Enhanced de Bruijn Graphs

Osman Guzide
Department of CIS
Shepherd University
Shepherdstown, WV 25443

Meghanad D. Wagh
Department of ECE
Lehigh University
Bethlehem, PA 18015

Abstract – This paper shows that by adding a small number of strategically placed edges, one can lower the diameter of de Bruijn graphs. The enhanced de Bruijn graph of dimension n on alphabet of size k has k^n vertices, a node degree bounded by k^2 and a diameter of $\lfloor n/2 \rfloor$. We obtain a simple routing algorithm and show that the average distance between a pair of nodes in the new graph is about 50% less than the original de Bruijn graph. We prove that the new graph is isomorphic to a direct product of two de Bruijn graphs. We also discuss embedding of de Bruijn graphs and Hypercubes into the new graph.

1 Introduction

There has been a considerable interest in de Bruijn graphs for applications ranging from distributed hash tables and peer-to-peer networks to parallel machines and optical architectures [1-4]. In this paper we show that by adding a small number of strategically placed edges, one can lower the diameter of the graph substantially. The new graph, which we call as the *Enhanced de Bruijn Graph* (EDB) retains all the good properties of the de Bruijn graph such as the constant node degree and ability to map many algorithmic structures including maximal binary trees and cycles of all lengths. We obtain the routing strategy and the diameter of the EDB and show that it is isomorphic to a product of two de Bruijn graphs. It is also possible to map a standard de Bruijn graph on EDB. In addition, we give an embedding of a hypercube in the EDB and characterize it by its dilation and congestion.

A generalized directed de Bruijn graph $\overrightarrow{DB}(k,n)$ has k^n vertices, each labeled with a string of n symbols from the set $Z_k = \{0, 1, \ldots, k-1\}$. n is called the degree of the de Bruijn graph. A vertex $(e_{n-1}, e_{n-2}, ..., e_1, e_0)$, $e_i \in Z_k$ of $\overrightarrow{DB}(k,n)$ is connected to all vertices $(e_{n-2}, e_{n-3}, ..., e_1, e_0, x)$, $x \in Z_k$. Clearly, the node degree of $\overrightarrow{DB}(k,n)$ is at most

k. A generalized bidirectional de Bruijn graph $DB(k,n)$ uses the same set of vertices as $\overrightarrow{DB}(k,n)$. It has all the edges of $\overrightarrow{DB}(k,n)$ and in addition, has edges from any vertex $(e_{n-1}, e_{n-2}, \ldots, e_1, e_0)$ to all vertices $(y, e_{n-1}, e_{n-2}, \ldots, e_1)$, $y \in Z_k$. Thus the edges in both $DB(k,n)$ and $\overrightarrow{DB}(k,n)$ connect the same pairs of vertices, but in the latter, they are bidirectional.

We now define the two new graphs which form the subject of this paper. The *directed enhanced de Bruijn graph* $\overrightarrow{EDB}(k,n)$ is defined on the same set of vertices as $\overrightarrow{DB}(k,n)$, but with a connectivity as follows. A vertex $(e_{n-1}, e_{n-2}, \ldots, e_1, e_0) \in \overrightarrow{EDB}(k,n)$ is connected to n^2 vertices $(e_{n-2}, e_{n-3}, \ldots, e_{\lfloor n/2 \rfloor}, y, e_{\lfloor n/2 \rfloor -2}, \ldots, e_0, x)$, for $x, y \in Z_k$.

Similarly, the *bidirectional enhanced de Bruijn graph* $EDB(k,n)$ is defined as a graph which has the same vertices as the graph $DB(k,n)$, but with each vertex $(e_{n-1}, e_{n-2}, \ldots, e_1, e_0) \in EDB(k,n)$ connected to $2n^2$ vertices $(e_{n-2}, e_{n-3}, \ldots, e_{\lfloor n/2 \rfloor}, y, e_{\lfloor n/2 \rfloor -2}, \ldots, e_0, x)$ and $(x, e_{n-1}, e_{n-2}, \ldots, e_{\lceil n/2 \rceil}, y, e_{\lceil n/2 \rceil -1}, \ldots, e_1)$, for all $0 \leq x, y < k$.

