二進制有向性 de Bruijn 網路的訊息廣播演算法
Broadcast Algorithms for Binary Directed de Bruijn Networks

林銘波
Ming-Bo Lin

國立台灣科技大學電子工程系

Department of Electronic Engineering
National Taiwan University of Science and Technology
E-mail: mblin@et.ntust.edu.tw

白明弘
Ming-Hong Bai

中央研究院資訊所

Institute of Information Science
Academia Sinica

中文摘要

近年來，由於 de Bruijn 網路具有 Bounded node degree ，它已經成為 Hypercube 網路之外的另外一種廣受歡迎的連結網路。在本論文中，我們提出一個在二進制有向性 de Bruijn 網路上的最佳化的一對所有的訊息廣播演算法。此外，使用任意兩個節點之間的距離觀念，我們建立了一個訊息傳播規則，並據以建立一個沒有重複訊息傳播的最佳化的所有對所有的訊息廣播演算法。這兩種演算法均善用了二進制有向性 de Bruijn 網路的輸入與輸出連線(edge)，因而達到最佳效果。

關鍵字： de Bruijn 圖、de Bruijn 網路、一對所有訊息廣播演算法、所有對所有訊息廣播演算法、最短路徑。

ABSTRACT

Recently, de Bruijn networks have been proposed as the alternatives to the popular hypercube networks as one kind of the new interconnection networks for highly parallel computers because they have bounded node degree that is an essential factor for a system to be scalable. In addition to scalability, message routing is an important factor that affects the system performance. In this paper, we introduce an optimal one-to-all broadcast algorithm for binary directed de Bruijn networks with the expected running time $O(\log N)$, where N is the number of nodes of the network. Based on the concept of distance between any two nodes, a message propagation rule is founded to guarantee that no redundant messages are received by any node on the network. In addition, an optimal all-to-all broadcast algorithm is proposed. Both of these algorithms make full use of incoming edges and outgoing edges of binary directed de Bruijn networks.

Keywords: de Bruijn graph, de Bruijn network, distance, one-to-all broadcast, all-to-all broadcast, and shortest path.

1 Introduction

Direct networks have been extensively used in highly parallel computer systems. One of these is the hypercube network, whose properties have been extensively studied in the literature [5]. However, due to its unbounded node degree, the scalability of the system based on this network is limited. This has motivated the interests of many researchers to search bounded-degree networks that can efficiently simulate hypercube computations [7, 8].

One of the most popular bounded-degree networks is the de Bruijn network based on the de Bruijn graph [5, 7, 9]. The attractive properties of a d-ary directed de Bruijn graphs with N nodes are that each node is of degree $2d$ and there are Nd edges. For simplicity and practical considerations, in this paper, we only consider binary directed de Bruijn graphs.

The major function of a routing procedure is to route messages to its right place at a reasonable amount of time. The most popular routing procedure is one-to-one routing, which routes a message from a source node to a given destination node. Two routing-related problems, specifically one-to-all broadcast and all-to-all broadcast, are essential to many scientific computations [2, 3]. In one-to-all broadcast, a given message is sent to all other processors of the system. It is usually required in the following important computations: matrix-vector multiplication, Gaussian elimination, shortest paths, and vector inner product problems. All-to-all broadcast is a generalization of one-to-all broadcast in which all processors initiate a broadcast simultaneously. It is commonly used in matrix operations, such as matrix multiplication and matrix-vector multiplication [4].

In this paper, we introduce an optimal one-to-all broadcast algorithm for binary directed de Bruijn networks with the expected running time $O(\log N)$, where N is the number of nodes of the network. Based on the concept of distance between any two nodes, a message propagation rule is founded to guarantee that no redundant messages are received by any node on

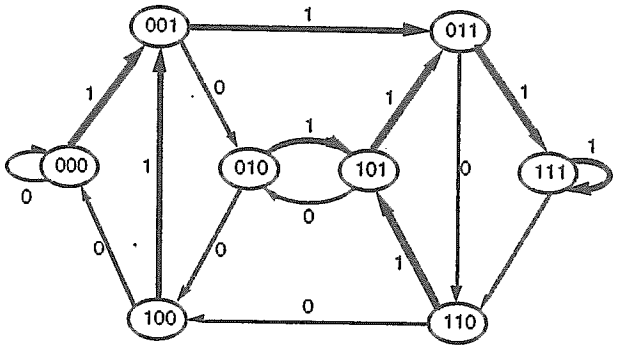

Figure 1: An example of binary 3-dimensional directed de Bruijn graph.

the network. In addition, an all-to-all broadcast algorithm is proposed. Both of these algorithms make full use of incoming edges and outgoing edges of binary directed de Bruijn networks.

