Efficient Broadcasting Protocols on the de Bruijn and similar Networks

Jean-Claude Bermond, Stéphane Perennes. *

I3S, C.N.R.S-U.R.A 1376, *Université de Nice-Sophia Antipolis*
930 Route des Colles- B.P 145 F - 06903 Sophia Antipolis
e-mails: bermond@unice.fr, sp@unice.fr

Abstract. Broadcasting is an information dissemination process in which a message is to be sent from a single originator to all members of a network by placing calls over the communication lines of the network. This is to be completed as quickly as possible subject to the constraints that each call involves only two vertices, each call requires one unit of time, a vertex can participate in only one call per unit of time, and a vertex can only call a vertex to which it is adjacent. The determination of the broadcast time has been done for several networks. Here we give new protocols which improve the known results for the de Bruijn network. The ideas can also be used for similar networks; we give as example the case of the Kautz network.

1 Introduction and Notations

Broadcasting (also called One to All) refers to the process of message dissemination in a communication network, whereby a message, originated by one node (called the *originator*) is transmitted to all the nodes of the network. This is to be completed as quickly as possible according to the considered model (see the surveys [4, 8]). Here we suppose that we are in the Store and Forward model, F_1, with constant time, which is also called *telephone model*. That is broadcasting is performed by placing a series of calls; a call involves only two vertices (the *sender* and the *receiver*); each call requires *one unit of time*; a vertex can be involved in only *one call* per unit of time; and calls occur between *adjacent nodes*.

Given a connected graph G and a message originator, vertex u, the broadcast time of u, denoted $b(u)$, is the minimum number of time units required to complete broadcasting from u. The broadcast time of the graph G, $b(G)$, is defined as the maximum of $b(u)$ taken over all the vertices u in G.

For any graph G with N vertices, $b(G) \geq \lceil \log_2(N) \rceil$, since the number of informed vertices can at most double during each unit of time. The problem of determining the value $b(G)$ is known to be NP-HARD (see [5]), but the values of $b(G)$ are known for many usual interconnection networks (like grids and hypercubes). However for the de Bruijn and related networks, the order of $b(G)$ is still to be found.

* This work has been supported by the French action RUMEUR of the GDR/PRC PRS and by the European HCM project MAP.

Let us recall the definition and basic properties of the de Bruijn and Kautz networks (see [3, 13] for more details).

The de Bruijn digraph $\mathcal{B}(d,D)$ of out-degree d and diameter D has as vertices the words of length D on an alphabet of d letters. Vertex $x_1 \dots x_D$ is joined by an arc to the vertices $x_2 \dots x_D \alpha$ where α is any letter from the alphabet. Between any pair of vertices $x_1 \dots x_D$ and $y_1 \dots y_D$ there exists a unique path of length D called *canonical* path:

$$x_1x_2x_3 \dots x_D,\ x_2x_3 \dots x_D y_1,\ x_3 \dots x_D y_1 y_2,\ \dots\dots, x_D y_1 \dots y_{D-1},\ y_1 \dots y_D$$

The Kautz digraph of out-degree d and diameter D, denoted $\mathcal{K}(d,D)$, has as vertices the words of length D on an alphabet of $d+1$ letters, with the property that $x_i \neq x_{i+1}$ for $1 \leq i \leq D-1$. Vertex $x_1 \dots x_D$ is also joined to $x_2 \dots x_D \alpha$, $\alpha \neq x_D$.

These two digraphs can also be defined as iterated line digraphs. Let us recall that the *line digraph* $L(G)$ of a digraph G has as vertices the arcs e of G, with an arc from e to f in $L(G)$ if and only if in G the initial vertex of f is the final vertex of e. In others words, e represents an arc of the form (x,y) and f an arc of the form (y,z). It is well known that $B(d,D+1)$ (resp. $K(d,D+1)$) is isomorphic to $L(B(d,D))$ (resp. $L(K(d,D))$ (see [3, 13]).

For a given digraph G, we denote $\mathcal{U}G$ the *underlying graph* associated to G (obtained by removing the orientation). The underlying de Bruijn (resp. Kautz) graph will therefore be denoted $\mathcal{UB}(d,D)$ (resp. $\mathcal{UK}(d,D)$).