Note that edges in both $\overrightarrow{EDB}(k,n)$ and in $EDB(k,n)$ connect the same pair of nodes, but in the latter's case, they are bidirectional. They both have a bounded node degree and have $\overrightarrow{DB}(k,n)$ and $DB(k,n)$ respectively as their subgraphs. This implies that any algorithms using the original de Bruijn graphs can run on these graphs without any performance loss.

In the rest of the paper, we let $k = 2$, i.e., the set of alphabet over which the vertex labels are defined is limited to $\{0, 1\}$. This has the important practical implication of limiting the node degree to 4 for the directed graph and to 8 for the bidirectional graph. However, most of the results obtained herein can be extended to any k.

2 Routing and diameter

Routing strategy and diameter are important properties of any interconnection network. A lower diameter and a good routing strategy imply lower communication delays in the peer-to-peer networks as well as in the parallel computers utilizing the network. When $k = 2$, each of the 2^n vertices in $\overrightarrow{EDB}(k,n)$ has an n-bit binary label and has edges to at most four other vertices. We now provide a simple routing from vertex $v_1 = (a_{n-1}, a_{n-2}, \ldots, a_0)$ to vertex $v_2 = (b_{n-1}, b_{n-2}, \ldots, b_0)$. Let x_i denote the vertices along the path. Then the following algorithm provides a path from v_1 to v_2.

1. Let $x_0 = v_1$ and $i = 0$.

2. Let $x_i = (c_{n-1}, c_{n-2}, \ldots, c_0)$.
   Go to the node $x_{i+1} = (c_{n-2}, c_{n-3}, \ldots, c_{\lfloor n/2 \rfloor}, b_{n-1-i}, c_{\lfloor n/2 \rfloor -2}, \ldots, c_0, b_{\lceil n/2 \rceil -1-i})$.

3. If $x_{i+1} = v_2$, then the path is complete.
   Otherwise increment i by 1 and go back to step 2.

As an illustration, consider $EDB(2,6)$. To go from $(a_5a_4\ldots a_0)$ to $(b_5b_4\ldots b_0)$ the path is: $(a_5a_4a_3a_2a_1a_0) \rightarrow (a_4a_3b_5a_1a_0b_2) \rightarrow (a_3b_5b_4a_0b_2b_1) \rightarrow (b_5b_4b_3b_2b_1b_0)$. Similarly, In another network, $EDB(2,7)$, the path is: $(a_6a_5a_4a_3a_2a_1a_0) \rightarrow (a_5a_4a_3b_6a_1a_0b_3) \rightarrow (a_4a_3b_6b_5a_0b_3b_2)$ $\rightarrow (a_3b_6b_5b_4b_3b_2b_1) \rightarrow (b_6b_5b_4b_3b_2b_1b_0)$.

Note that the path is complete when the destination is reached and one does not have to do all the $\lceil n/2 \rceil$ hops. But clearly the maximum distance between any pair of vertices of $\overrightarrow{EDB}(2,n)$ is $\lceil n/2 \rceil$. Further, the distance between vertices $(00\ldots0)$ and $(11\ldots1)$ is exactly $\lceil n/2 \rceil$. Therefore we have the following result.

Theorem 1. The diameter of $\overrightarrow{EDB}(2,n)$ is $\lceil n/2 \rceil$.

For $EDB(2,n)$, similar path can be worked out, but one may be able to shift in either direction. Consequently one has

Theorem 2. The diameter of $EDB(2,n)$ is $\lfloor n/2 \rfloor$.

In case of $EDB(2,n)$, it is also possible to get a shorter path than the one that is shown here by comparing the source and destination label strings and then using a set of left shifts followed by a set of right shifts and then doing a few more left shifts. But for most practical applications, such a complicated strategy does not pay off.

3 Mappings

Mapping of one graph on another is important for several reasons. Firstly, matching the algorithm structure to the network architecture minimizes communication overheads. In addition, if a well known topology can be mapped on a new architecture, then all the algorithms developed for the known topology may be quickly translated for the new architecture. With this in mind, we provide three results in this section. We show that the $\overrightarrow{EDB}(2,n)$ and $EDB(2,n)$ are isomorphic to products of de Bruijn networks. In addition, a hypercube with 2^n vertices may be mapped to $EDB(2,n)$ with a dilation bounded above by $\lceil n/4 \rceil$.

