Comparative Studies of Shortest Path Algorithms and Computation of Optimum Diameter in Multi Connected Distributed Loop Networks

Vasappanavara, Ramesh; Prasad, EV; Seetharamanath, MN

Abstract. Multi connected Distributed Loop (MCDL) networks with multiple hops offer smaller diameters, path lengths and better fault–tolerance. These networks have extensive uses in LAN, parallel processing, and multi processing environments. Results exist for finding shortest path between a single pair in O(δ) time where δ is diameter of distributed and in O(h/g + log h), and g is GCD(N, h), where N is number of nodes h is hop size, without prior knowledge of diameter. This paper critically analyses both these approaches and benchmarks these algorithms for suitability for adopting to MCDL Networks with large number of nodes. Results also exist for computing diameter in time O(log r) , where 0≤ r < h and h is hop length , for certain cases with restrictions imposed . But, in general, there is no closed form for diameter. In this paper we present a novel method to find out optimum diameter given N by employing curve fit using power series equations. We provide a closed form of expression for Opt Diameter = √N/2 for 1≤ N ≤112 ,rounded to nearest integer. The error is observed to be 0.035%.

Index Terms— ***Algorithm , Diameter , Distributed Systems, Optimum Diameter , Shortest path , Un directed double loop networks.***

1. INTRODUCTION

A distributed loop network G (n; h_1, h_2) [8] is an undirected graph with vertex set Zn = {0, 1 …. n-1} and edge at E = E_1 U E_2 where n, h_1, h_2 are positive integers.

E_1= {(u, u ± h_1) | u € Z_n}
E_2= {(u, u ± h_2) | u € Z_n}

[2] provides survey of Distributed Loop networks. A distributed loop network has two types of links h_1 and h_2 where every link h_1 connects u^{th} node to node (u ± h_1) mod n and link h_2 connects node u to node (u ± h_2) mod n. If either h_1 or h_2 is equal to 1, then we have a ring network with some additional links added homogenously to it. These networks are called Multi Connected Distributed Loop Networks (MCDL) and are denoted as G (n; 1, h). For example figure 1 is G (20; 1, 5). While designing the network, it is important to consider the performance requirements of the network in terms of network parameters such as; communication delays, shortest path lengths and average path lengths These parameters depend on various network characteristics such as; hop size, diameter. Hence, finding shortest path is of paramount importance in MCDL networks. A number of algorithms have been proposed in order to find the shortest paths for routing the packets in MCDL networks. In [7,12,13] authors have presented details regarding comparative studies of shortest path algorithms , fault-tolerance and delay performance studies for multi connected distributed loop networks. Several researchers also considered directed double loop networks [3, 8, 9], which have directions assigned to the four links of each node. In a directed double loop network, for each pair of links of same hop (i.e., a pair of links consisting of ±1-links only or consisting of ±h-links only), one of the two links is an inward arc and the other link is

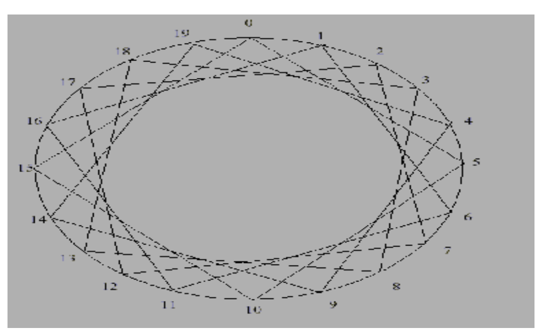

Fig 1. Distributed loop with n = 20 and h = 5

an outward arc, uniformly throughout the network. Raghavendra et al [8, 9 ,10] proposed a forward loop and backward hop (FLBH) network

Manuscript received May 10 , 2005.

Ramesh Vasappanavara* is with the Computer Science & Engineering Department, Gayatri Vidya Parishad Collge of Engineering,Visakhapatnam,India.(email:ramesh_vasappanavara@yahoo.com).

