Computer Science Journal of Moldova, vol.**19**, no.3(57), 2011

A $\sqrt{\frac{N}{G}}$ Method for Generating Communication Sets

Rupali Bhardwaj, V.S. Dixit, Anil Kr. Upadhyay

Abstract

In the fully meshed network, where every node is connected directly to every other node, network traffic is very high because in the fully meshed network, number of communication links is $\frac{N\times(N-1)}{2}$ and communication cost is $2 \times N \times (N-1)$, where N is total number of nodes in the network. To minimize network traffic, we propose an algorithm for generation of communication sets that allows any two nodes to communicate by traversing at most two nodes regardless of the network size by dividing the nodes in the system into subgroups of size G where $G \geq 1$, which are then organized into quorum groups of size $k_1 = \left(\sqrt{\frac{N}{G}}\, approx.\right)$ in a method similar to that used in Maekawa's algorithm except that now quorum groups are constructed out of subgroups instead of nodes. The performance analysis of the proposed partitioning algorithm shows that it significantly reduces network traffic as well as total number of communication links required for a node to communicate with other nodes in the system.

Keywords: Quorum; Coterie; Communication sets; Network traffic

1 Introduction

Every node is connected directly to every other node in the completely connected network, so that number of communication links is very high, $\frac{N\times(N-1)}{2}$, where N is total number of nodes in the network. The number of hops used in a completely connected network is $N \times (N-1)$

because each node can reach every other node using one hop only. In the proposed partitioning algorithm each node can communicate with other nodes by either one or two hops regardless of the network size by dividing the nodes in the system into subgroups of size G where $G \geq 1$, which are then organized into quorum groups of size $k_1 = \left(\sqrt{\frac{N}{G}}\, approx.\right)$ in a method similar to that used in Maekawa's algorithm [1] except that now quorum groups are constructed out of subgroups instead of nodes. The idea presented in this paper is that the entire system is divided into number of subsets equal to the number of nodes. Each node in the system is assigned a subset of size k, $(k = \sqrt{N}\, approx.)$. After that, $\left(\frac{N}{G}\right)$ subgroups are formed with G subsets per subgroup. Now subgroups organized into $\left(\frac{N}{G}\right)$ quorum groups of size $k_1 = \left(\sqrt{\frac{N}{G}}\, approx.\right)$ in a method similar to that used in Maekawa's algorithm [1]. Now, each such Quorum group will be associated with G subgroups. The intersection between every pair of quorum groups is exactly one subgroup instead of a node. The performance of proposed partitioning algorithm will be evaluated using network traffic, communication links, communication cost and routing table size as a criterion for evaluation. The major contributions of the paper include: (i) a novel approach for generating communication sets proposed by considering the concept of Quorums (ii) simulation results show that proposed algorithm performed better than Hajj's [2] scheme with respect to reducing the network traffic and the total number of communication links. The remainder of the paper is organized as follows: Section 2 features existing research work in the field of generation of communication sets. Section 3 outlines the model on which our algorithm is based. The proposed partitioning algorithm is presented in section 4. Section 5 discusses about the performance of the proposed algorithm. Finally, section 6 concludes the paper.

