Perfect Broadcasting in Unlabeled Networks

Krzysztof Diks * Evangelos Kranakis † Andrzej Pelc ‡

Abstract

We consider broadcasting a message from one node to all other nodes of an asynchronous totally unlabeled network: neither nodes nor links of the network have *a priori* assigned labels but they know the topology of the network. Nodes can send messages of arbitrary size and we are interested in minimizing the total number of messages. Broadcasting in an n-node unlabeled network is *perfect* if it uses $n-1$ messages (the minimum even in the labeled network). We show that the problem of deciding whether an arbitrary network admits perfect broadcasting from a given source, is NP-hard. We characterize regular networks in which perfect broadcasting from every node is possible, and give such broadcasting algorithms.

keywords: broadcasting, message complexity, unlabeled network.

* **Instytut Informatyki, Uniwersytet Warszawski, Banacha 2, 02-097 Warszawa, Poland. E-mail: diks@mimuw.edu.pl**
This work was done during the author's visit at the Université du Québec à Hull.
† **School of Computer Science, Carleton University, Ottawa, Ontario, K1S 5B6, Canada. E-mail: kranakis@scs.carleton.ca**
Research supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) grant.
‡ **Département d'Informatique, Université du Québec à Hull, Hull, Québec J8X 3X7, Canada. E-mail: Andrzej_Pelc@uqah.uquebec.ca**
Research supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

1 Introduction

The study of the computational power of anonymous networks is an important and well established research area in distributed computing (cf. [13] and the survey [8]). In an anonymous network, processors do not have distinct identities and execute identical algorithms. The impossibility of distinguishing processors yields symmetry in computations and restricts the power of the network both in terms of the class of functions that can be computed and the time and cost of computations. The study of anonymous networks was initiated in [1] and then pursued by many authors, both for specific networks, such as rings [2, 3, 11] and hypercubes [9] and for arbitrary networks [4, 10, 15]. Among the problems studied in this context are the following: Which functions a given anonymous network can compute? What is the message and bit complexity of such computation? Is it possible to perform leader election in the network?

An even more severe restriction on the amount of knowledge available to processors is the lack of *sense of direction* (cf. [12, 5]), when there is no "consistent" labeling of links. For rings, the sense of direction is equivalent to orientation and its computational aspects were studied, e.g., in [2]. Networks in which neither links nor nodes have *a priori* assigned labels, are obviously both anonymous and devoid of sense of direction. We will call such networks *totally unlabeled.*

In this paper we consider one of the fundamental tasks in distributed computing: *broadcasting.* (See, e.g., the survey [7] for references to the literature on broadcasting.) One node of the network, called the *source*, has a message which has to be transmitted to all other nodes. We assume that the network is totally unlabeled. This implies that a node sending a message to its neighbor knows neither its own identity nor that of the neighbor and cannot situate itself in the network. Moreover, a node sending a message does not know which neighbor will receive it. The only knowledge available to a node is the topology of the network, its own degree and an extra bit signifying "I am the source" (in case of the source) or "I am not the source" (in case of all remaining nodes). All other knowledge has to be acquired in the communication process. Nodes can send messages of arbitrary size and content, thus conveying already gained knowledge together with the source message. Only nodes that already got the source message can transmit. The network is asynchronous: every message sent on a link arrives to its destination in finite but unbounded time. Links are of FIFO type: messages sent on a given link arrive in the order of sending. All processors are identical and run the same algorithm.

We are interested in minimizing the number of messages needed to achieve broadcasting. In the naive algorithm every node relays the message to all neighbors except those from which it got it. Due to asynchronicity this may take as many as $2l-(n-1)$ messages in a network with n nodes and l links. On the other hand, the minimal number of messages is $n-1$, even in a labeled network. Thus we say that a broadcasting algorithm in an unlabeled network is *perfect* if it achieves this lower bound. We prove that, given a graph G and a source s, the problem of deciding whether there exists a perfect broadcasting in G from the source s is NP-hard. We do not know the complexity of the following (possibly easier) problem: given a graph G, does there exist a perfect broadcasting in G from every source? However, for regular graphs G, we precisely characterize graphs which have this property: these turn out to be *Moore graphs* (cf. [14]). Hence the latter decision problem restricted to regular graphs is polynomial.

The paper is organized as follows. In section 2 we give a precise definition of the model and of a broadcasting algorithm in a totally unlabeled network. Section 3 is devoted to the proof of the above mentioned NP-hardness result. In section 4 we characterize regular graphs admitting perfect broadcasting from every source. Section 5 contains conclusions and open problems.

2 The model

The network is represented as an undirected connected graph $G = (V, E)$ whose nodes are processors and edges are communication links. We assume that $|V| > 1$. For any node v, $I(v)$ denotes the set of edges incident to v and $d(v) = |I(v)|$ denotes the degree of v. All links from $I(v)$ are connected to v by distinct *ports*. Each port has two buffers: one for storing incoming and the other for storing outgoing messages. Before the start of broadcasting each node v knows the graph G, the degree $d(v)$ and knows if it is the source. There is no *a priori* labeling of links or nodes, i.e., v does not know which link from $I(v)$ joins it with which neighbor and where it is situated in G. For every v we fix an arbitrary local labeling of ports of v. For convenience, node v identifies its ports with respective links, i.e., this labeling is a bijection $\sigma_v : \{1, ..., d(v)\} \to I(v)$. It should be stressed that those labelings are chosen arbitrarily by all nodes and no "coherence" among them is assumed.

Messages placed in the outgoing buffer of a given port of v are sent to the neighbor joined with v by the respective link. Links are of FIFO type, i.e., messages transmitted through a given link arrive to the destination in the same order as they were placed in the buffer. The network is asynchronous: every message sent on a link arrives to its destination in finite but unbounded time. Whenever a message arrives at a node, it appears in the incoming buffer of the respective port and is received. We use two primitive operations:

$send_v(M, i)$ denotes the operation of sending message M by node v through port $\sigma_v(i)$.