Let us recall the previous bounds known on the broadcast time for de Bruijn and Kautz networks. Various lower bounds have been obtained on these networks, we give the ones concerning the de Bruijn network (they are analogous for the Kautz network). We first have the trivial general bound: $b(\mathcal{UB}(d,D)) \geq \lceil \log_2(d^D) \rceil = \lceil D \log_2(d) \rceil$. We can also use the results of [1] which give lower bounds for networks with bounded degree $\Delta = 2d$.

$$b(\mathcal{UB}(d,D)) \geq \lceil c_\Delta D \log_2(d) \rceil$$

where c_Δ is of the order $1 + \frac{\log_2(e)}{2^\Delta}$, where $e = exp(1)$. For example for $d = 2$, $c_4 = 1.137$. In the case of the binary de Bruijn network ($d = 2$), the bound has been improved in ([10]) leading to $b(\mathcal{UB}(2,D)) \geq 1.3117D$. These bounds have been recently improved in [11] with $b(\mathcal{UB}(2,D)) \geq 1.4404D$. In [11] the lower bounds for greater values of d and iterated line digraphs are also improved.

Concerning upper bounds, many partial results have been obtained for broadcasting on these networks (see [1, 2, 4, 6, 8]). For example, broadcasting protocols have been designed which give mainly the following bounds (valid for digraphs and graphs):

$$b(\mathcal{B}(d,D)) \leq \frac{d+1}{2}(D+1) \ \ [2] \tag{1}$$

$$b(\mathcal{B}(d,D)) \leq min(2D\lceil \log_2(d) \rceil, 3D\lceil \log_3(d) \rceil) \ \ [6] \tag{2}$$

$$b(\mathcal{B}(d,D)) \leq (\frac{5}{4}\lceil \log_2 d \rceil + 3)D \ \ [6] \tag{3}$$

Slight improvements of (3) can be found in [6]. For the Kautz digraph the bounds are less tight.

$$b(\mathcal{K}(d,D)) \leq \frac{d+3}{2}(D+1) \ [2] \tag{4}$$

$$b(\mathcal{K}(d,D)) \leq min(2D\lceil \log_2(d)\rceil, 3D\lceil \log_3(d)\rceil) \ [6] \tag{5}$$

Recently in [12] compound techniques were used to improve the bound for undirected de Bruijn and Kautz graphs, leading to

$$b(\mathcal{UB}(d,D)) \leq D\lceil \log_2(d)\rceil + D - 1 \tag{6}$$

$$b(\mathcal{UK}(d,D)) \leq D\lceil \log_2(d)\rceil + D - 1 \tag{7}$$

These results give an asymptotically optimal upper bound when d tends to infinity. However for small degrees they are not optimal; for example, when $d = 2$, it leads to $b(\mathcal{UB}(2,d)) \leq 2D - 1$ to be compared to $\frac{3}{2}(D+1)$ (protocol (1)).

The aim of this article is to provide a protocol with a broadcasting time which is the minimum of the value obtained in (6) and that given by the following one:

$$b(\mathcal{UB}(d,D)) \leq \lfloor (D+1)(\log_2(d) + F(d)) \rfloor \text{ where } F(d) \leq \frac{1}{2} \tag{8}$$

Remark. According to the respective values of d and D, the best bound will be induced either by (6) or (8). Note that, at least when $D > \log_2(d)$, (8) is better than (6).

We also improve the bound for the Kautz digraph and the Kautz graph.
The paper is organised as follows. First, in section 2 we recall the classical protocol for de Bruijn digraphs and shows how to extend it for the Kautz digraphs. In section 3 we give a simple protocol which can be applied to any underlying graph of a line digraph. Finally we mix the ideas to obtain the bound (8).

2 A protocol for de Bruijn and Kautz digraphs

We first give protocols for de Bruijn and Kautz digraphs as they contain some of the key ideas. The protocol for the de Bruijn digraph was already described in [2].

First idea When a vertex $x_1x_2\ldots x_D$ receives the message, it will inform its out-neighbors $x_2\ldots x_D\beta$ according to some policy. The choice of this policy is the key of the efficiency of the protocol. If we choose a policy independent from the vertex (like informing at first the neighbor with $\beta = 0$, then $\beta = 1$ and so on) then at least one vertex will be informed after Dd units of time. A better policy, as we will see after (used in [2, 1] and implicit in [9]) consists in a rule depending on the vertex $x_1\ldots x_D$. In the case of the de Bruijn network, let δ denote the d-arity of a vertex, that is $\delta = \sum_{i=1}^{i=D} x_i \ (mod\, d)$. Then $x_1\ldots x_D$ will inform successively $x_2\ldots x_D\delta$, $x_2\ldots$ $x_D\delta+1,\ldots$ and so on. Here there exist also destinations for which the delay on the canonical path is again Dd.

Second idea The message can reach a destination via different paths (we will use d of them). We want clearly to choose the path on which the delay is the smallest as possible, but the computation of such a parameter is too complicated. However, we will show that, with the policy defined above, one can compute the sum of the delays on the d paths and then the best delay will be bounded above by the average one.

