Evaluation of an Ant Colony Optimization Algorithm for Routing in MANETs

A thesis presented
by

Eseosa Stephanie Osagie

to
The Department of Computer Science
in partial fulfillment of the requirements
for the degree of
Master of Science
in the subject of

Computer Science

The University of Manitoba
Winnipeg, Manitoba
August 2007

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

Evaluation of an Ant Colony Optimization Algorithm for Routing in MANETs

BY

Eseosa Stephanie Osagie

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

MASTER OF SCIENCE

Abstract

Mobile Ad Hoc Networks (MANETS) have recently gained a lot of popularity amongst computer scientists and engineer. They are infrastructureless networks consisting of mobile nodes, with constantly changing topologies, that communicate via a wireless medium. Several methods have been proposed as solutions to the challenges associated with MANET routing particularly, more recently, the use of nature inspired algorithms. This thesis presents an improved approach, called PACONET, for routing in MANETs inspired by the Ant Colony Optimization (ACO) algorithm. ACO is a nature inspired algorithm mobile on by the foraging behaviour of ants. Ants have been found to be able to find the shortest path from their nest to food over time using very adaptive means. The performance of my routing algorithm is evaluated using simulation and is compared to an existing well known MANET routing algorithm, the Ad hoc On-Demand Distance Vector (AODV) protocol. Several performance metrics are considered in different scenarios with varying mobility levels and traffic load. The evaluation identifies situations for which the algorithm is well suited and offers direction for possible improvements.

Contents

List of Figures

Acknowledgments

I would like to begin by thanking my advisors, Dr. Parimala Thulasiraman and Dr. Ruppa K. Thulasiram for their guidiance and direction. I would also like to extend my heartfelt appreciation to my committee members.

I am deeply grateful to my family for their love and support along the way. Especially, my parents Prof and Mrs A. U Osagie, for their financial support making my studies possible.

I wish to thank my friends and collegues particularly Monirul Hasan and Jianping Wang for their many valauble suggestions that contributed to the success of this thesis.

Many thanks to David Edosa for words of encouragement and advice and a listening ear. Finally, I wish to express my loving thanks to my fiancé Ben Idiodemise.

This thesis is dedicated to God Almighty, for without Him nothing is.
Also to my wonderful parents, my caring brother, my loving fiancé, Ben
and my dear friend, David.

Chapter 1

Introduction

For many years now, nature has been a source of inspiration in the development of algorithms for solutions to combinatorial optimization problems. Combinatorial optimization algorithms are concerned with problems that are NP-hard, where there is no satisfactory heuristic to solve such problems. Metaheuristics (or 'beyond heuristics') is one way of solving such problems. There are many metaheuristics such as genetic algorithms, simulated annealing, and tabu search, to name a few. They offer solutions to a general class of computational problems. These algorithms demonstrate adaptive, robust and effective behaviours as observed in nature; where adaptive means that each technique improves its goal-achieving competence over time, robust means the techniques are is flexible and never completely breaks down while effective means they eventually find a satisfactory solution [1].

Recently, Swarm Intelligence (SI) (a class of nature inspired algorithms) has been used [43] to solve combinatorial optimization problems. The idea of swarm intelligence [15] is to design algorithms inspired by the collective behaviour of insects such as bees [48], termites [40], ants [14] and other animal societies that exist as decentralized, self-organized systems [15]. These insects live in a hostile, dynamic environment and co-ordinate and co-operate to survive. They communicate directly with one another or indirectly through the environment to accomplish tasks such as finding food.

Ant Colony Optimization (ACO) is one swarm intelligence technique inspired by the foraging behaviour of ants. It is a metaheuristic optimization algorithm in which a colony of artificial ants cooperate to find good solutions to difficult discrete optimization problems [14].

The observation and study of ant societies is now no longer attractive only to entomologists. Computer scientists and engineers have taken a keen interest in the study of their interaction. Ant societies, among other societies have fully distributed control, maintain autonomy of individuals, have collective and cooperative strategies and have an emergence of complex behaviors with respect to the single ant and self-organization. The simultaneous presence of these unique characteristics has made ant societies an attractive and inspiring model for building new algorithms and new multi-agent systems.

In nature, ants communicate using a stigmergic process (indirect communication by modifying the environment) . This sort of local interaction results in collective success in processes such as feeding. Ants are always able to determine the shortest path from their nest to food by following the trails they create using a chemical substance known as *pheromone*. In the ACO concept, artificial ants use virtual pheromone to update their path through a decision graph, (i.e. the path that reflects which alternative path an ant chooses at certain points). Ants of later iterations use the pheromone marks of previous ants as a means of orientation when constructing their own solutions, which ultimately result in focusing the ants on promising parts of the search space [1].

Ant algorithms are highly adaptive, robust and decentralised in nature which makes them well suited for several applications. This problem-solving method has also been used recently to solve many computational problems, particularly problems in wired (e.g [8]) and wireless networks (e.g [9]).

As the Internet has become increasingly popular, the need for flexibility while remaining connected has also grown. This has lead to the development of wireless networking options particularly Mobile Ad Hoc Networks. Simply put, a Mobile Ad Hoc Network is a network with nodes that are mobile and communicate via wireless connections [9] among themselves without the use of wired infratructures such as wireless access points. The nodes may join or leave the network at any time, there is no fixed topology. These networks have many application such as in disaster relief area where the existing infrastructure may have been damaged or whereever networks need to be formed on the fly.

The transmission range in MANETs is usually limited, so nodes within a network need to relay data packets over several hops to communicate. Thus, nodes in MANETs act as both hosts and routers. Since the nodes are mobile, designing an effective routing technique is a significant challenge. A routing technique that is appropriate for MANETs needs to be flexible enough to adapt to arbitrarily changing network topologies, and to support efficient bandwidth and energy management, since low-powered batteries operate the nodes.

