Enhanced Butterfly: A Cayley Graph with Node Degree 5

Osman Guzide
Department of CIS
Shepherd University
Shepherdstown, WV 25443
oguzide@shepherd.edu

Meghanad D. Wagh
Department of ECE
Lehigh University
Bethlehem, PA 18015
mdw0@lehigh.edu

Abstract

We describe a new interconnection network called the Enhanced Butterfly (EB_n). It is a Cayley graph with $n2^n$ nodes, each incident with five edges. This new network has a diameter n which compares favorably to the wrap-around butterfly (B_n) with the same number of nodes and a diameter of $\lfloor 3n/2 \rfloor$. B_n is a subgraph of EB_n. The simple and optimal routing algorithms for this network are presented.

1 Introduction

Performance of a distributed-memory message-passing parallel computer is often limited by the underlying interconnection network. The choice of the network is complicated by the conflicting requirements imposed on it. For example, for high communication throughput, one needs a completely connected network. But this implies a large node degree of each processor node and therefore high costs associated with the communication architecture within each processor. In addition, the complexity of physical connections of a complete network is overwhelming for a parallel processor of any realistic size. Thus in a useful interconnection network, the total number of edges is far fewer than in a complete network and each edge is chosen judiciously. A message is passed between unconnected nodes by routing it along the best possible path going through other nodes. Since the communication delay may be modeled by the length of such a path, networks may be evaluated by the worst path length encountered between any pair of nodes, known as the *diameter* of the network.

The interconnection networks with a fixed node degree are particularly attractive. De Bruijn networks [1], cube-connected cycles [2], wrap-around meshes [3], trivalent Cayley graphs [4] and wrap-around butterfly networks [3] are examples of fixed node degree networks. In these networks, the communication architecture of the processor and the associated communication hardware does not limit the size of the parallel machine. Thus for example, using processors with four communication edges, the largest hypercube one may build is limited to 16 processors. On the other hand, the same processors may be used to build arbitrarily large de Bruijn networks. But constant node degree networks have their own drawbacks. The de Bruijn networks are not *symmetric* and therefore do not allow simple task migration or fault avoidance. Mesh networks support some parallel algorithms well but in general, because of a large diameter, are ineffective in other applications. Trivalent Cayley graphs are relatively unexplored [5] and have not yet been proven to support many parallel algorithm mappings. Wrap-around butterfly networks are symmetric and support a large number of parallel algorithm embeddings [6, 7, 8]. They can also emulate other networks efficiently [9]. Unfortunately, they do not have the same low diameter as the de Bruijn networks.

In this paper we show that by adding one more edge between properly chosen pairs of nodes in a wrap-around butterfly, one can reduce its diameter by 50%. Addition of the new edges thus increases the node degree of the network from 4 to 5 while maintaining its symmetry. Note that such a technique has been used previously to obtain a *folded hypercube* with 50% smaller diameter by adding an additional edge between complementary hypercube nodes [10]. Similarly by increasing the node degree of the de Bruijn graph by four, an *enhanced de Bruijn graph* with 50% smaller diameter has been obtained [11].

In Section 2 we provide the definition and properties of the new network referred here as the *Enhanced Butterfly Network*. We obtain the explicit Cayley graph representation of the network. Section 3 gives the optimal path algorithm for this network. Finally, Section 4 provides a comparison of this network with other popular networks commonly used in parallel processing.

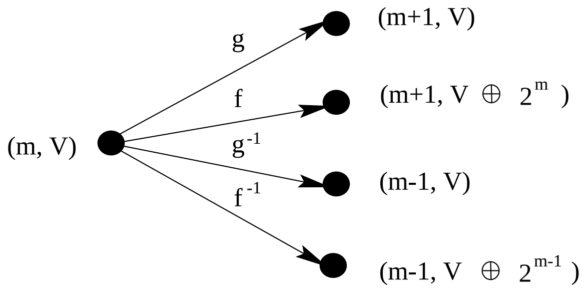

Figure 1: Connections from node $(m,\ V)$ in the butterfly network.

