Improving Network Performance using ACO Based Redundant Link Avoidance Algorithm

Chandra Mohan B[1], and Baskaran R[2]

**[1] Principal, John Bosco Engineering College
Chennai, Tamil Nadu, India**

**[2] Assistant Professor, Department of CSE, Anna University
Chennai, Tamil Nadu, India**

Abstract

In the wide spread internet, response time and pocket loss are inappropriate due to network traffic, as a result the network efficiency becomes worst and the system provides poor Quality of Service (QoS). An optimal routing protocol, especially multipath may avoid such traffic in the network. But existing routing protocols, both single path and multi path, concentrates only on finding the routes based on any one or some set of metrics, that not always suitable for dynamic, cloud natured network environment. Ant Colony Optimization (ACO) based multipath routing protocol was suggested as an alternate to this problem by many researchers. The multipath ACO also provides same set of link(s) for the source to destination, so that traffic merging again becomes a critical problem. This paper proposes an optimal solution to avoid the problem of traffic merging in the network by removing redundant link in the route. The Proposed algorithm, called 'Redundant Link Avoidance (RLA) algorithm', is an ACO based multi path routing methodology, avoiding copious link in the suggested routes of ACO multipath protocol.

Keywords: *Computer Networks, Routing, QoS, Swarm Intelligence, Ant Colony Optimization.*

1. Introduction

In October 1986, network had a series of congestion collapses, the data throughput from Lawrence Berkley Lab to University of California, sites separated by only two IMP hops, dropped from 32 kbps to 40 bps, i.e., 99.88% packet drop [1]. Due to this incident, the traffic management in computer network becomes critical problem in data communication. Several research efforts have proposed different approaches for traffic management problems, each one has its own strengths and limitations but optimality not yet achieved.

The following Table 1 shows the average response time and the average packet loss of each continent recorded on 12th February of 2010 at 12:40 Indian Standard Time (IST) [2]. It is unambiguous from these experimental results that the Average packet loss is reaches up to 17%. When considering response time, less than 150ms is an appreciate value for data communication, but all continents except North America have more than 150ms. Especially in Australia, the packet loss is 0% but response time is 169ms.

Table 1 Average Response Time and Average Packet Loss of each continent

Continent	Avg. Response Time (in ms)	Avg. Packet Loss (%)
Asia	386	17%
Australia	169	0%
Europe	205	5%
North America	142	8%
South America	138	0%

The loss of data packets and delay in data transmission in the computer network becomes obvious problem due to these reasons: {a} Bottleneck problem due to the growth of networks and increased link speeds; {b} Packet arrival rate exceeds the outgoing link capacity; {c} insufficient memory or buffer to store arriving packets, i.e., buffer of receiving node is not sufficient to store the incoming data packets; {d} network Burst; {e} Demand temporarily exceeds available resources at some point in the network; {f} Same route is used for more transmission, overloading; and {g} traffic merging.

Congestion in the computer network may be controlled [3] either by traffic management algorithms and / or by optimal routing. Traffic management provide optimal solution to solve the facts {a – c} but optimal routing may solve all the listed facts. All the existing traffic management and routing protocol concentrates to avoid the traffic by the way of solving one or few facts in the {a – e} except {f} and {g}. In which, the facts {d} and {e} may be controlled, only by multipath routing algorithm but when applying multipath there are high probability that the network becomes congested by the facts {f} and {g}. The combination of {f} and {g}, i.e., when two or more sources would simultaneously try to transmit packets to one destination via a single link, a high probability that the number of packets would exceed the packets handling capacity of the network and lead to heavy congestion.

In this paper, the proposed algorithm, swarm intelligence based multi path routing protocol with Redundant Link Avoidance (RLA) algorithm, focused to solve this important facts {f}, {g} and also the other facts {a-e}. This routing methodology ensures traffic free routing by avoiding copious link. Perspective of this routing methodology is sharing load on multiple routes to obtain optimal response time and at the same time avoiding packet drop due to overloading in the network.