More precisely, on $B(d, D)$ we will use the d following paths of length $D+1$ between $x_1 \dots x_D$ and $y_1 \dots y_D$: P_α for $\alpha \in \{0, 1 \dots d-1\}$ where

$$P_\alpha = x_1 \dots x_D,\ x_2 \dots x_D\alpha,\ x_3 \dots x_D\alpha y_1,\ \dots\dots,\ \alpha y_1 \dots y_{D-1},\ y_1 \dots y_D.$$

In fact we send the message to all the out-neighbors of the originator, and then consider the d canonical paths from the d out-neighbors to $y_1 \dots y_D$. Note that these paths have the property that their i^{th} internal vertices have all different d-arities. Indeed the i^{th} internal vertex of P_α is $x_{i+1} \dots x_D, \alpha, y_1 \dots y_{i-1}$ and differs from the i^{th} internal vertex of P_β only in the coordinate $D-i+1$.

The time to reach $y_1 \dots y_D$ on P_α is: $d_\alpha = \sum_{i=1}^{i=D+1} d_\alpha^i$ where d_α^i is the delay from $x_i \dots x_D, \alpha, y_1 \dots y_{i-2}$ to $x_{i+1} \dots x_D, \alpha, y_1 \dots y_{i-1}$. By the above property if $\alpha \neq \beta$, $d_\alpha^i \neq d_\beta^i$, then $\sum_\alpha d_\alpha^i = 1 + 2 + \dots d = \frac{d(d+1)}{2}$.

Hence the sum of the delays on the d paths P_α of length $D+1$ is:

$$\sum_\alpha d_\alpha = \sum_{i=1}^{D+1} \sum_\alpha d_\alpha^i = (D+1)\frac{d(d+1)}{2} \tag{9}$$

And so on one of the paths the delay is at most $\lfloor \frac{d+1}{2}(D+1) \rfloor$.

Strictly speaking, in the above scheme the message may arrive at some vertex more than once. By deleting redundant calls, a scheme can be obtained that completes the broadcast at the appropriate time.

That was the result proved in [2]; we state it as a proposition :

Proposition 1.

$$b(\mathcal{B}(d, D)) \leq \lfloor \frac{d+1}{2}(D+1) \rfloor$$

In the case of the Kautz digraph, we cannot use the d-arity like above because the letters are modulo $(d+1)$ and furthermore the canonical path is of length D or $D-1$. We will define a *"d-arity"* as a number modulo d, but we will work with the set of integers modulo d, with representatives $\{1, 2 \dots d\}$ denoted A_d^1. We define $\oplus$ as the addition in A_d^1 as follows: $i \oplus j = 1 + ((i-1) + (j-1)\ (mod\, d)\ )$. For example if $d = 4$ we have $2 \oplus 2 = 1 + (1 + 1\ (mod\, 4)\ ) = 3$, $3 \oplus 4 = 1 + (2 + 3\ (mod\, 4)\ ) = 1 + 1 = 2$.

Now for vertices of the Kautz graph, we define the d-arity $\delta(x)$ as follows:

For a word of length 2 $a_1 b_1$, the value $b_1 - a_1$ is, modulo $d+1$, different from zero. Let δ be the representative in A_d^1 of $b_1 - a_1$. We set $\delta(a_1 b_1) = \delta$. For $d = 4$ we have for example, $\delta(03) = 3$, $\delta(20) = 3$ as $2 + 3 = 0\ (mod\, 5)$, $\delta(41) = 2$ as $4 + 2 = 1\ (mod\, 5)$.

For a word x of even length $D = 2p$ with $x = a_1 b_1 a_2 b_2 \ldots a_p b_p$ we set:

$$\delta(x) = \bigoplus_{i=1}^{i=p} \delta(a_i b_i)$$

So, $\delta(032041) = 2$, as $3 \oplus 3 \oplus 2 = [(2+2+1) \ (mod\, 4)\ ] + 1 = 1 + 1 = 2$.

For a word x of odd length D, where $x = ya$ with y of even length $D-1$, we set $\delta(x) = \delta(y)$.

Note the key property of this d arity. If two words differ on one letter (other than the rightmost one in the case D odd) then their d-arities differ too. In particular $x_{i+1} \ldots x_d \alpha y_1 \ldots y_{i-1}$ and $x_{i+1} \ldots x_d \beta y_1 \ldots y_{i-1}$ have different d-arities.

During the protocol a vertex $x_1 \ldots x_D$ will inform successively the vertices: $x_1 \ldots x_D(x_D + \beta_1)$, $x_1 \ldots x_D(x_D + \beta_2)$, $\ldots$, $x_1 \ldots x_D(x_D + \beta_d)$ where $\beta_i \in A_d^1$ is equal to $\delta(x) \oplus i$ in A_d^1.

