

LOOK-AHEAD APPROACH TO MINIMIZE CONGESTION IN DATA NETWORKS

* D.S.Mahendran ** S.Sakthivel

* Lecturer (SGr in computer science, Aditanar college, Tiruchendur, INDIA

** Prof & Head, Department of CSE & Engg, Alagappa University, Karaikudi, INDIA

Email: dsmah65@yahoo.com

Abstract

This paper shows a new approach to minimize congestion in data networks. The routing algorithm guarantees a loss-free delivery of data packets from congested sources, and a deterministic bound on the route length in arbitrary topology networks. This work shows that routing decisions using Local Greedy method are not optimal, and the performance of the algorithm can be improved substantially by using new look-ahead measures. The contribution of this paper is to propose a new metrics to find an optimal path to minimize congestion in networks. In the proposed method time taken to find an optimal path is less when compared to Local Greedy algorithm. The objective is to minimize the congestion. The performance is studied computationally for various networks with different number of nodes under static traffic model. In all the experiments the proposed method shows better results.

Keywords: Congestion, Routing, Look-Ahead, Virtual Ring, Proximity.

1. Introduction

An efficient routing result in smaller average packet delays, which means that the flow control algorithm can accept more traffic into the network. On the other hand, an efficient flow control algorithm rejects excessive offered load that would necessarily increase packet delays by saturating network resources. It is clear that routing and congestion control are very much interrelated. Earlier models of static and dynamic routing problems have been well studied by Gallager [5], Bertsekas [4], Segall [14], and others. These models mainly consider the minimum number of hops a packet can travel from source to destination to minimize delay; however congestion and packet loss due to congestion are not addressed.

In nondeterministic routing techniques such as hot-potato routing [3], deflection routing [8] and convergence routing [12], [15], [17] ensure no packet loss due to congestion inside the network. The nondeterministic routing combines, in a dynamic fashion, the on-line routing decision with the instant traffic load inside the network. The dynamic behavior of deflection routing has been studied on some regular topologies such as the Manhattan street network [6], [11], and the hypercube. Convergence routing [15]

ensures that packets will reach their destinations without being routed on the same link twice. Thus it ensures a deterministic bound on the maximum route length in an arbitrary topology network [2].

A. Deficiencies of Local-Greedy Algorithm

The performance of convergence routing with the Local-Greedy routing strategy is not necessarily the best one since it considers only the local traffic conditions. Local-Greedy routing decisions may also increase delay and congestion, since the routing priorities are based only on distance (hops) and path length can also be increased due to default routing.

B. Related Work

In deflection routing, the packets are deflected from the shortest path to a random location in the network, while the convergence routing on the other hand, is guided by a global sense of direction and deflects the message only if locally seems like there is a “global improvement”. Global improvement and implicit self-routing is given by the method of “interval routing” of Santoro and Khativ[13].

In MetaNet routing [17] algorithm the packets will reach their destinations unless physical failure has occurred. This property is not provided by deflection routing, which means that the packets can deflect indefinitely inside the network. Therefore, in Baran’s “Hot-Potato”[3] routing there is hop-count field in each packet header, which is decremented by one after every hop. If the hop-count reaches zero the packet is discarded i.e. the packet may get lost due to congestion inside the network.

In “METANET Principles of an Arbitrary Topology LAN” [2], they assumed that the physical layout of the network is a tree, and a ring is embedded into an arbitrary topology network [16] using Euler Tree Traversal without the thread links. All the links of the tree are ring links and are part of tree embedded ring (TER).

In [1], a successful approach to deal with network delays, although it requires the computation of a large number of control parameters. The number of parameters is proportional to the round trip delays of the system. Reference [9] acknowledges the need to use a separate set of parameters for each delay’s values and the number of sources in the system. In [7], a congestion control system is developed to motivate the handling of feedback delays.