The remaining of this paper is organized in the following manner: Survey of various related work, its classification, and existing algorithm is explained in the Section Two. The proposed work and the algorithm used in this paper are described in Section Three. Conclusion and further studies are discussed in Section Five followed by the performance analysis of the proposed work is shown in Section Four.

2. Related Work

The basic problem of routing is, to find the path of lowest cost between any two nodes. Routing algorithms can be grouped into three major classes [4]: non-adaptive, adaptive and distributed. Non-adaptive algorithm, also static routing is computed in advance, offline using local information, their routing decision not based on measurements or estimates of the current traffic and topology. Adaptive algorithms, centralized routing will collect information from the entire subnet using global information in an attempt to make optimal solution. Distributed algorithm uses a mixture of local and global information.

Based on the discovery of no of path, the routing algorithms further divided as single-path and multi-path. Single path routing algorithm chooses only one ‘best’ shortest route between a given pair of source to destination for all of its traffic. Multi-path, the bifurcated routing algorithm finds several shortest paths that are almost equally good, between given source to destination. As the traffic is split over several paths, better performance can be obtained in multi-path routing.

There are several research papers [5 - 13] implemented variety of multi-path algorithms for various applications, in which Ant Colony Optimization (ACO) based multipath [13] routing provide better result than others due to its real time computation. In which, [5], [6] are few application of multi path that provides better performance in multimedia and data communication. In order to offer high video quality, a certain minimum end-to-end bandwidth has to be provided. In bandwidth-limited networks, it is difficult to guarantee such a bandwidth with a single path. To meet the bandwidth requirement, a multipath approach is proposed by Jiancong Chen et al [5]. Ka-Cheong Leung et al [6], explains implementation of multipath for avoiding two different traffic types, connection-oriented traffic and mixed traffic. Connection-oriented traffic consists of traffic from TCP connections only, while connectionless traffic comes from non-TCP connections. Mixed traffic is composed of both connectionless and connection-oriented traffic. For both multimedia and data communication, multipath avoids traffic induced by the facts {a – e} shown in section 1, but there are some more situations {f – g} that the network becomes congested when implementing multi path. This paper, further continuing with literature survey of the various multipath routing algorithms developed recently.

The selection of the routing paths is major design consideration that has a drastic effect on the resulting performance. Therefore, although many flow-control algorithms are optimal for a given set of routing paths, their performance can significantly differ for different sets of paths. Previous studies and proposals on multi-path routing in the previous context have focused on heuristic methods. In [7], a multi-path routing scheme, termed equal cost multi-path (ECMP), has been proposed for balancing the load along multiple shortest paths using a simple round-robin distribution. By limiting itself to shortest paths, ECMP considerably reduces the load balancing capabilities of multi-path routing; moreover, the equal partition of flows along the shortest paths, resulting from the round robin distribution, further limits the ability to decrease congestion through load balancing.

OSPF-OMP [8] allows splitting traffic among paths unevenly; however, the traffic distribution mechanism is based on a heuristic scheme that often results in an inefficient flow distribution. A. E. I. Widjaja [9] considered multi-path routing as an optimization problem with an objective function that minimizes the congestion of the most utilized link in the network; however, they focused on heuristics and did not consider the quality of the selected paths.

Israel Cidon et al [10] concluded two phenomena in the implementation of multi path routing protocol: 1) as the link capacity increases, the relative performance of multi-path routing decreases, 2) as the number of possible routes increases, the relative performance of multi-path routing also increases till k number of paths and when it exceeds the limit, the performance will be degraded. So to choose only k path is an important consideration for implementing multi path routing and the optimal value of k, may change in practice. Weijia Jia et al [11] proposed an approach that named as integrated routing, to select a subset of routers to carry out multi path routing (MPR), the rest of the routers execute Single path routing (SPR). With proper selection of MPR routers, shortcomings in the MPR will be rectified. This approach may provide better result but to implement this protocol in the network requires two tasks: router assignment and routing interface selection. The performance of this approach lies only in these two tasks and implementing these two tasks in the internet like environment is highly impractical.