2 Related Work

Maekawa's algorithm [1] is a distributed algorithm and total number of messages per mutual exclusion required $3\sqrt{N_k}$ messages: $\sqrt{N_k}$ messages to convey a request, $\sqrt{N_k}$ messages to obtain permissions, and $\sqrt{N_k}$ messages to release mutual exclusion. It is assumed that the nodes communicate only by passing messages instead of sharing of memory. Drawback of Maekawa's algorithm is that there is no procedure given for the construction of overlapped sets when $k-1$ is power of a prime number. Wassim EI-Hajj [2] presented a special network topology that is unique in terms of nodes interconnection, communication sets are designed by two techniques, when $p+1$ is the power of a prime number and when $p+1$ is not the power of a prime number. A distributed routing protocol is proposed by them after constructing the initial topology that allows any two sites to communicate with each other by traversing at most two nodes regardless of the network size. If k is power of a prime number then there exists a finite projective plane of order k. If either $k-1$ or $k-2$ is divisible by 4 and k is not a sum of two integral squares ($k \neq a^2 + b^2$), then finite projective plane of order k does not exist. $k \times (k+1) + 1$ lines are there in a finite projective plane of order k [3]. A method for creating a coterie with quorum size $k+1$, where k is a prime number is presented by K.T. Tseng, C.B. Yang [4]. In this paper [5], grid based quorums are constructed using paths that bear resemblance with billiard ball paths, through the resulting quorums of size $\sqrt{2N}$ as compared to $2\sqrt{N}$ of Maekawa's grid based method. Barbara [6] examined the relationship through pair wise non null intersections between weighted voting and sets of nodes. S. Rangarajan [7] proposed a distributed fault tolerant algorithm for the replica control problem that can be parameterized to achieve the desired balance between low message overhead and high data availability. M. Neilsen [8] introduced a new class of protocols within the unifying framework based on quorums which generalized all consensus protocols which used m rounds of message exchange. Load of a particular node is distributed over m identical nodes by partitioning the nodes into mutually overlapping subsets so that, through querying only a few

nodes, a node gets the partial system state information [9]. Advantage is that it significantly reduced the total number of messages required for a node to take scheduling decision.

3 The System Model

In our proposed algorithm, each node in the network allocates a communication set satisfying the following constraints:
A_1. $S_i \bigcap S_{j \neq \phi} \ \ \forall i, j \in 1, 2 \ldots N$.
A_2. S_i, $1 \leq i \leq N$ always contains i.
A_3. The size of $|S_i|$ is k for any i. That is,

$$|S_1| = |S_2| = |S_3| = \ldots\ldots = k$$

A_4. R_i is contained in k $S'_j s$ $\forall i, j \in 1, 2 \ldots N$.

Where set of subsets S_i is called coterie, R_i is referred to as requesting subset of S_i. According to Maekawa algortihm [1], for a network with N number of nodes, create N different sets of size k ($k = \sqrt{N}\, approx.$) such that N is represented as $N = k \times (k-1)+1$, where $k-1$ is a prime number. If N cannot be represented in this form, then some dummy sites have to be added for the construction to work.

Considering a system with 13 nodes, we see that the communication sets of the nodes 3, 7, 8 & 12 are as follows:
$S_3 = \{3, 7, 8, 12\}$, $S_7 = \{2, 7, 10, 13\}$,
$S_8 = \{1, 8, 9, 10\}$, $S_{12} = \{4, 5, 10, 12\}$.

Rule 1 states that there is at least one common node between the communication sets of any two nodes. Communication sets S_3 and S_7 have node n_2 in common as in Figure 1. Rule 2 states that each node should belong to its own communication set, i.e. node n_1 is part of communication set S_3. Rule 3 states that the size of each communication set is to be equal to k, i.e. size of S_3, S_7, S_8, S_{12} is 4 whenever $N = 13$. The fourth constraint states that each site should be contained in k other sets. Whenever communication sets are generated, each node can communicate with other nodes by either one or two hops regardless of the network size. So in case of S_3, node 3 will exchange its

