ITMR: An Experiment to ACO based Multipath Routing

Shalini Aggarwal, Pardeep Kumar, Shuchita Upadhyaya

***Abstract*: As the concepts of big data and cloud computing are growing day-by-day, the need of multipath routing is also getting more and more look into. Without using multipath routing the handling of big data is almost impossible. This paper is an attempt in the direction of finding and evaluating multiple paths in a network. This paper compares the similarities and differences between the ACO based approaches to routing viz., ITMR, AntNet and ABC algorithms. The ITMR has been evaluated for a network and the results are found as per expectation i.e. the algorithm is able to find the multiple paths for data transmission depending upon link utilization.**

Index Terms*: *ITMR, Link utilization, ACO, AntNet, ABC.

I. INTRODUCTION

Multipath routing is a technique in which multiple paths may be used in parallel to transmit the data via a network. It provides number of benefits viz., more bandwidth, fault resistant, better security, etc. There are three main components needed for the implementation of multipath routing viz., Multipath Calculation Algorithm, Multipath Forwarding, and Multipath End–Host protocol.

This paper considers the first component of the multipath routing implementation i.e. A multipath algorithm, which explores multiple paths between source and destination nodes depending on pertinent metric/s. Metrics can be throughput, delay, reliability, bandwidth, distance, security, cost, etc.

II. MULTIPATH CALCULATION ALGORITHM

A multipath calculation algorithm has number of characteristics. The major two characteristics are quality and quantity of paths.

The quality of a path does not always mean the best path; rather it depends on path specification. A path which fulfills the requirements of the flow specification will be termed as a better quality path. For example, if the flow specification is either low cost or higher bandwidth or higher throughput, then the multipath algorithm should be able to find multiple paths with least cost or available higher bandwidth or higher throughput, whichever is desirable. Multiple paths which are identified should have other attributes such as path quantity and path independence.

Path quantity means how many paths are generated by a multipath calculation algorithm. Although more paths means more overhead, yet the number of paths found by the algorithm should be considerable as one can use some selection criteria for selecting a limited number of paths depending upon the requirements. However, main focus of the algorithm should be on recognizing quality paths.

III. ACO BASED ROUTING

Traditional approaches to multipath routing yields good results, but in some cases it may be impractical to find and consider all possible paths for traffic transportation due to large overheads, and therefore, only a limited number of good paths need to be considered. In such cases nature inspired computational intelligence techniques provide optimistic solutions as the problem of finding and utilizing the multipath may become NP-hard.

Ant Colony Optimization (ACO) is a very natural approach to determine the routes to the destination as the ants are also doing the same thing as per the requirement of network routing. Therefore in this paper ACO [2] has been contemplated as one of the basis for identifying the routes from a source to a destination.

A multipath routing strategy with adaptation of ACO approach has been formulated in [3]. Proficient aspects of both ABC[4] and AntNet[2] variations of ACO [5] have been realized in the design of the strategy. Table I addresses various features pertinent to ACO based routing and their implementation strategy in ABC and AntNet. The third column derives the formulation of the ITMR algorithm [1] indicating a comparison as well as similarity with ABC and AntNet.

Revised Manuscript Received.

Dr. Shalini Aggarwal, Department of Computer Science, S.U.S. Govt. College, Matak Majri(Indri), Karnal(Haryana), India.

Dr. Pardeep Kumar, Department of Computer Science & Applications, Kurukshetra University, Kurukshetra(Haryana), India.

Dr. Shuchita Upadhyaya, Department of Computer Science & Applications, Kurukshetra University, Kurukshetra(Haryana), India.