Example 1. still with $d = 4$ $x = 032041$ we have, $\delta(x) = 2$. So $\beta_1 = 2 \oplus 1 = 2$, $\beta_2 = 2 \oplus 2 = 3$, $\beta_3 = 4$, $\beta_4 = 1$. As $x_D = 1$ the informed vertices will be successively 320413(as $1 + 2 = 3 \ (mod\, 5)$), 320414(as $1 + 3 = 4 \ (mod\, 5)$), 320410(as $1 + 4 = 0 \ (mod\, 5)$), 320412 (as $1 + 1 = 2 \ (mod\, 5)$).

Analysis of the protocol We have two different ways of performing the analysis. Either we consider that we have $d-1$ paths of length $D+1$, namely:

$P_\alpha = x_1 \ldots x_D, \ x_2 \ldots x_D\alpha, \ x_3 \ldots \ x_D\alpha, \ \ldots, \ \alpha \ldots y_{D-1}, \ y_1 \ldots y_D$ with $\alpha \notin \{x_D, y_1\}$. On P_α the total delay is: $d_\alpha = \sum_{i=1}^{D+1} d_\alpha^i$ where d_α^i is the delay to go from $x_i \ldots x_D \alpha y_1 \ldots y_{i-2}$ to $x_{i+1} \ldots x_D \alpha y_1 \ldots y_i$

We need to check that the delays d_α^i are well balanced, so that the sum of the delays d_α^i for a given i can be estimated. When $i = 1$ all the d_α^i clearly differ. When i is strictly greater than 2, as we already noted, the d_α^i are again different. So for $i \neq 2$ we have :

$$\sum_\alpha d_\alpha^i \leq \frac{(d-1)(d+2)}{2}$$

(Indeed we have to take $d-1$ distinct values among d values spanning $\{1, 2 \ldots d\}$, the worst case being if the value 1 is always missing)

When i is equal to 2, the delay d_α^2 satisfies $y_1 = \alpha + (\delta(x_2 \ldots x_D\alpha) \oplus d_\alpha^2))$. One can check that d_α^2 can be equal to d_β^2 only when D and d are even with $\alpha = \beta + \frac{d}{2}$. In this case d_α^i will not span $\{1 \ldots d\}$. However, in the worst case, the sum of the delays can be at most:

$$\sum_\alpha d_\alpha^2 \leq 2 + 4 + 4 + 6 + 6 \ldots + d + d = \frac{d(d+2)}{2} - 2$$

(Indeed the delays will span a set of the kind $c + 2j$). Using again the average argument of 9 page 4, we get for d odd $\frac{\sum_\alpha d_\alpha}{d-1} \leq \lfloor \frac{d+2}{2}(D+1) \rfloor$, and for d even $\frac{\sum_\alpha d_\alpha}{d-1} \leq \lfloor (D+1)\frac{d+2}{2} + \frac{d-2}{2(d-1)} \rfloor = \lfloor \frac{d+2}{2}(D+1) \rfloor$.

So altogether this leads to:

Proposition 2.

$$b(\mathcal{K}(d,D)) \leq \lfloor \frac{d+2}{2}(D+1) \rfloor \tag{10}$$

Another way is to send the message in at most $d+1$ units of time to the d vertices x_α where

$x_\alpha = x_3 \ldots x_D \alpha x_D$ for $\alpha \neq x_D$. Then for any vertex $x_D y_1 \ldots y_{D-1}$ we consider the d paths of length $D-1$ from x_α to $x_D y_1 \ldots y_{D-1}$. The same proof as above shows that one of this path has a delay of at most $\frac{d+1}{2}(D-1)$. So any vertex of the kind $x_D y_1 \ldots y_{D-1}$ can be reached in $\frac{d+1}{2}(D+1)$ units of time. Finally we can complete the broadcasting protocol in $d+1$ more time units. Therefore we have:

Proposition 3.

$$b(\mathcal{K}(d,D)) \leq \lfloor \frac{d+1}{2}(D+3) \rfloor \tag{11}$$

According to the values of D and d, (10) or (11) can be better, as (10)$-$(11)$= \frac{D+1}{2} - (d+1)$ can be either negative or positive.

3 A new protocol for undirected graphs

Third idea Till now our protocols use only the arcs of the digraph $B(d,D)$ (resp. $K(d,D)$). For the corresponding undirected graphs one can hope some improvements. We will use the arcs in the opposite direction, but only to accelerate the broadcasting to the out-neighbors.

