International Journal of Computer Engineering and Information Technology

VOL. 8, NO. 12, December 2016, 220–225
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

On a Product Network of Enhanced Wrap-Around Butterfly and de Bruijn Networks

Osman Guzide[1] and Weidong Liao[2]

[1, 2] Department of Computer Science, Mathematics and Engineering, School of Natural Sciences and Mathematics, Shepherd University, Shepherdstown, WV 25443, United States of America

[1]oguzide@shepherd.edu, [2]wliao@shepherd.edu

ABSTRACT

The topological structures of conventional Interconnection networks, such as Mesh, have recently been adopted in emerging areas such as wireless sensor networks, the pervasive computing and the Internet of Things. In this paper, we propose and analyze a novel Interconnection network topology called Enhanced Butterfly-de-Bruijn (denoted as EB_nD_n), which is the Product of Enhanced wrap-around Butterfly (denoted as EB_n) and de Bruijn Networks (D_n). We show that this Interconnection network may provide connections among $n2^n2^n$ nodes with a diameter of n, same as enhanced butterfly network, and a constant node degree of 11. Other desirable properties of our proposed product network as analyzed in this paper are 1): EB_nD_n is symmetric, even though it is a product of symmetric and non-symmetric networks; 2): EB_nD_n contains 2^n distinct copies of EB_n (enhanced wrap-around Butterfly Network; 3): EB_nD_n supports all cycle subgraphs. The proposed network topology structure may potentially be employed in parallel and distributed systems, as well as wireless sensor networks and the Internet of Things (IOT) environment.

Keywords: *Interconnection Networks, Topology, Product Networks, Enhanced Butterfly Network, de Bruijn Network.*

1. INTRODUCTION

As effective and efficient communication channels, the interconnection networks were historically employed to provide processor-to-processor and processor-to-memory connections for supercomputers. The topologies of these conventional Interconnection networks, such as Mesh, have also been adopted in wireless sensor network (Soparia & Bhatt, 2014) and the Internet of Things (Johnson) to provide connectivity for their ever-increasing number of intelligent nodes and agents. In either circumstance, as the number of computing nodes and intelligent devices increases, a suitable networking topology has to be adopted to connect I/O devices, memory units, caches and memory buffers, and processing cores and elements insider other computer systems. It therefore creates a demand and development of new interconnection topologies with larger cardinalities and other desirable properties, such as being symmetric, with lower diameter, and of constant degree.

Examining closely recently proposed Interconnection topologies with larger cardinalities and/or other desirable properties, one will derive that many of them are indeed the modifications, extension or enhancements to existing topologies, or the merge of two topologies so that the good properties of both topologies might be achieved. For instance, the cube-connected cycle, as proposed by Preparata and Vuillemin (Preparata & Vuillemin, 1981), is a merger of a Hypercube and a ring, which as a result possesses properties of a low diameter from the Hypercube and a low node degree from the ring. Hypercube is the product merged with the de Bruijn network (Ganesan & Pradhan, 1993) and the Butterfly network (Shi & Srimani, 1998). In both examples, the diameter of the resulting interconnection network topologies is the sum of diameters of the two child networks. Other examples of product interconnection networks include Scalable Twisted Hypercube (Alam & Kumar, 2011) and Torus-Butterfly Networks (Latifah & Kerami, Embedding on Torus-Butterfly Interconnection Network, 2012). The latter is the Cartesian product of Torus and Enhanced Butterfly.

Butterfly architecture is a popular interconnection network used early in parallel computing, and it is also used later in peer-to-peer networks (Datar, 2002) (Tang, Hu, Zhang, Zhang, & Ai, 2003). Many improvements and/or enhancements have been made to the original Butterfly topology. The wrap-around Butterfly (Wagh & Guzide, 2005) network (denoted as B_n) has 2^n times more nodes than regular butterfly network and is of the same diameter. Each wrap-around Butterfly network B_n (degree 4) has $n2^n$

nodes each labeled with a pair (I,X), where I is an integer between 0 and n-1 and X is a binary vector of length n. There are four edges from each node (I,X) going to (I+1,X), (I+1;X$\oplus 2^I$), (I-1;X) and (I -1;X $\oplus 2^{I-1}$) where $\oplus$ denotes eXclusive OR (XOR) operation of two vectors. Due to its properties of being symmetric and of small diameter of only |_3n/2_|, the wrap-around butterfly network is very attractive in many applications. It is employed to implement mappings of many signal processing algorithms such as the fast Fourier transform as well as many basic structures such as cycles and trees. In addition to wrap-around Butterfly network, research has been done on other modified versions of Butterfly networks. Examples are Extended Butterfly Networks (Guzide & Wagh, Extended Butterfly Networks, 2005) and Enhanced Butterfly Networks (Guzide & Wagh, Enhanced Butterfly : A Cayley Graph with Node 5 Network, 2007).