Several approaches for routing in MANETS have been proposed. Each approach deals with the dynamic aspects of MANETs in its own way. These approaches can be classified as either reactive, proactive or hybrid and are discussed in more detail in Chapter 2. All MANET routing algorithm share the same basic idea. Each approach attempts to obtain an optimal path by acquiring routing information about various paths available using control packets, using ACO for MANET routing, control packets correspond to ants. The nodes in the network send out ants (the frequency, process and number varies for each approach) in search of a path to a specific destination. As the ants travel the network they store the routing information in a table maintained by the nodes. The information stored is usually with respect to pheromone concentration. The optimal paths over time are determined by those with higher concentration. The ACO technique has been applied to routing in MANETs and has been shown in several cases (e.g. [9, 18], etc) to be a good technique.

In this thesis, I present an improved ACO algorithm, called PACONET, for routing in MANETs inspired by the work done by Islam [23]. This algorithm, unlike earlier work, focuses on the efficiency and effectiveness of the approach as a solution to the routing problem and uses a simulated ad hoc network environment. Islam's work studied the parallel implementation of the ACO algorithm as a solution to an irregular application such as MANETs. The algorithm was implemented in shared and distributed memory processors and he focused on parallel computing performance issues(e.g Speedup).

Goal of thesis

PACONET as earlier mentioned is based on Islam's Source Update algorithm [23] . A number of improvements were made to the Source Update algorithm inorder to create PACONET. These improvements are listed below:

- Node mobility is incorporated
- Dynamic route creation and maintenance in routing tables
- Route failure handling

The performance of the algorithm is evaluated using simulation with comparison to another existing MANET algorithm. Routing protocol performance issues such as end-to-end delay, packet delivery ratio and routing control overhead are also considered. The PACONET algorithm is explained in more detail in Chapter 4.

The rest of this thesis is organized as follows. In the next two chapters, the background and related work in ad hoc networks and ACO are discussed, respectively. In Chapter 4, the improved ACO algorithm is presented. The implementation, performance evaluation and results are described in Chapter 5. Conclusions are presented in Chapter 6 and in Chapter 7 I discuss possible future work.

Chapter 2

Ad Hoc Networks

A Mobile Ad Hoc Network (MANET) is a set of dynamic, mobile, self-organising nodes. Since the nodes have no fixed topology, they are also known as infrastuctureless networks [41]. These networks are formed on the fly and usually consist of heterogeneous nodes capable of independent communication. Devices such as laptop computers, personal digital assistants and mobile phones could be nodes in an ad hoc network. The responsibilities of organizing and controlling the network are distributed amongst the nodes. Interest and use of wireless mobile networks has been growing over the last few years. This can be attributed to the growing popularity of the internet and the increased need of people to be more flexible, free from wires and yet still remain connected to the network. Figure 2.1 is an example of a simple ad hoc network made up of three devices.

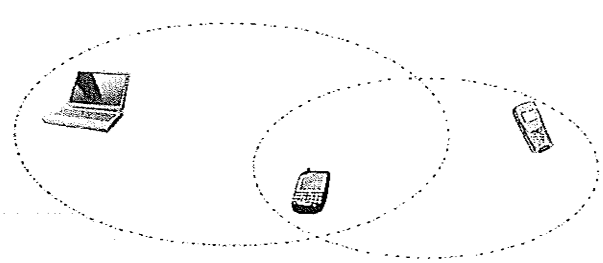

Figure 2.1: An example of an ad hoc network

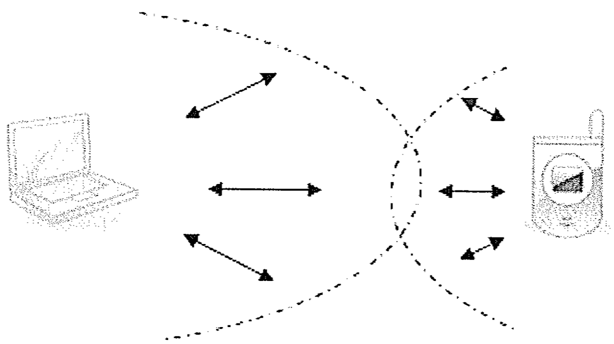

Figure 2.2: Limited Transmission between mobile computing devices

MANETs are usually characterised by limited transmission range because they are operated by limited battery powered devices. Figure 2.2 is an illustration of the limited transmission range between devices. To establish data communication within the network, nodes may need to relay data packets over several intermediate nodes. Therefore, nodes in MANETs operate as both hosts and routers. Take the example given in Figure 2.1. For the cellphone (on the right) to transmit information to the laptop (on the left), it would have to be transmitted via the personal digital assistant (in the middle), resulting in a multi-hop process(two hops).

When proposing techniques for use in these ad hoc networks such as developing a routing protocol, the features of the network have to be considered to ensure efficient solutions. Routing in MANETs is particularly challenging due to the constantly changing topology which can make paths, which were at one point the best, inefficient. Also, the lower bandwidth capacity and scarce energy resources affect the design of a suitable routing protocol. It is therefore, important to design a protocol that is adaptive, robust, self-healing and able to work in a localised way

2.1 MANET Routing Protocols

Several protocols have been proposed for the routing problem in MANETs. These protocols are usually categorized according to their design approaches as *proactive*, *reactive* or *hybrid* [47]. A proactive (table-driven) protocol is one in which the routing information is maintained for all the paths from source to destination, whether in use or not. In a reactive (on-demand) protocol only routes currently in use are maintained and they are created only when needed. Hybrid protocols, attempt to, fuse proactive and reactive routing to achieve better performance.

2.1.1 Proactive Routing

Proactive protocols continuously maintain up-to-date routes to every node in the network by periodically transmitting routing information throughout the network. Thus, transmission occurs without delay since a route is already known but the periodic updates result in more overhead and these protocols are not very scalable. Some well known proactive routing examples are briefly explained below.