2 Definition and Properties of the Enhanced Butterfly Network EB$_n$

The *Wrap-around Butterfly Network* of degree $n \geq 3$, B_n, is a graph with vertex set $Z_n \times \{0,1\}^n$. A vertex $(m,\ V)$ is connected to the four vertices shown in Fig. 1. We refer to these connections as the f, g, f^{-1} and g^{-1} edges. Clearly, the first and third edges are inverses of each other and so are the second and fourth[1]. Note that in this figure since $m \in Z_n$, $m+1$ and $m-1$ are evaluated modulo n. V is a n-bit vector whose bits are indexed 0 to $n-1$ with the rightmost bit being bit 0. Thus in this definition, 2^m refers to a length n vector with 1 in position m and 0's everywhere else. Similarly an all zero vector is often written simply as 0.

B_n is symmetric and has $n2^n$ vertices and $n2^{n+1}$ edges. Its node degree is 4 and its diameter is $\lfloor 3n/2 \rfloor$. Degree 4 Cayley graph recently proposed [12] is identical to B_n. Cube Connected Cycles is a subgraph of B_n. B_n supports many parallel algorithms well. It is known [13] that one can map on B_n (with dilation 1) cycles of almost all lengths except 6 and 10 when n is odd, and cycles of all even lengths except 6 and 10 when n is even. Similarly, B_n supports mapping of largest possible binary trees with relatively low dilation, e.g., dilation 2 for $n < 16$ [13].

The *Enhanced butterfly network* of degree n, EB_n is a graph with the same vertex set as B_n. Its connectivity is shown in Fig. 2. One may note that the set of edges of EB_n is a superset of the set of edges of B_n and the new h edge is bidirectional, i.e., $h^{-1} = h$.

Fig. 3 shows EB_3, the degree 3 Enhanced butterfly network. (Bottom row of nodes in this figure represents the same nodes as the top row. They are duplicated for clarity.) The vertical, diagonal and horizontal edges represent g, f and h edges respectively.

Clearly the degree of each node in EB_n is 5. The total number of edges in EB_n is $5n2^{n-1}$. Following theorem shows that EB_n can be viewed as a Cayley graph.

[1]By inverse edges we mean the following: If edge f (or g) goes from $(m_1,\ V_1)$ to $(m_2,\ V_2)$, then the edge f^{-1} (or g^{-1}) goes from $(m_2,\ V_2)$ to $(m_1,\ V_1)$.

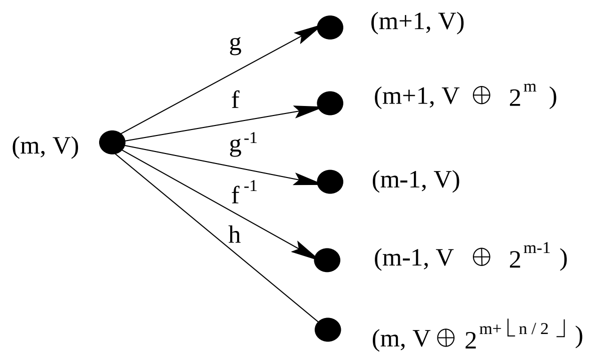

Figure 2: Connections from node $(m,\ V)$ in the Enhanced butterfly network.

Theorem 1. (Cayley Characterization) EB_n is a Cayley graph.

Proof. The vertices of EB_n form a group Γ under the following group operation $\circ$:
For any $(m_1,\ V_1)$, $(m_2,\ V_2) \in \Gamma$, define

$$(m_1,\ V_1) \circ (m_2,\ V_2) \ = \ (m_1 + m_2, V_1 \oplus \overset{m_1}{\overleftarrow{V_2}}).$$

In this group operation, $\overset{m}{\overleftarrow{V}}$ represents a vector obtained by left rotation of V by m times. Similarly, $\overset{m}{\overrightarrow{V}}$ will represent a right rotation of V by m places. Since we are dealing with vectors of length n in EB_n, m left rotations and $n-m$ right rotations are identical.