Dijkstra-old-touch-first (Dijkstra-OTF) with multipath extension [12] is an extended version of conventional Dijikstra's shortest path algorithm that computing all lexicographic- lightest paths from a source to every other node in the network, but it requires additional computational efforts. DiCaro et al [13] developed Swarm Intelligence (SI) based routing algorithm, called Ant-Net Routing using ant colony optimization technique with multi path capability. The ACO era starts with Schoonderwoerd et al [14], they proposed Ant based control (ABC) for telephone network, it continues with ant based algorithm for packet switched network proposed by Subramanian et al [15] and a co-operative Asymmetric Forward (CAF) model for routing with asymmetric costs are ant colony optimization based single path routing algorithm developed by Heusse et al [16].

ACO is a stochastic approach that has been proposed to solve different hard combinatorial optimization problems in almost all engineering fields such as traveling salesman problems, graph coloring problems, quadratic assignment problems, resource-constrained project scheduling problem [17], knowledge representation tools[18], Computer-assisted testing systems[19], Grid Workflow Scheduling Problem[20], Self-Structured P2P Information System[21], Intelligent transportation systems[22], Robot Path Integration[23], and Clustering and Data mining[24].

Initially, three different versions of AS were proposed, namely ant density, ant quantity and ant cycle [25]. In the ant density and ant quantity versions, the ants updated the pheromone directly after a move from one city to an adjacent city, in the ant cycle version the pheromone update was only done after all the ants had constructed the tours and the amount of pheromone deposited by each ant was sent to be a function of the tour quality. Now a day, when referring to ACO, it means to ant cycle since the two other variants were abandoned because of their inferior performance. Solution construction and Pheromone Update are the two phases of ACO implementation.

In ACO based routing algorithm, solution construction handles formation of routing table, Pheromone update used to choose the single optimal path based on density of pheromone, in case of single path routing; and the pheromone update desires the probability of load sharing along multiple path, in case of multi path routing. The ACO routing may provide feasible solution for traffic management problem, even though the overloading {f} and traffic merging {g} still is an issue in computer network. This paper concentrates to solve overloading and traffic merging by implementing RLA algorithm as creamy layer over existing ACO. The following section describes our proposed work in detail.

3. Proposed Work

The proposed routing protocol has two components, namely ACO Based Multipath Routing algorithm and redundant link avoidance algorithm. The main idea of ACO is to model the problem as the search for a minimum cost path in a graph. Artificial ants walk through this graph, looking for good paths. Each ant has a rather simple behavior so that it will typically only find rather poor-quality paths on its own. Better paths are found as the emergent result of the global cooperation among ants in the colony.

The behavior of artificial ants is inspired from real ants, they lay pheromone trails on the graph edges and choose their path with respect to probabilities that depend on pheromone trails and these pheromone trails progressively decrease by evaporation. In addition, artificial ants have some extra features that do not find their counterpart in real ants. In particular, they live in a discrete world and their moves consist of transitions from nodes to nodes. Also, they are usually associated with data structures that contain the memory of their previous actions. In most cases, pheromone trails are updated only after having constructed a complete path and not during the walk, and the amount of pheromone deposited is usually a function of the quality of the path. Finally, the probability for an artificial ant to choose an edge often depends not only on pheromones, but also on some problem-specific local heuristics.