Destination Sequenced Distance Vector (DSDV)

The most widely known example is the Destination Sequenced Distance Vector routing protocol (DSDV) [38]. The DSDV protocol is based on the Bellman-Ford [5] routing algorithm. In DSDV every node in the network maintains a routing table in which all of the possible destinations within that network and the number of hops to each destination are recorded. Each entry has a sequence number associated with it to indicate the most recent routes and to avoid forming routing loops. The routing table is updated periodically and as needed. Each new route broadcast contains the address of the destination, the number of hops to reach the destination, the sequence number of the information received regarding the destination, as well as a new sequence number unique to the broadcast. The route labeled with the most recent sequence number is always used. The drawback of DSDV is that it requires flooding which could result in wasted bandwidth.

Global State Routing (GSR)

The Global State Routing (GSR) protocol [10] improves upon DSDV by avoiding excessive flooding of packets using the link state (LS) routing approach. In the LS routing method, each node floods the link state information directly into the whole network (global flooding) once a link change between itself and any of its neighbors is detected. A node gets to know the whole topology by obtaining link information. Link state routing works well in static topology networks. The disadvantage of GSR is the large number of update messages that need to be sent to keep the routing table current.

Fisheye State Routing (FSR)

The Fisheye State Routing (FSR) [37] is derived from GSR. Instead of sending messages between all nodes as in GSR, it limits the amount of message passing by constraining messaging to closer nodes. That is, nodes more closely placed are considered more frequently for updates than those further away. Figure 2.3 illustrates the application of fisheye in a mobile, wireless network. The circles with different shades of grey define the fisheye scopes with respect to the center node (node 11). The scope is defined as the set of nodes that can be reached within a given number of hops. FSR handles mobility very well and destination nodes can easily be located within the network but FSR is not very scalable and the routing table storage is complex.

WRP

Wireless routing protocol (WRP) [34] solves some of the drawbacks of the protocls already mentioned and has a faster convergence in finding shortest paths between nodes. Each node in a WRP network maintains four tables; Distance table, Routing table, Link-cost table and Message Retransmission List (MRL) table. Nodes keep track of link changes by sending update messages only between neighbouring nodes. These update messages contain a list of updates (the destination, the distance to the destination, and the predecessor of the destination) and a list of responses indicating which nodes should acknowledge (ACK) the update. The messages are sent after processing updates from neighbors or detecting a change in a link to a neighbor. The nodes respond by updating their distance tables. If a node is not sending update messages, it sends out hello messages at regular intervals to ensure connectivity

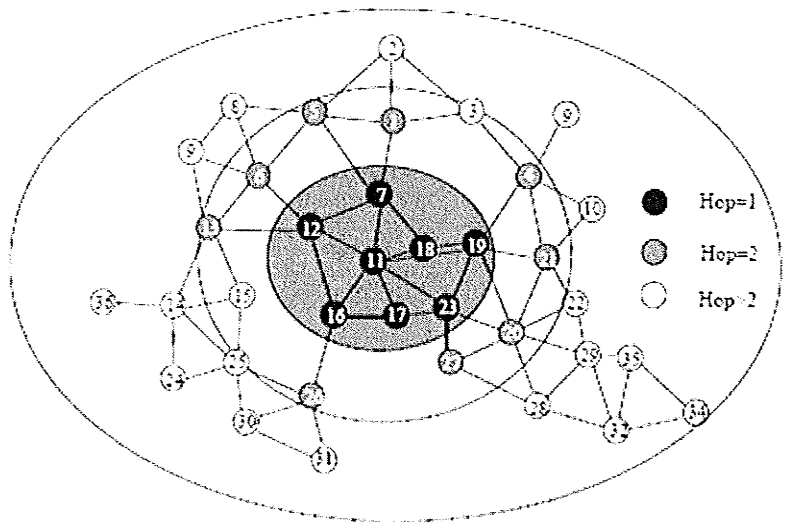

Figure 2.3: Scope of FSR [37]

to its neighbours. WRP achieves loop freedom by forcing each node to perform a consistency check of predecessor information reported by all its neighbors.

OLSR

The Optimised Link State Routing(OLSR) algorithm employs link state routing but reduces the overhead by using a mulitpath relaying (MPR) strategy. In this strategy, when a node wants to send topology updates, it selects a group of neighbouring nodes to retransmit the routing packets. These nodes are called the multipoint relays of the source node. A node is selected as a multipoint relay node if it is a one-hop neighbour able to reach two-hop neighbours. OLSR is able to determine optimal routes in terms of hop count but may experience increase overhead due to frequent two-hop hello messages.

Clusterhead Gateway Switch Routing (CGSR)

The Clusterhead Gateway Switch Routing (CGSR) [11] algorithm consists of a clustering scheme, called Least Cluster Change, which forms clusters and elects clusterheads using either lowest ID or maximum links. It also consists of gateway nodes, they are nodes, that are within more than one cluster, and therefore facilitate inter-cluster communication. The protocol uses a sequence number scheme to gain loop-free routes and avoid stale routing entries. In CGSR, a packet is routed alternating between clusterheads and gateways. Each node maintains two tables: a routing table with the distance vectors to other nodes, and a cluster membership table. Maintaining the cluster membership table takes up alot of bandwidth but its routing table size appears smaller when compared to other distance vector protocols.

