Ant based Algorithm for Load Balancing in Mobile Ad Hoc Networks

Rajneesh Kumar Gujral
Assoc. Professor, Computer Engineering Department, M. M. Engineering College, M. M. University, Ambala, India-133207

Manpreet Singh
Professor, Computer Engineering Department, M. M. Engineering College, M. M. University, Ambala, India-133207

Sanjeev Kumar Rana
Assoc. Professor, Computer Engineering Department, M. M. Engineering College, M. M. University, Ambala, India-133207

ABSTRACT

Mobile ad hoc networks are infrastructure-less networks consisting of wireless, possibly mobile nodes which are organized in peer-to-peer and autonomous fashion. Each node is also a router that forwards data packets to its proper destination. A new family of algorithms inspired by Swarm Intelligence has come into existence to provide route optimization through routing load distribution. In this paper, we have proposed an Ant based routing algorithm to ensure appropriate load balancing in mobile ad hoc network using AODV and Ant Colony Optimization metaheuristics.

Keywords: Ant Colony Optimization, Ad hoc Network, Load Balancing, Swarm Intelligence**.**

1. INTRODUCTION

Mobile Ad hoc Networks (MANETs) are communication networks built up of a collection of mobile devices which can communicate through wireless connections. The responsibilities of a routing protocol include exchanging the route information; finding a feasible path to a destination based on criteria such as hop length, minimum power required and lifetime of the wireless link gathering information about the path breaks; mending the broken paths expending minimum processing power and bandwidth; and utilizing minimum bandwidth. This task is particularly hard in MANETs due to mobility of the network elements.

Swarm intelligence is a relatively new approach to problem solving that takes inspiration from the social behavior of insects and of other animals. In particular, ants have inspired a numbers of methods and techniques among which the most studied and the most successful is the general purpose optimization technique known as ant colony optimization. Ant colony optimization (ACO) takes inspiration from the foraging behavior of some ant species. These ants deposit pheromone on the ground in order to mark some favorable path that should be followed by other members of the colony.

Ant colony optimization algorithms have been applied to many combinatorial optimization problems, ranging from quadratic assignment to fold protein or routing vehicles and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations [20]. It has also been used to produce near-optimal solutions to the travelling salesman problem. They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically; as the ant colony algorithm can run continuously and adapt to changes in real time.

There is in practice a large number of algorithms claiming to be "ant colonies", without always sharing the general framework of optimization by canonical ant colonies (COA). In practice, the use of an exchange of information between ants via the environment (a principle called "Stigmergy") is deemed enough for an algorithm to belong to the class of ant colony algorithms[1]. The two main characteristics of stigmergy that differentiate it from other forms of communication are the following:

- Stigmergy is an indirect, non-symbolic form of communication mediated by the environment insects exchange information by modifying their environment.
- Stigmergic information is local: it can only be accessed by those insects that visit the locus in which it was released (or its immediate neighborhood).

Examples of stigmergy can be observed in colonies of ants. In many ant species, ants walking to and from a food source deposit on the ground a substance called *pheromone.* Other ants perceive the presence of pheromone and tend to follow paths where pheromone concentration is higher. Through this mechanism, ants are able to transport food to their nest in a remarkably effective way.

Deneubourg et al. [3] thoroughly investigated the pheromone laying and following behavior of ants. In figure 1, an experiment known as the "double bridge experiment", the nest of a colony of Argentine ants was connected to a food source by two bridges of equal lengths. In such a setting, ants start to explore the surrounding of the nest and eventually reach the food source. Along their path between food source and nest, Argentine ants deposit pheromone. Initially, each ant randomly chooses one of the two bridges. However, due to random fluctuations, after some time one of the two bridges presents a higher concentration of pheromone than the other and, therefore, attracts more ants. This brings a further amount of pheromone on that bridge making it more attractive with the result that after some time the whole colony converges toward the use of the same bridge. This colony-level behavior, based on autocatalysis, that is, on the exploitation of positive feedback, can be exploited by ants to find the shortest path between a food source and their nest.

Goss et al. [4] considered a variant of the double bridge experiment in which one bridge is significantly longer than the other.