In “Virtual Node Algorithm for Data Networks” [10] they assumed that the physical layout of the network is a graph, and a ring is embedded using the above algorithm with thread links. The traversal ring on a graph is called graph embedded ring (GER).

C. Proposed Algorithm

In this paper the performance of Local-Greedy routing decisions can be improved substantially with a different distance measure that provides a look-ahead of the potential routes. The performance measure considered in this work is to find an optimal path, so that it minimizes the congestion in a network. Given a traffic load, optimization of a nondeterministic routing algorithm requires new techniques since actual

routes cannot be fixed, but altered, based on routing priorities and the actual traffic conditions. The stability and performance improvements are studied on various networks with different number of nodes and are discussed in section V.

The rest of this paper is organized as follows. Section II describes the network model. In section III we present the convergence routing model. Section IV shows the proposed algorithm. Performance of the proposed algorithm is discussed in section V. Conclusion of the algorithm is presented in section VI.

II. Network Model

A computer network is modeled as an undirected simple graph

$G = (N, E)$, where N is the set of nodes and E is the set of edges or links connecting the nodes. Each node has its own unique ID, denoted by a capital letter A, B, C, D, E... etc as in Fig. 1.

Fig .1 Given Network Topology

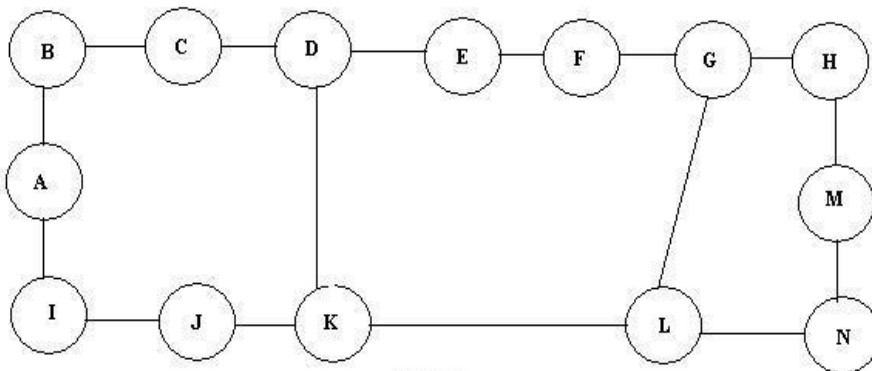


Figure 1

A. Virtual Ring Embedding

A virtual ring is embedded in a network by the Virtual Node Algorithm [10]. Such a virtual ring is called the graph embedded ring or GER, and the links are called ring links. The virtual ring links are numbered sequentially from 0 to $m-1$. The number associated with each ring link constitutes a virtual node (VN). Thus, m is the number of virtual nodes induced by the ring embedding (for example, in fig 2. $m=26$).

Fig .2 An Network Topology With Virtual Rings

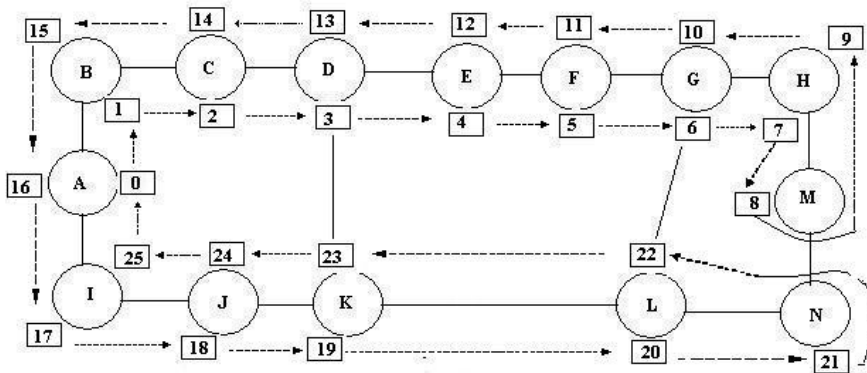


Figure 2

B. Forward Nodes

Let i, j and k be the virtual nodes. Then, j is called a forward node and (i,j) a forward link of i for destination k if and only if the following two conditions are true.

