Analysis and Comparison of Torus Embedded Hypercube Scalable Interconnection Network for Parallel Architecture

N. Gopalakrishna Kini, M. Sathish Kumar**, Mruthyunjaya H.S. ****

** Dept. of Computer Science and Engineering, Manipal Institute of Technology (Manipal University), Manipal, India*
*** School of EECS, Seoul National University, South Korea*
**** Dept. of Electronics and Communication, Manipal Institute of Technology (Manipal University), Manipal, India*

Summary
This paper analyzes an embedded architecture of torus network with the hypercube pertinent to parallel architecture. The product generated from torus and hypercube networks show how good interconnection network can be designed for parallel computation. The advantages of hypercube network and torus topology are used for product network known as Torus embedded hypercube network. A complete design analysis and comparison of this network with various other networks is given using network parameters.

Keywords:
Concurrent torus network, Data routing path, Embedded network, Hypercube network, Network parameters, Scalability.

1. Introduction

Massively parallel computing systems are placing a major emphasis on scalable networks with small diameters and bounded node degree. The hypercube is a network with a small network diameter, high connectivity and simple routing procedures. However, the node degree grows logarithmically with number of vertices making it difficult to build scalable architectures using the hypercube [1, 2]. On the other hand, Torus is a network with constant node degree and has a highly scalable architecture but has larger network diameter [1, 3]. The advantages of these two architectures can be combined by embedding the torus and hypercube networks to give rise to a torus embedded hypercube network as suggested in [4]. Such a combination results in a system which can be implemented with small node degrees, which implies a reduction in hardware cost per node. Also, a constant node degree results in a system that is scalable without having to modify the individual nodes [4].

Louri and Sung in [4] have reported a comparison between the torus embedded hypercube, which they refer to as OMMH, and the hypercube network. Based on their comparison, they conclude that the torus embedded hypercube is superior to the hypercube network. However, we feel that a more detailed and elaborate comparison of torus embedded hypercube network with other proposed networks would provide with more decisive conclusions and would be of great use in selecting network architectures for future high speed computations.

In this paper, we compare the torus embedded hypercube with other proposed and popular interconnection networks by considering the node degree, network diameter, total number of links and topological network cost.

2. Theoretical Consideration

2.1 Embedding Properties

In this section, we discuss the embedding of the torus and the hypercube networks to obtain the torus embedded hypercube network. While combining the torus and the hypercube network, several concurrent torus networks are used in the architectural design as shown in Fig.1. The black circles represent nodes in the individual torus. The ellipse drawn over the group of identical nodes of the concurrent torus form a hypercube. These nodes are connected according to the hypercube configuration while the torus configuration lies along with it [3, 4, 5].

Note that nodes of the hypercubes with similar addresses will be the nodes of individual torus.

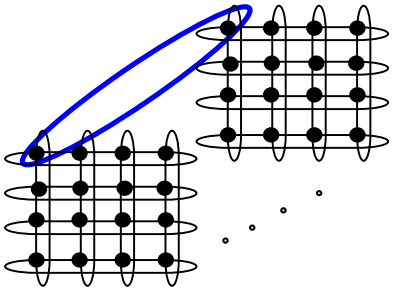

Fig. 1. Concurrent torus network.

Let l x m be the size of concurrent torus network and N be the number of nodes connected in the hypercube. Then the torus embedded hypercube network will be of size

Manuscript received January 5, 2009
Manuscript revised January 20, 2009

(*l, m, N*). Nodes with identical positions in the torus networks will form a group of *N* number of nodes. These nodes will be connected in the hypercube configuration. Such nodes can be addressed with three components; row number *i* and column number *j* of torus appended with the address of node *k* of hypercube.

Hence, a (*l, m, N*)–torus embedded hypercube network will have *l* x *m* x *N* number of nodes and a node will be addressed as (*i, j, k*) where $0 \le i < l$, $0 \le j < m$ *and* $0 \le k < N$. Thus, the data routing functions of hypercube and torus are combined together for two nodes in torus embedded hypercube network.