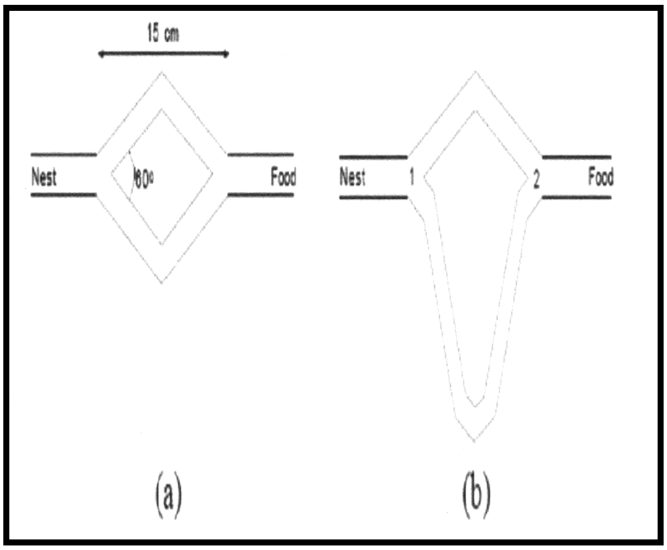


Figure 1: Experimental setup for the double bridge (a) Branches have equal lengths. (b) Branches have different lengths.

In this case, the stochastic fluctuations in the initial choice of a bridge are much reduced and a second mechanism plays an important role: the ants choosing by chance the short bridge are the first to reach the nest. The short bridge receives, therefore, pheromone earlier than the long one and this fact increases the probability that further ants select it rather than the long one. The model proposed by Goss and co-workers for explaining the foraging behavior of ants was the main source of inspiration for the development of ant colony optimization. In ACO, a number of artificial ants build solutions to an optimization problem at hand and exchange information on the quality of these solutions via a communication scheme that is reminiscent of the one adopted by real ants.

1.1 Metaheuristic Concept

A metaheuristics is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a metaheuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications. Examples of metaheuristics include simulated annealing [8], taboo search [9], iterated local search [10], evolutionary computation [2] and ant colony optimization [5] [7].

Ant Colony Optimization Metaheuristic: Ant colony optimization (ACO) has been formalized into a metaheuristic for combinatorial optimization problems by Dorigo and coworkers [6],[7]. The model of a combinatorial optimization problem is used to define the pheromone model of ACO. A pheromone value is associated with each possible solution *component;* that is, with each possible assignment of a value to a variable. The set of all possible solution components is denoted by C. In ACO, an artificial ant builds a solution by traversing the fully connected *construction graph* G (V,E), where V is a set of vertices and E is a set of edges. This graph can be obtained from the set of solution components C in two ways: components may be represented either by vertices or by edges. Artificial ants move from vertex to vertex along the edges of the graph, incrementally building a *partial solution.* Additionally, ants deposit a certain amount of pheromone on the components; that is, either on the vertices or on the edges that they traverse. The amount Δt of pheromone deposited may depend on the quality of the solution found. Subsequent ants use the pheromone information as a guide toward promising regions of the search space.

Algorithm 1: The Ant Colony Optimization Metaheuristic:

```
Step1:   Set parameters, initialize pheromone trails
Step2:   While (termination condition not met) do
             Construct Ant Solutions
Step 3:      Apply Local Search (optional)
             Update Pheromones
         End while
```

The ACO metaheuristic is shown in Algorithm 1. After initialization, the metaheuristic iterates over three phases: at each iteration, a number of solutions are constructed by the ants; these solutions are then improved through a local search (this step is optional) and finally the pheromone is updated.

1.2 ACO for Routing: General Principles

ACO routing algorithms [2][11] take inspiration from the behavior of ants in nature and from the related field of ACO to solve the problem of routing in communication networks. ACO routing algorithms involves a number of interesting properties as compared to traditional routing algorithms.

- They are adaptive by means of continuous path sampling and probabilistic ant forwarding which leads to an uninterrupted exploration of the routing capabilities.
- They are robust. This is because routing information is the result of the repeated sampling of paths. The use of sampling implies that routing information is based on direct measurements of the real network situation leading to enhancement in reliability.

The rest of paper is organized as follow: Section 2 deals with related work, Section 3 deals with the system model, The proposed algorithm is described in Section 4, Simulation results are discussed in Section 5 and Section 6 concludes the paper.