The ACO based multipath algorithm implemented in this paper, using forward and backward ants. Ants in each set possess the same structure, but they are differently situated in the environment, i.e., they can sense different inputs and they can produce different, independent outputs. Ants communicate in an indirect way, according to the dynamic paradigm, through the information they concurrently read and write on the network nodes they visit. While travelling toward their destination nodes, the forward ants keep memory of their paths and of the traffic conditions found. In single path routing protocol, an optimal path recorded by the ant was chosen as path between given source to destination. In multi path routing protocol, all paths are chosen for communicating from source to destination. As discussed in the previous section, some of the available path in the multi path routing protocol, heavily congested due to traffic merging problem.

The redundant link avoidance algorithm removes such links called replicating links from destination towards source using two data structures named available route array and selected route array. All available routes found by multipath routing protocol from source to destination is stored in the available routes, a two dimensional array, followed by initializing a selected route (one more two dimensional array), and two queues namely selected_ relationship and available_ relationship.

The first optimal route is copied from available route to selected route, then all links of this route is divided from destination towards source and stored into selected_ relationship queue and the same was removed available route. The same way, next route from available route is selected and divided. If front of available_relationship and front of selected_relationship are equal then this route is removed from available route, and deleted all association from available_relationship and the previous process is repeated until reaches the tail. If front of available_relationship and front of selected_relationship are not equal then next link is processed, and the above step is carried out until front reaches rear of the queue. The detailed algorithm is shown in the following section 3.1 and various test cases for the proposed work is shown in section 3.2 to 3.4.

3.1 Redundant Link Avoidance (RLA) Algorithm:

```
Procedure RLA(t, t_end, Δt)
  input t    // current time
  input t_end// time length of simulation
  input  Δt // time interval between ants generation

  // concurrent activity over the network
  for each i ∈ C do

  M ← InitLocalTrafficModel
  T^A ← InitNodeAvailRoutingTable
  T^S ← InitNodeSelectedRoutingTable

  // concurrent activity on each node
  while t ≤ t_end do
    in_parallel

      if (t mod Δt) =  0 then
        destination←SelectDestination
        (traffic_distribution_at_source)

        Launch_Forward_Ant (source, destination)

      end-if

      for each (Active_Forward_Ant [source, current,
      destination] ) do

          while (current ≠ destination)  do
            next_hop ← SelectLink (current, destination,
            link_queues, T^A)
              PutAntOnLinkQueue (current, next_hop)
              WaitOnDataLinkQueue(current, next_hop)
              CrossLink (current, next_hop)
              Memorize (next_hop, elapsed_time)
              current ← next_hop
          end-while
```

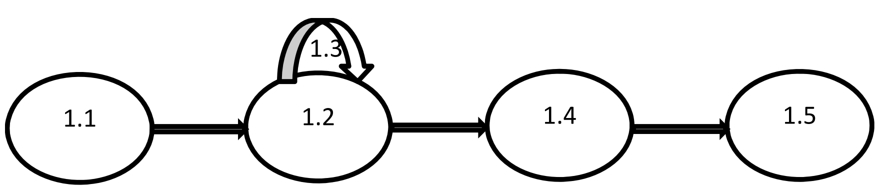


Fig 1. Data Flow in ACO based Routing
Where 1.1 Forward Ant Generation
1.2 Pheromone Matrix Formation
1.3 Removing Cyclic Path
1.4 Backward Ant Generation
1.5 Formatting Routing Table

LaunchBackwardAnt (destination, source, memory_data)

end for each

for each (ActiveBackwardAnt [source, current, destination]) do

while (current ≠ destination) do
next_hop ← PopMemory
WaitOnHighPriorityLinkQueue (current, next_hop)
CrossLink (current, next_hop)
from ← next_hop
current ← next_hop
UpdateTrafficModel (M,current,from,source, memory_data)
r←GetNewPheromone (M, current, from, source, memory data)
UpdateLocalRoutingTable (T^A, current, source, r)

end-while
end-foreach
end-in_parallel

$T^S[0] = T^A[0]$
T^{SL} = each link from Destination to source on T^S

for each (T^A [source, intermediate, destination]) do

$T^{AL}[]$ = each link from Destination to source on current $T^A[]$
$Current_1$ ← First Link in T^{AL}
$Current_2$ ← First Link in T^{SL}