Combining the data routing functions of torus and hypercube provides with the routing functions of the torus embedded hypercube [4] as

$$T_{h1}(i, j, k) = (i, \ (j+1) \ mod \ m, \ k) \quad (1)$$

$$T_{h2}(i, j, k) = (i, \ (m+j-1) \ mod \ m, \ k) \quad (2)$$

$$T_{h3}(i, j, k) = ((i+1) \ mod \ l, \ j, \ k) \quad (3)$$

$$T_{h4}(i, j, k) = ((l+i-1) \ mod \ l, \ j, \ k) \quad (4)$$

$$T_{Cd}(k_{n-1}.....k_{d+1}\,k_d\ \ k_{d-1}.....k_0) = (k_{n-1}...k_{d+1}\ \ \overline{k_d}\ \ k_{d-1}.....k_0) \quad (5)$$

We provide an example for the above discussion in the *APPENDIX* with a torus of size 2x2 and a 3-cube hypercube. For a (2, 2, 8) torus embedded hypercube network derived from that example,

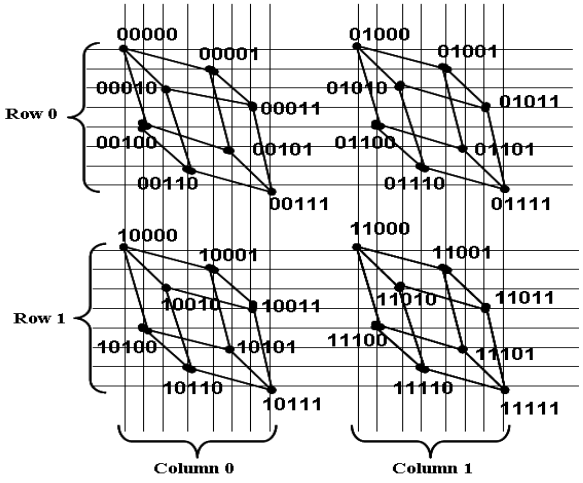

Fig. 5. A (2,2,8)-Torus Embedded Hypercube Network

a node with a five bit address has its left most bit representing row number, the next bit representing column number and the remaining bits representing the address of the hypercube as shown in Fig. 5.

Table-I demonstrates the data routing function of the torus embedded hypercube network. Data routing between a source node and a destination node by applying the data routing functions given in equations (1), (2), (3), (4) and (5) is shown in this table. The numbers written above the arrow mark in Table-I indicates the appropriate data routing function applied.

Table-I
Results of Intermediate Nodes Crossing to Establish a Data Routing Path

Source	Destination	Intermediate nodes passed with optimal communication path
00000	00111	00000 →(5) 00100 →(5) 00110 →(5) 00111
00000	01000	00000 →(1) 01000
00000	01111	00000 →(5) 00100 →(5) 00110 →(5) 00111 →(1) 01111
00000	10101	00000 →(5) 00100 →(5) 00101 →(3) 10101
00000	11111	00000 →(5) 00100 →(5) 00110 →(5) 00111 →(3) 10111 →(1) 11111
11111	00000	11111 →(5) 11011 →(5) 11001 →(5) 11000 →(4) 01000 →(2) 00000
01101	00000	01101 →(5) 01001 →(5) 01000 →(2) 00000

2.2 *Scalability of Torus Embedded Hypercube Network*

The torus embedded hypercube network is highly scalable. Scalability can be achieved in two ways [4]. Firstly, the dimension of the hypercube can be increased by keeping the size of concurrent torus same but increasing the number of concurrent toruses accordingly. Secondly, dimension of torus is expanded by keeping the size of the hypercube constant. Since node configuration is not required, scaling up the system using the latter method is preferable over the former.

The efficiency of the torus embedded hypercube network is evaluated using network parameters such as node degree, network diameter, total number of links in the network and topological network cost. The definitions of these network parameters can be found in [1,4]. It may be noted that the node degree has to be as less as possible because if the numbers of links are increased the number of I/O ports also increases per node. This results in greater cost of the network because of higher link complexity

3. Comparison Results and Discussions

3.1 Node degree

Table-II and Fig.6 gives the comparison of node degree as a function of the number of processors of various other popular networks [6, 7, 8, 9, 10] along with the torus embedded hypercube for parallel architecture.