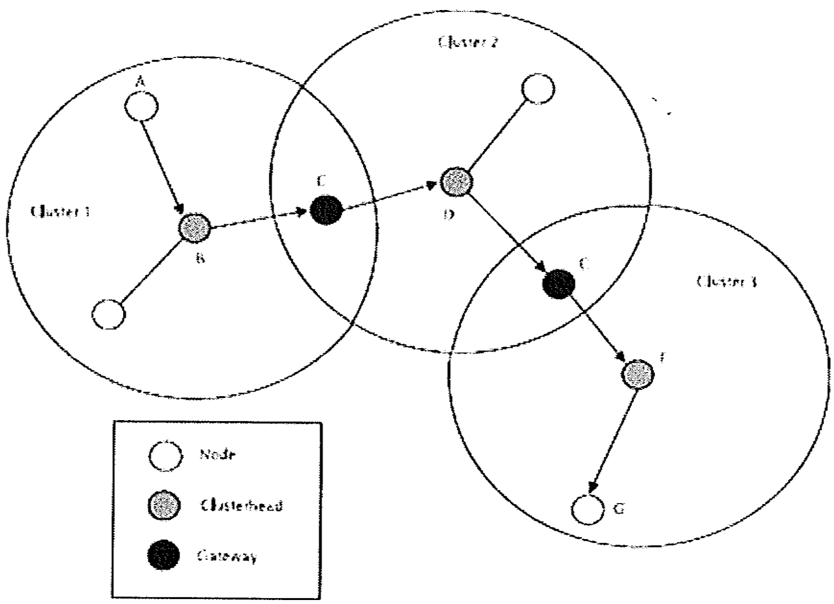

Figure 2.4: CGSR routing example [29]

2.1.2 Reactive Routing

A reactive routing protocol only searches for a route when it wishes to send data to a particular node. The route discovery is initiated by the source node and the route is maintained only by the nodes that form it. These protocols experience reduced overhead making them more scalable but they may also experience transmission delays due to the neccesity to search for a route when none exists.

AODV

The Ad hoc On-Demand Distance Vector (AODV) [39] protocol is an example of a reactive protocol. AODV improves on DSDV by creating routes on demand thus reducing the number of required broadcasts. In AODV, when a source node wants to communicate with a destination for which it has no valid route, it initiates a path discovery process to locate the destination node. It broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a recent route to the destination is located. AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. A route reply (RREP) packet is sent back to the node from which the RREQ originated, following the reverse path of the RREQ packet. If the source node moves, it reinitiates a route discovery. Routes are maintained by every node keeping track of its neighbours along a particular route and the use of periodic broadcasts (hello messages) by a node to inform other nodes of its neighbourhood. AODV allows both unicast and multicast routing but packets may experience some delay.

Dynamic Source Routing (DSR)

In Dynamic Source Routing (DSR) [24] the sender of the packet determines the sequence of nodes the packet should be routed through. Each node maintains caches of source routes to other nodes in the network. The protocol consists of two major phases: route discovery and route maintenance. A node wishing to communicate with some other node will first check its route cache for a recent route to the destination node. If a route exists, it uses it to send data. Where no route exists it initiates a route discovery by broadcasting a route request packet. The route request packet keeps a route record of all intermediate nodes that forward it if they have no routing information for its destination. If the intermediate node has routing information for the packet destination, it appends its routing information to it and sends it back to the source. However, if it is the actual destination, a route reply is generated with the route record of the route request and it is sent to the source.

Cluster Based Routing Protocol (CBRP)

Cluster Based Routing Protocol (CBRP) [33] groups nodes into clusters and appoints cluster-heads in each cluster. The cluster-heads communicate to find an efficient route from a source to a destination. The way a cluster-head is elected is an important issue in CBRP. When a node joins a network, it sends out hello messages and if it receives no reply from an existing cluster-head, it makes itself the cluster-head after it has obtained a bi-directional link to one or more neighbour nodes. A source node is able to find a route to a destination by sending route requests to neighbouring cluster-heads. If the destination node is not in the cluster then the cluster-head forwards the route request to other adjacent cluster-heads. When the destination receives the request, it sends a reply back to the source with the route recorded in the request packet as it traveled along the network. Packet routing is actually done using source routing but it incorporates route shortening. That is, when a node receives the reply from the destination to the source, it tries to find the farthest node in the route that is its neighbour. Thus, the route between source and destination can be shortened.

Temporally-Ordered Routing Algorithm (TORA)

The Temporally-Ordered Routing Algorithm (TORA) [36] is a scalable, distributed, source-routing protocol. The algorithm is based on the concept of link reversal and is able to provide multiple routes for any desired source/destination pair. It also uses the concept of localization of control messages by limiting them only to one-hop neighbours (adjacent nodes). The protocol performs three basic functions: route creation, route maintenance and route erasure. In the route creation phase, nodes use a 'height' metric to establish a directed acyclic graph (DAG) rooted at the destination. This DAG may be broken due to the mobility of nodes. Therefore, in the maintenance phase, the protocol attempts to reestablish and recreate a DAG rooted at the destination. The route erasure phase essentially involves flooding a broadcast clear packet (CLR) throughout the network to erase invalid routes. In some cases TORA may experience a temporary problem of instability due to oscillations similar to the 'count-to-infinity' problem in distance-vector routing protocols.

2.1.3 Hybrid Routing

Hybrid protocols combine the features of both proactive and reactive routing. In hybrid protocols, nodes typically exhibit proactive behaviour within their zones (group of neighbouring nodes) and reactive behaviour between zones.

Zone routing Protocols (ZPR)

Zone routing Protocols (ZPR) [19] as the name implies, are based on the concept of zones. A zone is measured in hop counts between the nodes of a network. A node can be part of several zones. Routing information is proactively maintained within each zone while routing between nodes in different zones is done reactively. ZRP refers to the local proactive routing component as the IntrA-zone Routing Protocol (IARP). The global reactive routing component is named the IntEr-zone Routing Protocol (IERP). Instead of broadcasting packets, ZRP uses the bordercasting concept which utilizes the topology information provided by IARP to direct query requests only to the border nodes of the zone. The bordercast packet delivery service is provided by the Bordercast Resolution Protocol (BRP). ZRP is actually a framework where any existing efficient proactive or reactive algorithm can be used as the IARP and IERP components respectively. A few papers have been written that analyse ZRP's performance and it has been shown to avoid excess bandwidth as in strict proactive routing. Also, long route request delays and network flooding as in purely reactive routing are avoided.