Table – I : Comparison of various ACO based approaches

Feature	**ABC Routing**	**AntNet Routing**	**ITMR**
Application	**Routing in circuit switched networks**	**Routing in packet-switched networks**	**Routing in packet-switched networks**
Ant Class	***Forward Ant* – Perform update as they travel through the network**	***Forward Ant (FA)* – Act as investigators** ***Backward Ant (BA)* – Update routing tables and other data structures**	***Forward Ant* – Act as investigator** ***Backward Ant* – Update routing table**
Structure of Ant	***Forward Ant:*** • **Source Address** • **Destination Address** • **Age (time of launch)**	***Forward Ant*:** • **Source Address** • **Destination Address** • **Sequence number** • **Identity – FA or BA** • **Stack (memory) : Identity of Nodes traversed and associated trip times** ***Backward Ant*:** **Inherits all fields of forward ant**	***Forward Ant*:** • **Source Address** • **Destination Address** • **Sequence number** • **Stack (memory) : Identity of nodes traversed and associated link utilization values of outgoing link (link on which ant is to be forwarded)** ***Backward Ant*:** • **Source Address** • **Destination Address** • **Sequence number** • **Path identification (all nodes falling on the current path as identified by destination)**
Pheromone representation	**Using probabilities initialized to 0.5 and updated according to age (age is calculated by the total steps that have been executed from the initialization of the ant)**	**Using probabilities initialized to 0.5 and updated according to a function (reinforcement) determined in terms of traffic statistics and previous probability value**	**Using link utilization values which are updated on the basis of current queued load, bandwidth and instant link utilization.**
Ant propagation basis	**According to probabilities in pheromone table**	***Forward Ant*: according to probabilities in pheromone table** ***Backward ant*; according to memory maintained by forward ant**	***Forward Ant*: according to link utilization values** ***Backward Ant*: According to routing decision taken at destination**

Feature	ABC Routing	AntNet Routing	ITMR
Pheromone updation	$p = (p\_old + \Delta p) / (1 + \Delta p)$ p = new probability Δp= probability increase *p_old* = old probability $\Delta p = ((d / age) + c)$ where: *age* = total steps that have been executed from the initialization of the ant *c* and *d* are constants.	Increase probability of channel that backward ant comes from, using: $P(k, n, d) = P(k, n, d) + r(1 - P(k,n,d))$ Where: P is the probability of k^{th} node to reach destination *d* through neighboring node *n*. $0 < r \leq 1$ *r* (reinforcement) is a function of time experienced by FA, average time for same destination memorized in trips table and standard deviation for same destination memorized in trips table. Decrease probability of other channels using: $P(k, i, d) = P(k, i, d) + r(P(k,i,d))$	$u_{new} = a * u_{old} + (1-a)*f$ u : Link Utilization factor (0 <= u <= 1) a : indicates how fast a node forgets current history a = capacity / load f : 0 or 1 (instantaneous Link Utilization)
Updation speed	Updation is faster and reliable, No delay	Updations are comparatively slow as they are performed by backward ants	Updations are faster and only required for routing table which are performed by backward ant
Location of path identification	Identified at destination	Identified at source after backward ant returns	Identified at destination and informed to source node by backward ant
Pheromone evaporation	By the concept of aging. Aging used to limit amount of pheromone deposited by each ant which steers evaporation.	By the concept of aging	Evaporation controlled by regulating link utilization
Data structures	• Pheromone table • Routing table	• Pheromone table • Routing table • Link queues • Statistical parametric model	• Pheromone table (Link utilization table) • Routing table
Pheromone Threshold	No Limit	No Limit	Limit value of pheromone more than a specified limit controls the movement of forward ants
No of ants used	No backward ants used	No of backward ants are equal to forward ants Number of forward ants varies in different propositions.	Limited number of backward ants; equal to number of paths assessed at destination for suitability.

IV. ITMR ALGORITHM

This section presents the ITMR algorithm as proposed by the same authors in [1].

ITMR Algorithm: Steps

i. At regular intervals, each node *i* generates forward ants destined for various destination nodes in the network.

ii. $u_{i,j}$ is initialized to *0* for all nodes ($u_{i,j}$ is the link utilization value for all links ***i-j*** where ***i*** is the current node and ***j*** is its neighbor).

iii. Repeat steps iv to x for each source-destination pair ***(s-d)***

iv. Update $u_{s,j}$ for all links ***s-j*** ,where ***j*** is a neighbor of ***s***.

v. Source generates and forwards Pert ants on all neighboring links for which $u_{s,j} < u_{thres}$.

vi. Repeat steps vii to ix for every node ***i*** reached by the ant till destination is reached excluding the link on which incoming ant arrived (can be identified from the last appended node *id* on forward ant).

vii. If identity of current node ***i*** is appended in forward ant received, node kills the ant (for avoiding loops).

viii. Node ***i*** updates $u_{i,j}$ for all links ***i-j*** where ***j*** is a neighbor of ***i***.

ix. If $u_{i,j} < u_{thres}$, node ***i*** appends its identity on the forward Pert ant along with the $u_{i,j}$ value and forwards the ant on link ***i-j***.

> [*All ants that reach destination with in a particular time identify the feasible multiple paths (following bandwidth and delay constraints) between source-destination pair. The destination applies a Fuzzy logic procedure to select and rank apposite paths*]

x. At the destination, selected forward ants will be converted to backward ants by inheriting the source-destination identities, sequence number and identities of all appended nodes.