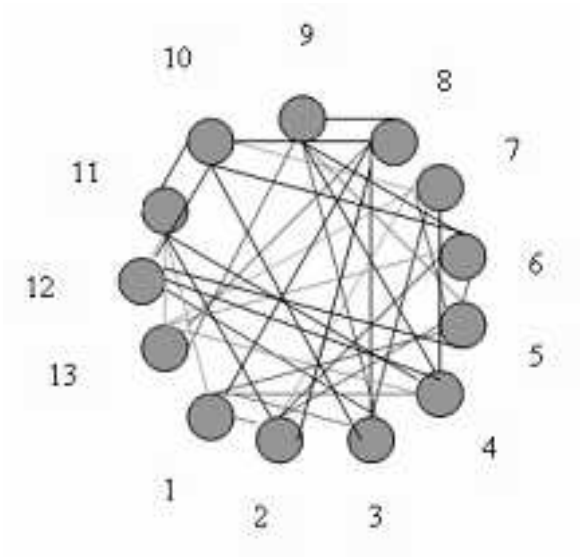

Figure 1. Network topology of size $N = 13$ nodes

state change messages only with node 7, 8 and 12. From these nodes it can also acquire the state information of nodes numbered 1, 2, 4, 5, 7, 8, 9, 10, 12 and 13 and can update its system state table also. So node 3 does not need to communicate explicitly with all the nodes except missing nodes 6, 11. So that total number of messages is equal to $2\times(k-1)+2\times$ *number of missing nodes.* Therefore, given a network of size N, our task is to generate a communication set for each node such that constraints A_1 through A_4 are satisfied.

4 The Proposed Partitioning Algorithm

Consider a system with N ($N \geq 2$) nodes, entire system is divided into number of subsets equal to the number of nodes. Each node in the system is assigned a communication set of size k, ($k = \sqrt{N}\, approx.$) according to algorithm 1 & 3. After that, we group these subsets into $\left(\frac{N}{G}\right)$ subgroups with G subsets per subgroup, where $G \geq 1$. We then

construct $\left(\frac{N}{G}\right)$ quorum groups such that each quorum group is made up of $k_1 = \left(\sqrt{\frac{N}{G}}\, approx.\right)$ subgroups; where each subgroup contains G subsets according to algorithm 1&3, except that now the quorum groups are constructed out of subgroups instead of nodes. Whenever Quorums are generated, each node can communicate with other nodes by either one or two hops regardless of the network size. Algorithm 2 calculates total number of missing nodes (worst case), through which each quorum communicated explicitly, because each quorum contained only partial system information.

4.1 Case 1: $N = k(k-1)+1$

Algorithm 1: Generating the communication sets where $k-1$ is a prime number
Result: Generate k groups of $(k-1)$ non-intersecting sets.
1. begin
2. Data K, count=2
3. Result: Generate a matrix B [K] [K-1]
4. for i = 0 to K do
5. for j = 0 to K-1 do
6. B[i] [j] = count++
7. end loop
8. input c = 1
9. Result: Calculate all $S[i]$ rows by performing operation on intersecting rows.
10. begin
11. for i = 0 to K do
12. S[c] [0] = 1
13. for j = 0 to K-1 do
14. S[c] [j+1] = B[i] [j]
15. end loop
16. c = B[i+1] [0]
17. end loop
18. end

19. input c = 2
20. Result: Calculate all $S[i]$ columns by performing operation on intersecting column.
21. begin
22. for j = 0 to K-1 do
23. S[c] [0] = 2
24. for k = 0 to K-1 do
25. S1[c] [k+1] = B [k+1] [j]
26. end loop
27. c = B[1] [j+1]
28. end loop
29. end
30. Result: Perform the operations of diagonal matrix
31. begin
32. set c = 0, s = 0, t2 = 2, z = 2
33. for k =1 to K-1 do
34. set t3 = 0, n = 0, r = 0
35. while (r < k-1) do
36. if (t3 < z-1) then
37. set t1 = B [t2] [++t3]
38. else
39. set t1 = B [0] [k]
40. set S1[t1] [0] = B[0] [k]
41. set c = 1+ n, j =1+ n, m1=1
42. for i =1 to k-1 do
43. S1[t1][m1] = B[i] [j-1]
44. m1++
45. c+ = k + s
46. P = c % (k-1)
47. if (P==0) then
48. set j = z
49. else
50. set j = P
51. end for loop
52. n = n + 1, r = r + 1