Table-II
Results of Node Degree Analysis

No. of processors / Network type	512	1024	2048	4096	8192	16384
n-cube Hypercube	9	10	11	12	13	14
Simple mesh	4	4	4	4	4	4
Torus	4	4	4	4	4	4
OMIN-Hypercube	10	11	12	13	14	15
OMIN- mesh	5	5	5	5	5	5
(16,16,N)– Torus embedded Hypercube	5 N=2	6 N=4	7 N=8	8 N=16	9 N=32	10 N=64
(l,m,16) - Torus embedded Hypercube	8	8	8	8	8	8

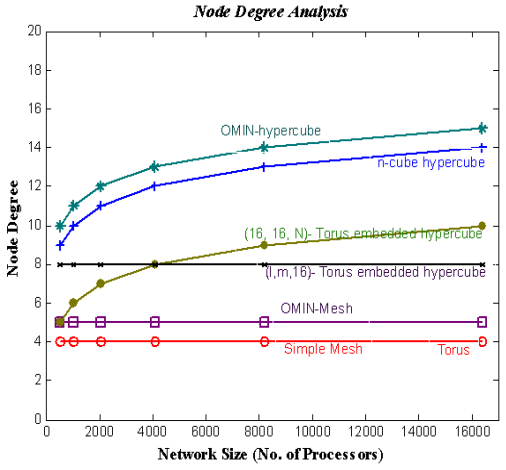

Fig. 6. Node degree analysis

The comparison shows that the n-cube hypercube and OMIN-hypercube interconnection networks are expensive and not suitable for parallel architecture. For these two networks, the node degree increases dramatically as the system expansion takes place. Obviously the cost per node also increases tremendously for these two networks as the system is scaled up.

3.2 Network diameter

Table-III and Fig. 7 gives the comparison of network diameter as a function of the number of processors of various other popular networks along with the torus embedded hypercube network .

It is preferred to have a network with a network diameter of minimum value. The result of comparison shows that OMIN-mesh network has the largest network diameter. Further, for this network, the network diameter increases tremendously as the system is scaled up.

Table-III
Results of Network Diameter Analysis

No. of processors / Network type	512	1024	2048	4096	8192	16384
n-cube Hypercube	9	10	11	12	13	14
Simple mesh	43	62	88	126	179	254
Torus	22	32	44	64	90	128
OMIN-Hypercube	19	21	23	25	27	29
OMIN- mesh	87	125	178	253	359	509
(16,16,N)– Torus embedded Hypercube	17 N=2	18 N=4	19 N=8	20 N=16	21 N=32	22 N=64
(l,m,16) - Torus embedded Hypercube	10	12	16	20	26	36

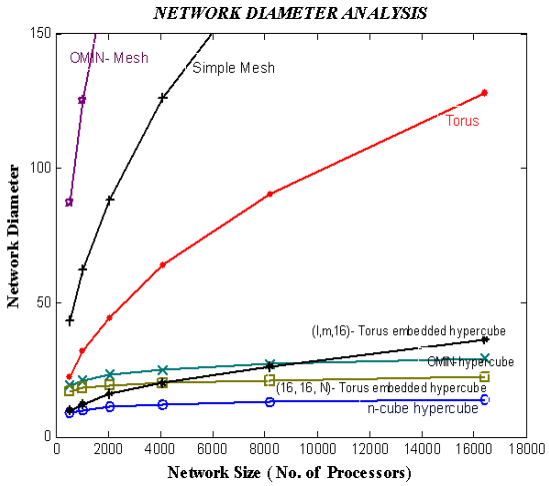

Fig. 7. Network diameter analysis

Worth to mention that due to their inferior performance as per the last two comparisons, the OMIN hypercube and OMIN mesh networks have been dropped from further comparisons

3.3 Total number of links

Table-IV shows the number of links with respect to the scaling of the parallel architecture for the different networks considered. Since the node degree is preferred to be as low as possible, the total number of links is also expected to be as small as possible.

With reference to Fig. 8, it is observed that torus embedded hypercube offers larger number of links than the

other networks under comparison. This is because every node of torus is a hypercube configuration and hence a larger number of links get reflected.