2. RELATED WORK

Gianni Di Caro et al. proposed AntNet, a routing algorithm based on the principle of ant colony optimization [12]. In AntNet, each node maintains a routing table and an additional table containing statistics about the traffic distribution over the network. The routing table maintains for each destination and for each next hop a measure of the goodness of using the next hop to forward data packets to the destination. These goodness measures, called pheromone variables, are normalized to one in order to be used by a stochastic routing policy. AntNet uses two sets of homogeneous mobile agents called forward ants and backward ants to update the routing tables. The forward ants use heuristics based on the routing table to move between a given pair of nodes and are used to collect information about the traffic distribution over the network. The backward ants retrace the paths of forward ants in the opposite direction. At each node, the backward ants update the routing table and the table containing statistics about the traffic distribution over the network. AntNet has been shown to perform better than Bellman-Ford, Open Shortest Path Forwarding (OSPF) routing protocols under varying and near saturation traffic loads.

Ant-Based Control (ABC) is an algorithm proposed by Schoonder woerd et al. [13] for load balancing in circuits witched networks. In ABC, the calls are routed using probabilistic routing tables that consist of next hop probabilities for each destination. The link costs are assumed to be symmetric and hence, only one-directional mobile agents are used for updating and maintaining the routing tables. The mobile agents use heuristics based on the routing tables to move across the network between arbitrary pairs of nodes. At each node along the path, the mobile agents update the routing tables based on their distance from the source node and the current state of the routing table.

Camara and Loureiro [16] described a novel routing protocol called Global Positioning System ANT-Like Routing Algorithm (GPSAL) which uses a Global Positioning System and mobile software agents modeled on ants for routing. Ants are used to collect and disseminate information about the location of nodes in the MANET while the GPS provides the physical location of a destination node. This algorithm focuses on position-based routing and so does not consider the concentration of pheromone on any paths.

Another ant based routing algorithm for wired networks has been proposed by Kuntz et al. [14]. The proposed algorithm differs from AntNet in terms of different loop detection behavior, simpler backward ant and different routing table update procedure.

Ad hoc Networking with Swarm Intelligence (ANSI) is a reactive routing protocol proposed by Rajagopalan and Shen [19] for mobile ad hoc networks. ANSI protocol uses two sets of mobile agents called forward reactive ants and backward reactive ants. The routing tables in ANSI contain an entry for each reachable node and next best hop while the ant decision tables store the pheromone values. In ANSI, the forward reactive ants are generated only when a node needs to transmit data to another node. The forward reactive ants broadcast while the backward reactive ants retrace the path of forward reactive ants and update the pheromone values at the nodes. The data packets choose the next hop deterministically i.e., the hop which contains the largest pheromone value is chosen as the next hop. ANSI has been shown to perform either better or comparable with AODV with respect to packet delivery and end-to-end delay [15].

ARAMA [17] is a proactive routing algorithm proposed by O. Hossein and T. Saadawi. The main task of the forward ant as in other ACO algorithms for MANETs is to collect path information. However, in ARAMA, the forward ant takes into account not only the hop count factor, as most protocols do, but also the links local heuristic along the route such as the node's battery power and queue delay. ARAMA defines a value called grade. This value is calculated by each backward ant, which is a function of the path information stored in the forward ant. At each node, the backward ant updates the pheromone amount of the node's routing table, using the grade value. The protocol uses the same grade to update pheromone value of all links. The authors claim that the route discovery and maintenance overheads are reduced by controlling the forward ant's generation rate. However, they do not clarify how to control the generation rate in a dynamic environment.

3. SYSTEM MODEL

Swarm intelligence boasts a number of advantages due to the use of mobile agents and stigmergy [17][18]. They are as follows:

1) Pheromone Table

Paths are implicitly defined by the pheromone tables which are kept locally at each node. An entry of the pheromone table *i* at node *i* contains a value indicating the estimated goodness of going from *i* over neighbor *n* to reach destination *d*. This goodness is a combined measure of path end-to-end delay and number of hops.

2) Reactive path setup

When a source node *s* starts a communication session with a destination node *d*, it broadcasts a reactive forward ant. At each node, the ant either unicast or broadcast accordingly whether or not the current node has pheromone information for *d*. If information is available, the ant chooses its next hop *n* with the probability *P* which depends on the relative goodness of *n* as a next hop, expressed in the pheromone variable parameter which controls the exploratory behavior of the ants.

3) Proactive path maintenance and exploration

During the course of a communication session, source nodes send out proactive forward ants to update the information about currently used paths and try to find better paths. They follow pheromone and update routing tables in the same way as reactive forward ants. Such continuous sampling of paths and pheromone updating by ants is the typical mode of operation in ant inspired outing algorithms.

4) Stochastic data routing

Nodes in Ant Hoc Net forward data stochastically according to the pheromone values. When a node has multiple next hops for the destination *d* of the data, it randomly selects one of them, with probability.