$receive_v$ denotes the function which waits for the first incoming message at v and returns the couple (M, i), where M is the message and i is the port number at which this message came.

Let s denote the source of broadcasting, Δ - the maximum degree of G and $\Sigma = \{\sigma_v : v \in V\}$ - the set of fixed local labelings of ports. We want to formalize the assumption that nodes can send arbitrary messages and the decision of a node at any point of broadcasting can be based on the entire knowledge that it acquired so far. This current knowledge of a node v increases with each incoming message: it is the previous knowledge of v plus the newly obtained knowledge. What knowledge can be gained by getting a new message? The following three items can be learned:

- the current knowledge of the sender in the moment of sending (including its degree);
- the local port number on which the sender sent the message;
- the local port number on which the message was received.

We define a *history* in the graph G to be any finite sequence of pairs $[a_1, b_1]$,..., $[a_j, b_j]$, where $j > 0$ and $a_i, b_i \in \{\text{source}, \text{dest}, 1, 2, ..., \Delta\}$, for all $i \leq j$. Let $\mathcal{H}$ be the set of all histories for graph G. A *chooser* for graph G is any function

$$\phi : \mathcal{H} \rightarrow P(\{1, 2, ..., \Delta\}) \times \{stop, wait\}.$$

The role of the chooser is the following: for any node v and any current history (i.e., knowledge) available at this node, it chooses the (local numbers of) ports on which a message (containing everything v knows) will be sent. It also selects one of the states *stop* or *wait*. The first one indicates that all actions of v should be terminated after sending messages on the chosen ports - no further messages will be read by the processor even if they arrive at some ports in the future; the second state means that the node is still active and it will read the next incoming message.

For any node v the *initial history* of v is defined as follows: $IH_v = [source, d(v)]$ if $v = s$ and $IH_v = [dest, d(v)]$ if $v \neq s$. Thus the initial history at node v specifies only the degree of v and whether v is or is not the source.

Fix a chooser ϕ. For every node v, $\mathcal{A}_\phi(v)$ denotes the following algorithm executed at v:

begin
  $CH_v := IH_v$; //initialization of current history//
  repeat
    $(X, Y) := \phi(CH_v)$;
    for all $i \in X$ **do**
      $send_v([CH_v, i], i)$; //the message containing the current history
      of v and the port number i on which it is sent, is sent on
      port i, for all i chosen by the chooser//
    if $Y = wait$ **then**
    begin
      $([H, out], in) := receive_v$;//a message is received on port in//
      $CH_v := CH_v \bullet H \bullet [out, in]$; //current history is updated
      by concatenating to its previous value the history obtained in the new
      message and the information about the local port numbers on which it was
      sent and received//
    end
  until $Y = stop$
end.

A *protocol* for the graph G with chooser ϕ is the family $\mathcal{A}_\phi = \{\mathcal{A}_\phi(v) : v \in V\}$. A protocol $\mathcal{A}_\phi$ is a *broadcasting algorithm* from source s iff the following conditions are satisfied:

1. $\phi(IH_v) = (X, Y)$, where $\emptyset \neq X \subset \{1, ..., d(v)\}$ and $s = v$, or $X = \emptyset$, $Y = wait$ and $v \neq s$ (in the beginning the source has to send a message to some neighbor, while all other nodes have to wait and cannot send anything (before obtaining the first message);

2. $\phi(CH_v) = (X, Y)$, where $X \subset \{1, ..., d(v)\}$ (for any current history at any node the chooser chooses a subset of available ports to send messages);
3. for any set of local labelings Σ every algorithm $\mathcal{A}_\phi(v)$ eventually assignes value *stop* to the variable Y.

Condition 3 is crucial in the above definition. In the beginning $Y = wait$ in all nodes except possibly the source. Hence the requirement that value *stop* be assigned means that each node has received a message and each node has finished its work. The requirement that this should happen for all labelings captures the full anonymity of broadcasting: eventually all nodes get the message, although at each transmission the choice to which neighbor a node sends a message is controled by the adversary. In an actual implementation of a broadcasting algorithm the original source message is appended to each "history" message specified above.

The *message complexity* of a broadcasting algorithm $\mathcal{A}_\phi$ for a given set Σ of local labelings, is the number C_Σ of operations *send* performed during the execution of the algorithm. The *message complexity* of $\mathcal{A}_\phi$ is the maximum of numbers C_Σ, over all sets Σ of local labelings. The algorithm $\mathcal{A}_\phi$ is *perfect* if its complexity is $|V| - 1$.

The descriptions of algorithms in the rest of the paper will be informal: for reasons of clarity of presentation we will not define the chooser formally. Indeed, in most cases, a node does not need to know the entire history to decide on which ports it should send messages. Likewise, it usually does not need to send its entire history but only some crucial facts that can be deduced from it. Here are some examples of such facts: "my degree is 5", "I got a message from the source via a path of length 4 from a neighbor of degree 3". Nodes can also send conditional orders to neighbors. Formally such orders are encoded in the chooser, e.g. "if your degree is 1 then stop, otherwise send a message to one neighbor and tell it to stop". In particular, each node knows from its history on which ports it already got or sent messages. Call all other ports of a node *free*. Thus the instruction "send a message on all your free ports" is legitimate and will be often used. We will use the phrase "a node stops after getting a message" if the value of the chooser is $(\emptyset, stop)$, i.e., when a node does not perform any further actions after getting this message. This way of describing algorithms in totally unlabeled networks better conveys their main idea and it will be clear in each case how to transform such an informal description into chooser definition.

3 Perfect broadcasting in arbitrary graphs

Consider the following decision problem:

PERFECT BROADCASTING (PB)

Input: connected graph $G = (V, E)$ and a node $s \in V$.