To see the associativity of this group operation, note that

$$\begin{aligned}&((m_1,\ V_1) \circ (m_2,\ V_2)) \circ (m_3,\ V_3)\\ &= (m_1 + m_2, V_1 \oplus \overset{m_1}{\overleftarrow{V_2}}) \circ (m_3,\ V_3)\\ &= (m_1 + m_2 + m_3,\ V_1 \oplus \overset{m_1}{\overleftarrow{V_2}} \oplus \overset{m_1+m_2}{\overleftarrow{V_3}}).\end{aligned}$$

and

$$\begin{aligned}&(m_1,\ V_1) \circ ((m_2,\ V_2) \circ (m_3,\ V_3))\\ &= (m_1,\ V_1) \circ (m_2 + m_3, V_2 \oplus \overset{m_2}{\overleftarrow{V_3}})\\ &= (m_1 + m_2 + m_3,\ V_1 \oplus \overset{m_1}{\overleftarrow{(V_2 \oplus \overset{m_2}{\overleftarrow{V_3}})}}).\end{aligned}$$

Therefore,

$$\begin{aligned}&((m_1,\ V_1) \circ (m_2,\ V_2)) \circ (m_3,\ V_3)\\ &= (m_1,\ V_1) \circ ((m_2,\ V_2) \circ (m_3,\ V_3)),\end{aligned}$$

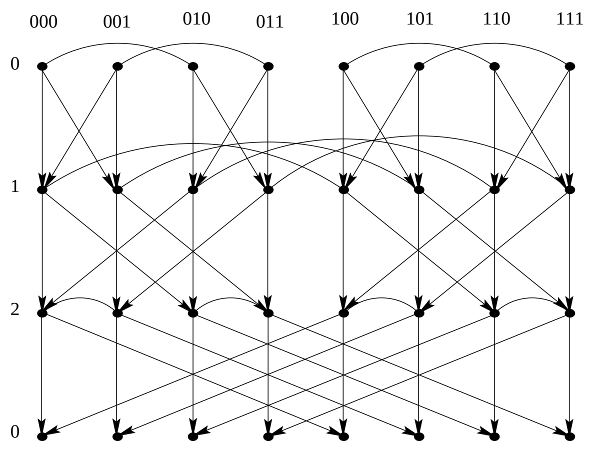

Figure 3: Enhanced Butterfly network of degree 3 (EB_3). The row numbers and column headings represent the two coordinates of the node labels.

showing that the group operation is associative. One may also verify that under this operation, $(0,\ 0) \in \Gamma$ is the identity element and every element $(m,\ V) \in \Gamma$ has the inverse $(-m,\ \overset{m}{\overrightarrow{V}})$. Therefore Γ is indeed a group.

To show that EB_n is a Cayley graph, we should be able to produce generators $\gamma \in \Gamma$ such that for any $x \in \Gamma$, vertices x and $x \circ \gamma$ are connected. One can check that the set of elements

$$S = \{(1,0),\ (-1,0),\ (1,1),\ (-1,2^{n-1}),\ (0,2^{\lfloor n/2 \rfloor})\}$$

are the five generators that correspond to the five edges from each node. Set S is also closed under inverse. Therefore network EB_n is a Cayley graph. ∎

This Cayley graph characterization implies the following result.

Corollary 1. *EB_n is a node symmetric network.*

Note that the group Γ is not Abelian. Theorem 1 may facilitate development of efficient mappings on EB_n. It may also be used to define the graph automorphism $\phi\ :\ \Gamma\ \rightarrow\ \Gamma$ that maps a specific node $(m_1,\ V_1)$ to an arbitrary node $(m_2,\ V_2)$:

$$\phi(m,\ V) = (m - m_1 + m_2,\ V_2 \oplus (\overset{m_2-m_1}{\overleftarrow{V_1 \oplus V}})).$$

This automorphism will be used later to obtain certain paths in EB_n.

A property of interconnection networks that is of great practical importance is the diameter of the network. Following theorem deals with that.