V. EXAMPLE NETWORK

The algorithm ITMR was executed for a network having 10 nodes and 15 edges as shown in figure 1. Assuming capacity of each link = 10, Source node = 1, and Destination node = 10. The example network has been tested by applying low, random and high loads on the links for finding all possible paths. A variation in threshold value of link utilization in all the three cases has been considered for this network to investigate the impact of link utilization on the number of paths identified.

CASE I: Low Load

Each edge is labeled with assumed input load. Further load on each link is opined to be smaller.

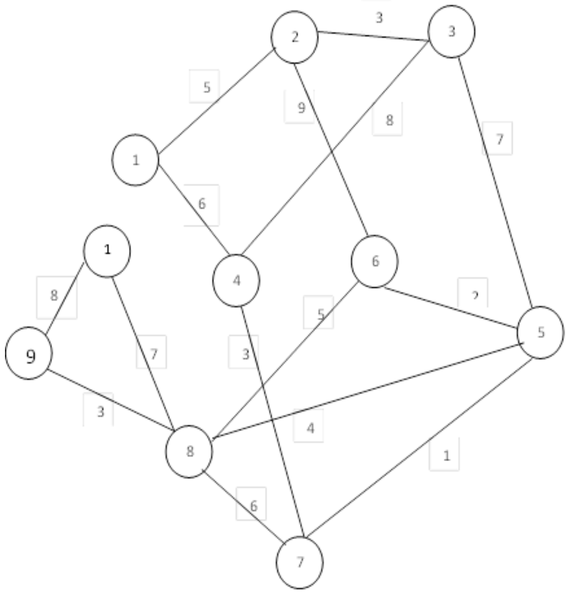

Fig. 1: Example network with Low load on links

The outcome of the execution of the algorithm in MATLAB under low load conditions found all possible forty loopless paths (verified by executing DFS on the network) consequentially.

When the algorithm was implemented using varying link utilization threshold value i.e. u_{thres}, it was observed that the paths found were same i.e. 40 for every u_{thres} varying from 0 to 1. The results of the experiment conducted on the network in fig. 1 are being shown in fig. 2.

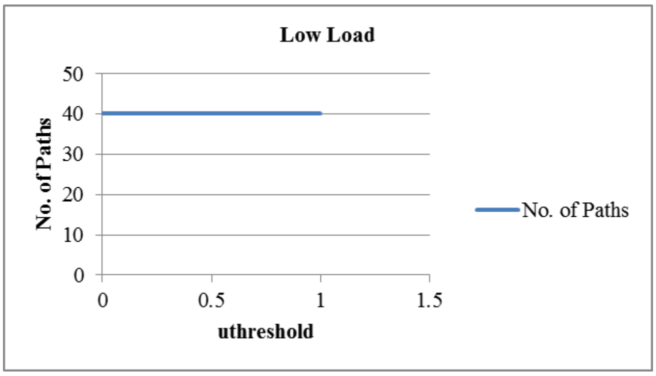

Fig. 2: No. of Paths for varying u_{thres} (Low Load)

CASE II: Random Load

Each edge in fig. 3 is labeled with assumed input load. Further load on each link is opined to be random i.e. smaller than capacity at some links and greater than capacity at some links.

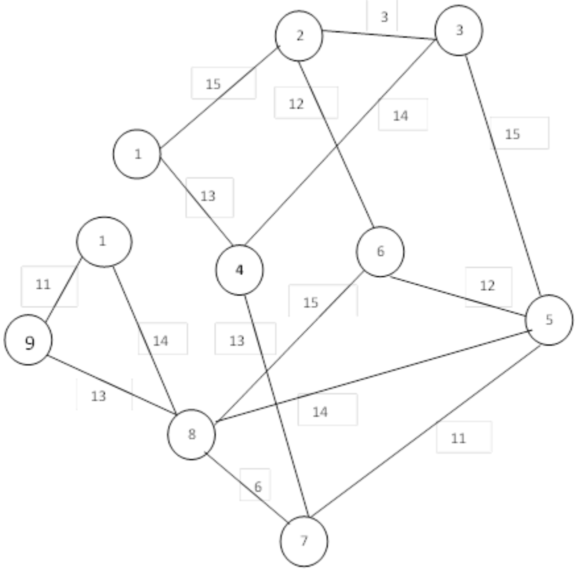

Fig. 3: Example network with Random load on links

The number of paths found in each case has been tabulated in table 2 and shown in fig. 4.