The rest of the paper is organized as follows. Section 2 summarizes some basic definitions that will be used throughout the paper. Section 3 develops a one-to-all broadcast algorithm for de Bruijn networks. Section 4 describes an all-to-all broadcast algorithm. The paper is concluded in Section 5.

2 Preliminaries

In this section, we review some basic facts which will be used throughout the paper.

Definition 1 *A binary k-dimensional directed de Bruijn graph, denoted as* $DDB(k)$, *consisting of* 2^k *nodes and* 2^{k+1} *directed edges, is defined as:* $DDB(k) = (V_k, E_k)$, *where* $V_k = \{0, 1, 2 \ldots, 2^k - 1\}$ *and* $E_k = \{\langle S, T\rangle | T = 2S \bmod 2^k$, *for all* $S, T \in V_k\} \cup \{\langle S, T\rangle | T = (2S \bmod 2^k)+1$, *for all* $S, T \in V_k\}$.

Let $X = x_k x_{k-1} \ldots x_1$, where $x_i \in \{0, 1\}$, be a node on a $DDB(k)$, then it is connected to two other nodes: $x_{k-1}x_{k-2} \ldots x_1 0$ and $x_{k-1}x_{k-2} \ldots x_1 1$, which are called left-child node and right-child node, respectively. The edges connected to node $x_{k-1}x_{k-2} \ldots x_1 0$ and $x_{k-1}x_{k-2} \ldots x_1 1$ are called 0 channel and 1 channel, respectively. Figure 1 shows an example of binary 3-dimensional directed de Bruijn graph along with the indications of 0 channels and 1 channels. The parallel computer system based on the $DDB(k)$ graph is called de Bruijn multiprocessor network and is denoted by the $DDB(k)$ network.

For convenience, let node X be an arbitrary node on a $DDB(k)$ network and $b \in \{0, 1\}$ then two operations: ShiftLeft and ShiftRight, are defined as follows:

$$ShiftLeft(x_k x_{k-1} \ldots x_1, b) = x_{k-1}x_{k-2} \ldots x_1 b \quad (1)$$

$$ShiftRight(x_k x_{k-1} \ldots x_1, b) = b x_k x_{k-1} \ldots x_2 \quad (2)$$

Based on this and the definition of $DDB(k)$, two parent nodes, P_1 and P_2, of any node X on a $DDB(k)$ network have node addresses: $P_1 = ShiftRight(X, 0)$ and $P_2 = ShiftRight(X, 1)$, respectively. In addition, node X has two child nodes, Y_1 and Y_2, with node addresses: $Y_1 = ShiftLeft(X, 0)$ and $Y_2 = ShiftLeft(X, 1)$, respectively.

The distance between any two nodes X and Y, denoted as $D(X, Y)$, is defined as the path with the minimum number of edges between nodes X and Y. For any $DDB(k)$ network, $D(X, Y) \neq D(Y, X)$ and $D(X, X) = 0$. The distance $D(X, Y)$ can be computed by using the following lemma [6].

Lemma 1
Let $X = (x_k x_{k-1} \ldots x_1)$ *and* $Y = (y_k y_{k-1} \ldots y_1)$ *be any two nodes on a* $DDB(k)$ *network and* $P = \{ShiftLeft(X, y_{k-c}), ShiftLeft(X, y_{k-c-1}), \ldots, ShiftLeft(X, y_1)\}$ *be a path on the network, where* $k \geq 1$, x_i *and* $y_i \in \{0, 1\}$, *for* $1 \leq i \leq k$, *and* c *is defined as*

$$c = \max\{s | 0 \leq s \leq k, x_s x_{s-1} \ldots x_1 = y_k y_{k-1} \ldots y_{k-s+1}\}$$

then P *is the shortest path between* X *and* Y *with length* $k - c$.