Theorem 3. The enhanced de Bruijn graphs are isomorphic to direct product of de Bruijn graphs. In particular,

$$\begin{aligned}
\overrightarrow{EDB}(k,n) &\equiv \overrightarrow{DB}(k,\lceil n/2 \rceil)\times\overrightarrow{DB}(k,\lfloor n/2 \rfloor).\\
EDB(k,n) &\equiv DB(k,\lceil n/2 \rceil)\times DB(k,\lfloor n/2 \rfloor).\\
\overrightarrow{EDB}(2,n) &\equiv \overrightarrow{DB}(2,\lceil n/2 \rceil)\times\overrightarrow{DB}(2,\lfloor n/2 \rfloor).\\
EDB(2,n) &\equiv DB(2,\lceil n/2 \rceil)\times DB(2,\lfloor n/2 \rfloor).
\end{aligned}$$

Proof. We prove only the first of these equivalences. Other proofs are similar. To show the isomorphism between $\overrightarrow{EDB}(k,n)$ and $\overrightarrow{DB}(k,\lceil n/2 \rceil)\times\overrightarrow{DB}(k,\lfloor n/2 \rfloor)$, define a mapping

$\phi : \overrightarrow{EDB}(k,n) \to \overrightarrow{DB}(k,\lceil n/2\rceil)\times\overrightarrow{DB}(k,\lfloor n/2\rfloor)$ as

$$\phi(e_{n-1},\ldots,e_0) = (e_{n-1},\ldots,e_{\lfloor n/2\rfloor}) \times (e_{\lfloor n/2\rfloor-1},\ldots,e_0).$$

One can easily show that the mapping ϕ is one-to-one and onto. Further, any edge in $\overrightarrow{EDB}(k,n)$,

$$(e_{n-1},e_{n-2},\ldots,e_1,e_0) \to (e_{n-2},e_{n-3},\ldots,e_{\lfloor n/2\rfloor},y,e_{\lfloor n/2\rfloor-2},\ldots,e_0,x), \qquad x,y\in Z_k$$

corresponds to the following edge in $\overrightarrow{DB}(k,\lceil n/2\rceil)\times\overrightarrow{DB}(k,\lfloor n/2\rfloor)$:

$$((e_{n-1},e_{n-2},\ldots,e_{\lfloor n/2\rfloor}),\ (e_{\lfloor n/2\rfloor-1},\ldots,e_1,e_0)) \to$$
$$((e_{n-2},e_{n-3},\ldots,e_{\lfloor n/2\rfloor},y),\ (e_{\lfloor n/2\rfloor-2},\ldots,e_0,x)), \qquad x,y\in Z_k.$$

Thus the two graphs $\overrightarrow{EDB}(k,n)$ and $\overrightarrow{DB}(k,\lceil n/2\rceil)\times\overrightarrow{DB}(k,\lfloor n/2\rfloor)$ are isomorphic. ∎

On the other hand, normal generalized de Bruijn network, $DB(k,n)$, can also be mapped on $EDB(k,n)$ as the following theorem shows.

Theorem 4. One can embed the $DB(k,n)$ in $EDB(k,n)$ with unit dilation, load and congestion.

Proof. Vertices of both $DB(k,n)$ and $EDB(k,n)$ are n bit strings made up from symbols in Z_k. We map vertices with same labels to one another and show that all the edges of the $DB(k,n)$ are still present in $EDB(k,n)$. Consider an edge $(\beta W, W\alpha)$ of $DB(k,n)$, where $\alpha,\beta\in Z_k$ and $W\in Z_k^{n-1}$. On the other hand, each edge of $EDB(k,n)$ has the form $(\beta U\gamma V, U\delta V\alpha)$ for $\alpha,\beta,\gamma,\delta\in Z_k$, $U\in {Z_k}^{\lceil n/2\rceil-1}$ and $V\in {Z_k}^{\lfloor n/2\rfloor-1}$. Note that different values of α and δ provide the different edges from vertex $\beta U\gamma V$. By choosing only those edges for whom $\delta=\gamma$, one can see that the chosen edges of $EDB(k,n)$ have the form $(\beta U\gamma V, U\gamma V\alpha)$. Finally, by representing string $U\gamma V$ by $W\in Z_k^{n-1}$, these edges take exactly the same form as the edges of a $DB(k,n)$. ∎

We now show an embedding of the Hypercube into $EDB(2,n)$.