Theorem 2. (Diameter) *The diameter of EB_n is n.*

Proof. Since EB_n is symmetric (Corollary 1), we only need to show that the distance from any arbitrary node $(m,\ V)$ to node $(0,\ 0)$ is at most n. We achieve that by specifying a path as follows:
case 1. If $m \leq \lfloor n/2 \rfloor$,

```
for $i = m$ to n-1 do
  if ($\lceil n/2 \rceil \leq i < m + \lceil n/2 \rceil$)
    if the ($i - \lceil n/2 \rceil$)th bit of $V$ is 1, then
      go along the $h$ edge, i.e.,
      replace $(i,\ V)$ by $(i,\ V \oplus 2^{i-\lceil n/2 \rceil})$
  if the $i$-th bit of current $V$ is 1, then
    go along the $f$ edge, i.e.,
    replace $(i,\ V)$ by $(i+1,\ V \oplus 2^i)$
  else
    go along the $g$ edge, i.e.,
    replace $(i,\ V)$ by $(i+1,\ V)$
```

case 2. If $m > \lfloor n/2 \rfloor$,

```
for $i = m$ downto 1 do
  if ($(m - \lfloor n/2 \rfloor) \leq i < \lceil n/2 \rceil$)
    if the ($i + \lfloor n/2 \rfloor$)th bit of $V$ is 1, then
      go along the $h$ edge, i.e.,
      replace $(i,\ V)$ by $(i,\ V \oplus 2^{i+\lfloor n/2 \rfloor})$
  if the ($i-1$)-th bit of current $V$ is 1, then
    go along the $f^{-1}$ edge, i.e.,
    replace $(i,\ V)$ by $(i-1,\ V \oplus 2^{i-1})$
  else
    go along the $g^{-1}$ edge, i.e.,
    replace $(i,\ V)$ by $(i-1,\ V)$
```

Clearly this algorithm transforms the first index of the label to 0. To see that the second index also gets converted to an all 0 vector, consider the two cases separately. in case 1, for each i, $m \leq i \leq n-1$, we convert a 1 at the i-th bit position of V using the f edge. In addition, for ($\lceil n/2 \rceil \leq i < m + \lceil n/2 \rceil$), we also convert the 1 bit at the $(i + \lfloor n/2 \rfloor)$ mod n-th bit position of V using the h edge. This affects bit positions 0 through $m-1$. Thus one can see that by the time i completes its entire iteration, all the bits of V would be zero. Therefore the final destination of this path is indeed $(0,0)$. The second case can be similarly proved.

To compute the path length, note that in the first case, one traverses h edges at most m times and f or g edges exactly $n-m$ times. Therefore the path length is at most n. Similarly in the second case, one uses h edges at most $n-m$ times and f^{-1} or g^{-1} edges exactly m times. Thus the path length is at most n in this case as well. ∎

Combined with the group automorphism, the path algorithm may be used to obtain path between an arbitrary pair of nodes in EB_n. To go from $(m_1,\ V_1)$ to $(m_2,\ V_2)$, one first finds an automorphism ϕ to map $(m_2,\ V_2)$ to $(0,0)$:

$$\begin{aligned}\phi(m,\ V) &= (m,\ V)\circ(m_2,\ V_2)^{-1}\\ &= (m-m_2, V\oplus \overset{\xleftarrow{m-m_2}}{V_2}).\end{aligned}$$

One then determines a path from $\phi(m_1,\ V_1)$ to $(0,0)$. By applying ϕ^{-1} to every node on this path, one may obtain the desired path between $(m_1,\ V_1)$ and $(m_2,\ V_2)$. (Note that $\phi^{-1}=(m_2,\ V_2)\circ(m,\ V)$.)

To illustrate this, consider the path in EB_{11} from $(1,10111011100)$ to $(5,10111100111)$. Since in this case, the function $\phi(1,10111011100)$ equals $(1,10111011100)(5,10111100111)^{-1}$ $=(7,11011000001)$, we first need to establish a path from (7,11011000001) to $(0,00000000000)$. But this is the exact path that was determined earlier. Therefore by translating each node according to ϕ^{-1}, one gets the desired path:

$$\begin{array}{llllll}(1,10111011100) & \xrightarrow{f^{-1}} & (0,10111011101) & \xrightarrow{g^{-1}}\\ (10,10111011101) & \xrightarrow{h} & (10,10111001101) & \xrightarrow{g^{-1}}\\ (9,10111001101) & \xrightarrow{h} & (9,10111000101) & \xrightarrow{g^{-1}}\\ (8,10111000101) & \xrightarrow{g^{-1}} & (7,10111000101) & \xrightarrow{h}\\ (7,10111000111) & \xrightarrow{g^{-1}} & (6,10111000111) & \xrightarrow{f^{-1}}\\ (5,10111100111) & & & \end{array}$$

It may also be noted that rather than translating each node one may simply choose to apply the same edge sequence as in the case of $\phi(m_1,\ V_1)$ to $(0,0)$ to get the final path.