Studies have also been done in merging Butterfly network with other network to yield desirable characteristics. For the network topology of NoC (Network on Chip), a new topology named Butterfly Clos Network (Liu, Xie, Liu, & Ding, 2014) was proposed to integrate modules that in the same layer but different dimensions into a new module. The authors claimed that, based on simulation analyses, the proposed Butterfly Clos Network has some desirable properties including the rich path diversity of Close Network and increased throughput and reduced delay in large network traffic compared to Butterfly Network. An early study on deriving nice topological characteristics from Butterfly Network and Close Network was also available in Flattened Butterfly (Kim, Dally, & Abts, 2007). The Flattened Butterfly is a high-radix topology that yields better path diversity than the regular butterfly network and approximately half the cost of a similar-performed Clos network on benign traffic.

The de Bruijn Network is a hypercubic network (of a fixed degree) that is derived from de Bruijn Graphs (Zhang & Lin, 1987) (Baker, 2011). It has been shown that a de Bruin graph (Baker, 2011), denoted as D_n, is a directed graph with d^n nodes labeled by n-tuples over a d-character alphabet. If d=2, D_n will have 2^n nodes that can be labelled with a bit representation (n bit) of the numbers 0, 1, …, 2^n-1.In the graph, vertices are connected if the label of one end is the left or right shifted sequence of the other end, or it is the left or right-shifted sequence of the other end and differs correspondingly in the first or last bit.

A Product Network between wrap-around Butterfly and de-Bruijn Networks was proposed in (Guzide & Weidong, 2016). The proposed network, as a product of butterfly (B_n) and de Bruijn (D_n) networks of degree **n**, is denoted as B_nD_n. The desirable characteristics of B_nD_n that it provides the connection among $n2^n2^n$ nodes with a diameter of |_3n/2_|, same as butterfly network, and a constant node degree of 8.

In this paper, we present a novel Interconnection network topology based on the Enhanced Butterfly network and de Bruijn network. We show that by proper integration of Enhanced Butterfly Network B_n and de Bruijn Network D_n (2^n distinct copies of B_n), it is possible to derive a new symmetric network from the product of a symmetric network and a non-symmetric network, The resulting network (denoted as EB_nD_n), which is the Product of Enhanced Butterfly (denoted as EB_n) and de Bruijn networks (D_n), will contain $n\times2^n\times2^n$ nodes. We also show that this product network possesses some desirable properties such as a constant node degree of **11** and a diameter of EB_nD_n is **n** (equal to that of EB_n). Then we show that EB_nD_n contains 2^n disjoint EB_n copies as subgraphs, and can therefore run up to 2^n independent algorithms designed for Butterfly networks. It also supports cycle and tree mappings much better than many other networks. As to the results for cycle subgraphs, we prove that regardless n is odd or even, cycles of all lengths up to $n\times2^n\times2^n$ are subgraphs of EB_nD_n..

The rest of this paper is organized as follows. Section 2 describes basic properties of a Enhanced Butterfly-de-Bruijn networks as a product of Enhanced Butterfly and de Bruijn Network. Section 3 analyzes and discusses the routing and diameter of this product network. Section 4 presents cycle subgraphs in the proposed product network. We summarize this article in Section 5.

2. ELEMENTARY PROPERTIES

Let Z_n denote the group of integers {0, 1, …, n-1} with the operation of addition modulo **n** and Z_n^2, the group of binary vectors of length **n** under the operation of bit-by-bit modulo 2 addition, of degree n ≥3, is defined as a graph on $\mathbf{n2^n2^n}$ nodes labeled by triples (p, X, Y) where $p\in Z_n$ and $(X,Y)\in Z_n^2$. As shown in Figure 1, a node in Enhanced Butterfly-de-Bruijn (EB_nD_n) is connected to eleven nodes. Let us denote the integer **p** and the vector components X and Y in a node label (p, X, Y) as the first, second and the third indices of the node, respectively. The eleven edges from the node of EB_nD_n are labeled e, f, g, f^{-1}, g^{-1}, h, i, h^{-1}, i^{-1}, j, and k, as shown in Figure 1.