53. end while loop
54. end for loop
55. end
56. Result: Display the no. of S[i] matrix.
57. for i =1 to n do
58. for j = 0 to k do
59. print S1[i] [j]
60. end loop
61. end loop
62. end

Algorithm 2: Find the all missing nodes
1. begin
2. input l, k, c
3. for t1 = 1 to N do
4. initialize j = 0
5. for m1 = 0 to k do
6. l = S [t1] [m]
7. k = 0
8. while k <= K do
9. c = 0
10. for a = 0 to 1000 do
11. if SS[a]==S1[l] [K] then
12. c =1
13. break
14. end if
15. end for loop
16. if c = 0 then
17. SS[j] = S1 [l] [k]
18. j++
19. end if
20. end while loop
21. end step 4 for loop
22. for i = 0 to 1000 do
23. if S[i] != 0 then
24. print SS[i]

25. else
26. break
27. end if
28. end for loop
29. print t1
30. print N-i
31. end for loop of Step 2
32. end

4.2 Case 2: $N \neq k(k-1)+1$

When $N \neq k \times (k-1)+1$, we need to find the value of a number M such that $M = k \times (k-1)+1$, where $M > N$. First, we create a degenerate set of S_i's in a similar way as in algorithm 1 and eliminate $M - N$ S_i's from this coterie as well as these nodes must be replaced from each quorum such that M_1 replaced by N_1, M_2 replaced by N_2 etc.

Algorithm 3: Calculation of D and replacement of D sites
1. begin
2. k = $\sqrt{N}$ and let M = k (k-1) + 1
3. if M < N then
4. k = k + 1
5. D = M – N
6. Replace these D sites, D_1, $D_{2.....}$ from coterie as well as from quorums by N_1, $N_2 \ldots$ in such a way that each quorum size should be k in such a way that R_i is contained in more than k S_j's, for all $i, j \in 1, 2 \ldots N$.
7. If there is a duplication of node in S_i then insert a new node (starting from the first node) in such a way that R_i is contained in k S_j's, for all $i, j \in 1, 2 \ldots N$
8. end

An example of such grouping strategy is discussed here. We consider a system with 28 nodes; Figure 2 groups nodes with $\sqrt{N}$ nodes per subset. Figure 3 shows subgroups that contain four subsets ($G = 4$)

per subgroup and there are seven quorum groups $\left(\frac{N}{G}\right) = 7$ as shown in Figure 4. Each quorum group consists of $k_1 = 3$, $k_1 = \left(\sqrt{\frac{N}{G}}\, approx.\right)$ subgroups. The intersection between every pair of quorum groups is exactly one subgroup. Now construct communication sets for $N = 28$ by algorithm 1 and 3.

$S_1 = \{1, 2, 3, 4, 5, 6\}$
$S_2 = \{2, 7, 12, 17, 22, 27\}$
$S_3 = \{3, 11, 12, 18, 24, 27\}$
$S_4 = \{4, 8, 15, 17, 24, 28\}$
$S_5 = \{5, 8, 16, 19, 22, 27\}$
$S_6 = \{6, 11, 15, 19, 23, 27\}$
$S_7 = \{1, 7, 8, 9, 10, 11\}$
$S_8 = \{2, 8, 13, 18, 23, 28\}$
$S_9 = \{2, 9, 14, 19, 24, 26\}$
$S_{10} = \{2, 10, 15, 20, 25, 27\}$
$S_{11} = \{2, 11, 16, 21, 26, 28\}$
$S_{12} = \{1, 12, 13, 14, 15, 16\}$
$S_{13} = \{3, 7, 13, 19, 25, 28\}$
$S_{14} = \{3, 8, 14, 20, 26, 27\}$
$S_{15} = \{3, 9, 15, 21, 22, 28\}$
$S_{16} = \{3, 10, 16, 17, 23, 26\}$
$S_{17} = \{1, 17, 18, 19, 20, 21\}$
$S_{18} = \{4, 9, 16, 18, 25, 27\}$
$S_{19} = \{4, 10, 12, 19, 26, 28\}$
$S_{20} = \{4, 11, 13, 20, 22, 26\}$
$S_{21} = \{4, 7, 14, 21, 23, 27\}$
$S_{22} = \{1, 22, 23, 24, 25, 26\}$
$S_{23} = \{5, 9, 12, 20, 23, 28\}$
$S_{24} = \{5, 10, 13, 21, 24, 27\}$
$S_{25} = \{5, 11, 14, 17, 25, 28\}$
$S_{26} = \{5, 7, 15, 18, 26, 6\}$
$S_{27} = \{1, 27, 28, 26, 6, 8\}$
$S_{28} = \{6, 7, 16, 20, 24, 28\}$