- (i) Distance (in hops) from j to k must be less than i to k .
- (ii) There exists a physical link (I,J) in the network, such that i is a virtual node of I and j is a virtual node of J . This can be either a ring or thread link.

For example, in Fig. 2, the forward nodes of 2 for destination 17 are 3, and 15. The corresponding forward links are $(2,3)$ and $(2,15)$.

Let the sets $N(i, k)$ and $L(i, k)$ denote the forward nodes and links of virtual node i for destination k , respectively. A forward path from i to k is a chain of forward links ie. a path of the form (v_1, v_2, \dots, v_n) where $v_1=i, v_n=k$, and $(v_l, v_{l+1}) \in L(v_l, k)$, where $l=1, \dots, n-1$.

Example: A forward path from 2 to 17 (in Fig. 2) is $(2,15,16,17)$ or $(2,3,14,15,16,17)$.

C. DIRECTED ACYCLIC GRAPH (DAG)

Fig.3 DIRECTED ACYCLIC GRAPH FOR VIRTUAL NODE 6 TO 21

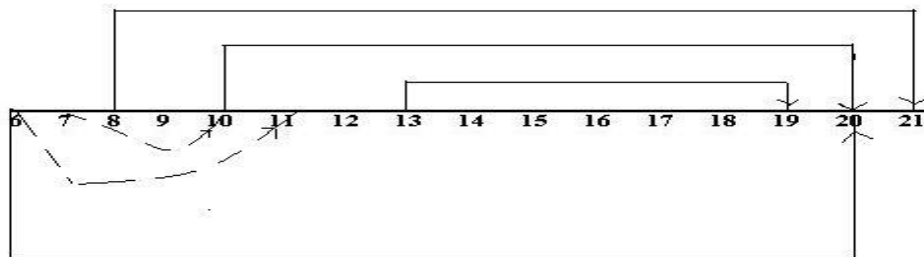


Figure 3

The virtual address is assigned to the nodes according to the virtual ring embedding results in a linear ordering of the nodes. The linear ordering of nodes is used for global sense of direction. DAG-based representation enables us to model and to formulate convergence routing precisely. The directed acyclic graph (DAG) for source i and destination k , denoted by DAG_i^k to be the union of all the forward paths from i to k .

i) **Ring links** is displayed as bold arrows. These links form a straight line from VN_6 to VN_{21} and show clearly the global direction imposed by the virtual ring embedding.

(ii) **Shortcut links** is displayed as dotted arrows. A shortcut link (i,j) indicates that a shortcut routing operation is possible from virtual node i to j .

Example: $(6,11)$ $(5,12)$ indicates the possible short cut from 6 to 11 via 10 and 5 to 12 via 11 respectively.

(iii) **Jump links** is displayed as normal arrows. A jump link indicates a possible jump from virtual node i to virtual node j .

Example: $(8,21)$ indicates a possible jump from 8 to 21.

III. CONVERGENCE ROUTING MODEL

In this section a new analytical model for the behavior of convergence routing at a node (switch) has been introduced. This model will enable us to determine the routing probabilities at each node for a given destination. At each node, the routing probabilities are computed according to an ordering of the forward links of this node for the destination. Precisely, let $|N(i, k)|=n_i^k$ be the number of forward nodes from node i to destination k . These nodes are ordered in its routing table according to their “closeness” to destination k . Note that such an ordering provides a priority assignment over the forward links, such that if node $j \in N(i, k)$ is in the y^{th} position, then the priority of the link (i,j) for destination k by $\pi_{i,j}^k$.