Theorem 5. There exists an embedding f of hypercube of dim n, H_n, into $EDB(2,n)$ with dilation less than or equal to $\lceil n/4\rceil$.

Proof. Define mapping ϕ of the vertices of H_n into $EDB(2,n)$ as follows. If vertex $(x_{n-1}x_{n-2}\ldots x_0)\in H_n$ has even parity, then map it to a vertex of $EDB(2,n)$ of the same label. Otherwise, map it to vertex obtained by rotating the label right by $\lceil n/4\rceil$ bits, i.e., to label $x_{\lceil n/4\rceil-1}\ldots x_1x_0\ldots x_{\lceil n/4\rceil+1}x_{\lceil n/4\rceil}$. Clearly this vertex mapping is one-to-one.

Consider an edge (u, v) in H_n. The binary strings u and v must be different in exactly one bit. Since their parity is different, they would be mapped to $EDB(2, n)$ vertices whose labels are rotated versions of one another by $\lceil n/4 \rceil$ rotations except they also differ in one bit. For example, for $n = 9$, if u and v differ in bit 1, then the vertices to which they are mapped will look like $\overline{u} = (a_8a_7a_6a_5a_4a_3a_2a_1a_0)$ and $\overline{v} = (a_2\overline{a_1}a_0a_8a_7a_6a_5a_4a_3)$. Recall that in $EDB(2, n)$, all edges look either like left shifts of labels except that the bits at positions 0 and $\lfloor n/2 \rfloor$ can be assigned arbitrary binary values; or as right shifts with bits at positions $\lceil n/2 \rceil - 1$ and $n - 1$ assigned arbitrary binary values. The single bit in which images of rotated u and v differ cannot be more than $\lceil n/4 \rceil$ away from the positions where the bits are changed in an edge. Thus one can lay out a path between $\overline{u}$ and $\overline{v}$ of a length not more than $\lceil n/4 \rceil$. ∎

One can prove that the mapping presented in Theorem 5 has a vertex congestion bounded by $2\lceil n/4 \rceil(\lceil n/4 \rceil - 1)$ and an edge congestion of $\max\{1, 2\lceil n/4 \rceil(\lceil n/4 \rceil - 1)\}$.

4 Average distance

As shown in Sec. 2, the diameter of the extended de Bruijn network $EDB(2, n)$, is $\lfloor n/2 \rfloor$ which is less than half of the diameter n of the standard de Bruijn network $DB(2, n)$. But in many applications, this worst distance is not as important as the average distance. We have therefore compared the average distance between any pair of vertices in the enhanced network $EDB(2, n)$ with the standard de Bruijn network $DB(2, n)$, and the product of de Bruijn networks $PDB(2, n)$ as defined in [5].

For each network, two curves are plotted corresponding to the directed and bidirectional networks. It can be seen that as expected, the average distance in a directional graph is larger than its bidirectional counterpart. The Average distance of enhanced de Bruijn networks is much superior to that of other networks. For example, in directional networks of degree 14 ($n = 14$), while de Bruijn and product de Bruijn networks have average distances of 12.36 and 10.84 respectively, the enhanced de Bruijn network has an average distance of only 6.58. For bidirectional networks, these numbers are 11.52, 9.30 and 6.26 respectively.

As a final comment, the enhanced de Bruijn networks presented here admit the same kind of algebraic models based on finite fields presented in [6]. Using these models, one would be able to explore the structural properties and mappings of these networks to a much greater depth.

5 References

[1] O. Yang and Z. Feng, "DBG MANs and their routing performance," *IEE Proc. on Communications*, vol. 147, pp. 32–40, Jan 2000.