Proof: First, we prove that there exists a path with length $k - c$ between any two nodes X and Y on the $DDB(k)$ network. Let $P = \{ShiftLeft(X, y_{k-c}), ShiftLeft(X, y_{k-c-1}), \ldots, ShiftLeft(X, y_1)\}$ be such a path on the $DDB(k)$ network, then node X can reach node $x_c x_{c-1} \ldots x_1 y_{k-c} y_{k-c-1} \ldots y_1$ through path P. However, $x_c x_{c-1} \ldots x_1 = y_k y_{k-1} \ldots y_{k-c+1}$; thus, $x_c x_{c-1} \ldots x_1 y_{k-c} y_{k-c-1} \ldots y_1 = y_k y_{k-1} \ldots y_{k-c+1}$ $y_{k-c} y_{k-c-1} \ldots y_1 = Y$. That is, P is a path with length $k - c$ from node X to Y.

Next, we prove that the length of the shortest path between nodes X and Y is $k - c$. To prove this by contradiction, assume that there is a path $Q = \{ShiftLeft(X, q_{k-r}), ShiftLeft(X, q_{r-1}), \ldots, ShiftLeft(X, q_1)\}$ from nodes X to Y with length less than $k - c$. Then node X can reach node $x_r x_{r-1} \ldots x_1 q_{k-r} q_{k-r-1} \ldots q_1$, that is, $x_r x_{r-1} \ldots x_2 x_1 q_r q_{r-1} \ldots q_1 = y_k y_{k-1} \ldots y_1$. Hence, $x_r x_{r-1} \ldots x_2 x_1 = y_k y_{k-1} \ldots y_{k-r+1}$. However, from the definition of c, we obtain $c \geq k - r$, which contradicts to the assumption that there exists a shorter path Q, that is, $0 < r < k - c$. □

As mentioned in Definition 1, each node on a $DDB(k)$ network has two parent nodes. However, only one of them can be used to transmit a message at any time to avoid redundant messages. This parent node is called the message parent node and defined as follows. A node Y is called the message parent node of node X on a $DDB(k)$ network if Y is one of the two parent nodes of X and $D(R, X)$ through node Y is shorter than $D(R, X)$ without through node Y, where

R denotes the source node of the message to be sent to node X.

Lemma 2 *The message parent node Y of any node X on a $DDB(k)$ network for any message source node R is unique.*

Proof: To prove this lemma by contradiction, let Z be the another message parent node of X. By the definition of message parent node, both nodes Y and Z are the parent nodes of node X and $D(R,Y) \leq D(R,Z)$ as well as $D(R,Z) \leq D(R,Y)$. Accordingly, $D(R,Z) = D(R,Y)$. To prove that node Y is the same as node Z, let $D(R,Z) = d = D(R,Y)$ and let $X = x_k x_{k-1} \ldots x_1$, $Y = y x_k x_{k-1} \ldots x_2$, $Z = z x_k x_{k-1} \ldots x_2$, and $R = r_k, r_{k-1} \ldots r_1$. Using Lemma 1, we have $r_d r_{d-1} \ldots r_1 = y x_k x_{k-1} \ldots x_{k-d+1} = z x_k x_{k-1} \ldots x_{k-d+1}$, which implies that $y = z$ and hence node Y is the same node as Z. □

The relationship between the distance and nodes on a $DDB(k)$ network is then established as the following lemma.

Lemma 3 *Let X and Y be any two nodes on a $DDB(k)$ network. Assume that node Z is the message parent node of node Y then $D(X,Z) < D(X,Y)$ and $D(X,Z) = D(X,Y) - 1$ if Y is not a child node of X.*

Proof: From Lemma 2, each node Y can only have one message parent node. Since the message parent node Z must also be a parent node of node Y, to reach node Y from node X the path must first go to one of the parent nodes of Y by the definition of the $DDB(k)$ network. Consequently, $D(X,Z) < D(X,Y)$ and thus $D(X,Z) \leq D(X,Y) - 1$. Since node Z is the message parent node of Y, there must exist a path from node X to node Y through node Z with length $D(X,Z) + 1$. Therefore, $D(X,Y) \leq D(X,Z) + 1$, that is, $D(X,Y) - 1 \leq D(X,Z)$. Combining this with $D(X,Z) \leq D(X,Y) - 1$, we obtain $D(X,Z) = D(X,Y) - 1$. □

3 One-to-All Broadcast Algorithm

In many parallel algorithms, a basic routing operation is to broadcast a message from one node to all other nodes in the system. This operation is called one-to-all broadcast. In this section, we present an $O(\log N)$ time one-to-all broadcast algorithm.