EV Prasad is with the Computer Science Department , JNTU College of Engineering Kakinada. Email: deevprasad@yahoo.co.in

MN Seetharamanath is with the Computer Science & Engineering Department,AndhraUniversityCollegeofengineering,Visakhapatnam ,India.(email:mns_nath@yahoo.com).

using +1- and −h- links and further discussed on reliable loop topologies and their performance studies. In [1], Arden and Lee proposed a chordal ring in which hop connections are formed by connecting node i to (i+h) mod N or to (i-h) mod N, depending on whether i is even or odd. In [5] Chen Bao-Xing et.al presented a method to compute the diameter of undirected double loop networks with certain restrictions, i.e when $q \le r$ and $a > (b+1)q+1$, where q, r, a and b satisfy $N = qh+r$, $0 \le r < h$, and $h-r = br + a$, $0 \le a < r$, and the algorithm runs in time O(log r), where $0 \le r < h$ and h is hop length. In our study we concentrate on algorithms , that do not need prior knowledge of diameter , i.e algorithm at [4] which has O (h/g + log n) time where g = GCD (n, h) and algorithm at[11] which has a time complexity of 'd' time steps where 'd' is the diameter of the network.

The MCDL network has a total of four link combinations denoted as (+1, +h), (-1, +h), (+1, -h), (-1, -h). The sub graphs formed by each of these link combinations can be represented as G_1 (n; +1, +h), G_2 (n; -1, +h), G_3 (n; +1, -h), G_4 (n; -1, -h). The shortest path from node l to m in G (n; +1, +h) can be represented as sp_{++} (l, m). Similarly shortest paths in the remaining link combinations can be represented. The sp (l, m) in MCDL is equal to sp (l, m) = Min[sp_{++}(l, m) + sp_{+-}(l, m) + sp_{-+}(l, m) + sp_{--}(l, m)]. Due to property of isomorphism among G_1, G_2, G_3 and G_4 [4], G_1 and G_4 are isomorphic and similarly G_2, G_3. Hence one does not need all four combinations but only two combinations are sufficient due to isomorphism. So only G_1, G_2 are sufficient for shortest paths.. The diameter of a network is defined to be the length of the maximum shortest path in the network. So we can represent the diameters of the MCDL is dia (n, h) = Max[sp(l, m)] for all l, m pairs in Z_n. The diameter of the network is dependent on the hop size and as the hop size varies the diameter value changes. The minimum of all diameters obtained with different hop sizes for a given 'n' is called optimal diameter, od(n) = $\mathrm{Min}_{\forall h}$ [dia(n,h)]. It may be noted that the 'od(n)' may be obtained at one or several values of h. There is no closed form equation for the shortest path, diameter and optimal diameter. Hence algorithms [4,6,11] exist. [6] requires knowledge of diameter to compute shortest path in O(δ) time , where δ is diameter. Algorithms at {4,11] do not require knowledge of diameter .They are compared for checking the suitability for application to large undirected double loop networks in the next section.

2. Evaluation of Algorithms

In any configuration, the paths through which the packets are routed from the source to the destination are of utmost importance [4,5] and if the packets were not routed through the shortest paths, the performance of the system would be less than expected [14]. Therefore, in most networks, the shortest path information is calculated, stored in some form or the other and used later for routing. We consider algorithms [4] and [11] in terms of their time complexities and suitability for un directed double loop networks.

2.1 GCD Based Algorithm

The time complexity of this algorithm, as mentioned in [4], is O(h/g + log (h)) where g = GCD (n, h) and h is hop size. This algorithm's time complexity is independent of the diameter. The algorithm computes shortest path only for (+h, +1) and (+h, -1) combination of links. Isomorphic property of network can be employed for finding the paths along the other two link combinations. Finally the minimum of these four is the shortest path. A detailed algorithm is found in [4]. Hence using isomorphic property, we need to calculate a total of 2*(n-1) paths as stated above. Therefore, total time to fill the array, is of the order O (2*(n-1)*(h/g + log (h)). To calculate the shortest path between one source destination pair, we need to access only one single array element i.e. c. If 'p' paths are calculated it is p*c. therefore, total time is O (2*(n-1)*(h/g + log (h) + p*c).

2.2 Computation of Optimal Diameter (OD) and hop-sizes(H) corresponding to 'OD' using Algorithm [4] given number of processors (N) as input.