Figure 2. Grouping nodes with $\sqrt{N}$ nodes per subset

$G_1 = (S_1, S_2, S_3, S_4)$
$G_2 = (S_5, S_6, S_7, S_8)$
$G_3 = (S_9, S_{10}, S_{11}, S_{12})$
$G_4 = (S_{13}, S_{14}, S_{15}, S_{16})$
$G_5 = (S_{17}, S_{18}, S_{19}, S_{20})$
$G_6 = (S_{21}, S_{22}, S_{23}, S_{24})$
$G_7 = (S_{25}, S_{26}, S_{27}, S_{28})$

Figure 3. Grouping subsets with G subsets per subgroup

$Q_1 = \{G_1, G_2, G_3\}$
$Q_4 = \{G_1, G_4, G_5\}$
$Q_6 = \{G_1, G_6, G_7\}$
$Q_2 = \{G_2, G_4, G_6\}$
$Q_5 = \{G_2, G_5, G_7\}$
$Q_7 = \{G_3, G_4, G_7\}$
$Q_3 = \{G_3, G_5, G_6\}$

Figure 4. Grouping subgroups with $\sqrt{\frac{N}{G}}$ subgroups per Quorum

5 Experimental Study

Performance of the proposed partitioning algorithm is evaluated using network traffic, communication links, communication cost and routing table size as a criterion for evaluation. According to simulation study, the proposed algorithm performs better than Hajj's [2] algorithm with respect to reducing the network traffic and the number of communication links. First, consider Hajjs [2] algorithm in some detail which is based on Maekawa's [1] algorithm.

5.1 Hajj's Algorithm

In Hajj's Algorithm [2], it is shown that how N sites in an ad-hoc network can be divided into communication sets with $\sqrt{N}$ nodes per communication set such that constraints A_1 through A_4 are satisfied. Communication between two nodes takes place either directly or through a third node, which will exist to connect them. An example is shown

in Figure 5 where there are 7 nodes and 7 communication sets with 3 nodes per communication set.
$S_1 = \{1, 2, 3\}$
$S_4 = \{1, 4, 5\}$
$S_6 = \{1, 6, 7\}$
$S_2 = \{2, 4, 6\}$
$S_5 = \{2, 5, 7\}$
$S_7 = \{3, 4, 7\}$
$S_3 = \{3, 5, 6\}$

Figure 5. Grouping nodes with $\sqrt{N}$ nodes per communication set

So in case of S_3, node 3 will exchange its state change messages only with node 5 and 6. From these nodes it can also acquire the state information of nodes numbered 1, 2, 5, 6 and 7 and can update its system state table also.

5.2 Simulation Result

5.2.1 Routing Table Size

Usually, each node stored a routing table of size $(N \times N)$ for making routing decisions. Now no routing table needs to be stored on any one of node, route is computed on demand.