A. COMPUTING ROUTING PROBABILITIES

Given a priority assignment and utilization values on the links, the computation method for the routing probabilities is discussed here. Note that convergence routing algorithm will switch a packet designated for k to the link (i,j) with priority y if all of the links with higher priority are busy and this link is available. Precisely, the probability $P_{i,j}^k$ that edge (i,j) is selected for destination k with priority y , is

$$P_{ij}^k = (1 - \rho_{ij}) \prod_{x: \pi_{i,x}^k > y} \rho_{ix} + \left(\prod_{x=1}^{n_i^k} \rho_{ix} \right) \frac{1 - \rho_{ij}}{\sum_{x=1}^{n_i^k} (1 - \rho_{ix})} \dots\dots\dots(1)$$

Note that according to the probability law, the sum of routing probabilities for a destination at an intermediate node/switch must add up to unity (ie. $\sum_{ij} P_{ij}^k=1$ at node i for destination k).

B. LOOK–AHEAD DISTANCE MEASURES

The look-ahead measures define a new notion of distance with a few properties. A look-ahead measure should favor or “credit” the paths if:

- (i) It is a shortest path to destination.
- (ii) The utilization of the link is less than the other forward links.

C. Minimum Proximity (MiP)

The minimum proximity (MiP) of node i to node k is defined as follows.

$$\text{MiP}_i^k = \min_{j \in N(i, k)} \text{MiP}_j^k (1 - \rho_{ij}) \quad \dots(2)$$

The Minimum Proximity (MiP) routing algorithm can be defined as the convergence routing algorithm that assigns the priorities of the forward links of node i by sorting the nodes in $N(i,k)$ in ascending order of their minimum proximities to destination k. Using this new definition, the computation of MiP can be done very efficiently. For each commodity k, we may find the corresponding DAG, and we may start computing minimum proximities from right to left, using (2).

IV. PROPOSED ALGORITHM

1. Input – The given network.
2. Insert virtual nodes using virtual node algorithm.
3. Find the forward nodes.
4. Compute the distance between each node in hops.
5. Assign initial utilization value for each link in the network.
6. Assign initial load / traffic to each link in the network.
7. Compute the minimum proximity (MiP) value.
8. Assign priority to each link.
9. Set up the routing table at each node.
10. Compute the routing probability for each link.
11. Compute the flow on each link using the equation.

$$f_{ij}^k = \begin{cases} = 0 & \text{if } (i, j) \text{ is a short cut link} \\ t_{ik} P_{ij}^k + \sum_{u \in U_i} f_{ui}^k P_{ij}^k + \sum_{u \in w_j} \sum_{u' \in U_w} f_{u'w}^k P_{wj}^k & \text{other wise .} \end{cases} \quad \dots (3)$$

12. Compute the new utilization value of the link using (4)

$$\rho_{ij} = \frac{\sum_k f_{ij}^k}{Z} \quad \text{----- (4)}$$

13. Maximum number of iterations performed or convergence, terminate the process otherwise go to step 7.
14. Display the congestion value.

A. STABILIZATION ISSUES

The ρ_{ij} values computed at iteration n of the algorithm are used to compute the routing probabilities in iteration $(n + 1)$. Thus, the utilization values may cause changing the priority assignments on the links and routing probabilities. In turn, routing probabilities may change the link utilization. The oscillation can be avoided, depending on the way the link utilization values are updated. Using a bias factor that considers the average of the current unbiased and all past biased utilization values as in equation (5), and guarantees a smoother change of the utilization at each iteration step.

$$\rho_{ij}(t) = \frac{1}{t} \left[\rho'_{ij}(t) + \sum_{l=1}^{t-1} \rho_{ij}(l) \right] \quad \dots (5)$$

V. PERFORMANCE ANALYSIS OF THE ALGORITHM

A. CONGESTION ANALYSIS

The performance measure used in our experiments is based on minimization of congestion to increase throughput. The congestion value obtained in various networks is given in table 1. In all the experiments the proposed algorithm performs better than the local-greedy algorithm. Fig4 shows the value of congestion obtained for LG and proposed algorithm for various network topologies. Fig5 shows the congestion value computed at each iteration for the various network topologies are given in Fig 1, the results are tabulated in table 2. Comparison between Minimum Proximity algorithm and Local-Greedy algorithm suggests that the Minimum Proximity MIP algorithm gives lower congestion value than Local-Greedy algorithm. Fig 5 shows that the proposed algorithm converges quickly than LG algorithm. Experiments indicate that performance improvement depends on 1) network topology, and 2) the network size.