3 Optimal Paths in EB_n

We will now describe the procedure to determine the optimal path between an arbitrary node $(m,\ V)$ and node $(0,0)$ in EB_n. Because of symmetry, this same algorithm may be extended to provide an optimal path between any arbitrary pair of nodes.

If $m\le\lfloor n/2\rfloor$, there are two possible paths to the final destination. The first path is the one described in case 1 of the proof of Theorem 2. The length D_L of this path is given by

$$D_L=(n-m)+weight(V\oplus(2^m-1)). \tag{1}$$

The second term in (1) represents the number of 1s in the m least significant bits of V.

The second possible path to go from (m,V) to $(0,0)$ is described by the following algorithm. It relies upon two constants, t_1 and t_2 satisfying $m\le t_1<\lceil n/2\rceil$ and $t_1+\lfloor n/2\rfloor\le t_2\le n$. Procedure to determine t_1 and t_2 is explained later.

```
for $i=m$ to $t_1-1$ do
  go from $(i,V)$ to $(i+1,V)$ along the $g$ edge
for $i=t_1$ downto $t_2+1$ mod $n$ (through 0) do
  if the $(i+\lfloor n/2\rfloor)$ mod $n$-th bit of $V$ is 1, then
     go along the $h$ edge, i.e.,
     replace $(i,\ V)$ by $(i,\ V\ \oplus\ 2^{i+\lfloor n/2\rfloor \ mod\ n})$
  if the $(i-1)$-th bit of $V$ is 1, then
     go along the $f^{-1}$ edge, i.e.,
     replace $(i,\ V)$ by $(i-1,\ V\ \oplus 2^{i-1})$
  else
     go along the $g^{-1}$ edge, i.e.,
     replace $(i,\ V)$ by $(i-1,\ V)$
for $i\ =\ t_2$ to $n-1$ do
  go from $(i,\ V)$ to $(i+1,\ V)$ along the $g$ edge
```

The length of this path, D_R, is given by

$$\begin{aligned}D_R&=(t_1-m)+(t_1+n-t_2)+(n-t_2)\\&+weight(V\oplus(2^{t_1+\lfloor n/2\rfloor}-2^{t_2-\lceil n/2\rceil}))\end{aligned}\tag{2}$$

Note that the weight function in (2) gives the number of 1's within bit positions $(t_2-\lceil n/2\rceil)$ to $(t_1+\lfloor n/2\rfloor)$ of V.

Constants t_1 and t_2 are chosen using the following procedure:

1. For $m\le i<\lceil n/2\rceil$, compute

$$\begin{aligned}H_L(i)&=2(i-m)+\\&weight(V\oplus(2^{i+\lfloor n/2\rfloor}-2^{m+1+\lfloor n/2\rfloor})).\end{aligned}\tag{3}$$

2. For $m\le i<\lceil n/2\rceil$, Choose the smallest integer $i'\ge i$ such that

   - $i'=i$, n is even and bit i' of V is 1.
   - $i'>i$ and either bit $i'-(n \bmod 2)$ or bit $(i'+\lfloor n/2\rfloor)$ mod n of V is 1.

   Note that for odd n, $i'>i$.
   Compute

$$\begin{aligned}H_R(i)=&\ 2(\lceil n/2\rceil-i')\\&+weight(V\oplus(2^{\lfloor n/2\rfloor-1}-2^{i'-(n \bmod 2)}))\end{aligned}\tag{4}$$

3. Choose t_1 equal to the value of i which minimizes

$$H=\min_{m\le i<\lceil n/2\rceil}(H_L(i)+H_R(i)) \tag{5}$$

   and choose $t_2=i'+\lfloor n/2\rfloor$ where i' denotes the corresponding integer as in step 2.