From Lemma 1, the distance between any two nodes on a $DDB(k)$ network can be computed directly from their node addresses. For example, the distance between two nodes $X = 00100$ and $Y = 10011$ on the $DDB(5)$ is 2 since the maximum length of suffix of X that matches the prefix of Y is $c = 3$. Therefore, the distance is $k - c = 5 - 3 = 2$.

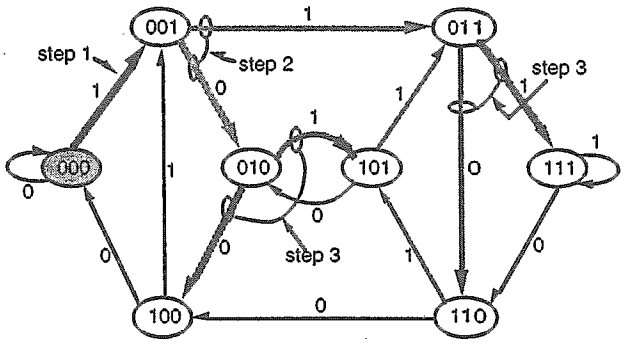

Figure 2: The basic idea of one-to-all broadcast.

On binary directed de Bruijn networks, each node has only two outgoing links. Consequently, the one-to-all broadcast algorithm must make full use of these two links so that a message can be distributed effectively. The key idea of the algorithm is based on the observation that once a node receives a message, it can then distribute the message out through its two outgoing links to two other nodes. These two nodes in turn distribute the received message in a similar way and so on until all nodes have the message.

An example is illustrated in Figure 2, in which the source node is node 000. As shown in the figure, the only outgoing link of node 000 is directed to node 001; hence, in step 1 node 000 only sends its message to node 001. In step 2, nodes 010 and 011 receive the message sent from node 001, and in step 3 the two nodes distribute the received messages to their child nodes: 100 and 101 as well as 110 and 111, respectively.

In order to guarantee the broadcast operation is correct and effective, the following rule is formulated.

Rule 1 *Let X, Y, and Z be any three nodes on a $DDB(k)$ network. Assume that Y is the left-child node of X and Z is the right-child node of X. Node R is the source node of a message. The node X distributes its received message $\mathcal{M}$ according to the following rules:*

1. *If $D(R,X) \geq D(R,Y)$, then X stop distributing $\mathcal{M}$ to Y; otherwise, X distributes $\mathcal{M}$ to Y.*
2. *If $D(R,X) \geq D(R,Z)$, then X stop distributing $\mathcal{M}$ to Z; otherwise, X distributes $\mathcal{M}$ to Z.*

The OnetoAllBroadcast algorithm is used to broadcast a message from a source node *root* to all other nodes on a $DDB(k)$ network. Since a node could not know where the message comes from before receiving it, every node on the $DDB(k)$ network must pay attention to its two parent nodes although Rule 1 guarantees only one of them can have the message passed from the source node. Once a node has the message, it

may compute $D(root, myid)$, $D(root, left\ child)$, and $D(root, right\ child)$ and then broadcast the received message in accordance with Rule 1.

Algorithm: OnetoAllBroadcast($myid, root$)
Input: The message source node *root* and any node *myid* of a $DDB(k)$ network.
Output: Every node has a copy of the message broadcast from node *root*.
begin
  if $myid \neq root$ **then pardo**
    receive message **from**
      ShiftRight($myid$, 0);**abort**; {left parent}
    receive message **from**
      ShiftRight($myid$, 1);**abort**; {right parent}
  parend {End of pardo}
  if $D(root, myid) < D(root$,**ShiftLeft**($myid$, 0))
    then send message **to**
      ShiftLeft($myid$, 0); {left child}
  if $D(root, myid) < D(root$,**ShiftLeft**($myid$, 1))
    then send message **to**
      ShiftLeft($myid$, 1); {right child}
end {End of OnetoAllBroadcast algorithm.}