5) Link Failures

Nodes can detect link failures (e.g. a neighbor has moved away) when unicast transmissions fail or when expected periodic pheromone diffusion messages were not received due to less available bandwidth at node.

4. PROPOSED ALGORITHM

Design of proposed ant based routing algorithm for load balancing is based on ACO routing algorithm used in wired networks. It deploys ant agents which follow and update pheromone tables in a stigmergic learning process. Data packets are routed stochastically according to the learned tables. It is reactive in the sense that nodes only gather routing information for destinations which they are currently communicating with, while it is proactive because nodes try to maintain and improve routing information as long as communication is going on.

We make a distinction between the path setup, which is the reactive mechanism to obtain initial routing information about a destination at the start of a session, path maintenance and improvement, which is the normal mode of operation during the course of a session to proactively adapt to network changes. Path maintenance and improvement is supported by pheromone diffusion process. The routing information obtained via stigmergic learning is spread between the nodes of the MANET in an information bootstrapping process to provide secondary guidance for the learning agents. Link failures are dealt with using a Backpressure Technique (Ants messages) [20].

Algorithm 2

The sequence of steps in proposed algorithm is as follows:

Step 1: Each network node launches forward ants to all destinations in regular time intervals.

Step 2: The ant finds a path to the destination randomly based on the current routing tables.

Step 3: The forward ant creates a stack, pushing in trip times for every node as that node is reached.

Step 4: When the destination is reached, the backward ant inherits the stack.

Step 5: The backward ant pops the stack entries and follows the path in reverse.

Step 6: The node tables of each visited node are updated based on the trip times.

Step 7: The message ant is generated as link failure occurs.

Following pseudo code describe the functions of a node when it behave as forward/backward ant during path finding between source to destination

```
if (Forward ant)
{ Get the next node based on the pheromone value
if (the link is available and no loop caused)
            then
            {
            Update forward ant with network status (stack)
            Send forward ant to the next node
            }
else if (no such link exist)
            {
            Create backward ant and load contents of forward
            ant to backward ant (queue).
            Send backward ant towards source along the same
            path as forward ant
            }
}
if (backward ant)
{ if current node is source node
 {Store path, kill backward ant and update routing table }
   else
 {
   Forward backward ant on to link available on queue
  update routing table
 }
   if (next node is not available)
        Kill backward ant
  else
  { if (link failure)
     then
           Update forward ant with network status as failure and
           stop sending information (data)
           Send message ant to the previous node regarding link
           failure and update of table for alternative path till path
           is recovered or restore
            }
 }
```

In proposed algorithm, routing information is organized in pheromone tables, each node *i* maintains one pheromone table *Ti*, which is a two-dimensional matrix. An entry T^{d}_{ij} of this pheromone table contains information about the route from node *i* to destination *d* over neighbor *j* as shown in figure 2. T^{d}_{ij} indicate the relative goodness of going over node *j* when traveling from node *i* to destination *d* as well as statistics information about the path and possibly virtual pheromone. Apart from a pheromone table, each node also maintains a neighbor table, in which it keeps track of which nodes it has a wireless link to.

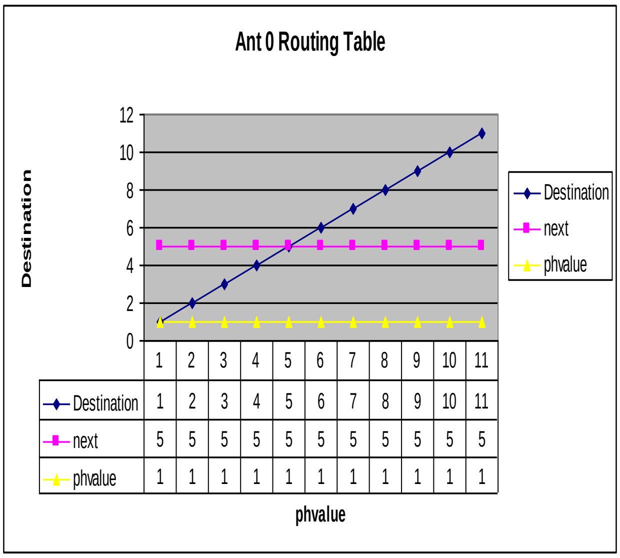


Figure 2: Routing Table of Ant 0

5. SIMULATION RESULTS AND DISCUSSION

We evaluated the performance of proposed algorithm NS 2 over several QoS parameters such as Packet Delivery Ratio, Throughput, End to End Delay, Routing Overhead and Jitter.

