International Journal of Advanced Research in
Engineering and Applied Sciences ISSN: 2278-6252

# DESIGN OF SIMULATOR FOR ANALYZING PERFORMANCE OF TELECOMMUNICATION NETWORK USING ANTS

Abhishek Bansal*

Ruchi Rewri**

Preetinder Kaur Mann***

**Abstract:** *All networks tend to become more and more complicated. They can be wired, with lots of routers, or wireless, with lots of mobile nodes. The problem remains the same: in order to get the best from the network, there is a need to find the shortest path as well as less congested path. The more complicated the network is, the more difficult it is to manage the routes and indicate which one is the best.*

*The Nature gives us a solution to find the shortest path. The real ants, in their necessity to find food and brings it back to the nest, lay down a chemical pheromone trail. Thus, the ants know where their nest is, and also their destination, based on the intensity of pheromone trail. On the computer the behavior of real ants is modeled by artificial ants for finding optimal path with in a network. This approach is implemented as Ant Colony Optimization metaheuristic technique.*

*The purpose of this paper is to provide a clear understanding of the ACO, by giving a proper and complete systematization of the subject. The simulation developed in .NET using C# language will be a support of a deeper analysis of the factors of the ACO based algorithm, its implementations in routing algorithm, mostly for switched-based telecommunication network, for analysis of network performance will be explained. Simulation results using ACO mode will be compared with non-ACO mode using graphs. Conclusions of recent studies will be given and resume the current employments of this great algorithm inspired by the Nature.*

***Keywords:*** *Component; Ant colony optimization (ACO); Ant ; Genetic Algorithm ; Set Covering Problem(SCP), Mobile Ad-Hoc Network(MANET)*

*M.tech, C.S.E Deptt. of Computer Science, B.M.S.C.E, Sri Muktsar Sahib, Punjab, India

**M.tech, C.S.E Dept. of Computer Science, C.D.L.U, SIrsa

***Assistant Professor, C.S.E, B.M.S.C.E, Sri Muktsar Sahib,Punjab,India

# I. INTRODUCTION

The present research has been carried out to study ACO (ant colony optimization) technique for load balancing in telecommunication networking, so that network capacity could be utilized efficiently. ACO is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs based on the behaviour of ants seeking a path between their colony and a source of food. The present chapter has been divided into four sections. Section 1.1 introduces about load balancing in telecommunication networks. Section 1.2 defines ACO technique which is the main concern of this research work and this section also divides into various sub sections such as definition of ACO metaheuristic technique, ACO problem solving paradigm, behaviour of the real ant and artificial ant, the simple ACO algorithm. Section 1.3 covers the brief introduction and use of ACO for load balancing.

**1.1 Load Balancing In Telecommunications Networks:**

Load balancing is a technique to spread work between two or more computers, network links, CPUs, hard drives, or other resources, in order to get optimal resource utilization, maximize throughput, and minimize response time. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. Load balancing can be useful when dealing with redundant communications links. For example, a company may have multiple Internet connections ensuring network access even if one of the connections should fail. A failover arrangement would mean that one link is designated for normal use, while the second link is used only if the first one fails. With load balancing, both links can be in use all the time. A device or program decides which of the available links to send packets along, being careful not to send packets along any link if it has failed. Use of multiple links simultaneously increases the available bandwidth. [1] In telecommunication networks, Calls between two points are typically routed through a number of intermediate switching stations, or nodes; in a large network, there are many possible routes for each such call. It is thus possible to relieve actual or potential local congestion by routing calls via parts of the network which have spare capacity. Load balancing is essentially the construction of call –routing schemes which successfully distribute the changing load over the system and minimize lost calls. Load balancers can be used to split huge data flows into several sub-flows and use several network analyzers, each reading a part of the original data [2].

### 1.2 PRELIMINARY INTRODUCTION OF ACO (Ant Colony Optimization)

In ACO, a set of software agents called artificial ants search for good solutions to a given optimization problem. To apply ACO, the problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a pheromone model, that is, a set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants. [3]

### 1.3 ACO FOR LOAD BALANCING

Of course it is possible to determine the shortest routes from every node to every other node of the network. In this way the average utilization of nodes will be minimized, but this is not necessarily the ideal way to avoid node congestion, as this has to do with how the traffic on the network is distributed. Controlling distributed systems like these by means of a single central controller has several disadvantages.