An example is shown in Figure 2. The message source node is node 000. From **OnetoAllBroadcast** algorithm, at step 1 node 000 obtains the result $D(000,000) < D(000,001)$ and $D(000,000) \geq D(000,000)$ and hence it transmits its message to node 001. At step 2, node 001 computes the distance and obtains the result $D(000,001) < D(000,010)$ and $D(000,001) < D(000,011)$ so that it transmits the message to both nodes 010 and 011. At step 3, nodes 010 and 011 obtain the results $D(000,011) < D(000,111)$ and $D(000,011) < D(000,110)$, $D(000,010) < D(000,101)$ and $D(000,010) < D(000,100)$, respectively. Thus, they transmit the received messages to the following nodes 100 and 101, 110 and 111, respectively. The nodes 100, 101, 110, and 111 do not relay the received messages furthermore because $D(000,100) \geq D(000,000)$ and $D(000,100) \geq D(000,001)$ for node 100; $D(000,101) \geq D(000,010)$ and $D(000,101) \geq D(000,011)$ for node 101; $D(000,110) \geq D(000,100)$ and $D(000,110) \geq D(000,101)$ for node 110; and $D(000,111) \geq D(000,110)$ and $D(000,111) \geq D(000,111)$ for node 111.

The following theorem provides the correctness of OnetoAllBoradcast algorithm.

Theorem 1 *If each node on a $DDB(k)$ network distributes the received message to its child nodes according to Rule 1, then*

1. *Every node on the $DDB(k)$ network can have a copy of message broadcast from source node R.*
2. *No duplicated messages can be received by any node Y on the $DDB(k)$ network.*

Proof: 1. Assume that node R is the source node of the message to be broadcast. For an arbitrary node Y on the $DDB(k)$ network, the following cases should be considered.

a It is valid trivially if $Y = R$.

b Node Y can receive the message if Y is the left-child node of R since $D(R,R) < D(R,Y)$. Similarly, node Y can receive the message if Y is the right-child node of R.

c In the case that Y is not a child node of R, assume that $D(R,Y) = d$, where $d > 1$. By Lemma 3, node Y can find a parent node Z such that $D(R,Z) < D(R,Y)$. That is to say, node Y can receive the message if node Z can and $D(R,Z) = d-1$. Similarly, node Z can find a parent node Z_1 and receive the message from it. This operation is repeated until node Z_{d-2} such that $D(R,Z_{d-2}) = 1$. In this case, node Z_{d-2} is a child node of R and thus can receive the message from R. Consequently, node Y can have the message from R.

2. From Lemma 2, every node on the $DDB(k)$ network can only have a message parent node, which means it could not receives more than one copy of the message at any time. Furthermore, by Rule 1, the message is broadcast according to the increasing distance from the source node to all other nodes. Hence, a node that already received the message could not receive it again. Thus, no duplicated messages can be received by any node. □

The following theorem gives the bound of running time for OnetoAllBroadcast algorithm.

Theorem 2 *The paths used by* **OnetoAllBroadcast** *algorithm are shortest paths.*

Proof: Let node X be the source node and Y be any node on the $DDB(k)$ network. Assume that the path used by the algorithm **OnetoAllBroadcast** to transmit a message from node X to Y is not a shortest path, that is, the path length r is greater than $D(X,Y) = d$. Let this path be $X \rightarrow Y_1 \rightarrow Y_2 \rightarrow \ldots \rightarrow Y_{r-1} \rightarrow Y$. From Rule 1, $D(X,Y_1) < D(X,Y_2) < \ldots < D(X,Y_{r-1}) < D(X,Y)$, so we have $D(X,Y) > r-1$, which implies that $D(X,Y) = d \geq r$ and contradicts to the assumption that $d < r$. □

The following corollary is immediately followed.

Corollary 1 *Algorithm* **OnetoAllBroadcast** *will terminate after at most $k = \log N$ iterations, where k is the dimension of the $DDB(k)$ network.*

Proof: The number of iterations of broadcast on the network is at most $k = \log N$ since algorithm **OnetoAllBroadcast** follows the shortest paths to distribute a message to all other nodes on the $DDB(k)$ network. □

4 All-to-All Broadcast Algorithm

All-to-all broadcast is an extension of one-to-all broadcast. In this case each node on the $DDB(k)$ network broadcasts a message to all other nodes. The basic idea of our algorithm is as follows. For each node on the network, it sends its messages to its two outgoing links and then receives messages from its two incoming links. In turn it combines the messages received and prepares for the next step of distributing operation. This operation is then repeated k times.

The AlltoAllBroadcast algorithm broadcasts a message from each source node to all other nodes on a $DDB(k)$ network. The **AlltoAllBroadcast** algorithm is given as follows.

