Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 4, Issue, 10, pp.203-206, October, 2012

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

LOOK-AHEAD APPROACH TO MINIMIZE CONGESTION IN DATA NETWORKS

Okafor Friday Onyema and Fagbounmi Griffin Siji

[1]Department Of Computer Science Michael Okpara University Of Agriculture, Umudike
[2]Department Of Computer Science Abia State Of Polytechnic, Aba

ARTICLE INFO

Article History:
Received 18th July, 2012
Received in revised form
14thAugust, 2012
Accepted 17th September, 2012
Published online 30th October, 2012

Key words:
Congestion, Routing, Look-Ahead, Virtual Ring, Proximity.

ABSTRACT

This paper presents 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.

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 [Bertsekas, (1992)], [Segall, 1977]. 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 [Baran,1964)], deflection routing [Greenberg, 1992] and convergence routing [Ofek, 1994] 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 behaviour of deflection routing has been studied on some regular topologies such as the Manhattan street network [Greenberg,1986] and the hypercube. Convergence routing [Yener 1994] 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 [Bao, 2003].

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.

***Corresponding author:** revmachi_4@yahoo.co.uk

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 it seems like there is a "global improvement". Global improvement and implicit self-routing is given by the method of "interval routing" [Santoro, 1995] In MetaNet routing algorithm [Yoram, 1995)] 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" routing [Baran, 1964] 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" [Bao, 2003], they assumed that the physical layout of the network is a tree, and a ring is embedded into an arbitrary topology network 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). The work of [Abraham, 2001)] shows 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. [Katabi, 2002)] acknowledges the need to use a separate set of parameters for each delay's values and the number of sources in the system. In [Mario, 2004] a congestion control system is developed to motivate the handling of feedback delays; while in "Virtual Node Algorithm for Data Networks" [Mahendran, 2007] it is 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).

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 below.

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.

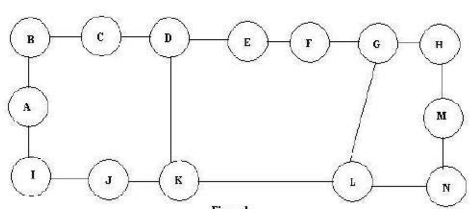

Fig .1 Given Network Topology

Virtual Ring Embedding

A virtual ring is embedded in a network by the Virtual Node Algorithm. 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).

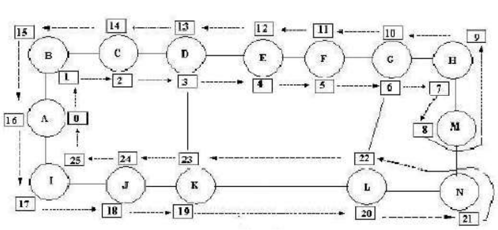

Fig . 2. Network Topology With Virtual Rings

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 $(v_1,v_2,\ldots.v_n)$ where v_1=i, v_n=k, and $(v_1,v_{l-1}) \in L(v_1,k)$, where l=1,…..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).

Directed Acyclic Graph (DAG)

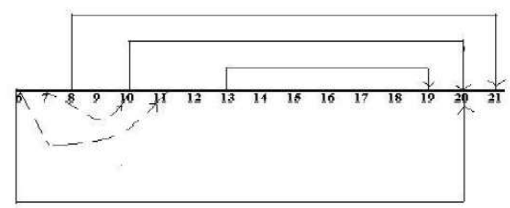

Fig.3 Directed Acyclic Graph For Virtual Node 6 To 21

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. Typical scenarios to illustrate are as follow:

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.

Convergence Routing Model

In this section a new analytical model for the behaviour 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 $\pi_{i,j}^k$.

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 that edge (i,j) is selected for destination k with priority y, is given as:

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

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. $\Sigma_{ij} P_{ij}^k = 1$ at node i for destination k).

Look–Ahead Distance Measures

The look-ahead measures define a new notion of distance with a few properties. A look-ahead measure should favour 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.

Minimum Proximity (MIP)

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

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

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 the minimum proximity algorithm.