We now state the result about the optimal path.

Theorem 3. (Optimal path) The path given above with the smaller of the lengths D_L and D_R is the optimal path in EB_n.

The proof of this theorem is omitted for brevity. We illustrate this procedure by computing the optimal path in EB_{11} from $(2, 10100010011)$ to $(0, 00000000000)$.

The first path between these nodes using the algorithm in Theorem 2 has a length given by (1) and equals $D_L = 11$. To obtain the second path, one computes $t_1 = 3$ and $t_2 = 5 + \lfloor 11/2 \rfloor = 10$ as specified earlier. Thus from (2), $D_R = 8$. Since $D_R < D_L$, we use the second path given below:

$$(2, 10100010011) \xrightarrow{g} (3, 10100010011) \xrightarrow{h} (3, 10000010011) \xrightarrow{g^{-1}} (2, 10000010011) \xrightarrow{f^{-1}} (1, 10000010001) \xrightarrow{f^{-1}} (0, 10000010000) \xrightarrow{f^{-1}} (10, 00000010000) \xrightarrow{h} (10, 00000000000) \xrightarrow{g} (0, 00000000000).$$

When $m > \lfloor n/2 \rfloor$, one can obtain the optimal path between any $(m,\ V)$ and $(0, 0)$ by simply transforming the problem using symmetry of the network into one that has $m \le \lfloor n/2 \rfloor$. To get the desired optimal path in this case, we first find the optimal path from $(-m,\ \overset{m}{\overrightarrow{V}})$ to $(0, 0)$. Note that since $m > \lfloor n/2 \rfloor$, $(-m) \bmod n \le \lfloor n/2 \rfloor$ and the optimal path algorithm described earlier is applicable in this case. Since this path is reversible (see footnote 1), this path is also the optimal path to go from $(0, 0)$ to $(-m,\ \overset{m}{\overrightarrow{V}})$. By multiplying (from the left) each node on this path by node $(m,\ V)$ (using the group operations in Γ defined earlier), one gets the optimal path from $(m,\ V)$ to $(0, 0)$. Alternately, after the path from $(-m,\ \overset{m}{\overrightarrow{V}})$ to $(0, 0)$ is obtained, one can read the sequence of edges used therein in reverse order and then employ the inverses of these edges to obtain the desired path from $(m,\ V)$ to $(0, 0)$.

We illustrate this procedure by computing the optimal path from node $(7, 11011000001)$ to node $(0, 00000000000)$ in EB_{11}.

Since $n = 11$, according to the procedure described above, we first have to get the optimal path from $(4, 10000011101)$ to $(0, 00000000000)$. For this, the first path has a length $D_L = 10$ given by (1). For the second path, t_1 and t_2 work out to be 4 and 10 respectively. Thus from (2), $D_R = 7$. Clearly the second path of length 7 is the optimal path. It is given by:

$$(4, 10000011101) \xrightarrow{f^{-1}} (3, 10000010101) \xrightarrow{f^{-1}} (2, 10000010001) \xrightarrow{g^{-1}} (1, 10000010001) \xrightarrow{f^{-1}} (0, 10000010000) \xrightarrow{f^{-1}} (10, 00000010000) \xrightarrow{h} (10, 00000000000) \xrightarrow{g} (0, 00000000000).$$

The optimal path from $(7, 11011000001)$ to $(0, 00000000000)$ is therefore given by

$$(7, 11011000001) \xrightarrow{g^{-1}} (6, 11011000001) \xrightarrow{h} (6, 11011000000) \xrightarrow{f} (7, 11010000000) \xrightarrow{f} (8, 11000000000) \xrightarrow{g} (9, 11000000000) \xrightarrow{f} (10, 10000000000) \xrightarrow{f} (0, 00000000000).$$

4 Conclusion

Wrap-around butterfly is a very useful network by virtue of its support of a wide variety of parallel algorithms, its ability to emulate other parallel architectures and its fixed node degree implying a low cost implementation. We have shown that with certain additional edges, one can get an *Enhanced Butterfly* network, EB_n. EB_n is symmetric, has $n2^n$ nodes, a fixed node degree of 5 and a low diameter n. It is a Cayley graph and therefore inherits all the nice properties of Cayley graphs including the symmetry. Table 1 compares EB_n with some popular networks used in parallel processing.