1. The controller usually needs current knowledge about the **entire** system, necessitating communication links from every part of the system to the controller.

2. These central control mechanisms **scale** badly, due to the rapid increase of processing and communication overheads with system size.

3. Failure of the controller will often lead to **failure** of the complete system.

4. There is the additional practical commercial requirement that centrally controlled systems may need to be owned by one single **authority**.

Further, the nature of distributed systems like these is highly dynamic, complex and stochastic, and their behavior can neither be predicted nor explained by reducing it to a single central controllable factor. A good decentralized control mechanism will not have the problems mentioned above. The field of artificial life has given us inspiration for such a mechanism that will be fully **distributed** way, is highly **adaptive** to network and traffic changes, uses lightweight mobile agents (called ants) for active path sampling, is **robust** to agent failures, provides multipath routing, and automatically takes care of data load spreading. Our approach ACO is inspired by the work of biologists studying social insects, who have uncovered the mechanisms controlling the foraging behaviors of ants. The most important method is the laying and sensing of trails of pheromones (specialized chemical substances) which are laid in amounts determined by local circumstances, and which by their local concentration subsequently directly influence an ant's choice of route.[1][2]

### 1.3.1 ACO as Metaheuristic

The term metaheuristic is derived from the composition of two Greek words. Heuristic is derived from the verb heuriskein which means “to find”, while the suffix Meta means “beyond, in an upper level”. Before this term was widely adopted, metaheuristics were often called modern heuristics. Metaheuristics are a general class of heuristics for solving optimization problems. [12]

“A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem.” [14].

ACO builds solutions to a given optimization problem, one solution component at a time, according to a defined set of rules (heuristics), i.e. starting with an empty solution, add solution components until a complete solution is built. ACO algorithms make use of past solutions in manipulating an artificial pheromone. The pheromone concentration of unique solution component reflects the estimated utility of this solution component.

**Steps for ACO Metaheuristics**

The main ACO metaheuristic consists of an initialization step and of three algorithmic components whose activation is regulated by the SCHEDULE_ACTIVITIES construct. This construct is repeated until a termination criterion is met. Typical criteria are a maximum number of iterations or a maximum CPU time. These three steps are independent and can operate in parallel or not, can act in a synchronized way or not. So, the designer can use his ideas to implement the three steps in any of his desired way keeping in view of the considered problem. [13]

**ACO METAHERUSTIC**

- Set parameters, initialize pheromone trails
- SCHEDULE_ACTIVITIES
- ConstructAntSolutions
- DaemonActions {optional}
- UpdatePheromones

- END_SCHEDULE_ACTIVITIES

In most applications of ACO to NP-hard problems however, the three algorithmic components undergo a loop that consists in (1) the construction of solutions by all ants, (2) the update of the pheromones, and (3) the (optional) improvement of these solution via the use of a local search algorithm. These three components are now explained in more details.

(1)Construction of Ant Solutions: This step is primarily involved with the concurrent and asynchronously movement of ants over the complete problem space in adjacent states and visiting the neighbour nodes of the construction graph. The movement is quite subjective by the effect of pheromone trails and heuristic information and the incrementally build-up of the solutions is being done.

(2)Updating of Pheromones: Either the concentration of pheromone on the matrix will increase due to ants' deposition of pheromone or the concentration will decrease due to the pheromone evaporation. Practically, the usage is that the more concentration of pheromone in the matrix is due to the components or connections being favoured by more ants or cause may be because that the solution produced by single ant is building a good solution for future ants. Also the pheromone evaporation is helpful with the forgetting because of this the rapid convergence towards a sub-optimal solution is avoided and also it helps in the favouring of the exploration of new paths in the defined search space.

(3)Daemonic Actions: These are the actions taken by the centralized team of the collection of ants which is not performed by single ants. These daemonic actions start the local optimization procedure, collect and analyse the global information which is used to decide the future decision of deposition of pheromone on the connections to bias the search space from a non-local perspective. Practically, what is happening is that the daemon observed all the path covered by all the ants in the colony and allow the ants that have built the best solutions to deposit additional pheromone on the components/connections [13].

