D\log n)$ for knowledge radius $c\log^4 n$, in the case of networks of bounded degree, indicates another gap for some positive knowledge radius. An interesting open problem is to locate the position of this gap. More precisely, what is the minimum knowledge radius for which broadcasting can be carried out in all networks of bounded degree and diameter D in time $O(D\,\mathrm{polylog}\, n)$?

References

[1] S. Albers, M.R. Henzinger, Exploring unknown environments, Proceedings of the 29th Symposium on Theory of Computing, 1997, pp. 416–425.
[2] B. Awerbuch, A new distributed depth-first-search algorithm, Inform. Proc. Lett. 20 (1985) 147–150.
[3] B. Awerbuch, O. Goldreich, D. Peleg, R. Vainish, A tradeoff between information and communication in broadcast protocols, J. ACM 37 (1990) 238–256.

[4] R. Bar-Yehuda, O. Goldreich, A. Itai, On the time complexity of broadcast in multi-hop radio networks: an exponential gap between determinism and randomization J. Comput. Syst. Sci. 45 (1992) 104–126.
[5] B.S. Chlebus, L. Gąsieniec, A. Gibbons, A. Pelc, W. Rytter, Deterministic broadcasting in unknown radio networks, Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'2000), pp. 861–870.
[6] B.S. Chlebus, L. Gąsieniec, A. Östlin, J.M. Robson, Deterministic radio broadcasting, Proceedings of the 27th International Collaboration on Automata, Languages and Programming (ICALP'2000), Lecture Notes in Computer Science, Vol. 1853, Geneva, Switzerland, 2000, pp. 717–728.
[7] M. Chrobak, L. Gąsieniec, W. Rytter, Fast broadcasting and gossiping in radio networks, Proceedings of the 41st Symposium on Foundations of Computer Science (FOCS 2000), Redondo Beach, CA, U.S.A., 2000, pp. 575–581.
[8] A. Czygrinow, M. Hańćkowiak, M. Karoński, Distributed $O(\Delta \log n)$-edge-coloring algorithm, Proceedings of the Ninth Annual European Symposium Algorithms (ESA 2001), Lecture Notes in Computer Science, Vol. 2161, pp. 345–355.
[9] G. De Marco, A. Pelc, Faster broadcasting in unknown radio networks, Inform. Process. Lett. 79 (2001) 53–56.
[10] X. Deng, C.H. Papadimitriou, Exploring an unknown graph, Proceedings of the 31st Symposium on Foundations of Computer Science, 1990, pp. 356–361.
[11] K. Diks, E. Kranakis, D. Krizanc, A. Pelc, The impact of knowledge on broadcasting time in radio networks, in: Proceedings of the Seventh Annual European Sympsoium Algorithms, ESA'99, Lecture Notes in Computer Science, Vol. 1643, Prague, Czech Republic, 1999, pp. 41–52.
[12] D.Z. Du, F.H. Hwang, Combinatorial Group Testing and its Applications, World Scientific, Singapore, 1993.
[13] R.C. Entringer, P.J. Slater, Gossips and telegraphs, J. Franklin Inst. 307 (1979) 353–360.
[14] S. Even, B. Monien, On the number of rounds necessary to disseminate information, Proceedings of the First ACM Symposium on Parallel Algorithms and Architectures, June 1989, pp. 318–327.
[15] U. Feige, D. Peleg, P. Raghavan, E. Upfal, Randomized broadcast in networks, Random Struct. Algorithms 1 (1990) 447–460.
[16] P. Fraigniaud, E. Lazard, Methods and problems of communication in usual networks, Discrete Appl. Math. 53 (1994) 79–133.
[17] P. Fraigniaud, A. Pelc, D. Peleg, S. Perennes, Assigning labels in an unknown anonymous network with a leader, Distributed Comput. 14 (2001) 163–183.
[18] L. Gargano, A. Pelc, S. Perennes, U. Vaccaro, Efficient communication in unknown networks, Networks 38 (2001) 39–49.
[19] L. Gąsieniec, A. Pelc, Broadcasting with a bounded fraction of faulty nodes, J. Parallel Distributed Comput. 42 (1997) 11–20.
[20] S.M. Hedetniemi, S.T. Hedetniemi, A.L. Liestman, A survey of gossiping and broadcasting in communication networks, Networks 18 (1988) 319–349.
[21] J. Hromkovič, R. Klasing, B. Monien, R. Peine, Dissemination of information in interconnection networks (broadcasting and gossiping), in: Ding-Zhu Du, D. Frank Hsu (Eds.), Combinatorial Network Theory, Kluwer Academic Publishers, Dordrecht, MA, 1995, pp. 125–212.
[22] J. Janssen, D. Krizanc, L. Narayanan, S. Shende, Distributed online frequency assignment in cellular networks, J. Algorithms 36 (2000) 119–151.
[23] D.W. Krumme, Fast gossiping for the hypercube, SIAM J. Comput. 21 (1992) 365–380.
[24] D.W. Krumme, G. Cybenko, K.N. Venkataraman, Gossiping in minimal time, SIAM J. Comput. 21 (1992) 111–139.
[25] S. Kutten, D. Peleg, Fault-local distributed mending, Proceedings of the 14th ACM Symposium on Principles of Distributed Computing, 1995, pp. 20–27.
[26] N. Linial, Locality in distributed graph algorithms, SIAM J. Comput. 21 (1992) 193–201.
[27] P. Panaite, A. Pelc, Exploring unknown undirected graphs, J. Algorthims 33 (1999) 281–295.
[28] D. Peleg, Deterministic radio broadcast with no topological knowledge, 2000, unpublished.