**Table 1 Congestion in various networks
for LG and MIP algorithm**

NODES	CONGESTION	
	LG ALG	MIP ALG
8	9.153	7.961
10	13.338	9.145
12	17.431	12.719
14	22.271	17.66
16	26.802	21.38
18	28.237	27.485
20	35.442	29.799
22	40.734	27.257
24	27.775	26.311
26	40.554	32.141
28	48.569	39.65
30	47.061	40.75
34	61.261	53.155

Fig 4. Comparison of Congestion Value for LG and MIP Algorithms

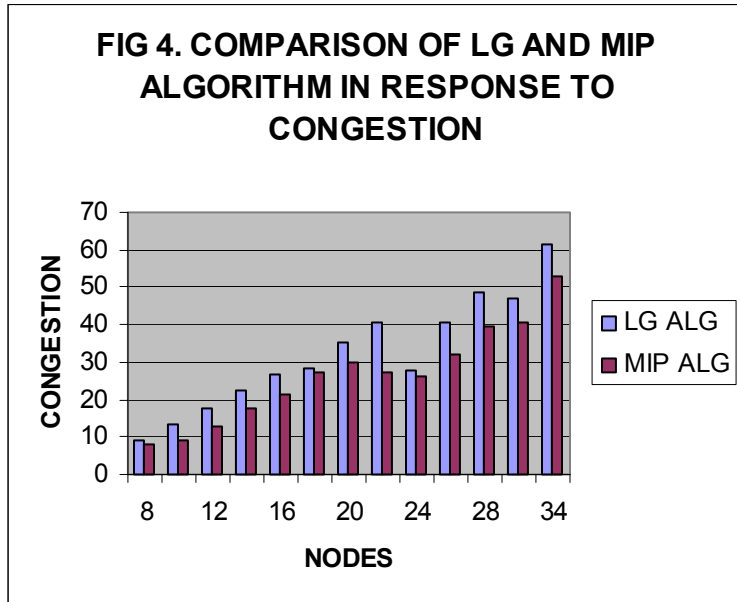
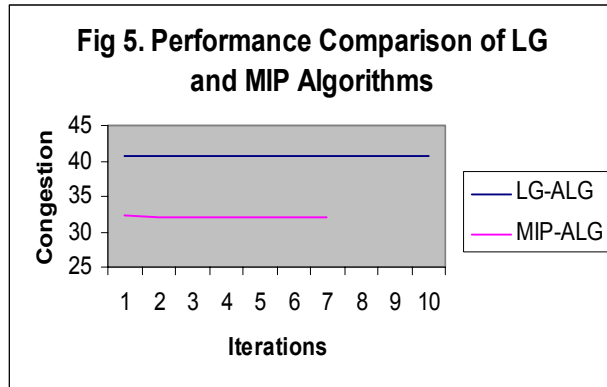


Table 2 Convergence rate for LG and MIP algorithm

ITERATIONS	CONGESTION	
	LG-Alg	MIP-Alg
1	40.596	32.168
2	40.554	32.141
3	40.554	32.141
4	40.554	32.141
5	40.554	32.141
6	40.554	32.141
7	40.554	32.141
8	40.554	
9	40.554	
10	40.554	

Fig 5. Comparison of Convergence rate for LG and MIP Algorithms for Network in Fig 1.