Packet Delivery Ratio (PDR): is the ratio of the data packets delivered to the destinations to those generated by the CBR sources. This metric characterizes both the completeness and correctness of the routing protocol.

$$PDR = \frac{\sum_{1}^{n} CBRrece}{\sum_{1}^{n} CBRsent} * 100$$

Average End to End Delay: is the average time taken by a data packet to reach from source node to destination node. It is ratio of total delay to the number of packets received.

$$Avg\_End\_to\_End\_Delay = \frac{\sum_{1}^{n} (CBRrecetime - CBRsenttime)}{\sum_{1}^{n} CBRrece} * 100$$

Throughput: is the ratio of total number of delivered or received data packets to the total duration of simulation time.

$$Throughput = \frac{\sum_{1}^{n} CBRrece}{simulationtime}$$

Normalized Protocol Overhead/ Routing Load: is the ratio of total number of the routing packets to the total number of received data packets at destination.

$$Routing\_Load = \frac{\sum RTRPacket}{\sum CBRrece}$$

Energy Consumption: Nodes consume power to transmit and receive the data. We can calculate this by subtracting the remaining energy with total initialized energy.

Table 1 Simulation Parameters

Simulation Time	Start Time	Finish Time	Sampling Interval
12 minutes (720 seconds)	0	715	0.5 seconds
Number of Nodes	**Routing Protocol**		**Traffic**
25	Antnet		CBR

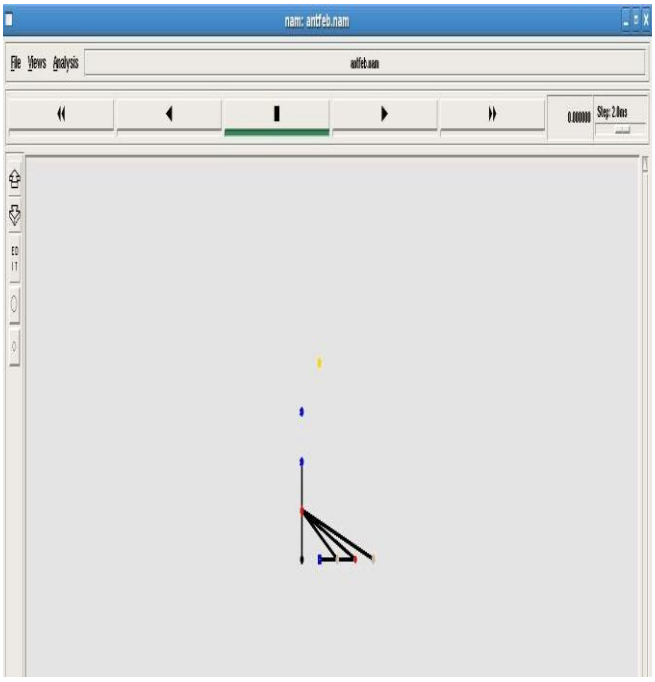


Figure 3: Scenario for Simulation

Tables 1 describe different simulation parameters. Figure 3 and figure 4 shows the scenario for simulation and data transfer from source node to destination node by proposed algorithm respectively**.**