Proposed Algorithm

- Input – The given network.
- Insert virtual nodes using virtual node algorithm.
- Find the forward nodes.
- Compute the distance between each node in hops.
- Assign initial utilization value for each link in the network.
- Assign initial load / traffic to each link in the network.
- Compute the minimum proximity (MIP) value.
- Assign priority to each link.
- Set up the routing table at each node.
- Compute the routing probability for each link.
- Compute the flow on each link using the equation

$$f_{ij}^k = \begin{cases} -0 \\ t_{ik} P_{ij}^k + \sum_{u \in U_i} f_{ui}^k P_{ij}^k + \sum_{u \in wi} \sum_{u' \in U_w} f_{u'w} P_{wj}^k \end{cases} \quad \text{………(3)}$$

- Compute the new utilization value of the link using the following equation.

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

Maximum number of iterations performed or convergence, terminate the process otherwise go to step 7.

- Display the congestion value.

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.

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

Performance Analysis of the Algorithm

A. Congestion Analysis / Results

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. Fig 4 shows the value of congestion obtained for LG and proposed algorithm for various network topologies. Fig 5 shows the congestion values computed at each iteration for the various network topologies 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

	CONGESTION	
NODES	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

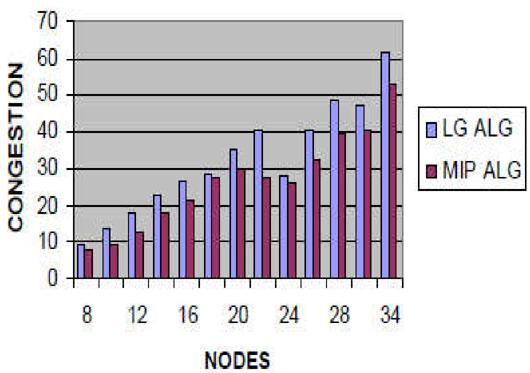

Fig 4. Comparison Of LG And MIP Algorithm In Response To Congestion

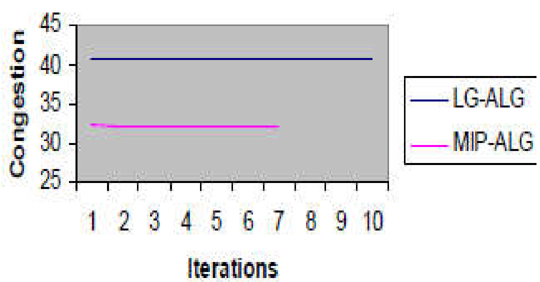

Fig 5 Performance Algorithm for LG and MIP Algorithm

Table 2. Iteration Results in Various Networks for LG & MIP Algorithms

	CONGESTION	
ITERATIONS	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	

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.

REFERENCES

Abraham S. P. and Kumar A., (2001:15-30) "*New Approach For Asynchronous Distributed Rate Control Of Elastic Session In Integrated Packet Networks*", Vol. 9(1), IEEE Trans. Networking, Califonia, USA.

Bao L. and Garcia-Luna L.L., (2003) "*Topology Management In Adhoc Networks",* Fourth ACM international symposium on mobile adhoc networking and computing,

Bao L. and Garcia-Luna L.L., (2003) "*Topology Management In Adhoc Networks",* Fourth ACM international symposium on mobile adhoc networking and computing,

Greenberg A. G and Goodman J., (1986:255-270) "*Sharp Approximate Models Of Adaptive Routing In Mesh Networks*", Teletraffic Annual Comput Performance,.

Greenberg A. G and Hajek B, (1992:1070-1081) "*Deflection Routing In Hypercube Networks"*, Vol. 40(6), IEEE Trans Common.

Katabi D et al (2002:89-102) "*Congestion Control For High Bandwidth Delay Product Networks*", Proceedings of ACM SIGCOMM.

Mahendran D. S and Sakthivel S, (2007:302-306) "*Virtual Node Algorithm For Data Networks*", Vol. 2(1), International Journal of Soft Computing.

Ofek Y. and Yung M., (1994) "*The Integrated Metanet Architecture: A Switch Based Multimedia LAN For Parallel Computing And Real-Time Traffic*", IEEE, INFOCOM.

Segall A (1977:85-95), "*The Modeling Of Adaptive Routing In Data Communication Networks*", Vol. 25(1), IEEE Trans commun.

Yener Y B et al (1994:169-175) "*Design And Performance Of Convergence Routing On Spanning Trees*", Vol. 94(1), IEEE GLOBCOM.

****