Let us recall that if G is a $d-$regular digraph of order N, then the arcs of its line digraph $L(G)$ can be partitioned into N bipartite subgraphs isomorphic to $K_{d,d}$. More precisely each $K_{d,d}$ of $L(G)$ correspond to a vertex of G in the following way. The $K_{d,d}$ associated to x has as first set of the bipartition, the vertices (called initial vertices of this $K_{d,d}$) of $L(G)$ which represent arcs in G entering x and as second set, those (called terminal vertices of this $K_{d,d}$) which represent arcs of G leaving x. By definition of $L(G)$ any initial vertex is joined to any terminal vertex. This property is known to be characteristic of line digraphs and has been noted by different authors (for example see [7]).

Example 2. The arcs of $B(d,D)$ can be partitioned into d^{D-1} $K_{d,d}$. Namely to each $D-1$ tuple $x_2 \ldots x_d$ is associated a $K_{d,d}$ with initial vertices $\{\alpha x_2 \ldots x_D | \alpha \in \{0 \ldots d-1\}\}$ and terminal ones $\{x_2 \ldots x_D \beta | \beta \in \{0 \ldots d-1\}\}$.

Now recall that we can broadcast from any vertex in $K_{d,d}$ to all the vertices in $\lceil \log_2(d) \rceil + 1$ time units with a protocol that we will call the *bipartite protocol for this* $K_{d,d}$. Indeed let the vertices of $K_{d,d}$ be respectively $A = \{a_0 \ldots a_{d-1}\}$ and $B = \{b_0 \ldots b_{d-1}\}$. The broadcasting protocol is as follows: at time 1, if the originator is a_i, it informs b_i; then at time $t \geq 2$, any vertex a_j(resp. b_j) who knows the message sends it to $b_{j+2^{t-2}}$ resp. $(a_{j+2^{t-2}})$. By induction one can easily shows that after time t, if a_i is the originator, the message is known by 2^{t-1} vertices of B, namely $b_i \ldots b_{i+2^{t-1}-1}$ and 2^{t-1} vertices of A, namely $a_i \ldots a_{i+2^{t-1}-1}$.

Example 3. For $d = 8$, let a_0 be the originator; at time 1 a_0 informs b_0; at time 2 a_0 informs b_1 and b_0 informs a_1; at time 3 a_0 and a_1 inform b_2 and b_3, and b_0, b_1 inform a_2, a_3; at time 4 a_0, a_1, a_2, a_3 inform b_4, b_5, b_6, b_7. Note that formally, at time 4 $b_0\, b_1\, b_2\, b_3$ can send the information to $a_4\, a_5\, a_6\, a_7$, but that is useless for our purpose. Furthermore, in what follows we will take advantage of the fact that $b_0\, b_1\, b_2\, b_3$ have finished their work at time 3.

Now we are able to define a protocol divided into phases. During a phase, each vertex, which has received the message as a terminal vertex of some $K_{d,d}$ in the preceding phase, sends it to all the terminal vertices of the $K_{d.d}$ in which it is an initial vertex. Each phase takes $\lceil \log_2(d) \rceil + 1$ time units. As defined the protocol cannot generate conflicts. Indeed at phase i, vertices which start a bipartite protocol are of the form $x_i \dots x_D *$ and none of them can be in the same $K_{d,d}$. Furthermore, if a vertex belongs to two $K_{d,d}$, in the first one as an initial vertex, in the second as a terminal vertex we cannot have conflicts, except if the vertex is a loop $a \dots a$ (in the de Bruijn graph) in which case we can consider it as being not a terminal vertex and work in $K_{d,d-1}$ instead of $K_{d,d}$. So we can effectively use this protocol. After i phases of this protocol, every vertex at distance at most i from the originator is informed. So we have:

Proposition 4.

$$b(\mathcal{UB}(d, D)) \leq D(\lceil \log_2(d) \rceil + 1) \tag{12}$$

$$b(\mathcal{UK}(d, D)) \leq D(\lceil \log_2(d) \rceil + 1) \tag{13}$$

Remark. Note that this protocol can be also defined on any underlying graph of a line digraph. The proof is similar as the one above. We state it as a proposition

Proposition 5. *If G is a d-regular digraph:*

$$b(\mathcal{U}L(G)) \leq D(L(G))(\lceil \log_2(d) \rceil + 1)$$

Combination of the ideas In fact we can still improve the bound if we mix all the ideas, that is we consider the broadcasting time on different paths in G, but using the bipartite protocol in order to inform the out-neighbors of a vertex in G.

In the directed protocol for the de Bruijn digraph the average time needed to inform the out-neighbors of a vertex is exactly $\frac{1}{d} \sum_{i=1}^{i=d} i = \frac{d+1}{2}$. We will see after that the average time is of order $\log_2(d) + \frac{1}{2}$ when we use the connections of $\mathcal{U}L(G)$.