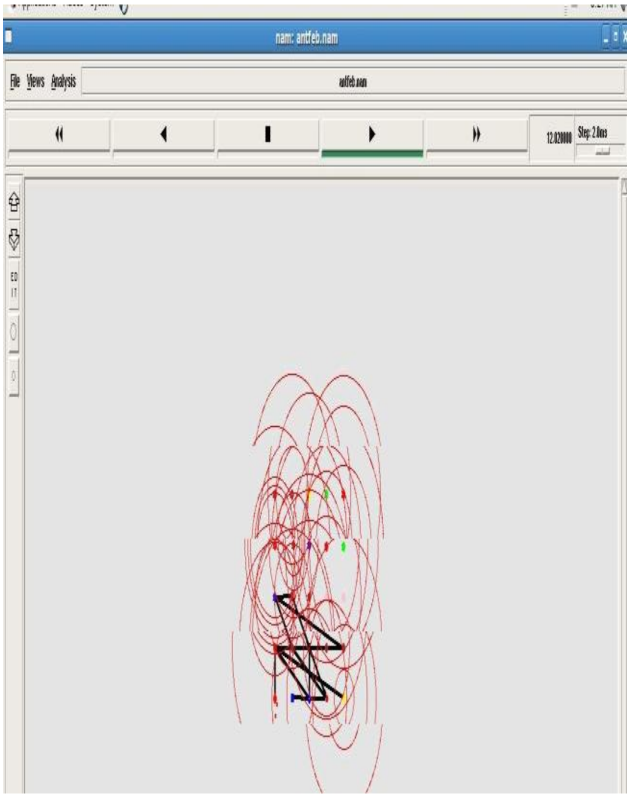


Figure 4: Data Transfer in Proposed Algorithm

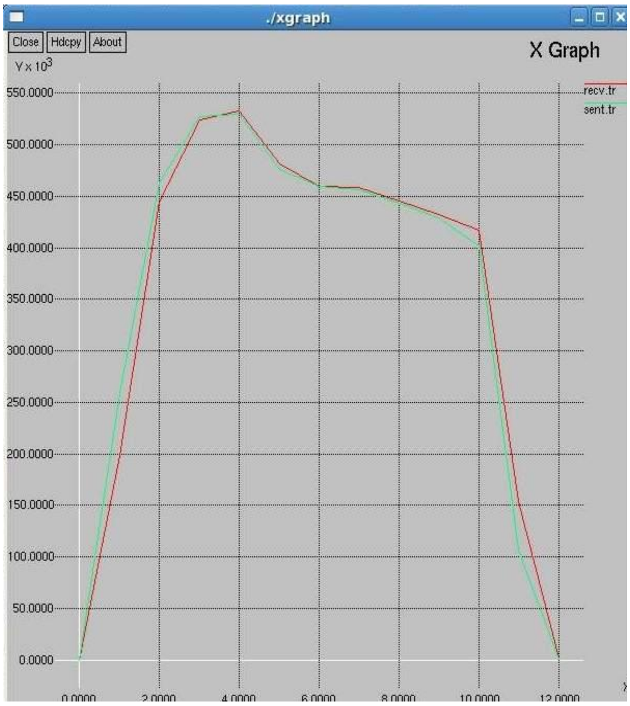


Figure 5: Packets Transmitted Vs Packets Received

As shown in figure 5, packet sent and received curves are very close to each other. If there is any variation in packet transmission that is directly reflected to packet received curve. It shows that ants maintain the balance between packet sent and packet received in order to reduce the packet losses. Figure 6 clearly highlights that in the beginning of simulation, energy consumption is very low but it increases as the ants participate in the transmission. Figure 7 and Figure 8 shows analysis of different QoS parameters for proposed algorithm.