VBR

Liang and Haas [26] propose a protocol for ad hoc network routing with a 'Dynamic Virtual Backbone'. The Virtual Backbone routing (VBR) scheme is a hybrid routing framework that combines the philosophy of the Zone Routing Protocol [19] with hierarchical routing. VBR limits the proactive component to the local routing zones only, as in ZRP, while the reactive component restricts route queries to within the virtual backbone. It uses the 'Distributed Database Coverage Heuristic' (DDCH) for backbone generation and maintenance and also for dynamic local zone construction and maintenance as the network topology changes. A node can join or leave the VB at any time depending on the movement of nodes and changes in the link topology. A VB node serves as a database queried by source nodes for the link-state information of nodes within the zone.

When a node wants to communicate with another node outside its zone, it sends a query to its nearest database (VB node). If it finds no route in the database, it broadcasts the route query to all other databases through the virtual links between the VB nodes. After receiving the route query, a database whose routing zone contains the destination, sends a route reply, thorough the reverse VB path, back to the query originating database. Each database along the way computes the best route segments that it can see within its zone; and appends them in the route reply packet.

Route queries are always directed to certain locations in the network, avoiding flooding in the network, which reduces the overall control traffic in the network. High node mobility in the network may pose a problem due to the operations that have to be performed to regenerate the VB. Also the reactive component tends to transmit a lot of control traffic.

Many of the conventional approaches mentioned above exhibit very good performance. But these protocols still face limitations due to their inability to easily adapt to very dynamic environments while maintaining localized control in the network and incoporating solutions to other factors such as link quality, traffic congestion etc. This has encouraged more research that has led to the use of nature inspired algorithms as a solution to the routing problem.

2.2 Summary

In this chapter, I defined what a mobile ad hoc network is and discussed a method of categorizing exsiting protocols as either reactive, proactive or hybrid. For each approach, I briefly reviewed some existing routing protocols.

Chapter 3

Ant Colony Optimization

Recently, several algorithms have been proposed as solutions to combinatorial optimization problems. These are NP-hard problems which are solved using approximate methods that return near-optimal solutions. The algorithms are thus heuristics, and have more recently been inspired by behaviours or processes in nature. A significant example of these algorithms is the field of 'Ant algorithms' inspired by several different aspects of the behaviour of ant colonies such as foraging, division of labor, brood sorting, and cooperative transport. In each case, ants coordinate their activities via *stigmergy*, a form of local indirect communication mediated by modifications of the environment [14].

In general, problem solving approaches that take their inspiration from nature (the social behaviour of insects and other animals) are termed as Swarm Intelligence(SI) [14]. Ant Colony Optimization(ACO) is a significant SI technique that has proved very successful and has been widely applied in the solution to static and dynamic problems. ACO has also been formalised into a metaheuristic. A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems [30].

Figure 3.1: Ants following a Pheromone trail

3.1 Ant Foraging

Several scientists such as Goss [17] have investigated the pheromone laying behaviour of ants. These studies have given insight into the foraging behaviour of ants showing that they are always able to determine, over time, the shortest path from their nest to a food source and also adapt easily to path disruptions that may occur.

This phenomena can be explained using Figures 3.1-3.4. As ants move along they deposit a chemical substance known as pheromone on their path. This substance evaporates over time. As more ants come along, they are attracted by the pheromone and continue along the trails of the previous ants. The ants are usually attracted towards a higher concentration of pheromones which enables them to choose shortest paths since these would retain a higher concentration each time than longer paths. This process of indirect communication is an example of stigmergy.

In Figure 3.1, the ants are seen to be moving in a straight line between their nest and food. This path is created by the ants following the trails created by preceeding ants. If an obstacle is placed on the path as in Figure 3.2, it would disrupt the existing pheromone trail. The ants then have to discover another path by randomly selecting to go either left or right as in Figure 3.3.

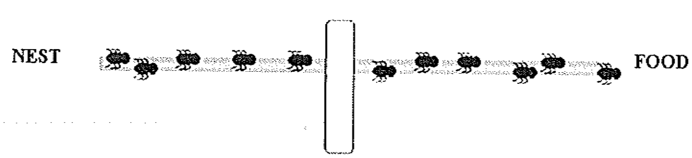

Figure 3.2: Obstructed Ant trail

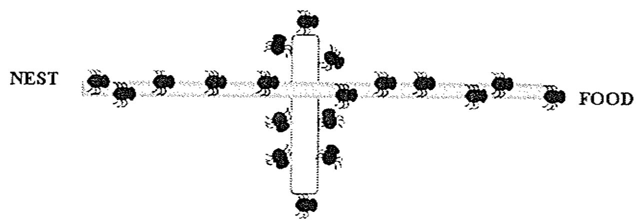

Figure 3.3: Alternative Path sampling

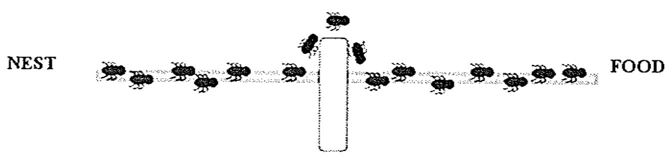

Figure 3.4: Adjusted Ant trail

The ants that choose the shorter path by chance will aquire a path between the nest and food that is faster than those taking the alternative path. This path will have a higher pheromone concentration and more ants will be attracted to it. Overtime all the ants will have chosen the shorter path, as in Figure 3.4

3.2 ACO Algorithms

Different ant colony optimization algorithms have been proposed and each has the same basic idea. The original ant colony optimization algorithm is known as 'Ant System' and was proposed by Dorigo, Maniezzo and Colorni. It was first applied as a solution to the traveling salesman problem (TSP)[12]. In the solution, an ant is placed on each city and it traverses from its current city, visiting other cities only once and returning to its origin when a tour is completed. The ants deposit pheromone on the edges connecting each city as they journey on. The pheromone concentration on the edges is constantly adjusted so that pheromone on unused edges eventually evaporate completely.