Table 2 - No. of Paths for varying u_{thres}(Random Load)

u_{thres}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1
No. of Paths	0	0	0	2	2	4	5	7	10	14	19	40

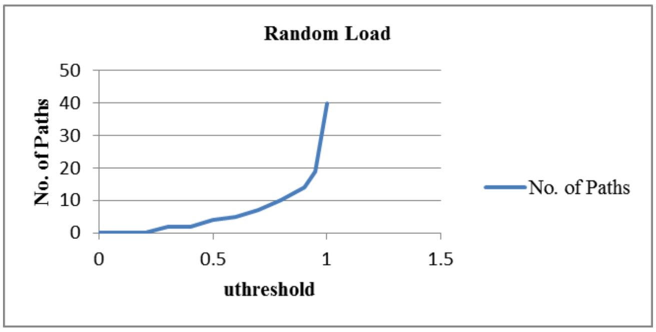

Fig. 4: Number of Paths for varying u_{thres} (Random Load)

CASE III: High Load

Each edge in figure 5 labeled with assumed input load. Further load on each link is opined to be high.

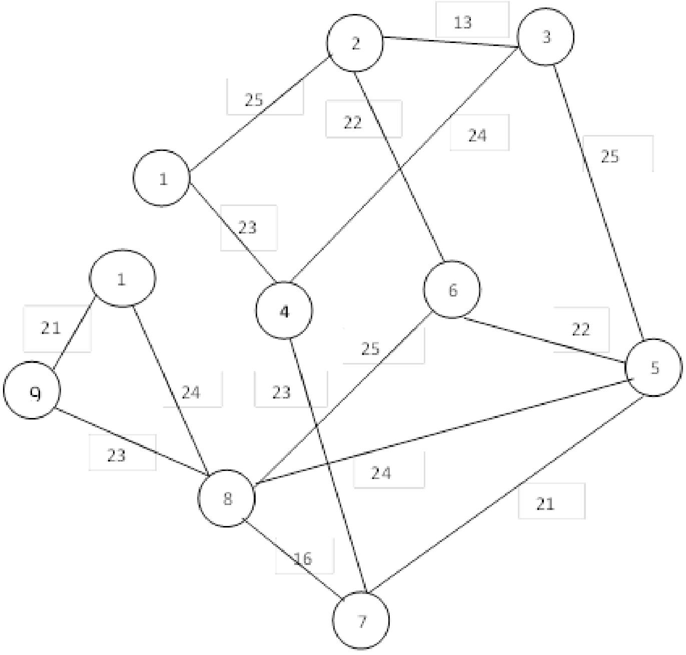

Fig. 5: Example network with high load on links

The number of paths found in each case has been tabulated in table 3 and has been shown graphically in fig. 6.

Table 3: No. of Paths for varying u_{thres}(High Load)

u_{thres}	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1
No. of Paths	0	0	0	0	0	0	2	2	2	4	6	40

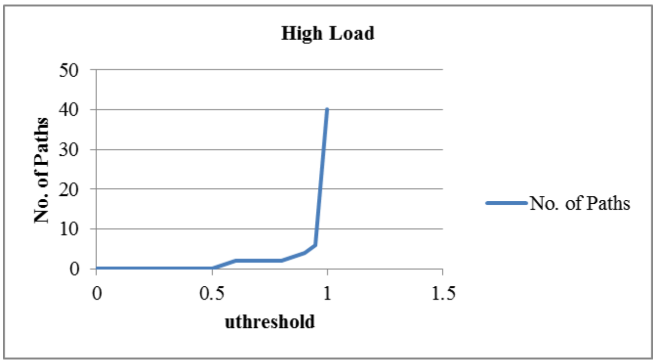

Fig. 6: No. of Paths for varying u_{thres} (High Load)

The comparison of effect of u_{thres} values on various types of loads i.e. low, random and high is depicted in the fig.7.