Algorithm: AlltoAllBroadcast$(myid, S, R)$
Input: The current node *myid*, the set of messages to be relayed S, and the set of received messages R of a $DDB(k)$ network. At beginning, each node has only its own message to be broadcast in S.
Output: Every node has a copy of the messages broadcast from all other nodes.

```
begin
  R = S;
  for i = 1 to k do
    send S to ShiftLeft(myid, 0);  {left child}
    send S to ShiftLeft(myid, 1);  {right child}
    S = ∅;
    receive message M from
      ShiftRight(myid, 0);  {left parent}
    S = S ∪ M;
    receive message M from
      ShiftRight(myid, 1);  {right parent}
    S = S ∪ M;
    R = R ∪ S;
  endfor
end {End of AlltoAllBroadcast algorithm.}
```

An example to illustrate the operations of algorithm **ALLtoAllBroadcast** is shown in Figure 3. The following theorem establishes the correctness of algorithm **AlltoAllBroadcast**.

Theorem 3 *Algorithm* **AlltoAllBroadcast** *will distribute the message of each node to all other nodes on the $DDB(k)$ network. That is, after the algorithm terminates, any node on the $DDB(k)$ network can receive all messages from all other nodes.*

Proof: For notational convenience, assume that the message sent by a node is identified by its node address. Based on this notation, in what follows we will

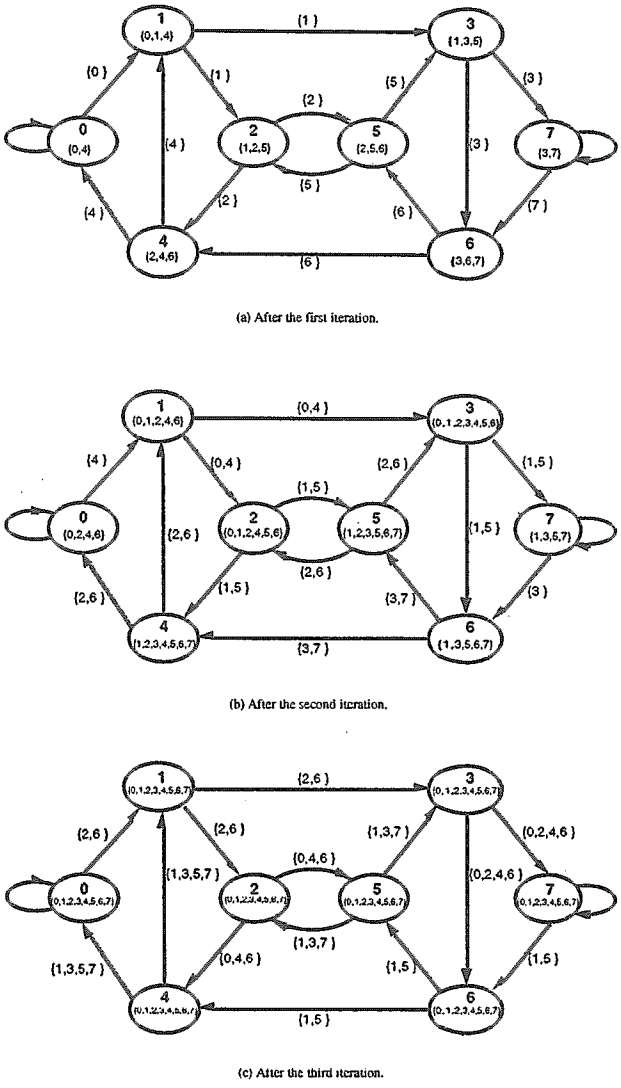

Figure 3: An example to illustrate the operations of AlltoAllBroadcast algorithm.

show that after d iterations the node X will broadcast its message to all other nodes with distance $\leq d$, where $1 \leq d \leq k$. Let nodes $X = x_k x_{k-1} \ldots x_1$ and $Y = x_{k-d} x_{k-d-1} \ldots x_1 y_d y_{d-1} \ldots y_1$ be any two nodes with distance $D(X,Y) = d$ on a $DDB(k)$ network, where $x_i, y_i \in \{0,1\}$, for $1 \leq i \leq k$. Assume that S_i is the set of nodes containing the message from node X after the ith iteration. Initially, let $S_0 = \{X\}$. Then $S_1 = \{X, x_{k-1}x_{k-2}\ldots x_1 0, x_{k-1}x_{k-2}\ldots x_1 1\}$ and $x_{k-1}x_{k-2}\ldots x_1 y_d \in S_1$ after the first iteration. Similarly, $S_2 = S_1 \cup \{x_{k-2}x_{k-3}\ldots x_1 00,$ $x_{k-2}x_{k-3}\ldots x_1 01, x_{k-2}x_{k-3}\ldots x_1 10, x_{k-2}x_{k-3}\ldots x_1 11\}$ and $x_{k-2}x_{k-3}\ldots x_1 y_d y_{d-1} \in S_2$ after the second iteration. Therefore, after d iterations, $x_{k-d}\ldots x_1 y_d \ldots y_1 \in S_k$, that is, node Y receives the message from node X after d iterations, where $1 \leq d \leq k$. As a consequence, all nodes will receive all messages from all other nodes on the $DDB(k)$ net-