Table-IV
Results of No. of Links Analysis

No. of processors / Network type	512	1024	2048	4096	8192	16384
n-cube Hypercube	2304	5120	11264	24576	53248	114688
Simple mesh	979	1984	4006	8064	16203	32512
Torus	1024	2048	4096	8192	16384	32768
(16,16,N) – Torus embedded Hypercube	1280 N=2	3072 N=4	7168 N=8	16384 N=16	36864 N=32	81920 N=64
(l,m,16) - Torus embedded Hypercube	2048	4096	8192	16384	32768	65536

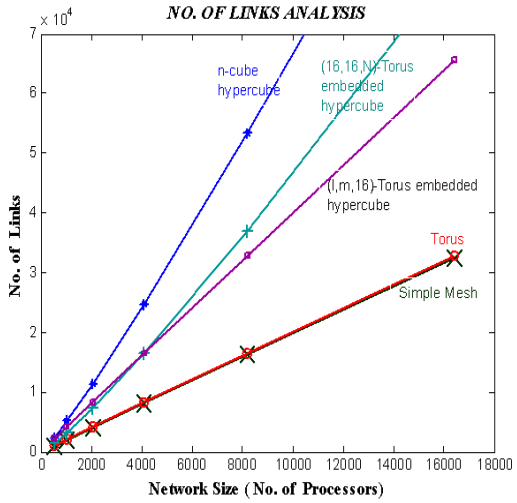

Fig. 8. Number of links analysis

3.4 Topological Network cost

The topological network cost analysis result is given in Table-V and Fig. 9. From this comparison, it is observed that the torus embedded hypercube network has a low network cost. Though this network cost is more than that of the n-cube hypercube, it has to be noted that the torus embedded hypercube has better values for all the other network parameters considered.

Table-V
Results of Network Cost Analysis

No. of processors / Network type	512	1024	2048	4096	8192	16384
n-cube Hypercube	20736	51200	123904	294912	692224	1605632
Simple mesh	42331	123008	354524	1016064	2900646	8258048
Torus	22528	65536	180224	524288	1474560	4194304
(16,16,N) – Torus embedded Hypercube	21760 N=2	55296 N=4	136192 N=8	327680 N=16	774144 N=32	1802240 N=64
(l,m,16) - Torus Embedded Hypercube	20480	49152	131072	327680	851968	2359296

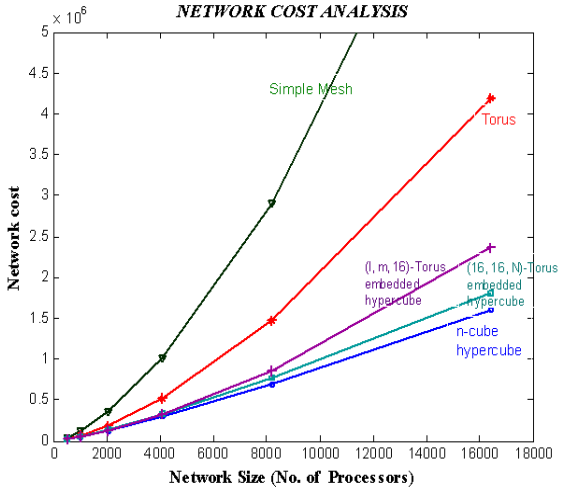

Fig. 9 . Network cost analysis

4. Conclusion

We have analyzed a torus embedded hypercube network and compared its network parameters with other networks. The results show that torus embedded hypercube interconnection network is highly scalable and any further configuration of existing nodes is not required. Due to the concurrent existence of multiple torus and hypercubes, this network provides a great architectural support for parallel processing. The parallel algorithms that have been designed for the hypercube and the torus interconnection networks can be combined together with minor modifications and applied on this embedded network. The growth of the network is observed to be more efficient.

Our comparison of network parameters have shown that as far as node degree, network diameter and number of link is concerned; the torus embedded hypercube has a clear advantage over the other networks. As far as the network cost is considered, the n-cube hypercube seems to be the most economical with the torus embedded hypercube being the second best. However, considering the better performance of the torus embedded hypercube in terms of the other network parameters other than the network cost, we conclude that the torus embedded hypercube could be considered as the best candidate for future high speed and capacity computational systems.

APPENDIX

To provide with an example, consider an eight concurrent torus network of size 2x2 each which forms 4 hypercubes with each hypercube containing 8 individual nodes. From the earlier discussion, this results in a (2, 2, 8*)*- torus embedded hypercube network with 2x2x8 = 32 nodes [4, 5].