When Ant Sytem was first developed and applied as a solution to TSP, the results were encouraging but not competitive in comparison to other more established approaches to the TSP. There were also a few disadvantages exhibited by the algorithm such as the excessive computation time and suboptimal convergence time. Much work followed this and several variants of the Ant System were developed. The most successful variants are the MAX-MIN Ant System [45] and the Ant Colony system [13].

3.3 Applications of ACO

The ACO meta-heuristic has been applied to several combinatorial optimization problems and so far has proved useful. The overall result that emerges from these applications is that, for many problems, ACO algorithms produce results that are very close to those of the best-performing algorithms, while on some problems they are the state-of-the-art [30].

3.3.1 Applications to NP-hard problems

ACO algorithms have been applied to a number of diferent NP-hard problems ranging from routing problems to assignment problems and scheduling problems. In each case the ACO algorithm's perfomance is compared to existing techniques to determine its usefulness. For example, the ACO algorithm for TSP was compared to and shown to perform better than both simulated annealing and genetic algorithms [12].

Subsequent applications of the ACO method were in problems such as the quadratic assignment problem [28] and the vehicle routing problem [6]. In the quadratic assignment problem, the problem is assigning n facilities to n locations so that cost is minimized. The results obtained showed that ACO outperformed genetic algorithms but not simulated annealing. In the Vehicle routing Problem, the objective is to find minimum cost vehicle routes such that:

1. Every customer is visited exactly once by exactly one vehicle;
2. For every vehicle the total demand does not exceed the vehicle capacity;
3. The total tour length of each vehicle does not exceed a given limit;

4. Every vehicle starts and ends its tour at the same position.

The results showed better performance than simulated annealing.

ACO algorithms also showed reasonable performance with the Job-Shop Scheduling Problem and Graph-coloring problem [15]. In the Job-Shop Scheduling Problem a given set of machines and set of job operations must be assigned to time intervals in such a way that no two jobs are processed at the same time on the same machine and the maximum time of completion of all operations is minimized while in the Graph-Coloring Problem, the problem is finding a coloring of a graph so that the number of colors used is minimal [1]

3.3.2 Applications to telecommunication networks

More recently, the ACO technique has also been applied to routing in communication networks. The routing problem in communication networks is to build and maintain routing tables to achieve optimum performance on the network based on its type (wired or wireless) and its purpose [12].

The routing concept based on ACO as described in [9] involves the acquisition of routing information through sampling paths using control packets corresponding to ants. The nodes in the network may simultaneously and independently generate these ants. Their task is to test a path to a specified destination, collecting information about its quality (e.g. end-to-end delay, number of hops, etc.) while moving from source to destination and back, and updating the routing information for intermediate nodes on the way back. The routing table at each node contains entries for each neighbour, which are constantly updated. These entries reflect pheromone values on the corresponding link. The entries are a measure of goodness of going through the neighbour to the destination and are updated by quality values calculated by the ants. The ants in turn use the routing table at each node to stochastically determine the next hop in their path to the destination based on the pheromone values. The higher the pheromone concentration on the links, the higher the probability that it will be chosen. These artificial ants, analogous to natural ants, are autonomous agents which, through updates and stochastic following of the links in the pheromone based routing tables, participate in a stigmergic communication process. The routing of data packets is very similar to the way ants are routed. In some cases, data for a particular destination may be spread over multiple paths, with more packets sent over the best paths, resulting in load balancing. The exploratory nature of the ants also allows them to keep track of less good paths by occasional sampling which can serve as backup paths in the event of failure or sudden congestion. The features of ACO have been exploited for routing in ad hoc networks. The major challenges are the high rate of change in the network topology and the limited bandwidth that restricts the generation of ant packets.

The first attempt at using ACO in communication was done by Schoonderwoerd et al. [42] on the Ant Based Control (ABC) algorithm for a wired circuit switched network. They were able to optimize performance in the network by balancing the load in the network. This work was followed by the AntNet algorithm [8] introduced for adaptive routing in packet switching networks. These protocols paved the way for the use of ACO in mobile ad hoc networks.

3.3.3 ACO Inspired MANET Routing Algorithms

In this section, I discuss the literature on MANET routing algorithms related to ACO.

Global Positioning System ANT-Like Routing Algorithm (GPSAL)

Camara and Loureiro [7] describe a novel routing protocol called Global Positioning System ANT-Like Routing Algorithm (GPSAL) which uses a Global Positioning System(GPS) 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.

The GPSAL algorithm presumes that all mobile hosts in a MANET have a GPS unit which provides the host its approximate three-dimensional position (latitude, longitude and altitude). Every host in the MANET has a routing table where each entry represents a known host and has information about its current location, previous location, timestamp of current location, timestamp of previous location and whether the host is a mobile or fixed host.

A mobile host willing to join the MANET must listen to the medium for a node to which it can send a request packet for its routing table. When the new mobile node receives this information, it can start routing and sending packets in the MANET. The routing protocol is based on the physical location of the destination host. Where there exists an entry in the routing table for a host, the best possible route is chosen using a shortest path algorithm. The route, which consists of a list of nodes and the corresponding timestamps, is attached to the packet which is sent to the first host in the list. If the host is not found in the routing table, the mobile node sends a message to the nearest fixed node, if available, which tries to find the destination node. Otherwise, the data packet is not delivered. Each host, on receipt, compares the routing information in its routing table with the header of the received packet. The entries with older information than in the packet received are updated. Routing information can be obtained locally from a neighbor node that periodically broadcasts only the changes that have occurred since the last broadcast or globally, and more rapidly, using mobile software agents modeled on ants (agents sent to random nodes in the network).