Algorithm:
1. Read 'n' ; // n is number of nodes in MCDL.
2. h $\leftarrow$ 2 // h is hopsize.
3. s $\leftarrow$ 0 // s is source node.
4. for d = 1 to n-1 // d is destination node.
do
a) sp_{++} (0,d)
b) sp_{-+}(0,d)
c) sp_{--}(0,d) $\leftarrow$ sp_{++} (0, n-d) // G_1, G_4 are isomorphic
d) sp_{+-}(0, d) $\leftarrow$ sp_{-+}(0, n-d) // G_2, G_3 are isomorphic
e) sp(0,d) = Min {a, b, c, d}
end for
5. dia_h = max (sp(0,d)) ; // After all the destinations have been considered, compare all the shortest paths obtained and assign the maximum to dia_h
6. h $\leftarrow$ h + 1; If (h < n-1) go to step 3.
7. od = Min (dia_h) ; // Compare all values of dia_h and assign the minimum value of diameter to 'od'. This is called 'h_{od}'.

Table : 1 for nodes 5 to 50 values of od & h_{od}

Nodes	od	h_{od}
5	1	2,3
6	2	2,3,4
7	2	2,3,4,5
8	2	2,3,4,5,6
9	2	2,3,4,5,6,7
10	2	4,6
11	2	3,4,7,8
12	3	2,3,4,5,6,7,8,9,10
13	2	5,8
14	3	3,4,5,6,8,9,10,11
15	3	3,4,5,6,9,10,11,12
16	3	4,6,10,12
17	3	3,4,5,6,7,10,11,12,13,14
18	3	4,5,7,11,13,14
19	3	4,5,7,8,11,12,14,15
20	3	8,12
21	3	6,8,13,15
22	3	6,16
23	3	5,9,14,18
24	4	4,5,6,7,9,10,14,15,17,18,19,20
25	3	7,18
26	4	4,6,7,10,11,15,16,19,20,22
27	4	4,5,7,8,10,11,16,17,19,20,21,22,23
28	4	5,6,8,11,12,16,17,20,22,23
29	4	5,6,8,11,12,17,18,21,23,24
30	4	8,12,18,22
31	4	7,9,12,13,18,19,22,24
32	4	7,9,12,20,23,25
33	4	5,6,7,9,13,14,19,20,24,26,27,28
34	4	6,28
35	4	10,25
36	4	8,10,26,28
37	4	8,14,23,29
38	4	16,22
39	4	7,11,28,32
40	5	6,7,9,11,12,15,16,17,23,24,25,2829,32,33,34
41	4	9,32
42	5	9,12,16,18,24,26,30,33
43	5	5,8,9,12,16,17,18,19,24,25,26,27,31,34,35,38
44	5	8,10,12,32,34,36
45	5	6,7,8,10,13,33,36,38,39,40
46	5	6,7,8,10,13,33,36,38,39,40
47	5	7,10,13,14,18,20,27,29,33,34,37,40
48	5	18,20,28,30
	5	9,11,14,35,38,40
50	5	9,11,14,19,21,22,28,29,31,36,39,41

2.3 Computation of Optimal Diameter (od) and hop -sizes corresponding to'od' using Algorithm [11] given number of processors (n) as input.

Algorithm :

1)i ← 0

2)s_i = { 0 }

3)I ← i + 1

4)s_i = set formed by adding 1, h, -h and -1 mod n to each element x in s_{i-1} such that if y € $s_{i-1,}$ y ∉ s_i

5)Mark distances to all elements of s_i by adding 1.

6)If s_i does not contain n elements, which is number of nodes then Goto 3.