for each ($Current_1 \neq$ Null, $Current_2 \neq$ Null)

if ($Current_1 \neq Current_2$) do
$Current_1$ ← next Link in T^{AL}
$Current_2$ ← next Link in T^{SL}
else-if
Flag = 0
end-if
end-foreach

if (Flag = 1)
$T^S = T^A$ [current]
end-if

end-for each

end-while

end-for each

end-procedure

3.2 Test case 1:

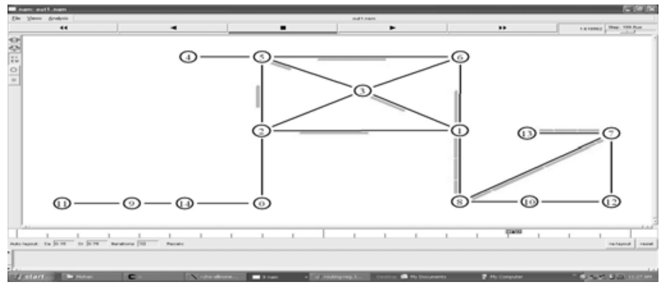


Fig 2 Data flow from source 13 to destination 5 (Test Case 1) using three selected path

Let consider Fig 2, the following table shows the ***available routes***, directed paths of existing ACO and ***selected routes***, directed routes of ACO with RLA from source 13 to destination 5:

Available Route (ACO):

- 13 – 7 – 8 – 1 – 6 – 5
- 13 – 7 – 8 – 1 – 3 – 5
- 13 – 7 – 8 – 1 – 2 – 5
- 13 – 7 – 8 – 1 – 3 – 2 – 5
- 13 – 7 – 8 – 1 – 3 – 6 – 5
- 13 – 7 – 8 – 1 – 6 – 3 – 5
- 13 – 7 – 8 – 1 – 2 – 3 – 5
- 13 – 7 – 8 – 1 – 6 – 3 – 2 – 5
- 13 – 7 – 8 – 1 – 2 – 3 – 6 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 6 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 3 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 2 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 3 – 2 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 3 – 6 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 6 – 3 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 2 – 3 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 6 – 3 – 2 – 5
- 13 – 7 – 12 – 10 – 8 – 1 – 2 – 3 – 6 – 5

Selected Route (ACO with RLA):

- 13 – 7 – 8 – 1 – 6 – 5
- 13 – 7 – 8 – 1 – 3 – 5
- 13 – 7 – 8 – 1 – 2 – 5

If packets are transmitted in all above 18 routes of available routes, then the link 3 – 5, 2 – 3 and 6 – 5 become heavy loaded and it may lead to packet drop. In order to avoid traffic on these links, our redundant link avoidance algorithm removes such link from routing table.

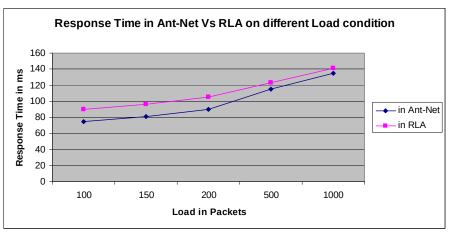
3.3 Test case 2:

The following table 3, shows the ***available routes***, directed routes of existing ACO and ***selected routes***, directed routes of ACO with RLA In multiple route routing the following are ***available routes*** from source 11 to destination 4:

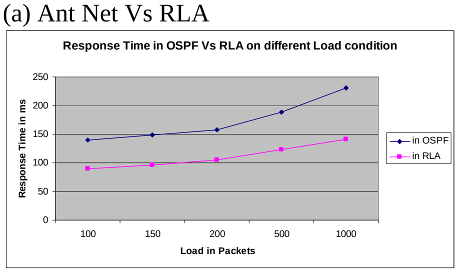
Available Route (ACO):