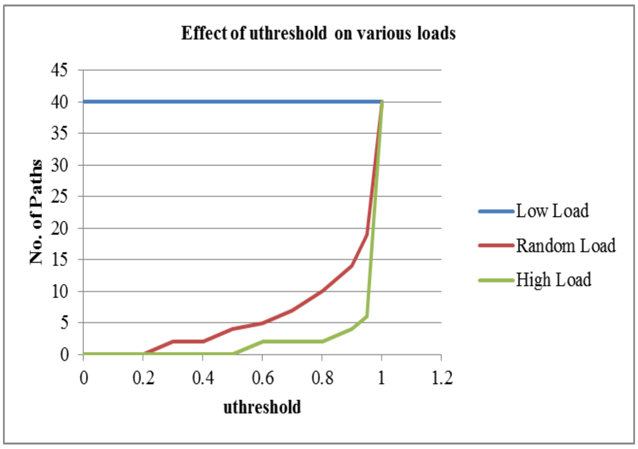

Fig. 7: Effect of u_{thres} with variations of loads

VI. INTERPRETATION OF THE RESULTS

It is interpreted from the example networks that when the lesser load is applied which is approximately same as the capacity, all or most of the possible paths have been found. In the opposite case i.e. load very high than capacity on all or most of the links; the identified paths are quite less. The paths found in case of high loads are the paths which are having comparatively lesser utilization. In case when random load is applied on the network, moderate number of paths is identified.

The paths having lesser utilization are used for the upcoming traffic. This is in accordance with the anticipation of results.

When u_{thres} (link utilization threshold) was varied on the various loads in example network, it was observed that in case of low load the effect of u_{thres} was nil as all the paths were found even when u_{thres} was very less. But in case of random and high load, higher value of u_{thres} provided more number of paths, while in case of lower value of u_{thres} either no paths were found or very less number of paths were found. These results are as per the anticipation.

VII. CONCLUSION

Finally it can be concluded that the ITMR is able to find quality multiple paths. The number of quality paths can be adjusted by setting the value of link utilization threshold. The link utilization threshold is directly proportional to the number of paths found. If link utilization threshold is less the number of paths identified is also less and so on. The ITMR can prove to be very useful for finding multiple paths in network routing.

REFERENCES

1. Shalini Aggarwal and Shuchita Upadhyaya, “Exploring Multiple Paths using Link Utilization in Computer Networks”, International Journal of Computer Science and Network (IJCSN), Vol. 5, Issue 5, October 2016, pp. 755-762.
2. Kwang Mong Sim and Weng Hong Sun, “*Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions*”, IEEE Transactions on Systems, Man, And Cybernetics—Part A: Systems And Humans, Vol. 33, No. 5, September 2003.
3. Shalini Aggarwal, Pardeep Kumar, Shuchita Upadhyaya, “An Approacg to Identify Multiple Paths in a Network”, International Journal of Scientific & Engineering Research, Volume 7, Issue 12, December-2016, pp. 397-401.
4. R. Schoonderwoerd, O. Holland, J. Bruten, and L. Rothkrantz: “*Ant-Based Load Balancing in Telecommunications Networks*”, Adaptive Behaviour, Vol. 5, No. 2, 1996.
5. Shalini Aggarwal, Pardeep Kumar, Shuchita Upadhyaya, “ACO Approaches to Routing”, International Journal of Control Theory & Applications, 9(40) 2016, pp. 257-266.

AUTHORS PROFILE

Dr. Shalini Aggarwal has completed her Ph.D. in Computer Science & Applications from Kurukshetra University, Kurukshetra. Before that she has done MCA, B.Sc.(Computer Science), from the same university. She has more than 20 publications in various reputed national and international journals. She has attended various national and international conferences for presenting her papers. She is currently working as an Assistant Professor since 2006. Her areas of interests are computer networks, optimization, etc.

Dr. Pardeep Kumar has completed her Ph.D. in Computer Science & Applications from Kurukshetra University, Kurukshetra. Before that he has done M.Sc.(Computer Science) Software, M.Sc.(Statistics & O.R.), B.Sc.(Computer Science), from the same university. He has more than 65 publications in various reputed national and international journals. He has attended various national and international conferences as resource person and for presenting the papers. He is currently working as an Associate Professor since 2012. Currently 6 students are doing Ph.D. under his guidance. His areas of interests are computer networks, optimization, cloud computing, privacy preservation, soft computing, yield management, etc.

Dr. Shuchita Upadhyaya has completed her Ph.D. in Computer Science & Applications from Kurukshetra University, Kurukshetra. Before that she has done MCA from the same university. She has more than 100 publications in various reputed national and international journals. She has attended various national and international conferences as resource person and for presenting the papers. She is currently working as a Professor. More than 10 students has completed their Ph.D. under her guidance. Her areas of interests are computer networks, privacy preservation, soft computing, computer graphics , etc.