## II. PROBLEM STATEMENT

In this paper work, the problem under study is to analyze the telecommunication network performance using ACO algorithm and to design the simulator to mimic the real behaviour of the system. To analyze the performance of telecommunication network for calls routing using artificial environment is much better than to test it on real system. The main problems faced while routing a call in telecommunication network are taken into consideration

1. To simulate a telecommunication network model for distribution of calls between nodes.
2. Read the fixed number of calls to be generated in telecommunication network between randomly chosen pairs of nodes.
3. Identify source node and destination node.
4. Find the optimal path for routing of call by launching a population of simple mobile agents with behaviours modelled on the trail laying abilities of ants.
5. Select the path of ant at each intermediate node according the intensity of simulated pheromones at each node.
6. List all the selected nodes from source to destination in the sequence in which they are generated to give the optimal path in memory of ant and then update the pheromone table accordingly.
7. Select the path for routing calls between nodes as a function of the pheromone intensity in pheromone table.
8. Analyse the performance of the network is by observing the average no of hops taken to complete the calls and number of calls lost.

**Problem Formulation Tool-Simulator**

A simulator is used as tool for the current research work. Simulator is designed for analyzing the performance of a telecommunication network using ACO technique. Fixed number of calls is generated between randomly chosen source and destination nodes. Whenever simulator is used as a research tool then there is need for generating random numbers that are conveniently and efficiently generated from a desired probability distribution. In the current research work random ( ) procedure is used to chose source and destination node between 0 to 29 nodes in British telecommunication SDH network topology.

**Random ( )**

This procedure generate the random number between 0 and 29, the generated number represent the source or destination nodes. Counter counts the number of generated random numbers, RN holds the generated random number, Ri is the number generated between 0 and 1. [1].

Step 1. [Initialization]

Counter: =0

Max: =10

Mean: =1.0

Step 2. Repeat steps 3 to 5 while Counter<=Max

Step 3. RN= (-log (1.0-Ri)/Mean)*29

[Generating a random number in RN using a function for generating random numbers]

Step 4. If RN>29

Set Counter: = Counter-1

Continue step 3.

Step 5.Store the RN in array.

Step 6. Exit.

## III. DESIGN OF SIMULATOR

Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the system or of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system [69]. Simulation can also deal with more complex systems and interactions between systems than analytical models. The main goal of this section is to present a novel complete simulation model for calls routing in telecommunication network inspired by ACO technique.

- Fixed numbers of calls are generated by randomly choice of source node and destination node.
- During a simulation run, before routing the actual call an ant is launched towards destination, this ant selects the optimal path and update the information in routing table accordingly.
- Finally, calls are routed throughout the network based on the information in updated pheromone table.
- Numbers of hops consumed for completing the calls are compared with traditional routing.
- Simulator is run m times, where m is number of calls to be made.

A simulated network models a typical distribution of calls between nodes; nodes carrying an excess of traffic can become congested, causing calls to be lost. In addition to calls, the network also supports a population of simple mobile agents with behaviours modelled on the trail laying abilities of ants. The ants move across the network between randomly chosen pairs of nodes; as they move they deposit simulated pheromones as a function of their distance from their source node, and the congestion encountered on their journey. They select their path at each intermediate node according the intensity of simulated pheromones at each node. Calls between nodes are routed as a function of the pheromone intensity at each intermediate node. The performance of the network is measured by the average no of hops taken to complete the calls and number of calls lost. In this study, a proposed ACO algorithm for load balancing in distributed systems will be presented ,then results of using the ant colony optimization (ACO) are compared with those achieved by using traditional fixed shortest path routes, (dijkstra's algorithm) used in network management. The ACO technique is shown to result in fewer call failures than the other methods, while exhibiting many attractive features of distributed control.

- **Network Topology**-The 30-node network topology of figure (1) was chosen because this is a realistic interconnection structure of a possible switch-based network. It is also the same topology as was used in (Appleby & Steward, 1994)[44], and is in fact the structure of the British Synchronous Digital Hierarchy (SDH) network. The network is cost-symmetric i.e. the congestion status over available paths is fully bidirectional which means that the congestion depends only on the state of the nodes in the path. Moreover, dealing with telephone networks, each call occupies exactly one physical channel across the path. Therefore, calls are not multiplexed over the links, but they can be accepted or refused, depending on the possibility of reserving a physical circuit connecting the caller and the receiver.