- 11 – 9– 14– 0 – 2– 5– 4
- 11– 9– 14– 0 – 2 – 3– 5 – 4
- 11 – 9– 14– 0 – 2– 1– 6– 5– 4
- 11 – 9 – 14 – 0 – 2 – 1 – 3 – 5 – 4
- 11 – 9 – 14 – 0 – 2 – 3 – 6 – 5 – 4
- 11 – 9 – 14 – 0 – 2– 1 – 6 – 3 – 5 – 4
- 11 – 9 – 14 – 0 – 2 – 3 – 1 – 6 – 5 – 4
- 11 – 9 – 14 – 0 – 2 – 1 – 3 – 6 – 5 – 4

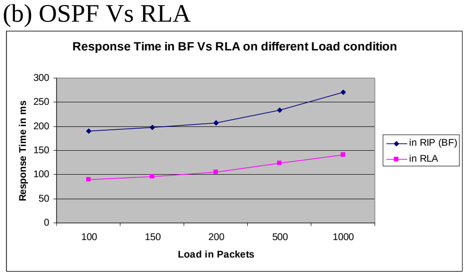
Selected Route (ACO with RLA):



(a) Ant Net Vs RLA



(b) OSPF Vs RLA



(c) BF (RIP) Vs RLA

Fig. 3: Response Time between various routing protocols Vs RLA on different Load condition

- 11 – 9 – 14 – 0 – 2 – 5 – 4

3.4 Test case 3:

The following table 4, shows the ***available routes***, directed routes of existing ACO and ***selected routes***, directed routes of ACO with RLA In multiple route routing the following are ***available routes*** from source 0 to destination 1:

Available Route (ACO):

- 0 – 2 – 1
- 0 – 2 – 3 –1
- 0 – 2 – 5 – 6 – 1
- 0 – 2 – 3 – 6 – 1
- 0 – 2 – 5 – 3 – 1

Selected Route (ACO with RLA):

- 0 – 2 – 1
- 0 – 2 – 3 –1
- 0 – 2 – 5 – 6 – 1

4. Result and Analysis

We implemented the proposed work in NS2 simulation, a widely used open source discrete event Network Simulator tool. There are many network simulation tools available, for example, GloMoSim, Omnet, Opnet, Qualnet. In which NS2 is preferred by almost all researchers and academicians, because it covers a very large set of applications, protocols, network types, network elements and traffic models.

The following events are scheduled in the Implementation, 1) Data communication from Source1 to destination for 8 sec, 2) the proposed work implemented for 8 Sec. By using AWK Programming, the packet loss and average response time of before and after implementing the proposed algorithm is measured from the trace file. These values are listed here:

- 100 packets loss in 8 sec simulation implementing existing ACO based multi-path algorithm
- 0 packets loss in 8 sec simulation implementing proposed ACO with RLA
- RTT is 0.1581 Sec in existing ACO multi-path algorithm
- RTT is 0.1586 Sec in proposed ACO with RLA

The detailed performance analysis is shown in the following section 4.1

4.1 Simulation Result of Redundant Link Avoidance Algorithm Based Routing

In this section, the result of RLA algorithm from source 13 to destination 5, described in Test case 1 is shown Fig 2. The data communication from source node 13 to destination node 5 through three selected route explained in case 1, also shown in the Fig 2. Comparison of response time in existing ACO (Ant Net) Vs proposed ACO with RLA algorithm is shown in the Fig. 3(a). Similarly, Comparison of response time in OSPF and SPI Vs proposed ACO with RLA algorithm is shown in the Fig.3(b) and Fig. 3(c) respectively.