Table 1: Comparison of EB_n with wrap-around $M \times N$ mesh, wrap-around butterfly (B_n), hypercube (H_n), de Bruijn network (DB_n), folded hypercube (FH_n) and enhanced de Bruijn network(EDB_n).

	No. of proc.	Node degree	Diameter	Symmetry
Mesh	MN	4	$(M+N)/2$	$\surd$
B_n	$n \times 2^n$	4	$\lfloor 3n/2 \rfloor$	$\surd$
H_n	2^n	n	n	$\surd$
DB_n	2^n	4	n	
$\mathbf{EB_n}$	$\mathbf{n \times 2^n}$	**5**	$\mathbf{n}$	$\surd$
FH_n	2^n	$n+1$	$\lceil n/2 \rceil$	$\surd$
EDB_n	2^n	8	$\lfloor n/2 \rfloor$	

Wrap-around butterfly network B_n is a subgraph of EB_n; consequently, EB_n supports all the embeddings that B_n supports including mappings of cycles and trees. It is possible that some of the mappings on B_n may be improved on EB_n. The lower diameter of EB_n also results into lower average distance between a random pair of nodes of the network. For example, when $n = 6$, EB_n has 20% lower average distance as compared to B_n. When $n = 12$, this gain rises to 22%. A path of length at most n between an arbitrary pair of nodes of EB_n is easy to implement. Design of an optimal path is also possible.

References

[1] M. Samatham and D. Pradhan, "The de bruijn multiprocessor network: A versatile parallel processing and sorting network for vsli," *IEEE Trans. Computers*, vol. 38, no. 4, pp. 567–581, 1989.

[2] F. Preparata and J. Vuillemin, "The cube-connected cycles: A versatile network for parallel computation," *Comm. ACM*, vol. 24, no. 5, pp. 30–39, 1991.

[3] F. Leighton, *Introduction to parallel algorithms and architectures: Arrays, trees, hypercubes.* M. Kaufman Pub., 1992.

[4] P. Vadapalli and K. Srimani, "Trivalent cayley graphs for interconnection network," *Information Processing Letters*, vol. 54, pp. 329–335, 1995.

[5] M. D. Wagh and J. C. Mo, "Hamilton cycles in trivalent cayley graphs," *Information Processing Letters*, vol. 60, no. 4, pp. 177–181, 1996.

[6] S. Bhatt, F. Chung, J. W. Hong, F. Leighton, and A. Rosenberg, "Optimal simulation by butterfly networks," in *Proc. 20th Symp. Theory Comput.*, pp. 192–204, 1988.

[7] A. Gupta and S. E. Hambrusch, "Embedding complete binary trees into butterfly networks," *IEEE Trans. Computers*, vol. 40, pp. 853–863, Jul 1991.

[8] R. Feldmann and W. Unger, "The cube-connected-cycle is a subgraph of the butterfly network," *Parallel Processing Letters*, vol. 2, no. 1, pp. 13–19, 1992.

[9] S. Bhatt, F. R. K. Chung, J. Hong, F. Leighton, B. Obrenic, A. L. Rosenberg, and E. J. Schwabe, "Optimal emulation by butterfly-like networks," *JACM*, vol. 43, pp. 293–330, Mar 1996.

[10] A. El-Amawy and S. Latifi, "Properties and performance of folded hypercubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 2, pp. 31–42, Jan 1991.

[11] O. Guzide and M. D. Wagh, "Enhanced de bruijn graphs," in *Proc. of the the 2005 Int. Conf. on Algorithmic Math and Comp. Sc*, (Las Vegas, NV), pp. 23–28, 2005.

[12] P. Vadapalli and K. Srimani, "A new family of cayley graph interconnection networks of constant degree four," *IEEE Trans. Parallel and Distributed Systems*, vol. 17, no. 1, pp. 26–32, 1996.

[13] M. D. Wagh and O. Guzide, "Mapping cycles and trees on wrap-around butterfly graphs," *SIAM J. on Comput.*, vol. 35, pp. 741–765, 2006.