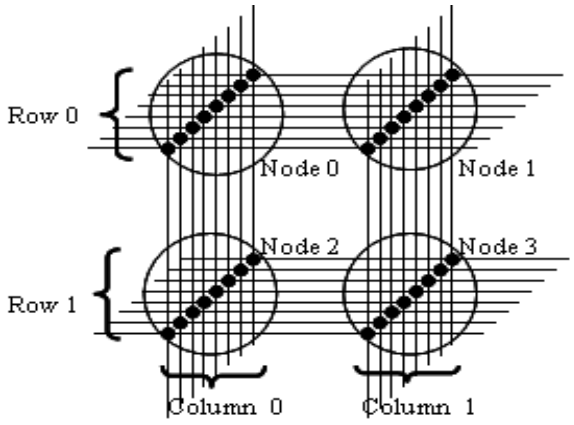

Fig. 2. A (2, 2)- concurrent meshes of torus network.

A diagrammatic representation of the above example is given in Fig. 2. As shown in this figure, a (2, 2, 8)-torus embedded hypercube network can be derived in which *N* nodes (with respect hypercube) of Node *i* (with respect to concurrent torus) are connected in a hypercube configuration.

As is well known, the data routing functions of torus network [1] are

$$T_1(i, \ j) = (\ i, (j+1) \ mod \ m\ ) \quad (1.a)$$
$$T_2(i, j) \ = (\ i, (j\text{-}1) \ mod \ m\ ) \quad (1.b)$$
$$T_3(i, \ j) = ((\ i+1) \ mod \ l, j\ ) \quad (1.c)$$
$$T_4(i, \ j) = ((\ i\text{-}1) \ mod \ l, j\ ) \quad (1.d)$$

where *i* and *j* are row and column numbers respectively. According to these data routing functions, the following permutation cycles can be generated for a 2 X 2 torus network as given in Fig. 3 and in equations 1.a to 1.d.

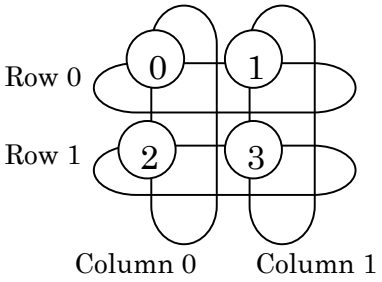

Fig. 3. A 2 X 2 torus network.

.

$T_1 = (0 \quad 1) \ (\ 2 \quad 3)$
$T_2 = (3 \quad 2) \ (\ 1 \quad 0)$
$T_3 = (0 \quad 2) \ (\ 1 \quad 3)$
$T_4 = (3 \quad 1) \ (\ 2 \quad 0)$

The data routing function of hypercube network [1] is

$$C_d \ (k_{n-1}.....k_{d+\ 1} \quad k_d \quad k_{d-1}.....k_0\ ) = (k_{n-1}...k_{d+\ 1} \quad \overline{k_d} \quad k_{d-1}.....k_0) \quad (2)$$

for $d = 0, 1, \ldots.n\text{-}1$ where k_j for ($j = 0$ to n-1) is the binary representation of node address k and $n = log_2(N)$ where N is the total number of nodes in the hypercube.

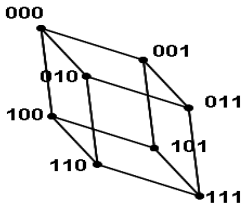

Fig. 4. A 3-cube hypercube

According to the above hypercube data routing function the following permutation cycles are generated for a 3-cube structured network as shown in Fig. 4 and equation (2)..

$C_0 = (0 \quad 1) \ (\ 2 \quad 3) \ (4 \quad 5) \ (6 \quad 7)$
$C_1 = (0 \quad 2) \ (\ 1 \quad 3) \ (4 \quad 6) \ (5 \quad 7)$
$C_2 = (0 \quad 4) \ (\ 1 \quad 5) \ (2 \quad 6) \ (3 \quad 7)$

References

[1] K. Hwang, *Advanced Computer Architecture: Parallelism, Scalability, Programmability***.** New York McGraw-Hill, 1993.

[2] N. Gopalakrishna Kini, "Analysis of parallel algorithm for matrix multiplication on hypercubes", *National Conf. on*

Advanced Computing, PSG college of Technology, Coimbatore, India,, page 123, Feb 1-2, 2002.

[3] Kim Jong-Seok, Lee Hyeong-Ok and Heo Yeong-Nam**,** "Embedding among HCN(n,n), HFN(n,n) and hypercube" **,** *Proceedings of Eighth International conference on parallel and distributed systems (ICPADS)-2001* pp 533 – 540, 26-29 June 2001.