5. Conclusion

Network traffic is a common problem in all type of network, of topology, of protocol. It requires more attention, as it will diminish all major metric of network. Especially in internet like huge, dynamic and cloud like network, traffic is an unavoidable condition. This paper analyses the various factors that causes for congestion and provides optimal solution called redundant link avoidance algorithm over the existing ant colony optimization based routing protocol. When comparing the existing routing protocol with the proposed work, the response time get delayed and this delay in transmission time is a negligible, only 5 ms but all packet loss is avoided.

It will perform even better, when implementing the same work for some complex network with different topologies and high packet loss. As shown in the test case 2 of section 3.3, there are few situations, the multi route routing protocol may act as single route routing protocol. It may lead to poor response time. But in the huge resource centric world wide web, it is promised that it never will ensue.

Acknowledgement:

The First Author, Chandra Mohan B would like to thanks Dr. Sridharan R, Assistant Professor, Department of Electronics and Communication Engineering, College of Engineering, Anna University, Chennai for his constant support and help.

Both Authors would like to thanks Dr. Srivatsa, Professor (Retired), Department of Electronics and Communication Engineering, Madras Institute of Technology, Anna University, Chennai and Dr. Kannan. A, Professor, Department of Computer Science and Engineering, College of Engineering, Anna University, Chennai

References

[1] Jacobson. V, "Congestion Avoidance and Control", ACM SIGCOMM Computer Communication Review, Vol 25, p157 – 187, 1995

[2] Real time traffic monitoring available at http://www.internettrafficreport.com/details.htm

[3] Andrew S. Tanenbaum, "Computer Networks", Prentice Hall of India, 3rd Edition, 1998

[4] Larry L. Peterson & Bruce S. Davie, "Computer Networks – A Systems approach", Morgan Kaufmann Series, 2000

[5] Jiancong Chen, S.-H. Gary Chan, and Victor O. K. Li, "Multipath Routing for Video Delivery Over Bandwidth-Limited Networks", IEEE Journal on Selected Areas in Communications, Vol. 22, No. 10, December 2004

[6] Ka-Cheong Leung, and Victor O.K. Li, "Generalized Load Sharing for Packet-Switching Networks I: Theory and Packet-Based Algorithm", IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 7, July 2006

[7] J. Moy, "OSPF Version 2," IETF RFC 2328, Apr. 1998

[8] C. Villamizar, "OSPF optimized multi-path (OSPF-OMP)," Internet Draft, 1998.

[9] A. E. I. Widjaja, "Mate: MPLS adaptive traffic engineering," Internet Draft, Aug. 1998.

[10] Israel Cidon, Raphael Rom, and Yuval Shavitt, "Analysis of Multi-Path Routing", IEEE/ACM Transactions On Networking, Vol. 7, No. 6, December 1999

[11] Weijia Jia, Dong Xuan and Wei Zhao, "Integrated Routing Algorithms for any cast Messages", IEEE Communications Magazine, January 2000

[12] Joao Luis Sobrinho, "Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet", IEEE / ACM TRANSACTIONS ON NETWORKING, VOL. 10, NO. 4, AUGUST 2002

[13] Di caro G., and Dorigo M. (1998), 'Extending AntNet for best effort quality of service routing', First international workshop on ant colony optimization

[14] Schoonderwoerd R., Holland O., Bruten J., (1997) 'Ant like agents for load balancing in telecommunication networks', In proceedings of the first int. conf. on autonomous agents, pp209-216, New York, ACM Press

[15] Subramanian D., Druschel P., and Chen J., (1997) 'Ants and reinforcement learning: A case study in routing in dynamic networks', In proceedings of the 15th int. joint conf. on artificial intelligence, pp823-838, San Francisco, Morgan Kaufmann

[16] Heusse M., Snyers D., Guerin S., and Kuntz P. (1998), 'Adaptive agent driven routing and load balancing in communication networks', Advances in complex systems, 1(2), pp 237-254.