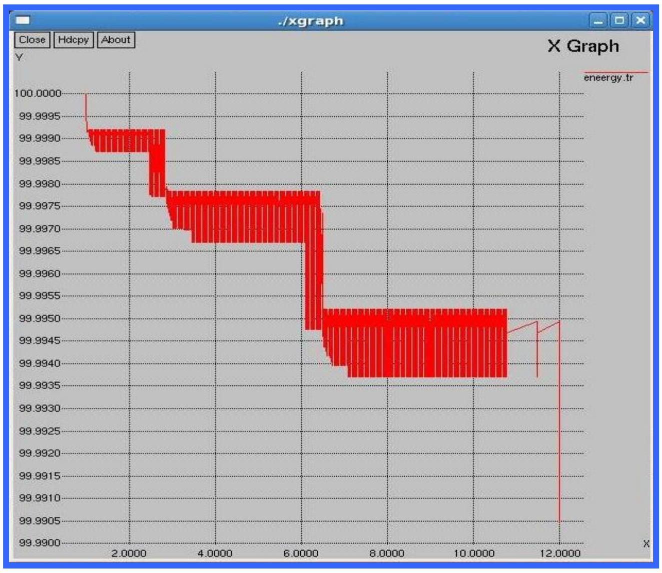


Figure 6: Energy Consumption

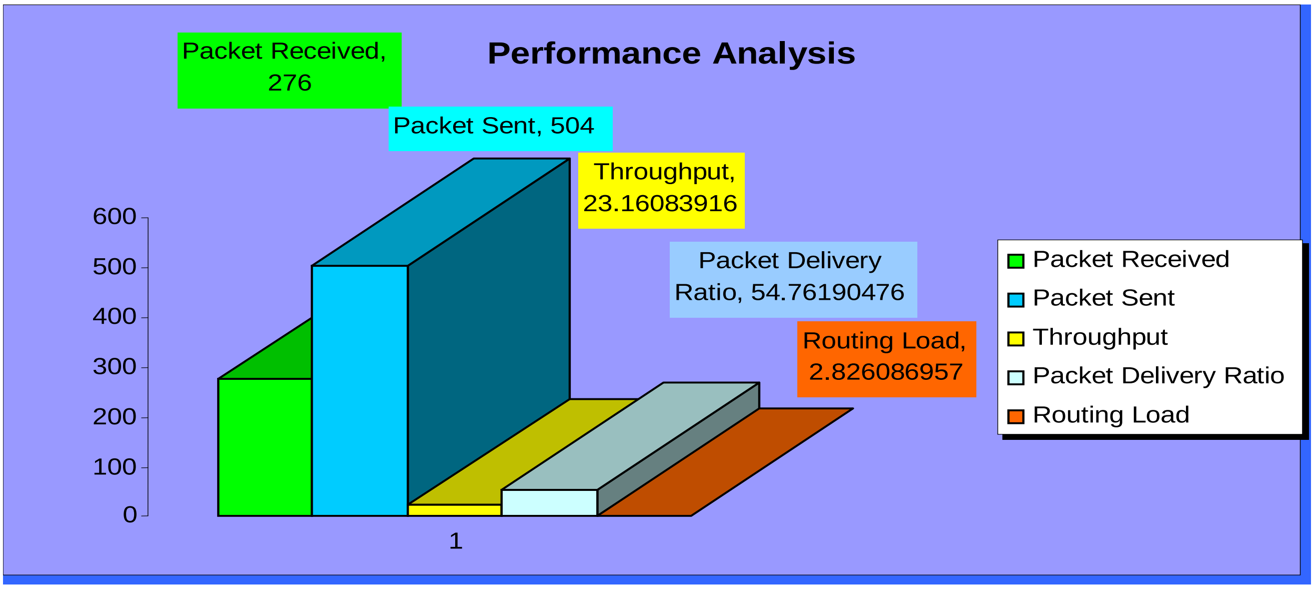
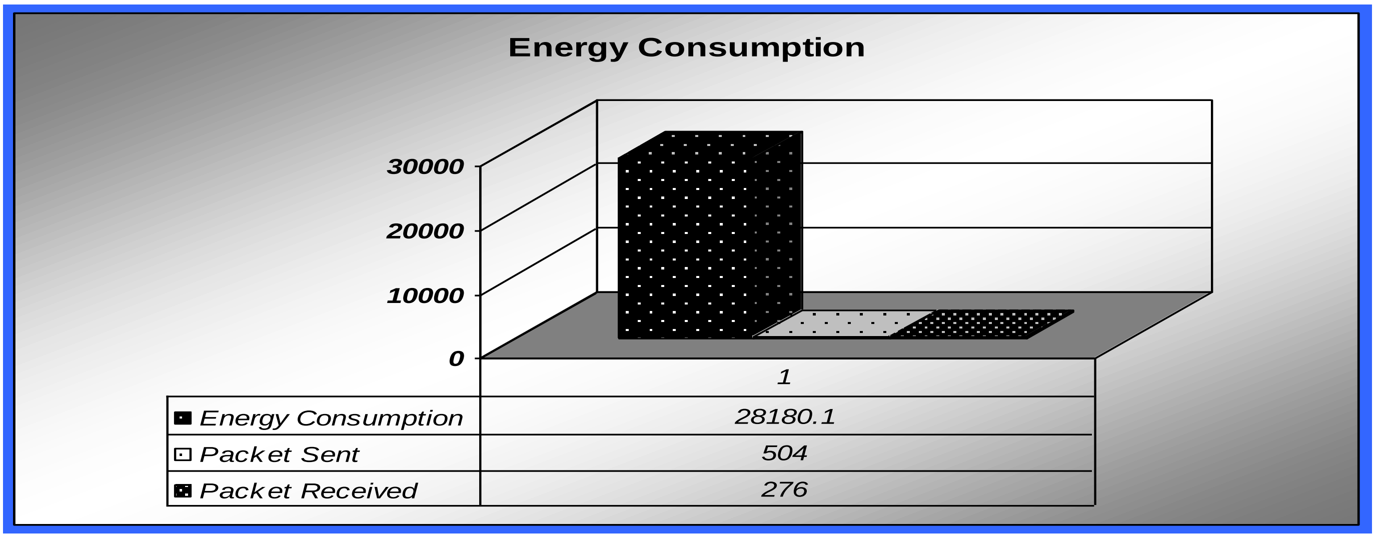