Question: does there exist a perfect broadcasting algorithm in G from source s?

The following decision problem is known to be NP-hard (cf. [6]):

EXACT COVER BY 3-SETS (X3C)

Input: set X with $|X| = 3m$, for some positive integer m, and a family $\mathcal{F}$ of 3-element subsets of X.

Question: does $\mathcal{F}$ contain an exact cover of X, i.e., a subfamily $\mathcal{F}' \subset \mathcal{F}$ such that every element of X belongs to exactly one set from $\mathcal{F}'$?

Theorem 3.1 *The problem* **PERFECT BROADCASTING** *is NP-hard.*

Proof: We will reduce the problem **X3C** to **PB**. Consider an instance of the former problem in which $X = \{x_1, ..., x_{3m}\}$, $\mathcal{F} = \{F_1, ..., F_k\}$. For any such pair $(X, \mathcal{F})$, let $G_{X,\mathcal{F}} = (V, E)$ be the graph defined as follows:

$$V = V_1 \cup V_2 \cup V_3 \cup V_4 \cup \{s\},$$

where $V_1 = \{x_1, ..., x_{3m}\}$, $V_2 = \{F_1, ..., F_k\}$,$V_3 = \{y_1, ..., y_k\}$, $V_4 = \{z_1^1, z_2^1, z_2^2, ..., z_k^1, z_k^2, ..., z_k^k\}$, and

$$E = E_1 \cup E_2 \cup E_3 \cup E_4,$$

where $E_1 = \{\{x_i, F_j\} : i = 1, ..., 3m; j = 1, ..., k; x_i \in F_j\}$, $E_2 = \{\{F_i, y_i\} : i = 1, ..., k\}$, $E_3 = \{\{y_i, z_i^j\} : i = 1, ..., k; j = 1, ..., i\}$, $E_4 = \{\{s, y_i\} : i = 1, ..., k\}$.

Example

Instance of **X3C**: $X = \{x_1, ..., x_9\}$, $\mathcal{F} = \{F_1, ..., F_5\}$, where $F_1 = \{x_1, x_2, x_3\}$, $F_2 = \{x_3, x_4, x_5\}$, $F_3 = \{x_4, x_5, x_6\}$, $F_4 = \{x_5, x_7, x_8\}$, $F_5 = \{x_7, x_8, x_9\}$, and corresponding graph $G_{X,\mathcal{F}}$ (cf. Figure 1).

We will now prove that for any pair $(X, \mathcal{F})$ as above, $\mathcal{F}$ contains an exact cover of X, if and only if, there exists a perfect broadcasting algorithm in $G_{X,\mathcal{F}}$ from source s.

Let $\mathcal{F}'$ be an exact cover of X. Below we give the description of a perfect broadcasting in $G_{X,\mathcal{F}}$ from source s.

1. s sends a message to every neighbor and then stops;
2. neighbors of s send a message on each free port and then stop //these are nodes y_i//;
3. neighbors of y_i (for $i = 1, ..., k$) of degree 1 stop after getting the message //these are nodes z_i^j//;
4. the remaining neighbor F_i of y_i (for each $i = 1, ..., k$) does the following:
   if $F_i \notin \mathcal{F}'$ **then** F_i stops after getting the message
   else F_i sends a message on each free port and then stops
   //a node F_i deduces whether it belongs to $\mathcal{F}'$ from the degree of node y_i from which it got the message. Indeed, this degree equals $i + 2$//.
5. all three neighbors that got a message from $F_i \in \mathcal{F}'$ stop after getting the message.

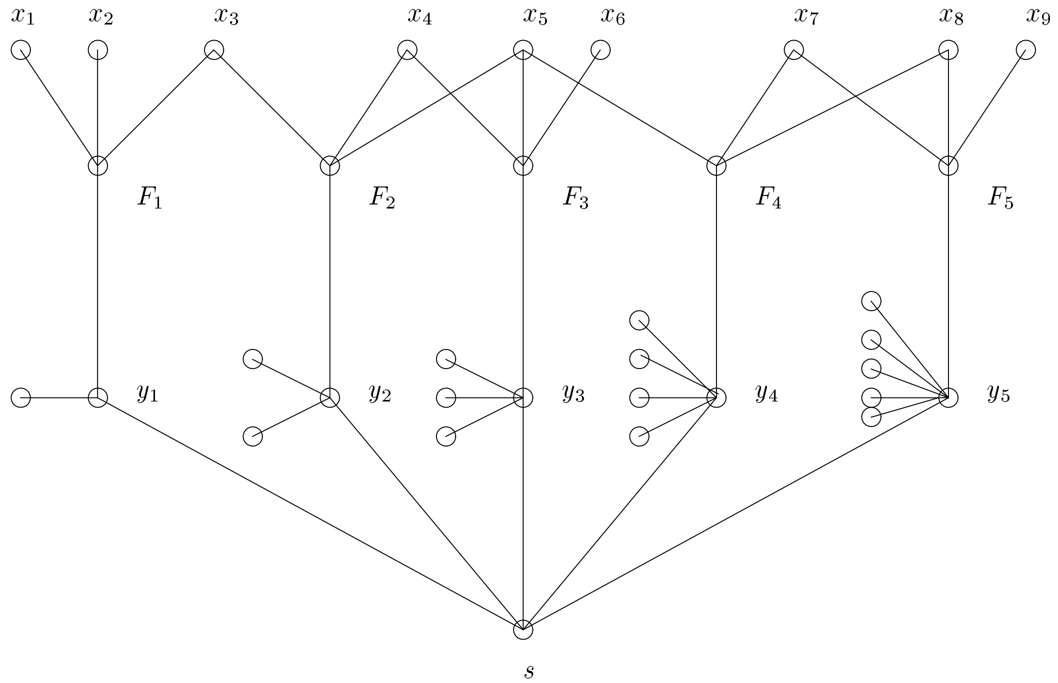


Figure 1: Graph $G_{X,\mathcal{F}}$. The vertices of degree 1 adjacent to y_i are numbered z_i^j, $1 \leq j \leq i$.