[17] Wei-Neng Chen, Jun Zhang, Henry Shu-Hung Chung, Rui-Zhang Huang, and Ou Liu, "Optimizing Discounted Cash Flows in Project Scheduling—An Ant Colony Optimization Approach", IEEE TRANSACTIONS ON SYSTEMS,

MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 40, NO. 1, p64 – 77, JANUARY 2010

[18] Pedro C. Pinto, Andreas N¨agele, Math¨aus Dejori, Thomas A. Runkler, and Jo˜ao M. C. Sousa, "Using a Local Discovery Ant Algorithm for Bayesian Network Structure Learning", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 13, NO. 4, p767, AUGUST 2009

[19] Xiao-Min Hu, Jun Zhang, Henry Shu-Hung Chung, Ou Liu, and Jing Xiao, "An Intelligent Testing System Embedded With an Ant-Colony-Optimization-Based Test Composition Method", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 39, NO. 6, NOVEMBER 2009

[20] Wei-Neng Chen, and Jun Zhang, "An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 39, NO. 1, JANUARY 2009

[21] Agostino Forestiero, and Carlo Mastroianni, "A Swarm Algorithm for a Self-Structured P2P Information System", IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 13, NO. 4, AUGUST 2009

[22] Minglu Li, Hongzi Zhu, Member, Yanmin Zhu, and Lionel M. Ni, "ANTS: Efficient Vehicle Locating Based on Ant Search in Shanghai Grid", IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 58, NO. 8, OCTOBER 2009

[23] Girma S. Tewolde, and Weihua Sheng, "Robot Path Integration in Manufacturing Processes: Genetic Algorithm Versus Ant Colony Optimization", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 38, NO. 2, MARCH 2008

[24] Parag M. Kanade and Lawrence O. Hall, "Fuzzy Ants and Clustering", IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 37, NO. 5, SEPTEMBER 2007

[25] Dorigo M and Stutzle T, "Ant Colony Optimization", MIT Press, First Edition

Authors

[*] **Mr. B. Chandra Mohan** received his Diploma in Electrical and Electronics Engineering, B.Tech Degree in Information Technology and Master degree in Computer Science and Engineering from Directorate of Technical Education, Madras University and Anna University in the years 1993, 2003 and 2006 respectively. Presently he is pursuing Research in Anna University Chennai. He worked both industry and academic in India and Abroad about 15 years. Presently he is serving as Principal in John Bosco Engineering College, Tiruvallur, Tamilnadu, India. His experience includes as Faculty in Anna University Chennai, as Tech In-charge in Kuwait National Petroleum Company, as Principal in PTR College of Engg and Tech. His present research included Networks, Data Mining and Swarm intelligence. He has published dozen of papers in National, International Conferences and Book series. His paper is awarded as best paper in the IEEE International Conference (SNPD 2008). He is a reviewer in IEEE JSAC, Elsevier and few international journals. He delivered more than 50 Special Lectures in National, International Seminars, Workshops and Faculty Development Programs. He has worked in Kuwait. He is a life member of Institution of Electronics and Telecommunication Engineers (IETE), Indian Society for Technical Education (ISTE), Computer Society of India (CSI) and International Association for Engineers (IAEng).

[**]**Dr. R. Baskaran** received his B.Tech in Electrical and Electronics Engineering, Master Degree in Computer Science and Engineering, Madras University and Doctorate from Anna University in the years 2000, 2001, and 2007 respectively. He is now working as an Assistant Professor in Department of Computer Science and Engineering, College of Engineering Guindy, Anna University Chennai. His present research includes Networks, Database, Data mining and warehousing and Image Processing. He presented more than 50 Special Lectures in National, International Seminars, Workshops and Development Programs. He is an expert in Data mining. He has published 35+ papers in International, National Journals and Conferences. He is a reviewer in IEEE/ ACM JSAC, Elsevier and many international journals. He is a life member of Institution of Electronics and Telecommunication Engineers (IETE), Indian Society for Technical Education (ISTE), Computer Society of

India (CSI) and International Association for Engineers (IAEng).