Let us focus on the following problem: One initial vertex of a $K_{d,d}$ has to send the message to all the terminal vertices of its $K_{d,d}$. Let s_i be the time at which b_i is not only informed but has finished to participate in the local bipartite protocol. That means that at time $s_i + 1$, b_i will be able to start a protocol in the bipartite $K_{d,d}$ for which it is an initial vertex. What will be important is to find a protocol which minimise the average of s_i (respectively to the terminal half of the $K_{d,d}$), denoted $\overline{b}_{out}(K_{d,d})$.

Proposition 6. *Let $d = 2^{k-1}(2+\delta)$ with $0 \le \delta \le 2$, then we have:*

$$0 \le \delta \le 1 \quad \overline{b}_{out} = k + \tfrac{2\delta+1}{\delta+2} \tag{14}$$
$$1 \le \delta \le 2 \; \overline{b}_{out} = k + 1 + 2(\tfrac{\delta-1}{\delta+2}) \tag{15}$$

Proof. In the case $\delta \le 1$, let the protocol be the one of the bipartite graph during the k first steps. At this point 2^{k-1} terminal and initial vertices are informed. Then informed terminal vertices stop to participate to the algorithm. The 2^{k-1} informed initial vertices inform 2^{k-1} new terminal vertices at time $k+1$, and the $\delta 2^{k-1}$ remaining terminal vertices at time $k+2$. So

$s_0 = s_1 = \ldots = s_{2^{k-1}-1} = k$, $s_{2^{k-1}} = \ldots = s_{2^k-1} = k+1$ and $s_{2^k} = \ldots = s_{d-1} = k+1$. That leads to $\overline{b}_{out} = \frac{k2^{k-1}+(k+1)2^{k-1}+(k+2)\delta 2^{k-1}}{(\delta+2)2^{k-1}} = k + \frac{2\delta+1}{\delta+2}$.

In the case $\delta \ge 1$, after k steps of the algorithm, $(2-\delta)2^{k-1}$ terminal vertices will stop and $(\delta-1)2^{k-1}$ terminal vertices will continue to participate to the algorithm one step more. Then at time $k+1$, $(\delta-1)2^{k-1}$ new initial vertices, and 2^{k-1} new terminal vertices will be informed. At this stage, the $(\delta-1)2^{k-1}$ terminal vertices acting during the preceding step and the 2^{k-1} newly informed terminal vertices will stop. So altogether $\delta 2^{k-1}$ vertices have finished to participate to the protocol at this time. To complete the protocol the $\delta 2^{k-1}$ initial vertices having the message inform the $\delta 2^{k-1}$ remaining terminal vertices which finish at time $k+2$. The average time is then: $\overline{b}_{out} = \frac{k(2-\delta)2^{k-1}+(k+1)\delta 2^{k-1}+(k+2)\delta 2^{k-1}}{(\delta+2)2^{k-1}} = k+1+2\frac{\delta-1}{\delta+2}$.

Remark. One can prove that this is protocol provides an optimum value of $\overline{b}_{out}$, but we omit the proof.

Example 4. For $d = 4$, $s_0 = s_1 = 2$ $s_2 = s_3 = 3$; for $d = 5$ $s_0 = s_1 = 2$ $s_2 = s_3 = 3$ $s_4 = 4$; for $d = 6$ $s_0 = s_1 = 2$ $s_2 = s_3 = 3$ $s_4 = s_5 = 4$; for $d = 7$ $s_0 = 2$ $s_1 = s_2 = s_3 = 3$ $s_4 = s_5 = s_6 = 4$.

Remark. For $d = 2$, (3) we have $s_0 = 1$ $s_1 = 2$ $(s_2 = 3)$ which are the same values as that used in proposition 1. For $d = 4$, $\overline{b}_{out} = 2.5$ which is also the average obtained in the directed protocol : $\frac{1+2+3+4}{4}$.

To continue we need to compare $\overline{b}_{out}(d)$ with the maximal value which is $\lceil \log_2(d) \rceil + 1$. In fact the reader can check that $\overline{b}_{out}(d) = \log_2(d) + F(\delta)$ with $F(\delta) = 2 - \log_2(\delta+2) + \frac{\delta-1}{\delta+2}$ for $\delta \le 1$, and $F(\delta) = 2 - \log_2(\delta+2) + 2\frac{\delta-1}{\delta+2}$ for $\delta \ge 1$. Easy computation shows that $\log_2(\frac{4}{3}) \le F(\delta) \le \frac{1}{2}$.