Since $\mathcal{F}'$ is an exact cover of X, every node x_j gets a message from exactly one node F_i and hence broadcasting is perfect.

Now suppose that there exists a perfect broadcasting in $G_{X,\mathcal{F}}$ from source s. Let $\mathcal{A}_\phi$ be such a broadcasting. Every node, except s must get exactly one message. We first prove the following two claims:

Claim 1. Every node x_i must stop after getting the message.

Proof: Suppose not and let x_i be a node that sends a message to its neighbor. Let F_j be the neighbor that received this message. Consider two cases.

1. F_j stops after getting the message.
The node y_j can get the message only from the source. It is also the only node node from which nodes z_j^l, $l = 1, ..., j$ can get a message. Hence y_j must send messages. There exists a local labeling of ports of y_j such that the chooser indicates F_j as a neighbor to which y_j sends a message. Thus F_j gets two messages, from x_i and from y_j. Contradiction.

2. F_j sends at least one message.
There exists a local labeling of ports of F_j such that the chooser indicates y_j as a neighbor to which F_j sends a message. As before, y_j must send messages. Now there exists a local labeling of ports of y_j such that the chooser indicates s as a neighbor to which y_j sends a message. Contradiction.

Claim 2. Every node F_j either does not send any messages or sends messages on all free ports.

Proof: Suppose not and let F_j be a node that sends messages on some but not all free ports. By Claim 1, F_j gets a message from y_j and only from this node. Fix a neighbor x_i of F_j. There exists a local labeling of ports of F_j under which x_i does not get a message from F_j. Hence some other neighbor F_l of x_i must send messages on some (or all) its free ports, in order for x_i to get the source message. Thus there exist local labelings of ports of F_j and F_l such that the chooser indicates x_i as the node that gets the message both from F_j and F_l. Contradiction.

Let $\mathcal{F}'$ be the family of those nodes F_j that send messages. Elements of $\mathcal{F}'$ are pairwise disjoint, for otherwise a common element x_i would get messages from two neighbors. Since all nodes in X must get a message, the union of all sets in $\mathcal{F}'$ must cover X. This implies that $\mathcal{F}'$ is an exact cover of X. □

Consider the following decision problem closely related to **PB**.

UNIVERSAL PERFECT BROADCASTING (UPB)

Input: connected graph $G = (V, E)$.

Question: does there exist a perfect broadcasting algorithm in G from every source?

UPB is not harder than **PB**, as there is an obvious reduction from **UPB** to **PB**. We do not know what is the complexity of **UPB**; however, the result from the next section implies in particular that **UPB** is polynomial when restricted to regular graphs. It also remains open whether **PB** is polynomial for regular graphs: the possibility of varying degrees was crucial in our reduction.

4 Perfect broadcasting in regular graphs

In this section we give an exact characterization of regular graphs in which perfect broadcasting can be done from every node and show such a broadcasting whenever it exists.

4.1 Terminology and preliminaries

The *girth* of a graph G is the length of the shortest cycle in G. Assume $d \geq 2$ and $g \geq 3$. A regular graph of degree d and girth g will be called a (d,g)-*graph*. For given d and g, a (d,g)-graph with the minimal possible number of nodes is called a (d,g)-*cage*, cf. [14].

For $d = 2$ and any $g \geq 3$, a (d,g)-cage is the cycle of length g.
For $g = 3$ and any $d \geq 3$, a (d,g)-cage is the clique of size $d+1$.
For $g = 4$ and any $d \geq 3$, a (d,g)-cage is the complete bipartite graph $K_{d,d}$.

Suppose that $d \geq 3$ and $g \geq 5$. Let $f(d,g)$ denote the number of nodes of a (d,g)-cage.

Theorem 4.1 *[14] Let*

$$f_0(d,g) = \begin{cases} \frac{d(d-1)^r-2}{d-2} & \text{if } g = 2r+1 \\ \frac{2(d-1)^r-2}{d-2} & \text{if } g = 2r \end{cases}$$

Then $f(d,g) \geq f_0(d,g)$.

If $f(d,g) = f_0(d,g)$, a (d,g)-cage is called a *Moore graph* or a *minimal* (d,g)*-graph*. The existence of (d,g)-cages, for all $d \geq 2$ and $g \geq 3$, was proved by Erdös and Sachs (cf. [14]). However, minimal (d,g)-graphs exist only for some values of d and g.

Theorem 4.2 *[14] A minimal* (d,g)*-graph exists only if (i)* $g = 5$ *and* $d = 3, 7$ *or (possibly) 57, or (ii)* $g = 6, 8, 12$.

4.2 Regular graphs admitting perfect broadcasting

The following is the main result of this section.

Theorem 4.3 *For every* $d \geq 2$, *a regular graph* G *of degree* d *admits perfect broadcasting from every node, if and only if, either*

- $d = 2$ *(*G *is a cycle), or*
- $d \geq 3$ *and* G *is either the clique* K_{d+1} *or the complete bipartite graph* $K_{d,d}$, *or*
- $d \geq 3$ *and* G *is a minimal* (d,g)*-graph, for some* $g \geq 5$.