Note that edges of EB_nD_n are bidirectional. In particular, for u, v $\in EB_nD_n$, f(u) = v implies $f^{-1}(v)$ = u, g(u) = v implies $g^{-1}(v)$ = u, h(u) = v implies $h^{-1}(v)$ = u, i(u) = v implies $i^{-1}(v)$ = u, e(u)=v implies e(v)=u, j(u)=v implies j(v)=u, and , k(u)=v implies k(v)=u . Since every node of EB_nD_n has a fixed node degree of 11, there is a total of $11\times n\times2^n\times2^n$ edges in EB_nD_n.

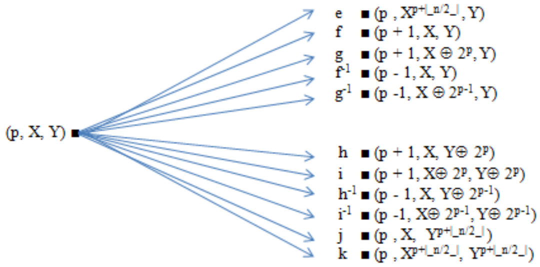

Fig. 1. Connections from node (p, X, Y) in Enhanced Butterfly-de-Bruijn Networks

EB_nD_n is a symmetric network. As the Product of Enhanced Butterfly and de Bruijn Networks, EB_nD_n, is closely related to the Enhanced Butterfly network denoted as EB_n. Recall that EB_n is a graph on $n2^n$ nodes, each with a label (p, X), where $p \in Z_n$ and $X \in Z_n^2$. By comparing the EB_n with the above definition of EB_nD_n, one can observe the following.

Theorem 1. EB_nD_n contains 2^n disjoint copies of EB_n subgraphs.

Proof. Partition the nodes of EB_nD_n in 2^n sets based on the first and last indices. Denote the set of nodes in a partition where all nodes have the same last index Y by EB_nD_n (*, *, Y). To show that the subgraph on nodes EB_nD_n (*, *, Y) is isomorphic to B_n, define a mapping (p,X) : EB_nD_n (*,*,Y) $\varphi_{p:}$ EB_nD_n (*, *, Y) --->B_n as φ_p (p, X, Y) = (p, X). It is clear that φ_p (.) is a one-to-one mapping. Further, the edges within EB_nD_n (*,*,Y) are exactly mapped onto the edges of EB_n. For example, consider the edge (p, X, Y) --> (p, X ⊕ 2^p) of EB_nD_n (*,*,Y). By using the mapping φ_p (.), this edge translates to the edge (p, X ⊕ 2^p)-> (p,X) of B_n. Therefore, index Y has 2^n member of Enhanced Butterfly network (EB_n).

Theorem 1 implies that 2^n instances of any algorithm designed to run on Enhanced Butterfly networks (EB_n) can be run on EB_nD_n networks without suffering any performance degradation. In addition, these instances can exchange information using the additional links present in EB_nD_n networks. This structure also suggests a possibility of being able to map other algorithms from EB_n networks onto EB_nD_n networks.

In Section 4, we can see how to exploit the relationship between EB_nD_n and EB_n to develop cycle mappings on EB_nD_n.

3. ROUTING AND DIAMETER

Diameter and the availability of feasible routing strategy are both important properties of any interconnection network. For the sake of efficient implementation of parallel algorithms, it is desirable to have a low diameter and a good routing strategy. In this section we provide an algorithm and strategy to obtain paths between nodes of the proposed EB_nD_n network and analyze its diameter.

3.1 Routing Strategy

To proof the diameter of EB_nD_n network with accepting the network symmetric and without loss of generality, one present a simple path algorithm that goes from node (p, X, Y) to (0, 0, 0) in an EB_nD_n network as follows.

Simple Path Algorithm to Go from (p, X, Y) to (0, 0, 0) in n Hops.