When a mobile computer in a MANET is powered on, it may be unaware of the physical location (current or past) of other hosts, but, this information can be gathered when a computer receives a message to be forwarded to another node or as a result of its own route discovery. . The process of updating routing information is essential for good performance of the algorithm but there is an overhead associated with this process that can be controlled by the number of ants in the MANET. The possible options for determining where an ant should be sent to collect more up-to-date information are to choose the node with the oldest information in the routing table or the farthest known node in the MANET or just any node. GPSAL algorithm focuses on position-based routing and so does not consider the concentration of pheromone on any paths.

Ant-AODV

Marwaha et al. [31] propose a routing scheme that combines the on-demand routing potential of the Ad Hoc On-demand Distance Vector (AODV) routing protocol with a distributed topology discovery mechanism using ant-like agents. This hybrid protocol, known as Ant-AODV, reduces the latency of route discovery and the end-to-end delay when compared to the basic AODV protocol.

In the AODV routing protocol, a node that wishes to send a message to a destination node for which it has no valid route broadcasts a route request (RREQ) packet to its neighbors. The neighbors in turn forward the request to their neighbors till it reaches the destination or an intermediate node with an up-to-date route to the destination. AODV avoids loops using sequence numbers maintained by each node, and each has a broadcast ID. When either the destination node or intermediate node receives the RREQ, they respond by sending a route reply packet (RREP) to the source. AODV uses HELLO broadcasts to inform nodes of others in their neighborhood. If the movement of a node is detected by its upstream neighbor, it propagates a route error message (RERR) to its active upstream neighbors to inform them.

The Ant-AODV hybrid routing protocol tries to overcome the characteristic faults of AODV and ant-based routing. In ant-based routing techniques, route establishment is dependent on ants visiting a node so data packets cannot be sent until an ant reaches a node and provides a route. They also do not maintain local connectivity, so when a route breaks, the source is unaware and keeps sending data packets which end up being lost. AODV, on the other hand experiences delays in route discovery which result in longer times for connection establishment. In Ant-AODV, the ants independently provide routes to the nodes but the nodes can also find recent route entries to destinations by launching on-demand route discovery via AODV. RREQs receive quicker replies due to increased connectivity of all the nodes, thus reducing route discovery latency. Source nodes have the flexibility of switching from longer routes to newer and shorter ones provided by the ants. Ant-AODV sends route error messages (RERR) to upstream nodes to inform them of local link failures. It uses HELLO broadcasts to maintain neighbor information and uses a routing table to select a randomly chosen next hop from a list of neighbors to avoid previously visited nodes.

However, the performance of the algorithm shows no significant increase in the fraction of packets delivered compared to AODV.

Probabilistic Emergent Routing Algorithm(PERA)

Baras and Mehta [4] present a Probabilistic Emergent Routing Algorithm (PERA) for mobile ad hoc networks. Route discovery in the algorithm is done by broadcasting ants towards the destination creating multiple paths, and entries at each node's routing table that represent the probability of choosing neighbouring nodes as next hops.

The algorithm uses three kinds of agents - regular forward ants, uniform forward ants and backward ants. Uniform and regular forward ants are agents (routing packets) that are of unicast type. These agents proactively explore and reinforce available paths in the network. Both ants are created the same but routed differently. They create a probability distribution value in each node's table for its neighbors. The probability or goodness value at a node for its neighbor reflects the likelihood of a data packet reaching its destination by taking the neighbor as a next hop.

Backward ants are used to propagate the information collected by forward ants through the network and to adjust the routing table entries according to the perceived network status. Nodes proactively and periodically send out forward, regular and uniform ants to randomly chosen destinations. Thus, regardless of whether a packet needs to be sent from a node to another node in the network or not, each node creates and periodically updates the routing tables to all the other nodes in the network. (The algorithm assumes bidirectional links in the network and that all the nodes in the network fully cooperate in the operation of the algorithm.)

A broadcast of regular forward ants is done at the start of the session and does not follow any pheromone value. Data packets are routed deterministically based on the maximum probability (highest pheromone value) of neighbouring nodes.

ARA

Gunes et al. [18] were the first to propose a routing algorithm based on the idea of AntNet for MANETs. Their algorithm, Ant-colony based Routing Algorithm (ARA) relies on a backward and forward ant and consists of 3 phases: route discovery, route maintenance and route failure handling.

Route Discovery Phase: As the name implies, in this phase new routes are created. The creation of new routes is done using a forward ant (FANT) and a backward ant (BANT) as can be seen in Figures 3.5 and 3.6. A FANT is an agent which establishes the pheromone track to the source node while the BANT enforces it. Each FANT is implemented as a small packet with a unique sequence number. The FANT is broadcast by the sender and is relayed by its neighbors. Any node receiving a FANT for the first time creates a record in its routing table. The records in the routing table consist of (destination address, next hop, pheromone value). The node interprets the source address of the FANT as destination address, the address of the previous node as the next hop and computes the pheromone value depending on the number of hops the FANT needed to reach the node. When the FANT reaches the destination node, a BANT is created and sent to the source node. The BANT like the FANT has the task of establishing a track to the destination node. When the sender receives the BANT from the destination node, the path is established and the data packets can be sent.

Route Maintenance: This phase of the algorithm responsible for the improvement of the routes during communication. ARA needs no special packets for this but when pheromone tracks for the source and destination nodes are established, subsequent data packets are used to maintain the path. As in nature, the paths do not keep their initial pheromone values forever. When any node relays a data packet to a neighbour node, towards the destination it increases the pheromone value of the entry, i.e., the path of the destination is strengthened by data packets. The evaporation process of pheromone is simulated by regularly decreasing the pheromone values.