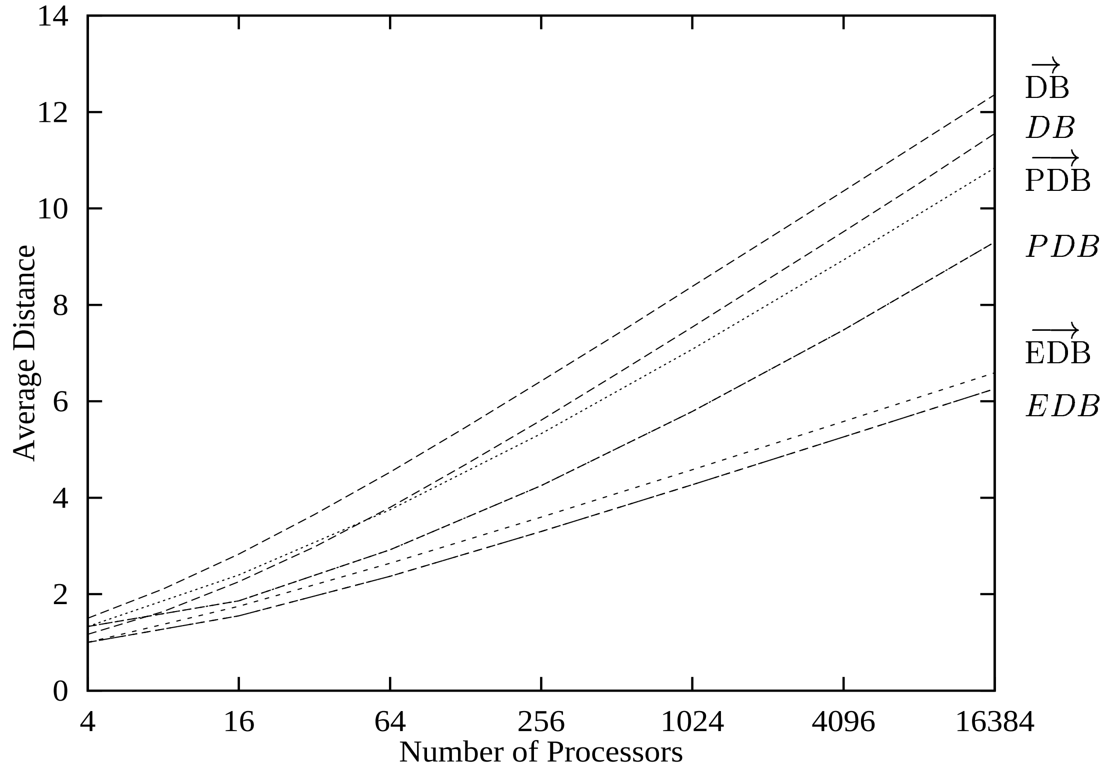

Figure 1: Average distances in (Standard, Product and Enhanced) de Bruijn graphs.

[2] D. Wei, S. Rajasekaran, and K. Naik, "Efficient algorithms for selection and sorting of large distributed files on De Bruijn and hypercube structures," *Int. J. of Foundations of Computer Science*, vol. 14, pp. 1129–1146, Dec 2003.

[3] A. Datta, S. Girdzijauskas, and K. Aberer, "On de Bruijn routing in distributed hash tables: There and back again?," in *Proc. of the Fourth Int. Conf. on Peer-to-Peer Computing*, 2004.

[4] W. Z. Song, X. Li, Y. Wang, and W. Wang, "dBBlue: low diameter and self-routing Bluetooth scatternet," *Journal of Parallel and Distributed Computing*, vol. 65, pp. 178–190, Feb 2005.

[5] A. L. Rosenberg, "Product-shuffle networks: toward reconciling shuffles and butterflies," *Discrete Applied Math*, vol. 37/38, pp. 465–488, 1992.

[6] M. Wagh and J. Mo, "An analytical setting and mappings on the product of generalized de bruijn graphs," in *Proc. of ISCA the 10th Int. Conference on Parallel and Distributed Computing Systems*, (New Orleans), pp. 253–257, October 1–3 1997.