1. If $p \le \lfloor n/2 \rfloor$,

   for *r* = p to *n-1* do

   if ($\lceil n/2 \rceil \le r \le p + \lceil n/2 \rceil$)

   if the (*r*- $\lceil n/2 \rceil$)th bit X is 1 and the bit Y is 0, then
   go along the *e* edge, i.e.,
   else if the (*r*- $\lceil n/2 \rceil$)th bit Y is 1 and the bit X is 0, then
   go along the *j* edge, i.e.,
   else if the (*r*- $\lceil n/2 \rceil$)th bit X is 1 and the bit Y is 1 , then
   go along the *k* edge, i.e.,

   if the *r*th bit X is 1 and the bit Y is 0, then
   go along the *g* edge, i.e.,
   else if the *r*th bit Y is 1 and the bit X is 0, then
   go along the *h* edge, i.e.,
   else if the *r*th bit X is 1 and the bit Y is 1 , then
   go along the *i* edge, i.e.,
   else if the *r*th bit X is 0 and the bit Y is 0 , then
   go along the *f* edge, i.e.,

2. If $p > \lfloor n/2 \rfloor$,

   for *r* = p downto *1* do

   if ($p - \lfloor n/2 \rfloor) \le r \le \lceil n/2 \rceil$)

   if the (*r*+ $\lfloor n/2 \rfloor$)th bit X is 1 and the bit Y is 0, then
   go along the *e* edge, i.e.,
   else if the (*r*+ $\lfloor n/2 \rfloor$)th bit Y is 1 and the bit X is 0, then
   go along the *j* edge, i.e.,
   else if the (*r*+ $\lfloor n/2 \rfloor$)th bit X is 1 and the bit Y is 1 , then
   go along the *k* edge, i.e.,

   if the(*r-1)*th bit X is 1 and the bit Y is 0, then
   go along the g^{-1} edge, i.e.,
   else if the (*r-1)*th bit Y is 1 and the bit X is 0, then
   go along the h^{-1} edge, i.e.,
   else if the (*r-1)*th bit X is 1 and the bit Y is 1 , then
   go along the i^{-1} edge, i.e.,
   else if the (*r-1)*th bit X is 0 and the bit Y is 0 , then
   go along the f^{-1}edge, i.e.,

Clearly this simple path algorithm transports the first index of the label to all vectors. We may show that the second and third index also get converted to an all vectors as follows. Note that in steps 1 of the algorithm, when the first index goes up and reaches 0, the second and third indexes will change and eventually it will reach (0,0,0). In each hop, a different bit of the second and third indexes will be modified. Together these steps use n hops and in the end, all the n bits of the second and third indexes will be modified to reach (0, 0, 0). Within these first n hops, the first index will become 0, then the second and third indexes will follow. Eventually all three indexes would change and it arrives at (0, 0, 0). This algorithm therefore provides a path of length at most n between any pair of nodes in EB_nD_n.

To illustrate this algorithm, consider the path from node (4, 111111, 111111) to node (0, 000000, 000000) in EB_6D_6. According to step 1 of the path algorithm, one would use 6 hops along edges i or k as follows:

(4. 111111, 111111) → (3, 111011, 111011) → (2, 110011, 110011) → (2, 110010, 110010) → (1,100010, 100010) → (1,100000, 100000) → (0,000000, 000000).

It should be noted that the above algorithm may not give an optimal path between the two nodes. It is nevertheless a simple algorithm and suffices to specify the diameter of EB_nD_n as the following theorem shows.

3.2 Diameter of Enhanced Butterfly-de-Bruijn

Theorem 2 (Diameter of EB_nD_n). Diameter of EB_nD_n, the Product Network of Enhanced Butterfly Networks (EB_n) and de Bruijn (D_n), is **n**.

Proof: As shown by the path algorithm given above, a path of length at most n exists between any pair of nodes in EB_nD_n. Hence, to prove the theorem we merely have to show that **n** is also the lower bound on the diameter. According to Theorem 1, we know that EB_nD_n contains 2^n copies of EB_n subgraphs. Consider two nodes of EB_nD_n which lie in the same copy of a EB_n subgraph. It is obvious that the shortest path between the nodes uses only the edges of that subgraph. The distance between these two points in EB_nD_n is the same as the distance between the corresponding points of the graph EB_n. Therefore the diameter of EB_nD_n cannot be less than the diameter **n** of EB_n.

Theorem 2 shows that even though the Product of Enhanced Butterfly Networks and de Bruijn EB_nD_n has 2^n times as many nodes as an enhanced Butterfly EB_n, its diameter is the same as that of EB_n.