5.2.2 Communication Links

The number of links in a completely connected network is $\frac{N \times (N-1)}{2}$; number of links required by our algorithm is $N \times \left(\sqrt{NG} - 1\right)$.

$$Gain = \left(\frac{N-1}{2 \times \left(\sqrt{NG} - 1\right)} \right) \tag{1}$$

5.2.3 Communication Cost

The number of hops used in a completely connected network is $N \times (N-1)$ because each node can reach every other node using one hop. In

the proposed algorithm each node can communicate with other nodes by either 1 or 2 hops. So, total number of hops is = $N\times$(no. of 1 hop + 2 $\times$(no. of 2 hops)) = $2 \times N \times \left(\sqrt{NG} - 1\right)^2$.

$$Loss = \left(\frac{N-1}{2 \times \left(\sqrt{NG} - 1\right)^2} \right) \tag{2}$$

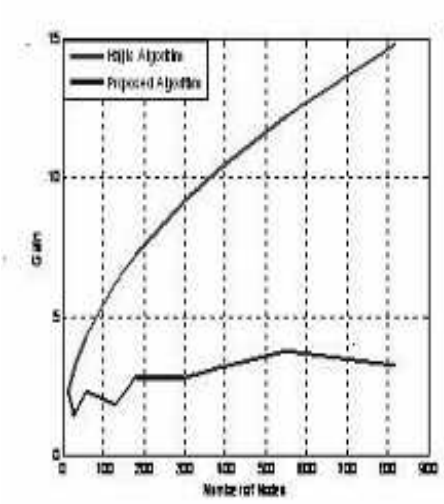

Figure 6. Gain

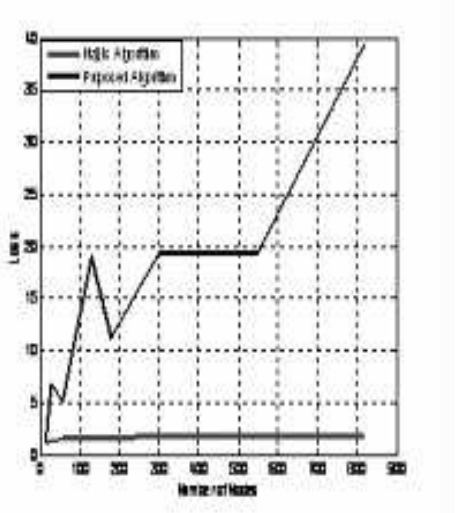

Figure 7. Loss

5.2.4 Network Traffic

Quorum contained only partial system information, so that some node information is missed through which each quorum communicated explicitly. If quorums explicitly communicate with these nodes, then the required number of messages will be $2 \times \{[(k-1) \times G]\} + 2\times$ *number of missing nodes* according to the proposed partitioning algorithm where, as in traditional systems, the number of required messages is $[2 \times (N-1)]$ and in Hajj's algorithm [2], the number of required messages is $[2 \times (k-1)+$ *number of missing nodes* $\times 2]$.

5.2.5 Analysis

It can be clearly seen from Figure 9 that network traffic is less in the proposed algorithm as compared to Hajj's algorithm [2], as N increases, traffic increases almost in a linear fashion. If communication sets are generated using the proposed algorithm, then the number of missing nodes is less as compared to Hajj's algorithm [2] (Figure 8). But in terms of gain (Figure 6) and loss (Figure 7), Hajji's algorithm [2] performs better than the proposed algorithm. As N increases, the gain increases almost in a linear fashion, while the loss is bounded by a constant in case of Hajj's [2] algorithm.