Figure 7: Simulation Based Analysis of QoS Parameters



	1
■ Energy Consumption	28180.1
□ Packet Sent	504
▨ Packet Received	276

Figure 8: Analysis of Energy Consumption by 25 Nodes

6. CONCLUSION

In this paper, we have proposed an ant based routing algorithm to ensure appropriate load balancing in mobile ad hoc network. Our algorithm mostly selects the optimal paths for transmission of packets from source to destination. The load balancing in network is attained by using multiple paths to send packet from source to destination. The simulation results also confirm that proposed algorithm also achieve the objectives of path setup and dynamic path maintenance, thereby, reducing the congestion in network and improving bandwidth utilization. Analytic proof and models of the swarm-based algorithm performance remain topics of future research.

7. REFERENCES

[1] A. Ajith, G. Crina and R. Vitorino 2006, Stigmergic Optimization, Studies in Computational Intelligence, vol. 31, 299.

[2] Marco Dorigo, Mauro Birattari, and Thomas Scutzle 2006, Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique, IRIDIA - Technical Report Series Technical Report No. TR/IRIDIA/2006-023.

[3] J.Deneubourg, S. Aron, S. Goss, and J. Pasteels 1990, The self organizing exploratory pattern of the Argentine ant, Journal of Insect Behavior, vol. 3.

[4] S. Goss, S. Aron, J.-L. Deneubourg, and J. M. Pasteels 1989, Self-organized shortcuts in the Argentine ant, Naturwissenschaften, vol. 76.

[5] M. Dorigo, V. Maniezzo, and A Colomi 1996, Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics , vol. 26(1), 29-41,

[6] M. Dorigo and T. SCutzle 2004, Ant Colony Optimization, MIT Press, Cambridge, MA.

[7] M. Dorigo, G. Di Caro, and L. M. Gambardella 1999, Ant algorithms fordiscrete optimization, Artificial Life, vol. 5(2), 137-172.

[8] V. Cerny 1985, A thermo dynamical approach to the traveling salesman problem, Journal of Optimization Theory and Applications, vol. 45(1), 41-51.

[9] F. Glover and M. Laguna 1997, Tabu Search, Kluwer Academic Publishers.

[10] H. R Lourenc, O. Martin, and T. Srutzle 2002, Iterated local search, Handbook of Metaheuristics, International Series in Operations Research & Management Science, F. Glover and G. Kochenberger, Eds. Kluwer Academic Publishers, vol. 57, 321-353.

[11] S.S. Dhillon, P. Van Mieghem 2007, Performance analysis of the AntNet algorithm, Computer Networks, vol. 51,2104-2125.

[12] Gianni Di Caro and Marco Dorigo 1998, AntNet: distributed stigmergetic control for communication networks, Journal of Artificial Intelligence Research voI.9, 317-365.

[13] Ruud Schoonder woerd, Owen Holland, Janet Broten 1997, Ant like agents for load balancing in telecommunication networks, Proceedings of the First International Conference on Autonomous agents.

[14] M. Heusse, D. Snyers, S. Guerin and P. Kuntz 1998, Adaptive agent-driven routing and load balancing in communication networks, Rapport technique de l'ENST de Brestagne, RR-98001-iasc.

[15] S. Rajagopalan and C. Shen 2005, ANSI: a unicast routing protocol for mobile ad hoc networks using swarm intelligence, Proceedings of the International Conference on Artificial Intelligence, 24-27.

[16] D. Camara, and A. Loureiro 2000, A novel routing algorithm for ad hoc networks, Proceedings of 33rd Annual Hawaii International Conference on System Sciences,1-8.

[17] T. White 1997, "Swarm intelligence and problem tele communications", Canadian Artificial Magazine ,Spring.

[18] Kawabata, K.; Suzuki, T.; Hayama, T.; Kobayashi, H. 1996, Distributed intelligent control structure for multi-legged walking robots, Proceeding of 4th International Workshop on Advanced Motion Control, (AMC '96-MIE), vol. 1, 231 -236.

[19] A. Colorni, M. Dorigo, and V. Maniezzo 1992, "An investigation of some properties of an ant algorithm", Proceeding of Parallel Problem Solving from Nature Conference (PPSN 92), Brussels, Belgium, 509-520.

[20] Manpreet Singh 2010, GRAAA: Grid Resource Allocation Based on Ant Algorithm, Journal of Advances in Information Technology, vol. 1(3), 133-135.