It is interesting to note that the simple path algorithm presented in Section 3.1 uses edges *e,f, g, h, i, j* and *k* only. As a result, if one were to construct a directed graph EB_nD_n which uses only these 7 edges, then the node degree would drop to 5. Nevertheless the diameter would remain unchanged at **n**.

4. CYCLE SUBGRAPHS

As indicated in Section 2, EB_nD_n contains 2^n disjoint copies of Enhanced Buttefly networks (EB_n). We use this fact to obtain larger cycle subgraphs of EB_nD_n by merging smaller cycle subgraphs located in these copies. To facilitate this, we first restate the following result from (Guzide & Wagh, Enhanced Butterfly : A Cayley Graph with Node 5 Network, 2007) that relates to the cycle subgraphs of butteries.

Theorem 3 (Cycle subgraphs of EB_n) (Guzide & Wagh, Enhanced Butterfly : A Cayley Graph with Node 5 Network, 2007). Cycles of all lengths L are subgraphs of EB_n including when:

a. odd L when n is even.
b. odd L less than n.

Herein we do not discuss designs of cycles in EB_n. It is sufficient to note that for lengths smaller than **4n**, these cycles are generated using a template given in (Wagh & Guzide, Mapping Cycles and Trees on Wrap-Around Butterfly Graphs, 2005). For larger lengths, one first obtains a cycle subgraph of length L' such that n|L' and L – L' is a small even number ≤ 2(n-1). One can then attach up to (n-1) additional pairs of nodes to this cycle to get the length L cycle. Cycle subgraphs of length L' are obtained by judiciously picking edges h and i (see Fig. 1 for the edge naming convention) to form the cycle. Recall that EB_nD_n contains 2^n distinct copies of EB_n, each made up of those nodes of EB_nD_n which have the same first index. By identifying cycle subgraphs in these copies of EB_n and merging them together, one can get the desired cycle subgraphs of EB_nD_n. One needs the following two lemmas to carry out this merging.

Lemma 1 (Cycle merging using edges g). Given a node pair u, v ϵ EB_nD_n connected by an h-edge, i.e., h(u) = v, there exists another node pair z; w ϵ B_nD_n, also connected by an h-edge, i.e. w = h(z) such that g(u) = z and g(v) = w. Further, the four nodes v, u, w, and z are distinct.

Proof: Let u = (p, X, Y). Then, v = h(u) = (p+1, X, Y⊕ 2^p). One can verify that the nodes z = (p + 1, X, Y) and w = h(z) = (p +1+1,X,Y) satisfy the required conditions of the lemma. Fig. 2 illustrates the connections specified in Lemma 1. Note that nodes u and v in this figure belong to the same copy of EB_n (they have the same second index) and w and z to another copy. One can relate node pairs in different copies of EB_n by h edges as well. This is given in Lemma 2.

Lemma 2 (Cycle merging using edges f). Given a node pair u, v ϵ EB_nD_n connected by an i-edge, i.e., i(u) = v, there exists another node pair z,w ϵ EB_nD_n, also connected by an i-edge, i.e. w = i(z) such that f(u) = z and f(v) = w. Further, the four nodes v, u, w, and z are distinct.

Proof: One can verify that the nodes u = (p, X.Y), v = (p+1,X, Y⊕ 2^p), z = (p + 1, X,Y⊕ 2^p) and w = (p+1+1, X, Y⊕ 2^p) satisfy the required conditions of the lemma.

Before we identify the cycle subgraphs of B_nD_n, we first present a result that imposes a fundamental limit on the cycle subgraphs of B_nD_n.

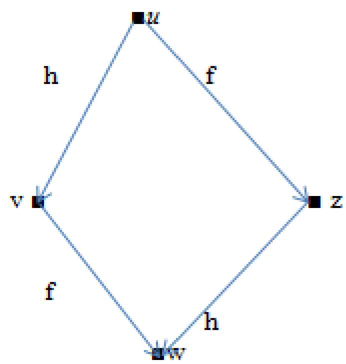

Fig. 2. The connection of four nodes in EBnDn.

Theorem 4 (Impossible cycle subgraphs of EB_nD_n). Cycles of the following lengths L are never subgraphs of EB_nD_n:

Proof: For an even n, partition the nodes of EB_nD_n into two sets based upon on whether the sum of the first two indices of a node is even or odd. Clearly, all EB_nD_n edges go only between these sets and no nodes in the same set are connected. Thus EB_nD_n is a bipartite graph for even n and therefore cannot support odd length cycle subgraphs.