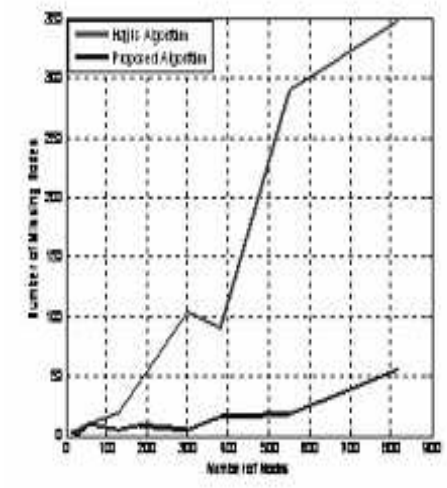

Figure 8. Missing Nodes Information

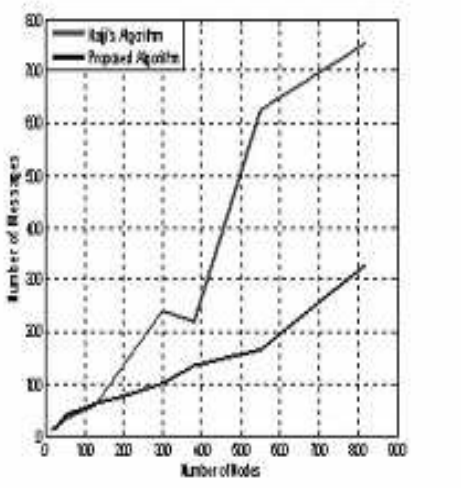

Figure 9. Network Traffic

6 Future work and conclusion

It is shown that from the perspective of the network traffic and partial system information, the proposed algorithm provides a significant performance over the traditional one. We illustrated by simulation that this protocol reduces both network traffic and number of messages communicated explicitly with missing nodes. Now, we will propose an algorithm for load balancing problem in P2P system using concept of

this paper.

References

[1] M. Maekawa. *A $\sqrt{N}$ algorithm for mutual exclusion in decentralized systems*, ACM Trans. Computer Systems,Vol. 3, pp.145–159, May 1985.

[2] Wassim El-Hajj, Hazem Hajj, Zouheir Trabelsi. *On Fault Tolerant Ad Hoc Network Design*, IWCMC'09, June 21-24, 2009,

[3] A. A. Albert, S.R. *An Introduction to Finite projective Planes*, New York: Holt, Rinehart, Winston, 1968.

[4] K.T. Tseng, C.B. Yang. *A $\sqrt{N}$ method for generating coteries*, M. Tech. thesis paper.

[5] Divyakant Agrawal, Omer Egecioglu, Amr El Abbadi. *Billiard quorums on the grid*, Information Processing Letters 64 (1997) 9–16, Elsevier Science.

[6] Hector Garcia-Molina, Daniel Barbara. *How to Assign Votes in a Distributed System*, J. ACM, Vol. 32, No. 4, pp. 841–860, 1985.

[7] Sampath Rangarajan, Sanjeev Setia, and Satish Tripathi. *A Fault-Tolerant Algorithm for Replicated Data Management*, IEEE Transactions on parallel and distributed systems, VOL. 6, NO. 12, December 1995.

[8] Mitchell L. Neilsen, Masaaki Mizuno. *Decentralized Consensus Protocols*, supported by National science Foundation under Grant CCR-8822378.

[9] Md. Abdur Razzaque1 and Choong Seon Hong. *Dynamic Load Balancing in Distributed System: An Efficient Approach*, funded by the "Korean Government (MOEHRD)" (KRF-2006-521-D00394), 2006.

A $\sqrt{\frac{N}{G}}$ Method for Generating Communication Sets

Rupali Bhardwaj, V.S.Dixit, Anil Upadhyay

Received January 27, 2011
Revised May 10, 2011

Rupali Bhardwaj
Institution: Krishna Institute of Engineering and Technology,
Mahamaya Technical University, Noida, India
Address: Department of MCA, KIET, Ghaziabad, India
E–mail: *rupalibhardwaj*09@*gmail.com*

Dr. V.S.Dixit
Institution: Atmaram Sanatan Dharmshala, Delhi University,Delhi,India
Address: Department of CS, ARSD College, Delhi University, Delhi, India
E–mail: *veersaindixit*@*rediffmail.com*

Anil Upadhyay
Institution: Mata Rajkaur Institute of Engineering and Technology,
Mahrishi Dayanand University, Rohtak, Haryana, India
Address: Department of Applied Science, MRKIET, Rewari, India
E–mail: *anilupadhyay*2005@*gmail.com*