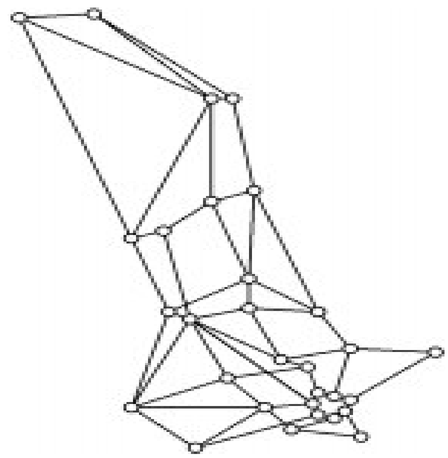


**Figure (1) Network Topology used for Simulation.**

- **Network Configuration**-Network is modelled as graph with nodes and links as shown in figure (2).When a new call arrives at a node then can be accepted or rejected, depending on the possibility of available number of connections which depends on the node capacity.  A call is routed according to the ant routing table, which have information about which link to use to follow the path towards its destination node. When connection is available, the call is routed. On new call's arrival, if there is not enough node capacity to hold it, the call will be discarded.

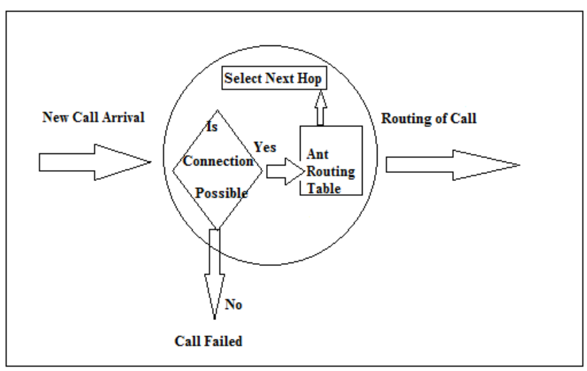



**Figure (2) Node Model**

- **Pheromone (Ant Routing) Table Initialization-**A pheromone table exists for every node of the network. The ant routing table at each node is organized in the form (Next hop, Pheromone Probability). The uniform probabilities assigned to all the neighbours indicate that nothing is known about the state of the network initially. For example, if the network topology is as shown in figure 3, then each node has routing table where all nodes have the normalized random values.

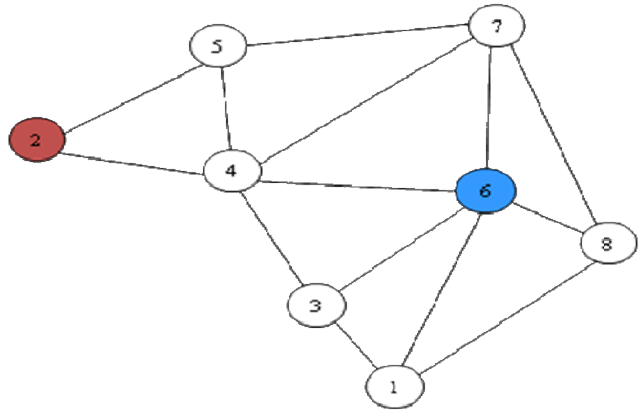



**Figure (3) Example of network topology with source node 6 and destination node 2**

The probabilities in the pheromone tables represent the strength of the pheromone trail. The higher the probability the stronger the pheromone trail. To compute the shortest routes needed to route a call from one node in the network to another node, the next step from a node towards the destination node is always determined by using the pheromone table of the node. The next hop is the node with highest probability in the row that represents the destination. By that, calls are always routed to their destination along the path with highest probability. Since this path represents the strongest pheromone trail to the destination, it should be the shortest known path.

Table 1 illustrates that if the network topology is as shown in figure (3) with source node 6 and destination node 2, then pheromone table (ant routing table) at node 6 is as follows:

| **Next Hop** | **Pheromone Probability** |
|---|---|
| 1 | 0.045 |
| 3 | 0.100 |
| 4 | 0.520 |
| 7 | 0.325 |
| 8 | 0.010 |

**Table 1 Pheromone Table at Node 6**

In Table 1 next hop can be node 4 because it has highest probability.