Now consider a length L, L < n, cycle subgraph of EB_nD_n. Since L < n, there exists an integer k, 0 ≤k < n such that no node on the cycle has the form (n-k-2, X, Y). Replace each node (p, X, Y) of the cycle by node ((p-k -1) mod n, X, Y). Note that this does not change the cycle connectivity. The new cycle of the same length L will not have any node whose first is n - 1. Consequently, this new cycle will not use any wrap-around edges (i.e., edges that go between nodes whose first or second index changes from n-1 to 0). Thus along this cycle, the sum of the first two indices of the nodes traversed alternates between odd and even. This implies that the cycle length L < n must be even.

To obtain cycle subgraphs of lengths greater than $n2^n$, one may first design cycle subgraphs on multiple copies of EB_n, and then merge them using Lemma 1 or Lemma 2. Recall that the cycles in each copy of EB_n use only h or i edges. The merging process using Lemma 2 is illustrated in Fig. 3.

By removing the i-edges and adding f-edges between points u = (p,X,Y), v = (p+1,X, Y⊕ 2^p), z = (p+1,X,Y) and w = (p + 1+1, X , Y Y⊕ 2^p). It should be clear from this figure that the merging of cycles in p-th and p+1-th copies of EB_n is possible if there exists an edge (p, X,Y) -> (p+ 1;X X, Y⊕ 2^p) in the first cycle and an edge (p + 1,X,Y) -> (p + 1+ 1X, Y⊕ 2^p) in the second cycle. However, the index r of all the nodes in a cycle may be incremented by the same amount without destroying the cycle connectivity. Thus the only requirement for merging the cycles in the two copies of EB_n is the existence of an edge (p, X, Y) ->(p+1, X , Y⊕ 2^p) in the first cycle and an edge (p + 1, X , Y⊕ 2^p) -> (p+1+1,X , Y 2^p) in the second cycle for arbitrary Y_1 and Y_2. If one of these cycles has a length of at least 2^n -1, then this requirement can be easily met if the other cycle has an *i* edge. If the other cycle has only h edges, then one may use Lemma 1 in place of Lemma 2 in this proof. Thus cycles of all possible lengths 3≤L ≤$n2^n2^n$ are subgraphs of EB_nD_n.

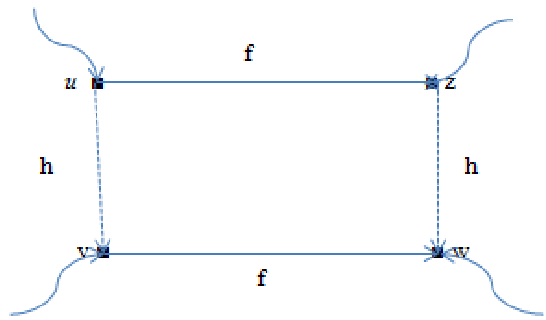

Fig. 3. Merging cycles in two EBn subgraphs.

5. CONCLUSIONS

This paper describes a new Interconnection network that carries some desirable features. The proposed network EB_nD_n (Enhanced Butterfly-de-Bruijn), which is the Product of Enhanced wrap-around Butterfly (EB_n) and de Bruijn Networks (D_n), contains 2^n distinct copies of EB_n and therefore can run 2^n different algorithms designed for butteries without any slowdown. We show that the interconnection between these copies preserves all the desirable properties of the Butterfly network. EB_nD_n is a symmetric network with $n2^n2^n$ nodes and a constant node degree. It has a diameter equal to that of EB_n, i.e., n. We have obtained a comprehensive solution to the problem of cycle subgraphs of EB_nD_n. The novel product interconnection network topology as presented and discussed in this paper can be adopted not only in conventional parallel computing environment, but also in other areas such as wireless sensor networks and the Internet of Things (IOT).

6. ACKNOWLEDGMENTS

The research reported herein is supported by Research Enhancement Awards from NASA WV Space Grant Consortium. We greatly appreciate NASA WV Space

Grant Consortium Committee for the generous grant support.

REFERENCES

[1] Alam, J., & Kumar, R. (2011). STH: A highly Scalable and Economical Topology for Massively Parallel Systems. Indian Journal of Science and Technology, 4(12), 1737-1748.

[2] Baker, J. (2011). De Bruijn Graphs and Their Applications to Fault Tolerant Networks. Oregon: Oregon State University (Master Thesis).