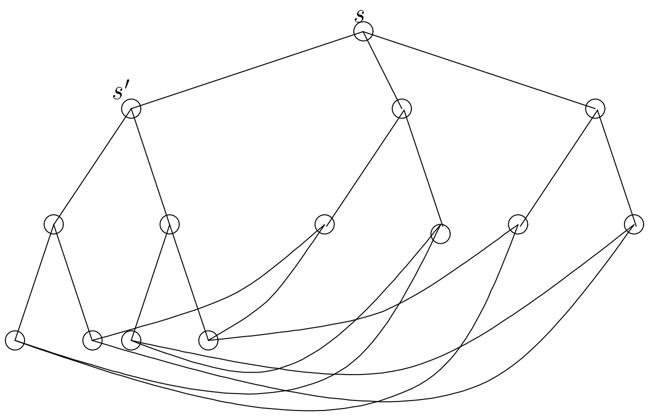
Proof: We first show a perfect broadcasting algorithm for all graphs specified in the theorem.

• $d = 2$ (cycle of length g).
The source message is sent on the cycle in one direction, together with the information how many nodes have already been visited. When this number becomes $g-1$ the next node getting the message stops.

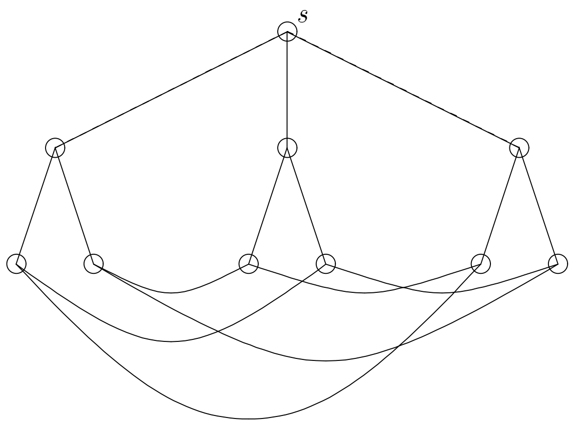
• $d \geq 3$ and G is the clique K_{d+1}.
The source sends a message to all neighbors and then stops. All other nodes stop after getting a message.

• $d \geq 3$ and G is the complete bipartite graph $K_{d,d}$.
The source sends messages to all neighbors and then stops. The message sent on port 1 contains the order "send a message on all free ports telling recipients to stop and then stop yourself". Messages sent on all other ports contain the order "stop". Other nodes execute orders.

• $d \geq 3$, $g \geq 5$ and G is a minimal (d,g)-graph.
Let s be any source in G. Consider two cases.



(a)



(b)

Figure 2: Broadcasting in a minimal (d, g)-graph. Straight (respectively, curved) lines represent edges of the broadcasting tree (respectively, remaining edges of the graph). In figure (a) $d = 3, g = 6$; in figure (b) $d = 3, g = 5$.

Case 1. The girth of G is even, $g = 2r$.
Here is the description of a perfect broadcasting from s (cf. Figure 2 (a)):

1. s sends a message on port 1 with the information "you are on level 0" //The neighbor s' that gets this message becomes a "twin source"//;
2. both s and s' send messages on all free ports with information "you are on level 1";
3. any node that got a message "you are on level i", for $i = 1, ..., r-2$, sends a message on all free ports with information "you are on level $i+1$" and then stops; any node that got a message "you are on level $r-1$", stops.

Since $g = 2r$, no pair of nodes on levels $i < r-1$ have common neighbors. Thus every node gets at most one message. On the other hand, upon completion of the above algorithm, the number of nodes that get the source message is

$$2 + 2(d-1) + 2(d-1)^2 + ... + 2(d-1)^{r-1} = 2\frac{(d-1)^r - 1}{d-2} = f_0(d, g),$$

i.e., all nodes get it.

Case 2. The girth of G is odd, $g = 2r + 1$.
The algorithm is similar as above. The source s sends messages on all free ports with information "you are on level 1" and then step 3 is executed, until level r instead of $r-1$ (cf. Figure 2 (b)). A similar argument as before proves that every node gets at most one message and the number of informed nodes is

$$1 + d + d(d-1) + ... + d(d-1)^{r-1} = \frac{d(d-1)^r - 2}{d-2} = f_0(d, g),$$

i.e., all nodes get the message.

We have shown that all graphs specified in the theorem admit perfect broadcasting from any node. Now we prove the opposite implication. Assume $d \geq 3$ and $g \geq 5$. (For $d = 2$ G is a cycle, for $g = 3$ G is a complete graph, and for $g = 4$ G is a complete bipartite graph.) Suppose that perfect broadcasting from every source is possible in a regular graph $G = (V, E)$ and let $\mathcal{A}_\phi$ be such a broadcasting algorithm. Since every node other than the source gets exactly one message and the source only sends but does not receive messages, the set of edges on which messages are transmitted in the execution of $\mathcal{A}_\phi$ yields a spanning tree of G rooted at the source s of broadcasting and with edges directed from the root towards leaves. Such a tree will be called *the broadcasting tree* of $\mathcal{A}_\phi$. For a given source, this tree depends on the algorithm $\mathcal{A}_\phi$ and on the set of local labelings $\Sigma = \{\sigma_v : v \in V\}$. Fix an algorithm $\mathcal{A}_\phi$ and denote by $T_{\Sigma,s}$ the tree corresponding to $\mathcal{A}_\phi$ when s is the source and Σ is the set of local labelings. For any node v in G denote by T_v the (maximal) subtree of $T_{\Sigma,s}$ rooted at v. In particular, $T_s = T_{\Sigma,s}$.

We will use the following properties of $T_{\Sigma,s}$.

Claim 1. The number of children of s in $T_{\Sigma,s}$ is either 1 or d.

Proof: Suppose not and let $u_1, ..., u_l$, for $l > 1$, be neighbors of s in G to which s sends a message and let u be a neighbor of s in G to which s does not send a message. Since u must get the source message, it is in one of the subtrees T_{u_i}, for $i = 1, ..., l$. Let j be any element of $\{1, ..., l\}$ different from i and consider a local labeling σ'_s of ports of s which interchanges values of $\sigma_s(\{s, u\})$ and $\sigma_s(\{s, u_j\})$ leaving all remaining values as in σ_s. (As mentioned in section 2, for a fixed node - s in this case - ports are identified with edges.) Let $\Sigma' = \Sigma \setminus \{\sigma_s\} \cup \{\sigma'_s\}$ and consider the execution of algorithm $\mathcal{A}_\phi$ for the set of local labelings Σ'. Since $\mathcal{A}_\phi$ is a perfect broadcasting, the tree T_{u_i} will be constructed as before. However, node u which is in this tree gets the second message directly from s, contradiction.