**Parameter Initialization-**The following parameters should be populated for simulation:-

- ACO Mode ( On/Off) - Switches the algorithm on and off
- Simulation speed - 1 tick p/s - 1,000 tick p/s (or as near speed as the system can run)
- Total calls to make - Number of completed calls before simulation termination
- Maximum concurrent calls - Number of calls allowed at the same time
- Node capacity - The number of calls a node can route at once
- Call duration - The length (in ticks) of a call
- Remove Loop (Checked, Unchecked)-Removes the problem of laying pheromone trail in a cycle.

## IV. SIMULATION ALGORITHM

The simulator imitates behaviour of two algorithms. ACO based algorithm and conventional routing algorithm. Here the overall working of simulator is briefly described in an algorithmic notion. The algorithm described below presents high level design of simulator.

**Step1.** Initialize all simulator parameters whose maximum limits are stored in global module. The main parameter is number of calls to be made is provided as starting point to simulator. (i.e. 100, 200, 500 etc.).Given number of calls are generated with random source and destination.

**Step2.** If ACOMODE = = TRUE, then ACO based algorithm [From Main form Module] gets executed and calls routing take place on the basis of updated pheromone table.

Else

Calls routing take place on the basis of available pheromone table.

End if

[Mention the algorithm for which you want to simulator call by On/Off the ACO mode].

**Step3.** Result (Avg. no. of hops taken between source and destination to complete calls) produced by selected algorithm for the current simulation is returned to drawpanel module.

**Step4.** The selected algorithm is run until all calls get completed. Then a graph is plotted between avg no. of hops taken and no. of calls made using dot net charting feature of dot net platform.

**Step5.** Exit.

**4.1 PROPOSED ACO ALGORITHM**

This section describes ACO algorithm in detail. Firstly, we will explain the intelligent agents that will be used in the algorithm.

**Forward Ants-**Each node s periodically sends a forward ant Fs→d to a randomly chosen destination node d throughout the network. The task of the forward ant is to discover a feasible, low-cost path to the destination and to keep track of visited nodes. Every ant will represent a package sent from s to d through the network and each forward ant contains the following fields; source id and destination id and stack memory. When destination is reached then forward ant is transformed into backward ant and transfers its stack to backward ant.

**Backward Ants-**The backward ant Bd→s uses the memory from the forward ant. The task of the backward ant is to go back to the source node s along the same path as the forward ant but in the opposite direction and to update the routing tables on this path. When source node is reached then backward ant get die.

***Steps for ACO Algorithm***

**W**hen a call is to be made in network then before placing a call on network, ants are launched throughout the network which does not only select the shortest as well as less congested path, update the pheromone table (ant routing table), the higher the probability of pheromone, the most probable shortest path.

Then calls are routing according to pheromone table and hence less number of call failures as well as calls are routed through shortest path. The load balancing is achieved in the way. The algorithm each ant follows is below:-

ACO based algo (source, destination)

**Step1.** Forward ant is created and begins to explore routes to destination.

**Step2.** Forward ant is launched to connected neighbour nodes. And select the next node on the basis of pheromone probability.

**Step3.** Initialize stack with source node;

**[** stack is a memory to hold statistics of visited nodes]

**Step4.** If a cycle is detected, the cycle node are popped from ant's stack.

**Step5.** If current node == Destination, then generate backward ant and transfer stack to it.

goto step6.

Else

current node is pushed to stack and forward ant is further launched to connected nodes.

go to step4.

Endif

**Step6.** Backward ant pop the nodes form stack maintained by forward ant and move to next node towards source.

**Step7.** If current node = = source, then backward ant die and go to step8.

Else

update the pheromone table at current node and go to step 6.

**Step8.** Exit.

**4.1.1 Algorithm for Pheromone Table Updation**

**Step1.** [Initialize] Generate the value for variable new_value using random number generator.

**Step2.** [Sum] Add probabilities of all possible links. Store the sum in variable Total.

**Step3.** [Add the new_value to probability of selected node.]

$^{P}$selectednode =$^{P}$selectednode +new_value

**Step4.** [Sum] Add probabilities of all possible links after updating probability of selected node. Store the sum in variable New_Total.