[3] Datar, M. (2002). Butterflies and Peer-to-Peer Networks. Proc. of the 10th European Symposium on Algorithms (pp. 310-322). Berlin Heidelberg: Springer.

[4] Ganesan, E., & Pradhan, D. K. (1993, September). The Hyper-de Bruijn networks: Scalable versatile architecture. IEEE Trans. on Parallel and Distributed Systems, Vol. 4, pp. 962-978.

[5] Guzide, O., & Wagh, M. D. (2005). Extended Butterfly Networks. Proc. of the The 18th Int. Conf. on Parallel and Distributed Computing Systems, (pp. 109-113). Las Vegas, NV.

[6] Guzide, O., & Wagh, M. D. (2007). Enhanced Butterfly : A Cayley Graph with Node 5 Network. ISCA PDCS, (pp. 224-229).

[7] Guzide, O., & Weidong, L. (2016). Topological and Structural Properties of Butterfly-de-Bruijn: A Product Network of Wrap-Around Butterfly and de Bruijn Networks. International Journal of Applied Science and Technology, Vol. 6, No. 3.

[8] Johnson, S. (n.d.). TechTarget IoT Agenda. Retrieved September 15, 2016, from Using mesh networking to interconnect IoT devices: http://internetofthingsagenda.techtarget.com/feature/Using-mesh-networking-to-interconnect-IoT-devices

[9] Kim, J., Dally, W. J., & Abts, D. (2007). Flattened Butterfly : A Cost-Efficient Topology for High-Radix Networks. ACM SIGARCH Computer Architecture News, 35, 126-137.

[10] Latifah, E., & Kerami, D. (2012). Embedding on Torus-Butterfly Interconnection Network. International Journal of Apllied Information Systems (IJAIS), Vol. 4. No. 12, 38-41.

[11] Liu, H., Xie, L., Liu, J., & Ding, L. (2014). Application of Butterfly Clos-Network in Network-on-Chip. The Scientific World Journal.

[12] Preparata, F. P., & Vuillemin, J. (1981, May). The cube-connected cycles: a versatile network for parallel computation. Communications of the ACM, pp. 300-309.

[13] Shi, W., & Srimani, P. K. (1998). Hyper Butterfly network: A scalable optimally fault tolerant architecture. Proceedings of the First Merged International Parallel Processing Symposium and Symposium on Parallel and Distributed Processing (pp. 732-736). Orlando, FL: IEEE Press.

[14] Soparia, J., & Bhatt, N. (2014, February). A Survey on Comparative Study of Wireless Sensor Network Topologies. International Journal of Computer Applications, 87, 40-43.

[15] Tang, Y., Hu, Z., Zhang, Y., Zhang, L., & Ai, C. (2003). A Practical Peer-Performance-Aware DHT. 2nd Int. Workshop on Agents and Peer-to-Peer Computing (pp. 193-200). Melbourne, Australia: Springer Berlin Heidelberg.

[16] Wagh, M. D., & Guzide, O. (2005). Mapping Cycles and Trees on Wrap-Around Butterfly Graphs. SIAM Journal on Computing, 35, 741-765.

[17] Zhang, F., & Lin, G. (1987). On the de Bruijn-Good Graphs. Acta Math. Sinica 30 (2), 195-205.

AUTHOR PROFILES:

Osman Guzide, Dr. Osman Guzide is an associate Professor of Computer and Information Science in Department of Computer Science, Mathematics and Engineering at Shepherd University, Shepherdstown, WV 25443, USA. He obtained his Ph.D. in Computer Science and Engineering from Lehigh University, Bethlehem, Pennsylvania, USA in 2000 with dissertation titled “Derivatives of Butterfly Interconnection Networks for Parallel Processing”. Dr. Guzide is an active researcher in interconnection network topology, parallel and distributed processing, wired and wireless networking, and security.

Weidong Liao, Dr. Weidong Liao is an associate Professor of Computer and Information Science in Department of Computer Science, Mathematics and Engineering at Shepherd University, Shepherdstown, WV 25443, USA. Dr. W. Liao obtained his Ph.D. in Computer Science in 2004 from Kent State University, Kent, Ohio, USA. His research interests are in Internet computing, distributed database systems, software engineering, distributed and pervasive systems, and computer architecture. Dr. Liao is a member of the IEEE.