Proposition 7. *For any degree $d = 2^{k-1}(2+\delta)$ with $0 \le \delta \le 2$:*

$$\overline{b}_{out}(d) = \log_2(d) + F(\delta) \text{ with } \log_2(\frac{4}{3}) \le F(\delta) \le \frac{1}{2}$$

Now we are ready to prove our main theorem.

Theorem 8.

$$b(\mathcal{UB}(d,D)) \le \lfloor (D+1)\overline{b}_{out}(d) \rfloor$$
$$b(\mathcal{UK}(d,D)) \le 3 + 2\lceil \log_2(d) \rceil + \lfloor (D-1)\overline{b}_{out}(d) \rfloor$$
$$b(\mathcal{UK}(d,D)) \le (D+1)(\overline{b}_{out} + \frac{3}{2(d-1)})$$

Sketch of Proof. Consider the following protocol. First the originator sends to its out-neighbors according to the bipartite protocol described in proposition 11. During the protocol suppose a vertex receives the message as a terminal vertex of some $K_{d,d}$, then at first, it participates as a terminal vertex in the bipartite protocol of this $K_{d,d}$, and when it has finished its work in this $K_{d,d}$ it starts the bipartite protocol in the $K_{d,d}$ for in which it is an initial vertex. The protocol is similar of the one of proposition 5 except that we do not use synchronous phases as a vertex starts sending messages as an initial vertex as soon as possible. The only difference is that there might be conflicts. But we will show after that it is not a problem.

Furthermore we use the bipartite protocol like in the proposition 1 in such a way that the "delays" differ if the initial vertices differ in one coordinate. To do that, it suffices to label the initial vertex $\alpha x_2 \dots x_D$ as a_i where i is the d-arity of $\alpha x_2 \dots x_D$ in the case of the de Bruijn graph (and the d-arity minus 1 in the case of the Kautz graph). We label a terminal vertex $x_2 \dots x_D \beta$ by b_j with $j = \beta$ for the Bruijn graph ($j = \beta - x_D - 1$ for the Kautz graph).

In the case of the de Bruijn graph a vertex can received the information on d paths of length $D+1$ and similarly as in the protocol for digraphs the delay between the time where $x_{i+1} \dots x_D \alpha y_1 \dots y_{i-1}$ starts the bipartite protocol as initial vertex and the times where its neighbor $x_{i+2} \dots x_D \alpha y_1 \dots y_{i-1} y_i$ is ready to start a new bipartite protocol are all different. And, when α varies, the average value of the delays taken on the d paths, is $\overline{b}_{out}$. So $b(\mathcal{UB}(d,D)) \le \lfloor (D+1)\overline{b}_{out} \rfloor$.

For the Kautz graph we can consider $d-1$ paths with $\alpha \notin \{x_D, y_1\}$. In that case the average will be taken on the $d-1$ greatest values that is we will replace, for $d = 2^{k-1}(\delta+2)$, $\overline{b}_{out}$ by $\frac{d\overline{b}_{out}-k}{d-1}$. This is $k + \frac{d}{d-1}\frac{2\delta+1}{\delta+2}$ for $\delta \le 1$ and $k + \frac{d}{d-1}\frac{3\delta}{\delta+2}$ if $\delta \ge 1$. So we always have at most $(D+1)(\overline{b}_{out} + \frac{3}{2(d-1)})$.

An other way is to first send the message from $x_1 \dots x_D$ to $x_\alpha = x_3 \dots x_d \alpha x_D$ in at most $\lceil \log_2(d) \rceil + 2$ units of time. Then using the d paths from x_α to $x_D y_1 \dots y_{D-1}$ and the properties of d-arities, $x_D y_1 \dots y_{D-1}$ is informed in at most $(D-1)\overline{b}_{out}$ more units of time, and finally in $\lceil \log_2(d) \rceil + 1$ more units of time we inform all the vertices.

Now let us examine the possibility of conflicts (that is when a vertex is involved into different calls at the same time).

First suppose that in the same $K_{d,d}$ two initial vertices say a_i (resp. a_j) are ready to start the protocol at time t (resp. $t+h$). When a_j is ready to start the protocol, suppose it has already received the information from a_i, then the action of a_j is covered by that of a_i and we simply delete a_j from the $K_{d,d}$ for which it is a terminal vertex. If a_j is ready to start the protocol, but it has not still received the message from a_i, we can combine the action of a_i and a_j to accelerate the process. It suffices to modify the protocol in such a way that any a_j (resp b_j) informed informs the first vertex not already informed having an index greater than or equal to the one it was supposed to inform. For example suppose a_i and a_j are both ready to start at the same time then a_i inform b_i and a_j inform b_j Then on the next step a_i and a_j inform b_{i+1} and b_{j+1} or for example if $b_{i+1} = b_j$ a_i inform b_{i+2} and a_j b_{j+2} and so on.. In the case $d = 2^k$ the terminal vertices will have been informed at time $k-1$ and k instead of k and $k+1$.