[4] A. Louri and H. Sung, "An Optical Multi-Mesh Hypercube: A Scalable Optical Interconnection Network for Massively Parallel Computing," *IEEE J. Lightwave Technology*, vol. 12, pp. 704–716, Apr. 1994.

[5] J .-F. Fang, J.-Y. Hsiao and C.-Y. Tang, "Embedding Meshes and TORUS Networks onto degree-four chordal rings", *IEE Proc.-Comput. Digit. Tech.,* Vol. 145, No. 2, March 1998.

[6] Francis Zane, Philippe Marchand, Ramamohan Paturi and Sadik Esener, "Scalable Network Architectures Using the Optical Transpose Interconnection System (OTIS*) ", Journal of Parallel and Distributed Computing* 60, pp 521-538, 2000.

[7] Hashem Hashemi Najaf-abadi, Hamid and Sarbazi-azad, "An Empirical Comparison of OTIS-mesh and OTIS-hypercube Multicomputer Systems under Deterministic Routing", *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)* Page(s):8 pp 4-8 April 2005.

[8] K. Day and A. Al-Ayyoub, "Topological Properties of OTIS Networks*", IEEE Trans. On Parallel and Distributed systems,* Vol. 13, No. 4, pp. 359-366, April 2002.

[9] N. Gopalakrishna Kini , M. Sathish Kumar and B.S. Satyanarayana, "Enhancement of Speed in Massively Parallel Computers using Optical Technology", *Second National Conference on Frontier Research Areas in Computing Sciences (e_proceedings) ,Department of Computer Science, Bharathidasan University, Tiruchirapalli, Tamilnadu,* India, Oct 5-6, 2007.

[10] N. Gopalakrishna Kini , M. Sathish Kumar and B.S. Satyanarayana, "Free-Space Optical Interconnection Networks for High Performance Parallel Computing Systems", *IEEE Sponsored International conference on Recent Applications of Soft Computing in Engineering & Technology RASIET-07,Institute of Engineering & Technology, Alwar, Rajasthan,* India, pp 134-137, Dec 22-23, 2007.

N. Gopalakrishna Kini, graduated in E & C Engineering from Mysore University, Mysore in the year 1987 and received M. Tech degree in Computer Science and Engineering, Manipal Institute of Technology, Manipal in the year 1993. Presently he is pursuing Ph.D. in E & C Engineering under Manipal University, Manipal. His area of research interest is Advanced Computer architecture, Optical communication, Operating systems and Parallel Algorithms. Currently he is Reader in the Department of Computer Science and Engineering at Manipal Institute of Technology, Manipal.

M. Sathish Kumar graduated in E&C Engineering from the Institution of Engineers (India) in the year 1991 and received M.E degree in E&C Engineering from College of Engineering, Anna University, Chennai in the year 1998 and PhD in E&C Engineering from National Institute of Technology, Karnataka, India in the year 2004. He was with the Optical Communication System Lab of School of Electrical Engineering, Seoul National University from September 2006 to August 2007 as an International Scholar Exchange Fellow of the Korea Foundation for Advanced Studies. His areas of research interests include optical CDMA, multilevel modulation schemes for optical communications and nano photonic devices for optical communications. He has authored/ coauthored over 40 publications in journals/conference proceedings. Currently he is a BK 21 Contract Professor in Seoul National University, South Korea.

H. S. Mruthyunjaya has completed his bachelor degree in Electronics and Communication Engineering from Mysore University in 1988 and obtained his masters degree in Electronics and Control Systems Engineering from Birla Institute of Technology and Science, Pilani in 1994. He has a Ph.D in Electronics and Communication Engineering conferred by Manipal University. He is currently serving as a Professor in the Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal, India where he joined as a Lecturer in the year 1998. He has done research on countering non-linear effects and other noises in WDM all-optical networks by employing error control coding techniques. His areas of major interests are the Optical Fiber Communication systems, Fiber Optics, Photonic Crystal Fibers, WDM networks and systems, Electromagnetic theory & General areas of Digital Communication Systems. He has authored or co-authored over eight technical papers in refereed journals and seven International conference proceedings. He is a Fellow of the Institution of Engineers (India) and member of Indian Society for Technical Education.