**Step5.** Calculate ratio r = New _Total / Total.

**Step6.** Multiply each entry in pheromone table by r.

**Step7.** Stop.

**4.1.2Algorithm for Next Node Selection**

**Step1.** [Sum]

Calculate sum(s) of all connected neighbour node probability for a node K.

**Step2.** [Select]

Generate random number r form the internal (0, s).

**Step3.** [Loop]

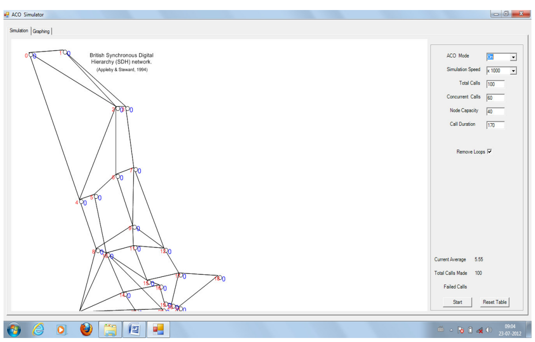
Go through the neighbour node and sum the probability form 0. When value of sum exceeds the value of r the procedure stops and current node become the selected one.

## V. RESULT ANALYSIS

The following tests will illustrate how the ACO based algorithm affects the routing of calls in telecommunication networks. These tests will show the effectiveness of the algorithm against the system running without ACO mode. Since it is possible to switch nodes on and off, a number of test comparisons will be done to show how ACO based algorithm can improve the routing of a network when paths are no longer valid and new routes have to be chosen. The value of simulator parameters is provided on the main form of ACO simulator.

**Test 1: Number of calls to be made 100**

Simulation of 100 calls using ACO mode




Simulation of 100 calls using non ACO mode

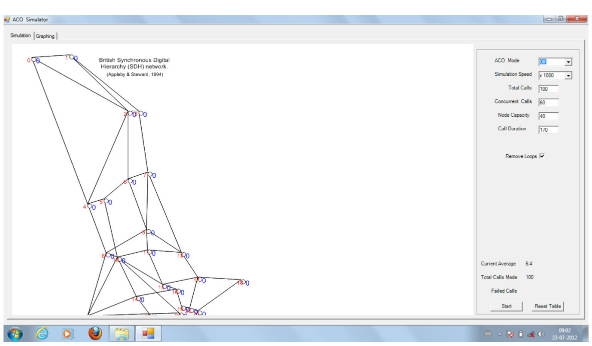
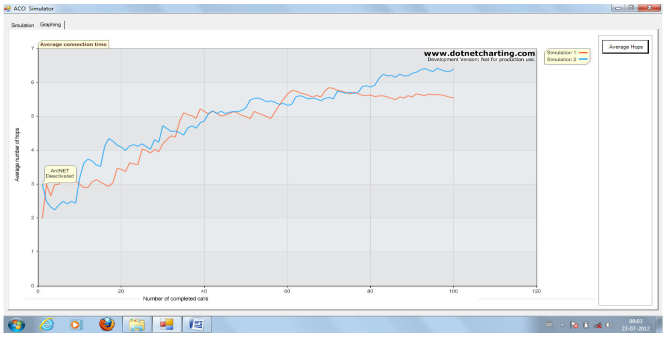



Chart for simulation of 100 calls




The first test contains two simulations:

- Simulation 1 (**Orange**) - ACO mode ON
- Simulation 2 (**Blue**) – ACO mode OFF

From this simulation, it is clear that even by the first 100 calls completed; ACO mode has reduced the average number of hops by approximately 1.5 nodes. At the end of the simulation optimal route is obtained, resulting in ACO improving network performance by almost 1.5 hops.

## VI. CONCLUSION AND FUTURE WORK

**Conclusion:** As it can be seen that that present telecommunication networks suffer from network congestion which causes calls, put on the network, to fail. Better novel algorithms are required to minimize the effect of congestion. In this research, the use of Ant Colony Optimization with network performance analysis has been considered and the performance for a telecommunication network in term of throughput and bandwidth consumption has been presented.

The new simulator algorithm depicts the relationships between network call routing and ACO by combining the benefit of ACO algorithm, i.e., selecting the most suitable paths in the network, with less congestion and a more stable network connection. ACO algorithm is designed to achieve high performance in a network.

Simulation results demonstrated that the proposed ACO algorithm could significantly improve the performance of the network in term of, throughput, number of calls failures and bandwidth consumption.