Now it can happen also that some vertex b_i is implied in the protocol of some $K_{d,d}$, say B_1 as a terminal vertex and also as the initial vertex of another $K_{d,d}$, say B_2. Then as we want only to inform the out-neighbors, it is useless that b_i works in B_1 because his future action in B_2 has already been done. So we delete it as a terminal vertex of B_1.

In summary either some vertices are useless as terminal vertices and we work with a $K_{d,d'}$ with $d' \leq d$ and the bipartite protocol is faster, or some vertices can conjugate their actions as initial vertices to accelerate the process.

The following table displays the time of our protocol, and compares it with previous one. For this purpose we have listed the coefficient $A(d)$ appearing as the coefficient of $(D+1)$ in the best previous bound (see [6]) , $\overline{b}_{out}$ the coefficient of $D+1$ in theorem 8 and the coefficient $F(d)$ which gives the gap with the trivial lower bound $\log_2(d)D$.

d	2	3	4	5	6	7	8	9	10	11	12
$A(d)$	1.5	2	2.5	3	3.5	4	4.5	4	5.5	6	5.5
$\overline{b}_{out}(d)$	1.5	2	2.5	2.8	3	3.28	3.5	3.66 ...	3.8	3.9	4
$F(d)$	0.5	0.415	0.5	0.478	0.415	0.478	0.5	0.497	0.478	0.45	0.415

4 Conclusion

We have designed a new protocol for broadcasting in the de Bruijn and Kautz graphs. The bounds obtained are better than all of those known before. The ideas and tools can be used for any networks obtained as the undirected graph of a line digraph (butterflies networks and so on). The gap with the trivial lower bound is always bounded by $\frac{D}{2}$ so for large diameters our protocol is nearly optimal. For very small degrees $d = 1, 2, 3, 4$ our protocol is the same as for digraphs, and there are still improvements to be found perhaps using paths of greater length. Finally let us note that recently A. Marchetti and X. Munoz (lecture at IWIN 95, Marseille July 1995) have used similar ideas and improved the bounds for general line digraphs, leading also to improvements for many de Bruijn graphs.

References

1. J-C. Bermond, P. Hell, A.L. Liestman, and J.G. Peters. Broadcasting in bounded degree graphs. *SIAM journal of Discrete Mathematics*, 5:10–24, 1992.
2. J-C. Bermond and C. Peyrat. Broadcasting in de Bruijn networks. In Congressus Numerantium 66, editor, *Proceedings of the 19th S-E conference on Combinatorics, Graph theory, and Computing, Florida*, pages 283–292, 1988.
3. J-C. Bermond and C. Peyrat. De Bruijn and Kautz networks: a competitor for the hypercube? *In Hypercube and Distributed Computers*, pages 279–294, North-Holland 1989
4. P. Fraigniaud and E. Lazard. Methods and problems of communication in usual networks. *Discrete Applied Mathematics*, 53:79–133, 1994.
5. S.L Johnsson M. Garey. *Computer and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, San Francisco CA, 1979.
6. M-C. Heydemann, J. Opatrny, and D. Sotteau. Broadcasting and spanning trees in de Bruijn and Kautz networks. *Discrete Applied Math.*, 27-28:297–317, 1992.
7. M-C. Heydemann and D. Sotteau. A note on recursive properties of de Bruijn, Kautz and FFT digraphs. *IPL*, 53:255–259, 1995.
8. J. Hromkovic, R. Klasing, B. Monien, and R. Peine. Dissemination of information in interconnection networks (broadcasting and gossiping). Technical report, University of Paderborn, February 1993. To appear as chapter in the book:" Combinatorial Network Theory".
9. M-R. Jerrum and S. Skyum. Families of fixed degree graphs for processor interconnection. *IEEE Trans Comput*, C-33(2):190–194, 1984.
10. R. Klasing, B. Monien, R. Peine, and E. Stohr. Broadcasting and gossiping in the butterfly and de bruijn networks. *Discrete Applied Mathematics*, 53:183–197, 1994.
11. S. Perennes. Bound on the broadcasting time of the de Bruijn and Butterfly networks. In preparation.
12. S. Perennes. Broadcasting and Gossiping on de Bruijn Shuffle exchange and similar networks. Technical Report 93-53, I3S, Sophia-Antipolis, October 1993.
13. J. Rumeur. *Communications dans les Réseaux de Processeurs.* Masson, 1994.

This article was processed using the LaTeX macro package with SIROCCO style