7)End

The algorithm was used for 'n' between 5 and 1000. to find the 'od' and the results are tabulated in Table 1. From the table it can be seen that as the size of the network increases (i.e. 'n' increases), the value of 'od' increases as expected. This is because the processors are farther apart and the paths required become longer. It can also be seen that the values of 'hod' the hop sizes where OD occurs also increase with 'n'. This indicates that for larger networks, larger hop sizes should be preferred and these larger hops have greater chance of providing shorter paths. We have fitted a linear graph into these values of 'od' versus n as given in Figure 2. The equation od = 0.017924*n + 6.1869 fits it and is of linear relation line. The values of 'od' for n = 5 to 1000 have been calculated using the above equation. Similarly, we have also fitted a non-linear power series graph, Fig 3, into these values of 'od' as od = 0.6943 * n0.5029 . The values of 'od' for n = 5 to 42 have been calculated using the above equation, rounding the value of 'od' to the nearest integer. We have drawn errors in linear and non linear fits in Fig:4 and observed that non-linear approximation gives a better results. Hence this non-linear approximation fit has been extended to the range of 1 to 1000.The respective equations for the ranges and the corresponding errors have been tabulated as below in Table 2. We have also observed that row1 of table2 can be approximated to governing equation Opt dia = Round(√ N)/2 for 1≤ N ≤112 ,rounded to nearest integer. The error is observed to be 0.035%

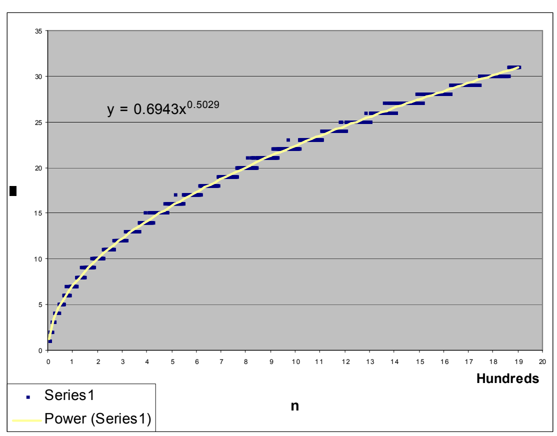

Fig 2. Linear approximation

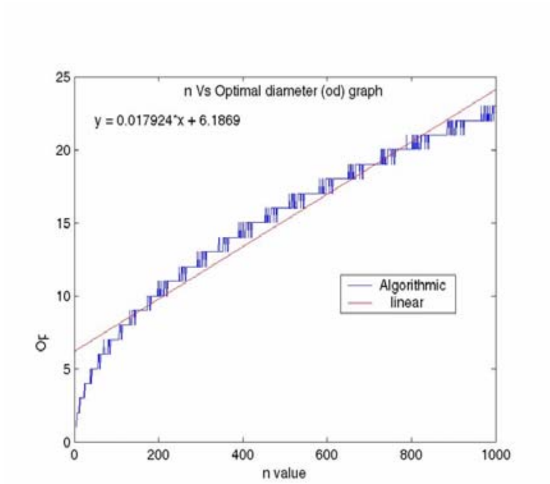

Fig:3 Non-Linear approximation

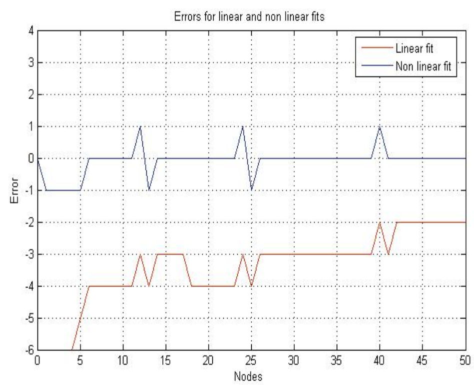

Fig 4 : comparison of linear and non linear fits

Slno	Nodes Range	Equation (power series) Opt.dia	avg% error
1	1-112	0.672*X^0.5109	0.035
2	113-172	0.5928*X^0.5122	0.03
3	173-262	0.8179*X^0.4561	0.05
4	263-300	1.9566*X^0.3148	0.05
5	301-418	1.9393*X^0.3148	0.008
6	419-454	0.067*X^0.8788	0
7	455-480	14*X^7E-11	0.08
8	481-516	0.06815*X^0.8788	0
9	517-610	16.203*X^-0.0019	0.01
10	611-682	0.523*X^0.5381	0.03
11	683-758	23.185*X^-0.0384	0
12	759-838	24.266*X^-0.0365	0.01
13	839-972	18.271*X^0.0134	0.01
14	973-1000	0.2359*X^0.6557	0

Table 2: Average % error for computing Optimum diameter given number of node (n) using algorithm and governing equation given by power series.