Claim 2. If a node v is not a leaf in $T_{\Sigma,s}$ and has less than $d - 1$ children, then all its neighbors in G, except its parent, are in T_v.

Proof: Suppose not. Let u be a neighbor of v in G other than its parent in $T_{\Sigma,s}$ and outside of T_v. Let w be the closest ancestor of v in $T_{\Sigma,s}$ such that u is in T_w. Consider a local labeling σ'_v of ports of v which interchanges values of $\sigma_v(\{v, u\})$ and $\sigma_v(\{v, u'\})$, where u' is one of the children of v, and leaves all remaining values as in σ_v. Let $\Sigma' = \Sigma \setminus \{\sigma_v\} \cup \{\sigma'_v\}$ and consider the execution of algorithm $\mathcal{A}_\phi$ for the set of local labelings Σ'. Node u will get two messages: via a path in T_w and directly from v. Contradiction.

Claim 3. The sum of lengths of any two edge-disjoint paths (one of which may be empty) from any node x to leaves of T_x, is at most $g - 1$. In particular, the height of $T_{\Sigma,s}$ is at most $g - 1$.

Proof: Suppose not. Let $v_0 = x, v_1, ..., v_{l_1}$ and $u_0 = x, u_1, ..., u_{l_2}$ be such paths and assume $l_1 + l_2 \geq g$. W.l.o.g. assume that $l_1 \geq l_2$. Let l be the depth of x in $T_{\Sigma,s}$ and let $s = y_0$, y_1 ,..., $y_l = x$ be the path in $T_{\Sigma,s}$ from s to x. Let $y_{l+1} = v_1$. Let $w_0, w_1, ..., w_{g-1}, w_0$ be consecutive nodes of a cycle of length g in G. If $l > g - 2$, we consider a set of local labelings Σ' in which

$$\sigma'_{w_0}(\{w_0, w_1\}) = \sigma_{y_0}(\{y_0, y_1\}), \sigma'_{w_1}(\{w_1, w_0\}) = \sigma_{y_1}(\{y_1, y_0\}), ...,$$

$$\sigma'_{w_{g-2}}(\{w_{g-2}, w_{g-1}\}) = \sigma_{y_{g-2}}(\{y_{g-2}, y_{g-1}\}), \sigma'_{w_{g-1}}(\{w_{g-1}, w_{g-2}\}) = \sigma_{y_{g-1}}(\{y_{g-1}, y_{g-2}\}),$$

$$\sigma'_{w_{g-1}}(\{w_{g-1}, w_0\}) = \sigma_{y_{g-1}}(\{y_{g-1}, y_g\}).$$

If w_0 is the source and Σ' the set of local labelings, w_0 will get a message in the execution of $\mathcal{A}_\phi$. Contradiction.

Assume $l \leq g - 2$. If $l_2 = 0$ the argument is as above, so assume $l_2 > 0$. Since $d \geq 3$, there exists a node z and a simple path $z_0 = z$, z_1, ..., $z_l = w_0$ in which every node is different from w_1 and w_{g-1}. Consider a set of local labelings Σ'' in which

$$\sigma''_{z_i}(\{z_i, z_{i+1}\}) = \sigma_{y_i}(\{y_i, y_{i+1}\}), \sigma''_{z_{i+1}}(\{z_{i+1}, z_i\}) = \sigma_{y_{i+1}}(\{y_{i+1}, y_i\}),$$

for $i = 0, ..., l - 1$, and

$$\sigma''_{w_0}(\{w_0, w_1\}) = \sigma_{v_0}(\{v_0, v_1\}), \sigma''_{w_1}(\{w_1, w_0\}) = \sigma_{v_1}(\{v_1, v_0\}), ...,$$

$$\sigma''_{w_{l_1-1}}(\{w_{l_1-1}, w_{l_1}\}) = \sigma_{v_{l_1-1}}(\{v_{l_1-1}, v_{l_1}\}),$$

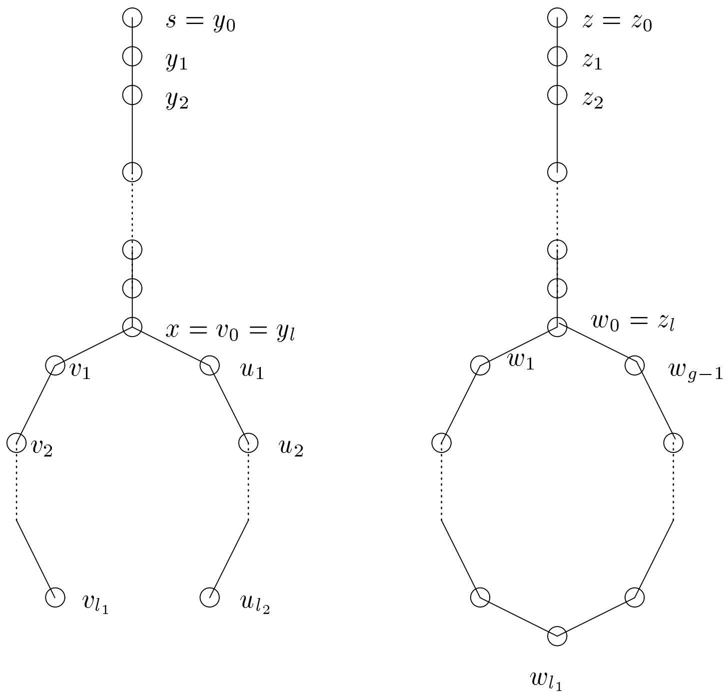


Figure 3:

$$\sigma''_{w_0}(\{w_0, w_{g-1}\}) = \sigma_{v_0}(\{v_0, u_1\}), \sigma''_{w_{g-1}}(\{w_{g-1}, w_0\}) = \sigma_{u_1}(\{u_1, v_0\}), ...,$$
$$\sigma''_{w_{l_1+1}}(\{w_{l_1+1}, w_{l_1}\}) = \sigma_{u_{g-l_1-1}}(\{u_{g-l_1-1}, u_{g-l_1}\}).$$