work after k iterations. □

Before continuing to analyze the running time of AlltoAllBroadcast algorithm, the following definition [4] is required.

Definition 2 *1. Startup time (t_s) is the time required to handle a message at the sending node. This includes the time to prepare the message, the time to execute the routing algorithm, and the time to establish an interface between the node and the router.*
2. Per-character transfer time (t_c) is the time to transmit a character through a link.

Theorem 4 *Assuming that each node broadcasts at most one message, the expected running time of algorithm* **AlltoAllBroadcast** *is $O(N)$, where N is the number of nodes of a $DDB(k)$ network.*

Proof: Let each message has a fixed length with m characters. Then the execution time of AlltoAllBroadcast algorithm is:

$$T_{AlltoAllBroadcast} = \sum_{i=1}^{\log N} (t_s + 2^{i-1} t_c m) \quad (3)$$
$$= t_s \log N + t_c m(N-1) = O(N) \quad (4)$$

where 2^{i-1} is the maximum number of messages to be broadcast at the ith iteration. In the above calculation, each node is assumed to have the same startup time t_s and to be started up simultaneously in every iteration. □

5 Conclusion

In this paper, we have proposed algorithms for one-to-all broadcast and all-to-all broadcast. Both algorithms make full use of the incoming edges and the outgoing edges of binary directed de Bruijn networks. Since each edge of the underlying network is unidirectional, a node could not be connected directly to its parent nodes. To construct our one-to-all algorithm, the distance between any two nodes is defined and a message propagation rule is founded to guarantee that no redundant messages are received by any node on the network. The resulting algorithm is optimal and without any redundant messages passed through any node. Finally, an optimal all-to-all broadcast algorithm is also proposed.

References

[1] J. C. Bermond and C. Peyrat, "de Bruijn and Kautz networks: a competitor for the hypercube," *Hpercube and Distributed Computers: Proceedings of the First European Workshop,* pp. 279-293, October 1989.

[2] J. C. Bermond and P. Fraigniaud, "Broadcasting and gossiping in de Bruijn networks," *SIAM Journal on Computing,* Vol. 23, No. 1, pp. 212-225, February 1994.

[3] E. Ganesan and D. K. Pradham, "Optimal broadcasting in binary de Bruijn networks and hyper-de Bruijn networks," *Proceedings of Seventh International Parallel Processing Symposium,* pp. 655-660, April 1993.

[4] Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis,"*Introduction to Parallel Computing: Design and analysis of Algorithms,*" Redwood City, California: The Benjamin/Cummings Publishing Company, Inc., 1994.

[5] F. Thomson Leighton,"*Introduction to Parallel Algorithms and Architectures: Arrays, Tree, and Hypercubes,*" San Mateo, California: Morgan Kaufmann Publishers, 1992.

[6] Z. Liu, "Optimal routing in the de Bruijn networks," *Proceedings of the 10th International Conference on Distributed Computing Systems,* pp. 537-544, June 1990.

[7] M. R. Samatham and D. K. Pradham, "The de Bruijn multiprocessor network: a versatile parallel processing and sorting network for VLSI," *IEEE Transactions on Computers,* Vol. 38, No. 4, pp. 567-581, April 1989.

[8] Eric J. Schwabe, "On the computational equivalence of hypercube-derived networks," *Proceedings of the 2nd Annual ACM Symposium on Parallel Algorithms and Architectures,* pp. 388-397, July 1990.

[9] Kumar N. Sivarajan and Rajiv Ramaswami, "Lightwave networks based on de Bruijn graphs," *IEEE Transactions on Networking,* Vol. 2, No. 1, pp. 70-79, February 1994.