2.4 Algorithm [11]

This algorithm [11] does not give the number of ‘h’ and ‘1’ links used; rather it provides us shortest path length. This information about the number of h and 1 links is necessary for the implementation of various fault-tolerant algorithms [4,5]. This algorithm fills an array of size ‘n’ with an integer which is the shortest distance from node 0.The time complexity of this algorithm is O (d) where d is the diameter of the network, however for the computation of the shortest path the knowledge of the diameter is not required. Once the array is filled, the k^{th} element gives the shortest path from node 0 to node the k. This can be extended to find the shortest path between nodes u and v basing on the observation that shortest path from u to v is equal to shortest path from 0 to (v-u) mod n. . Once the array is filled, the number of array elements that need to be accessed is variable and can be a maximum of d elements, where d is the diameter. Therefore, the total time required to calculate shortest path is d*c, where c is the time taken to access one single array element. The total time to fill the array and calculate one shortest path is O (d + d*c). If ‘p’ paths are to be calculated, time is d + d*p*c. As diameter is dependent on number of processors and hop size, the time complexity increases with increase in diameter and hence the number of processors.

3.0 Time Complexities

For [11], time required to fill up the array is O (d). For [4], the time to fill the array is the time required to calculate the shortest paths O (h/g + log h). These algorithms were implemented on IBM iv Pentium Machines and the time actually consumed by these algorithms computed by varying number nodes up to 6000. . The results are at Fig 5. It is observed that algorithm [4] performs better and displayed a flat response , indicating the independence of diameter for computing the shortest path for large n. The algorithm at [11] time consumed raises exponentially confirming O(δ) nature of the algorithm. Its performance is better only for networks with less number of nodes.

Charts 1 and 2 show the times we calculated for paths p = 50 and p = 200 respectively where p is number of shortest paths computed.. The network considered is 200 nodes and hop sizes were h= 4, 5, 6, 8, 10. The diameters obtained were, 16 and 14 respectively. ‘D’ indicates algorithm at [11] and ‘h/g+log h’ implies algorithm at [4]. Charts 3 and 4 show the time performance of the two algorithms for a network having 200 processors, with hop sizes 4 and 10 respectively. Here we have considered number of path lengths as parameter .We can see that time taken in [11] increases as the number of nodes increases, where as [4] with isomorphic property, gives a flat response, independent of the diameter and number of nodes.

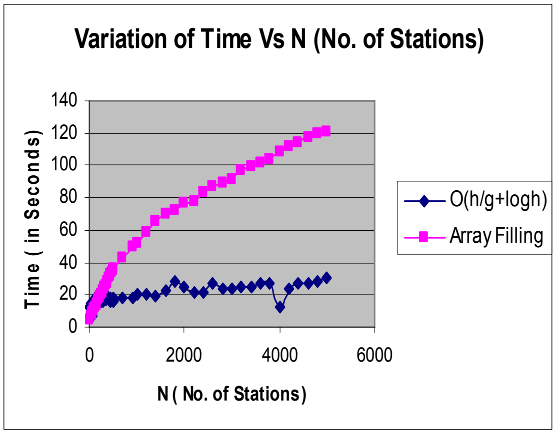

Fig 5: Time units consumed by algorithms [11 , 4]