VI. CONCLUSION

The Local-Greedy routing algorithm has certain deficiencies due to its limited use of information about the traffic load conditions across the network during the routing process. As a remedy, a new look-ahead measure is proposed. The look-ahead measure is embedded into the proposed algorithm that simulates the behavior of the Local-Greedy and proposed routing algorithms on arbitrary networks. A series of experiments were performed for various network topologies and in all the experiments the proposed algorithm gives minimum congestion value. It is also observed that the proposed algorithm converged quickly than Local-Greedy algorithm.

VII. REFERENCES

- [1] S.P.Abraham and A.Kumar, "New approach for asynchronous distributed rate control of elastic session in integrated packet networks", IEEE Trans. Networking, vol 9, pp. 15-30, Feb 2001.
- [2] L.Bao and J.J.Garcia-Luna-Aceves, "Topology Management in Adhoc Networks". Fourth ACM international symposium on mobile adhoc networking and computing, June 13,2003.
- [3] P.Baran, "On distributed Communication Networks", IEEE Trans Commn Syst pp1-9,Mar 1964.
- [4] D.P. Bertsekas – "Dynamic models of shortest routing algorithms for communication networks with multiple destinations" IEEE Trans Automat Contr, Feb 1982 AC-27 pp. 60-74.

- [5] D.P.Bertsekas and R.G Gallager , Data Networks. PHI 1992.
- [6] A.G. Greenberg and J. Goodman, “Sharp approximate models of adaptive routing in mesh networks”, Teletraffic Annual Comput Performance, pp. 255-270, 1986.
- [7] DirceuCavendish, Mario Gerla, and Saverio Mascolo, “A control theoretical approach to congestion control in packet networks”, IEEE/ACM Trans Networking, vol 12 , no 5, pp. 893-906, Oct 2004.
- [8] A.G. Greenberg and B. Hajek, “Deflection routing in hypercube networks”, IEEE Trans Common Vol. 40, No. 6, pp. 1070-1081, June 1992.
- [9] D.Katabi, M.Handley, and C.Rohrs, “Congestion control for high bandwidth delay product networks”, in Proc ACM SIGCOMM, pp 89-102, Aug 2002
- [10] D.S.Mahendran and S.Sakthivel, “Virtual Node Algorithm for Data Networks”, International Journal of Soft Computing, (2) 2: 302-306, Jan 2007.
- [11] N.F.Maxemchuck, “Routing in the manhalthan Succd”, IEEE Trans Commn –vol AC27 , pp 60-74, Feb 1982.`
- [12] Y. Ofek and M.Yung, “The integrated Metanet architecture: A switch based multimedia LAN for parallel computing and real-time traffic”, presented at IEEE, INFOCOM '94, 1994.

- [13] N.Santoro and R.Khativ , “ Labeling and implicit routing in networks”, Compt Journal vol 28, pp 259-273, 1985
- [14] A.Segall, “The modeling of adaptive routing in data communication networks”, IEEE Trans commun, vol.COM-25,pp 85-95, Jan 1977.
- [15] B. Yener Y. Ofek and M. Yung, “Design and performance of convergence routing on spanning trees”, IEEE GLOBCOM – 94, pp.169-175, 1994.
- [16] B.Yener and Y.Ofek, “Topological design of loss-free switch – based LAN’s” in Proc IEEE, INFOCOM '94, pp. 802-811. 1994.
- [17] Yoram Ofek and Modi Yung, “Meta Net Principles of an Arbitrary Topology LAN” IEEE Trans Networking Vol. 3 No. 2, pp. 169 – 180, Apr. 1995.



D.S.Mahendran received the M.Sc physics and PGDCA degrees from Madurai Kamaraj University and M.Phil degree in Computer Science from Alagappa University, Karaikui. He is working toward the Ph.d degree in computer science at Alagappa University. His area of interests includes QOS guarantees in wired networks and optimization.



S.Sakthivel received the Ph.d degree from Anna University, Chennai, India. He is currently working as a Professor Head, in the Department of CSE& Engg, Alagappa University, Karaikudi, India. He has thirty years of teaching and research experience.

His research interest includes Computer algorithms, Computer Networks and Image Processing.