If z is the source and Σ'' the set of local labelings, w_{l_1} will get two messages: via path w_0, w_1 ,..., w_{l_1-1}, w_{l_1} and via path w_0, w_{g-1}, ..., w_{l_1+1}, w_{l_1}, (cf. Figure 3). Contradiction.

Claim 4. Every node v in $T_{\Sigma,s}$ which is different from s and is not a leaf, must have exactly $d-1$ children.

Proof: Suppose not. All neighbors of v in G that are neither its parent nor its children in $T_{\Sigma,s}$ have to be in T_v, in view of Claim 2. Since the height of T_v is smaller than $g-1$, this yields a cycle in G of length smaller than g. Contradiction.

Consider any perfect broadcasting $\mathcal{A}_\phi$. If the source has only 1 child s', this child must have at least one child t. In this case the algorithm $\mathcal{A}_\phi$ can be modified to an algorithm in which the source (s') has at least two children (s and t) one of which (s) stops after getting the message and all the others behave as before. This modified algorithm is also a perfect broadcasting and in this algorithm the source has more than 1 child. Hence, by Claim 1, we may assume that the source has d children. Denote by h the height of $T_{\Sigma,s}$. Since the girth of G is g, we have $h \geq \frac{g-1}{2}$. If $T_{\Sigma,s}$ is a complete $(d-1)$-ary tree, we have $h = \frac{g-1}{2}$ by Claim 3. By Theorem 4.1, G is then a minimal (d, g)-graph with odd g.

Hence, from now on, assume that $T_{\Sigma,s}$ is not a complete $(d-1)$-ary tree. Consider any path from the source to a leaf on level h of $T_{\Sigma,s}$ and let $z_0 = s$, ..., z_p be consecutive nodes on this path, such that none of the trees T_{z_0}, T_{z_1}, ..., $T_{z_{p-1}}$ is a complete $(d-1)$-ary tree but T_{z_p} is such a tree. Let h_i be the height of tree T_{z_i}. Thus $h = h_0$. For every $i = 1, ..., p$, let z_i^1, ..., z_i^{d-1} be siblings of z_i in $T_{\Sigma,s}$ (cf. Claim 4).

Claim 5.

1. For every $i = 0, ..., p-1$, we have $h_i > \frac{g-1}{2}$.
2. For every $i = 1, ..., p$, all trees $T_{z_i^j}$, $j = 1, ..., d-1$, are complete $(d-1)$-ary trees of the same height $g - 3 - h_i$.
3. For every $i = 1, ..., p$, the neighbors in G of every leaf of $T_{z_i^j}$, $j = 1, ..., d-1$ different from its parent in $T_{\Sigma,s}$ are leaves of level h_i in T_{z_i}.

Proof: We first show that $h_0 > \frac{g-1}{2}$ and all trees $T_{z_1^j}$, $j = 1, ..., d-1$, are complete $(d-1)$-ary trees of height $g - 3 - h_1$. Suppose that $h_0 \leq \frac{g-1}{2}$. Since T_{z_0} is not a complete $(d-1)$-ary tree, there exists a leaf x in T_{z_0} whose depth d in $T_{\Sigma,s}$ is less than h_0. Let y be a neighbor of x in G different from the parent of x in $T_{\Sigma,s}$. The depth d' of y is at most h_0. Hence the edge joining x and y closes a cycle of length at most $d + d' + 1 < g$, contradiction.

Hence $h_0 > \frac{g-1}{2}$. It follows that the height of each of the trees $T_{z_1^j}$, $j = 1, ..., d-1$, must be less than $\frac{g-1}{2} - 1$. Hence no pair of leaves in $T_{z_1^j}$ can be adjacent in G, as this would yield a cycle of length less than g. Hence all leaves from $T_{z_1^j}$ must have all their neighbors in T_{z_1}. Notice that all leaves in $T_{z_1^j}$ must be on the same level. (Otherwise, if two leaves x and y were on different levels, say x on a smaller level than y, it would be possible to find a set local labelings in which T_{z_1} would remain unchanged but x would have children. Its neighbors in T_{z_1} would get two messages, contradiction.) Let k_j be the height of $T_{z_1^j}$. We have $h_0 + 1 + k_j \leq g - 1$, by Claim 3, and $h_0 + 1 + k_j \geq g - 1$ because otherwise we would have a cycle of length less than g. Hence $h_0 + 1 + k_j = g - 1$ which implies $k_j = g - 3 - h_1$. Thus leaves in trees $T_{z_1^j}$ are adjacent (in G) only to leaves from level h_1 in T_{z_1}. This concludes the proof of Claim 5 for $i = 0$. The proof for larger i is similar.

We now consider two cases depending on the parity of g.

Case 1. $g = 2r$

We have $h_p + 1 > \frac{g-1}{2} = r - \frac{1}{2}$, hence $h_p + 1 \geq r$. By Claim 3, $h_p = r - 1$. Hence the height of $T_{z_p^j}$ is $g - 3 - h_p = r - 2$. By Claim 5, the height of $T_{z_i^j}$ is by one less than the height of $T_{z_{i+1}^j}$, for $i = 1, ..., p-1$. It follows that the number of nodes in $T_{\Sigma,s}$ corresponds to the lower bound from Theorem 4.1 and edges of G that are not in the tree join only leaves. Hence G is a minimal (d, g)-graph with even g (cf. Figure 4).