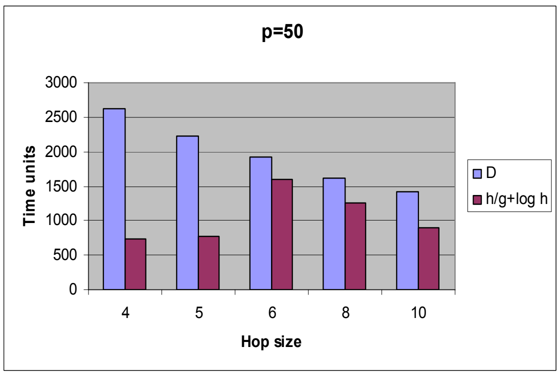

Chart 1: Time units (delay) vs. hop size, n=50

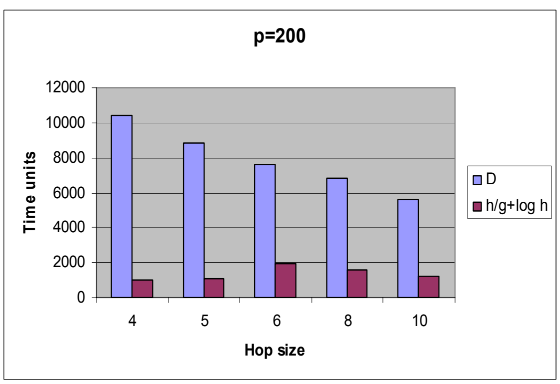

Chart 2: Time units (delay) vs. hop size, n=20

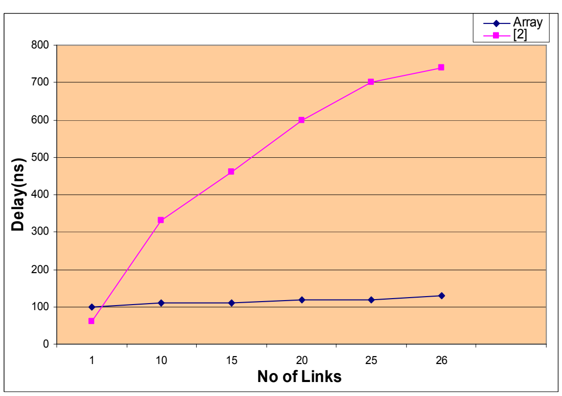

Chart 3:Delay vs. no of links traversed n=200, h=4

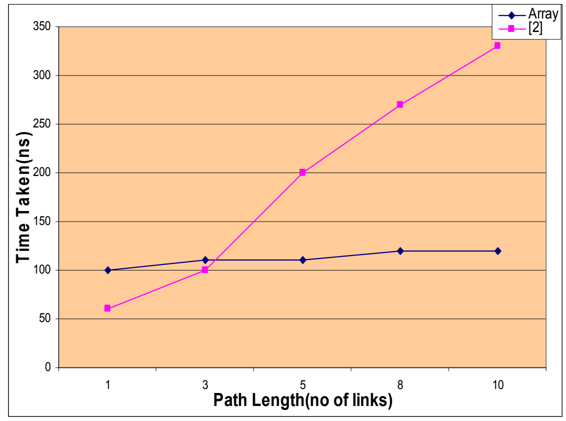

Chart 4: Delay vs. no of links traversed (path lengths) n=200, h=10

4.0 Conclusions

This paper considered MCDL networks , of the type G(n,+h,+1)G(n,+h,-1)G(n,-h,+1) G(n,-h,-1). We have critically analyzed the shortest path algorithms at [4 ,11]], that do not demand prior knowledge of diameter, for their suitability for adoption practical implementations on MCDL networks. Algorithm at [4] takes O[h/g+log(h)] where g = GCD(n,h) which is independent of diameter and computes shortest path in terms of type and number of links. Algorithm at [11] takes 'd' (diameter) time steps. Since diameter is dependent on the number of processors and hop size, the time complexity increases with increase in number of processors. Algorithm at [4] owing to diameter independence and flat and constant response times with increase in no of processors and path lengths is apt for computing shortest paths in MCDL networks than [11]. We have presented a novel method to find out optimum diameter, given n , the number of nodes , by employing curve fitting using power series. It is observed that the predicted values from the curve fitting approach and from the algorithm are almost same with an average error of 0.05 %

ACKNOWLEDGMENT

Authors would like to thank Gayatri Vidya parishad college of Engineering for providing all the facilities and encouragement and Prof(Dr.) NB Venkateswarlu for his valuable suggestions.

5.0 References

[1] B.W. Arden , H. Lee, Analysis of chordal ring networks , IEEE Trans. Compt. C-30(4)(1981)291-295.

[2] J.C. Bermond , F.Comellas , D.F.Hsu, Distributed loop computer networks : a survey . Journal of Parallel and Distributed Computing 24(1995) 2-10.

[3] F.T. Boesch, J.F Wang. Reliable circulant networks minimum transmission delays . IEEE Trans. Circuits Systems CAS-32(1985)11286-1291.