First, it has been shown that numbers of hops needed to complete the fixed number of calls get reduced by 1.5 hops if ACO was using as routing mode. That leads to much more improvement in the throughput, than the traditional routing protocols in telecommunication networks.

Additionally, there was gain in the bandwidth consumption by adopting ACO with calls routing by comparison with traditional routing algorithm. Since as the number of fixed calls is increased, then even average number of hops consumed remains constant.

Furthermore, the proposed algorithm that indicate of combining network calls routing with ant colony optimization can validate load balancing as well by reducing the number of call failures by 4% than the traditional schemes. Hence ant colony optimization technique performs better that of conventional routing algorithms.

**FUTURE WORK:** One possible direction for future work is to adopt the proposed ACO algorithm in wireless Ad-hoc based networks. In Ad-hoc networks the receivers, as well as the intermediate nodes, can connect anywhere, anytime and still stay connected with the rest of group of nodes Ad- hoc is an autonomously self-organised network that does not have fixed communications.

Another captivating possibility is to use ‘probabilistic routing’ of calls during peak hours. Here routes of calls, or perhaps a proportion of calls, would not be chosen according to the largest probabilities in the pheromone tables, but randomly according to these probabilities.

A mechanism that is assumed not to be used by natural ants, but could be useful here, is laying ‘anti-pheromone’. One could let ants directly decrease probabilities in the pheromone tables in particular circumstances, rather than increase them.

The other idea for future work is as ACO algorithm for routing does not make use of information bootstrapping, but rely on pure ant’s memory. On the other hand, in quasi-stationary situations it might be effective to make use of also some form of information bootstrapping. Therefore, proposed ACO algorithm, in the sense that pheromone estimates are updated by using both the ant travelling time and the estimates coming from the other nodes along the path, and that are carried by the backward ant.

Another possible area for future work is this proposed algorithm could be implemented to additional protocols using different meta-heuristics like particle swarm optimization (PSO) that is a population-based stochastic method for solving uninterrupted and distinct optimization problems. In particle swarm optimization, basics software agents, called particles, transfer in the search area of an optimization problem. The position of a particle depicts a candidate solution to the optimization problem at hand. Each particle investigate for proper location in the optimization area by replacing its direction according to rules originally motivate by behavioural of bird gathering that used to solve computational optimization problem.

Extending ant-like algorithms to situations like personnel recruitment, autonomous office which are not found in nature will also increase our understanding of the abstract and general abilities of such algorithms over and above those applications found in nature. This increase of knowledge will in turn support and inform biologists studying social insects.

## REFERENCES

[1] System Simulation with Digital Computer: Books by Nar Singh Deon

[2]http://web.cs.mun.ca/~donald/msc/node84.html#Shan7

[3]http://web.cs.mun.ca/~donald/msc/node84.html#Zeig9

[4]http://web.cs.mun.ca/~donald/msc/node84.html#Gajs8

[5].[ Angus, D. , 2006] "Niching for Population-based ant Colony Optimization" 2nd International IEEE Conference on e-Science and Grid Computing, Workshop on biologically-inspired Optimisation Methods for Parallel and Distributed Architectures: Algorithms, Systems and Applications.

[6] [Bonabeau et al., 2000] E. Bonabeau, M. Dorigo, and G. Theraulazß. *Inspiration for Optimization from Social Insect Behaviour*. Nature, vol. 406, 2000.

[7] J.-L. Deneubourg, S. Aron, S. Goss, and J. M.Pasteels. The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3:159–168, 1990.

[8] Beckers, R., Holland, O.E., & Deneubourg, J.L. (1994). From local actions to global tasks:stigmergy and Collective Robotics. In R.A. Brooks, & P. Maes(Eds.) *Artificial Life IV*, Cambridge, MIT Press. p. 181-189.

[9] [Dorigo et al., 1991] M. Dorigo, V. Maniezzo, and A. Colorni. Positive Feedback as a Search Strategy.t*echnical Report 91-016*, Dipartimento di Elettronica, Politecnico di Milano, Italy,1991.

[10][Dorigo et al., 1991a] M. Dorigo, V. Maniezzo, and A. Colorni. Ant System: An Autocatalytic Optimizing Process. *Technical Report 91-016*, *Dipartimento di Elettronica*,Politechnico di Milano, Italy, 1991.