Case 2. $g = 2r + 1$

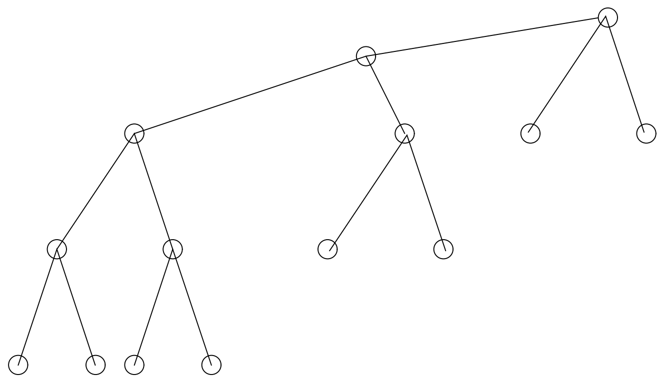


Figure 4: Another broadcasting tree for the minimal $(3, 6)$-graph.

Since $h_{p-1} = h_p + 1 > \frac{g-1}{2}$, we have $h_p \geq r$ and, in view of Claim 3, $h_p = r$. Hence the height of $T_{z_p^j}$ is $g - 3 - h_p = r - 2$. By Claim 5, the height of $T_{z_i^j}$ is by one less than the height of $T_{z_{i+1}^j}$, for $i = 1, ..., p-1$. It follows that if we consider the tree $T_{\Sigma,s}$ as rooted in z_p we will get a complete $(d-1)$-ary tree of height r. Edges of G that are not in this tree join only leaves. Thus G is a minimal (d, g) graph with odd g (cf. Theorem 4.1). □

Both the degree and the girth of a graph can be found in time $O(n^2 d)$, hence this time is sufficient to check if a regular graph is a minimal (d, g)-graph. Thus our characterization implies that the problem **UPB** described in section 3 is polynomial when restricted to regular graphs.

5 Conclusion

We investigated broadcasting in networks in which neither nodes nor links have *a priori* assigned labels and, consequently, the choice of the neighbor to which a node sends a message is controled by the adversary. We showed that, in spite of this restriction of knowledge available to processors, broadcasting can be sometimes achieved efficiently. We were interested in message complexity of the broadcasting process, assuming worst case behavior of the adversary. In particular we showed for which regular networks broadcasting can be done as efficiently as when the network is fully labeled.

On the other side of the spectrum, it is not hard to find examples of graphs in which lack of *a priori* labeling of links and nodes is very harmful for the efficiency of broadcasting. Take a complete graph on n nodes and add to each node a separate new neighbor of degree 1. The resulting $2n$-node graph has the property that (under the unlabeled scenario) broadcasting from each of its nodes must use $\Theta(n^2)$ messages, in fact at least one message must travel on every link.

Thus, for any network topology, it is important to know how vulnerable it is to knowledge restriction caused by lack of *a priori* labeling, from the point of view of communication tasks. One of the interesting open problems in this context is to find good broadcasting algorithms for important classes of totally unlabeled networks, i.e., algorithms whose message complexity is optimal or at least close to optimal. Further directions of research might concern message complexity of other communication tasks, such as all-to-all communication or voting, in the context of totally unlabeled networks.

References

[1] D. Angluin, Local and Global Properties in Networks of Processors, Proc. 12th Annual ACM Symposium on Theory of Computing (1980), 82-93.

[2] H. Attiya, M. Snir and M. Warmuth, Computing on an Anonymous Ring, Journal of the ACM, 35 (1988), 845-875.

[3] H. Attiya and M. Snir, "Better Computing on the Anonymous Ring, Journal of Algorithms 12 (1991), 204-238.

[4] P. Boldi, S. Shammah, S. Vigna, P. Gemmell, B. Codenotti, and J. Simon, Symmetry Breaking in Anonymous Networks: Characterizations, Proc. 4th Israel Symposium on Theory of Computing and Systems (1996), 16-26.

[5] P. Flocchini, B. Mans and N. Santoro, "Sense of Direction: Formal Definitions and Properties", Proc. 1st Int. Conf. on Structural Information and Communication Complexity (1994), 9-34, Carleton University Press, 1995.

[6] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman and Co. 1979.

[7] S.M. Hedetniemi, S.T. Hedetniemi and A.L. Liestman, A survey of Gossiping and Broadcasting in Communication Networks, Networks 18 (1988), 319-349.

[8] E. Kranakis, Symmetry and Computability in Anonymous Networks: A Brief Survey, Proc. 3rd Int. Conf. on Structural Information and Communication Complexity (SIROCCO'96), Carleton University Press, 1997, 1-16.

[9] E. Kranakis and D. Krizanc, Distributed Computing on Anonymous Hypercube Networks, Journal of Algorithms 23 (1997), 32-50.

[10] E. Kranakis, D. Krizanc and J. van der Berg, Computing Boolean Functions on Anonymous Networks, Information and Computation, 114 (1994), 214-236.

[11] S. Moran and M. K. Warmuth, Gap Theorems for Distributed Computation, Proc. 5th ACM Symposium on Principles of Distributed Computing (1986), 131-140.

[12] N. Santoro, Sense of Direction, Topological Awareness, and Communication Complexity, ACM SIGACT News 16 (1984), 50-56.

[13] G. Tel, Introduction to Distributed Algorithms, Cambridge University Press, 1996.

[14] P.-K. Wong, Cages - A Survey, Journal of Graph Theory 6 (1982), 1-22.

[15] M. Yamashita and T. Kameda, Computing on an Anonymous Network, Proc. 7th Annual ACM Symposium on Principles of Distributed Computing (1988), 117 - 130.