[4] N. Chalamaiah and B. Ramamurthy, "Finding shortest paths in distributed loop networks", Information rocessing letters, 67, pp. 157-161, 1998.

[5] CHEN Bao-Xing, Wenjun Xiao , Behrooz Parhami Diameter formulas for class of undirected double loop networks, Journal of Interconnection Networks, Vol. 6, No. 1 (2005) 1-15

[6] Krishnendu Mukhopadhyaya and Bhabani P. Sinha, "Fault-Tolerant Routing in Distributed Loop Networks", IEEE Trans. on Computers, Vol. 44, No. 12, 1995.
[7] G Praveen Babu, Ramesh Vasappanavara and Nimmagadda Chalamaiah "Comparative Study of shortest path routing algorithms in MCDL Networks" Proceedings of Asia Pacific International Conference ObComAPC-2004: Parallel and Distributed Computing Technologies, (December, 13-15, 2004:Vellore Institute of Technology, Vellore, India.
[8] Raghavendra and M Gerla,AAvizienis, "Reliable loop topologies for large local loop networks", IEEE Trans on Computers, 34(1) pp 46-55,1985.
[9] C.S Raghavendra , J.A. Silvester, double Loop Network architecture – a performance study , IEEE Trans. Comm COM-3392) (1985) 185-187
[10] S. Raghavendra , J.A. Silvester, A survey of multi-connected loop topologies for local computer network. Comput. Networks ISDN Systems 11(1986) 29-42
[11 Raj Katti and V.V. Bapeswara Rao, "An array based technique for routing messages in distributed loop networks", presented at : ISCAS 2002 IEEE International Symposium on Circuits and Systems, 26 -29 May, 2002, pages V 9-12, Raj Katti and V V Bapeswara Rao.
[12] Ramesh Vasappanavara, Sasikiran Kandula and Nimmagadda Chalamaiah "Fault Tolerance Studies" for wormhole routing in Multi connected Double Loop Networks", Asian Applied Computing Conference AACC-2004, Khatmandu, Nepal. (October, 28-31, 2004, Springer Veriag LNCS 3435, pp198-204)
[13] Ramesh Vasappanavara, Sasikiran Kandula and Nimmagadda Chalamaiah "Delay Performance Simulation Studies in MCDL Networks" International Conference on Systemics, Cybernetics, and Informatics ICSCI-2005, Hyderabad, India. (January, 06-09, 2005 Proceedings pp948-953).
[14] Suresh Chalasani and Rajendra V Boppana , " A Frame work for Designing a Deadlock-free wormhole routing Algorithms", IEEE Tr on Parallel and Distributed systems Vol 7 No 2 Feb 1996.

Authors

Ramesh Vasappanavara received his B.Tech degree in Electronics and Communication Engineering from Andhra University (AU) in 1977. He received M.Tech degree in Computer Science from Indian Institute of Technology, Kharagpur in 1984. He has worked as Commander, Naval Systems for and Defense R&D for twenty-five years prior to joining as Professor and Head of the Department, Department of Computer Science, GVP college of Engineering, Visakhapatnam, India in 1999. His research interests include parallel and distributed systems, multimedia and network security

EV Prasad received his PhD from Indian Institute of Technology Chennai India in 1990. He has served as Professor and head of Department of Computer Science at College of Engineering Anantapur , Andhra Pradesh India. Since June 2005 , he is working as Vice Principal JNTU College of Engineering Kakinada . He has several research publications to his credit. He has served as member review committee for IEEE Transaction on Reliability . His areas of research interests include fault tolerant studies , Software engineering , and soft computing.

M N Seetaramnath received his PhD from Andhra University Visakhapatnam in the year 1972. He has worked as Systems Programmer at Electronics Corporation of India Limited. Subsequently in the year 1977, he joined as Asst Professor in the Department of Computer Science at Indian Institute of Technology Kharagpur . Since 1985, he is Professor Department Computer Science and Engineering Andhra University College of Engineering Viakhapatnam. His areas of interest include Operating Systems, Systems Programming, Parallel and Distributed Computing, and Data Warehousing and Data Mining.