[11] [Dorigo et al., 1996] M. Dorigo, V. Maniezzo, and A. Colorni. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, andCybernetics – Part B*, 26(1), pages 29–41, 1996.

[12] [Blum and Roli, 2003] C. Blum and A. Roli. Metaheuristics in Combinatorial Optimization:Overview and Conceptual Comparison. *ACM Computing Surveys*, Vol. 35, No. 3, pages 268–308, 2003.

[13] [Dorigo and Stützle, 2003] M. Dorigo and T. Stützle. The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances. In F. Glover and G. Kochenberger,(Eds.), *Handbook of Metaheuristics*, Kluwer Academic Publishers, pages 251-285, 2003.

[14] [Metaheuristics Network Website, 2000] *Metaheuristics Network Website 2000*. http://www.metaheuristics.net.

[15] J. Pan and Wang, “An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem”, *Innovative Computing, Information and Control, 2006. ICICIC '06. First International Conference,* vol.1, no., pp.210-213, Aug. 2006.

[16] D. Merkle, M. Middendorf and H. Schmeck, “Ant colony optimization for resource-constrained project scheduling”, *Proceedings of the Genetic and Evolutionary Computation Conference*, pp.893-900, 2000.

[17] C. Blum, “ACO applied to group shop scheduling: a case study on intensification and diversification”, *Proceedings of ANTS*, *Lecture Notes in Computer Science*, vol. 2463, pp.14-27, 2002.

[18] T. Stützle, “MAX-MIN Ant System for the quadratic assignment problem”, *Technical Report AIDA-97-4, FB Informatik, TU Darmstadt, Germany*, 1997.

[19] R. Lourenço and D. Serra “Adaptive search heuristics for the generalized assignment problem”, *Mathware & soft computing,* vol.9, no.2-3, 2002.

[20] K. I. Aardal, S. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino and Antonio. Sassano, “Models and solution techniques for the frequency assignment problem”, *A Quarterly Journal of Operations Research*, vol.1, no.4, pp.261-317, 2001.

[21] Y. C. Liang and A. E. Smith, “An ant colony optimization algorithm for the redundancy allocation problem (RAP)”, *IEEE Transactions on Reliability*, vol.53, no.3, pp.417-423, 2004.

[22] Y. Bin, and Y. Zhen, "An ant colony optimization model: The period vehicle routing problem with time windows", *Elsevier Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 2, pp. 166-181, 2011.

[23] E. Angelelli and M. G. Speranza, "The periodic vehicle routing problem with intermediate facilities", *European Journal for Operations Research*, vol.137, no.2, pp.233-247, 2002.

[24] S. C. Ho and D. Haugland, "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries", *Computers & Operations Research*, vol.31, no.12, pp.1947-1964, 2004.

[25] W. P. Nanry and J. W. Barnes, "Solving the pickup and delivery problem with time windows using reactive tabu search", *Transportation Research Part B*, vol.34, no. 2, pp.107-121, 2000.

[26] S. C. Hong and Y. B. Park, "A heuristic for bi-objective vehicle routing with time window constraints", *International Journal of Production Economics*, vol.62, no.3, pp.249-258, 1999.

[27] R. Hadji, M. Rahoual, E. Talbi and V. Bachelet "Ant colonies for the set covering problem", *Abstract proceedings of ANTS*, pp.63-66, 2000.

[28] V Maniezzo and M Milandri, "An ant-based framework for very strongly constrained problems*", Proceedings of ANT*, pp.222-227, 2002.

[29] G. Leguizamon, Z. Michalewicz and Martin Schutz, "An ant system for the maximum independent set problem", *Proceedings of the Argentinian Congress on Computer Science*, vol.2, pp.1027-1040, 2001.

[30] R. S. Parpinelli, H. S. Lopes and A. A Freitas, "Data mining with an ant colony optimization algorithm", *IEEE Transactionon Evolutionary Computation*, vol.6, no.4, pp.321-332, 2002.

[31] Xiao. M.Hu, J. ZHANG, and H. Chung, "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, pp. 659-669, 2009.

[32] L. Wang and Q. D. Wu, "Linear system parameters identification based on ant system algorithm",