ARA uses a simple method which is also used in the route discovery phase to prevent loops. Duplicate ants are recognized based on the source address and the

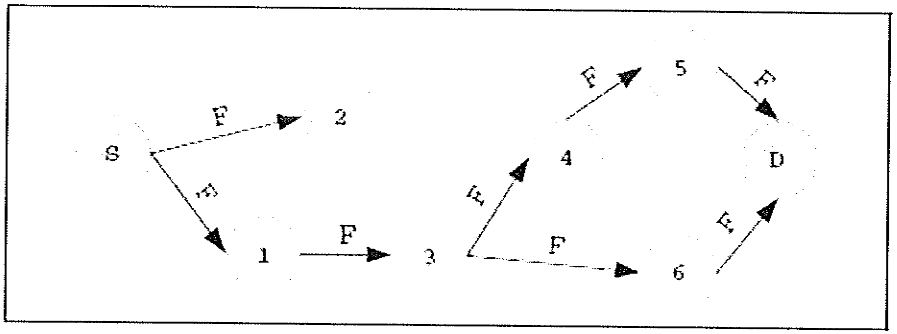

Figure 3.5: Route Discovery Phase by Forward Ant

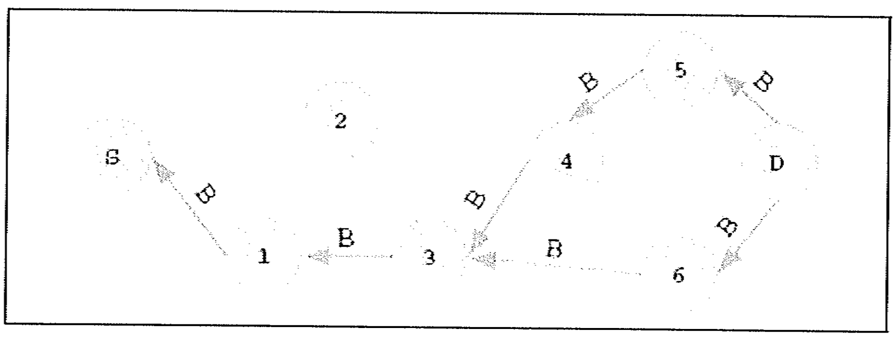

Figure 3.6: Route Discovery Phase by Backward Ant

sequence number. Whenever a node receives a duplicate ant, it sets the DUPLICATE_ERROR flag and sends the packet back to the previous node. The previous node deactivates the link to this node so that packets are no longer sent in that direction.

Route Failure Handling: This phase handles routing failures caused especially by node mobility. ARA identifies a route failure through a missing acknowledgement. If a node receives a ROUTE_ERROR message for a certain link, it first deactivates the link by setting the pheromone value to 0. It then searches for an alternative link in its routing table. If there is a second link it sends the packet via this path, otherwise it informs its neighbors that they should relay the packet. The packet can either be transported to the destination node or backtracking continues to the source node.

Where the packet is unable to reach its destination, the source has to initiate a new route discovery phase.

ARA reduces routing overhead but it is not scalable and cannot detect cycles.

Accelerated Ants-Routing(AAR)

The Accelerated Ants-Routing algorithm discussed by Matsuo and Mori [32] uses ant-like agents that go through the network randomly, without a specific destination, updating pheromone entries pointing to their source (the source of the agents is represented as destination in the routing table entries). It is an enhancement of ARA and a modification of the Ants-Routing algorithm [46]. The Accelerated Ants-Routing algorithm is proposed to accelerate the converging speed of the routing table to obtain the optimum routing paths.

Ant Routing Algorithm for Mobile Ad hoc network(ARAMA)

Ant Routing Algorithm for Mobile Ad hoc network (ARAMA) [22], is a combination of on-demand and table driven algorithms. The focus of this technique is on optimizing different Quality of Service (QoS) parameters, other than number of hops. Such parameters include energy, delay, battery power, mobility etc. The paper proposes a path grading enforcement function that can be modified to include these QoS parameters. One of the important attributes of this algorithm is that the lifetime of the ad hoc nodes have been extended by using a fair distribution of energy across the network.

Mobile Ant Based Routing

A two-layered concept is proposed by Hessenbüttel [21] for routing in large-scale mobile ad-hoc networks as represented in Figure 3.7. It uses a Topology Abstracting Protocol (TAP) which abstracts the dynamic and irregular topology of a MANET to create a topology with logical routers and logical links. The logical routers are a collection of nodes built by grouping geographically closeby nodes together and the logical links are multi-hop paths between selected logical routers.

The underlying routing protocol, Mobile Ant Based Routing (MABR), runs on the abstract topology. Its responsibilities include updating the logical router table and determination of logical paths for packet routing over the abstract topology. The protocol consists of two; forward and backward ants that are used in the update process. The forward ant is dispatched at regular intervals from every logical router to destinations chosen at random. It changes to abackward ant when it reaches its destination. This ant travels in the reverse direction over the recorded path determining the quality of the path using the data stored in the link cost tables. It updates any logical router it reaches with the logical link cost encountered so far (i.e. marks entries with artificial pheromone). Naturally, this pheromone decays with time.

The Straight Packet Forwarding (SBF) protocol used in the lower layer is responsible for forwarding packets over the logical link to the next logical router which it does in a greedy manner. That is, it sends packets to the node with the closest coordinates defined by MABR. It also deals with failed forwarding of packets.

AntHocNet

AntHocNet [9] is a hybrid algorithm that combines the reactive setup with proactive route probing, maintenance and improvement.

This algorithm uses ant agents that adhere to and update their pheromone tables by a process based on stigmergy. Each node maintains a local pheromone table which basically defines the paths. The value of each entry in the table is a measure of goodness of taking a particular path. It is a combined measure of the path end-to-end delay and number of hops. The reactive path setup entails a source node broadcasting a reactive forward ant when it wants to start communication with a destination node for which it has no routing information. It uses the routing information to determine the probability of following its next hop depending on the relative goodness. The broadcast results in duplicate copies of the same ant but a node only accepts the first and disposes of the rest, thus it sets up only a single initial path. Additional paths are added by the proactive path exploration and maintenance mechanism. The path

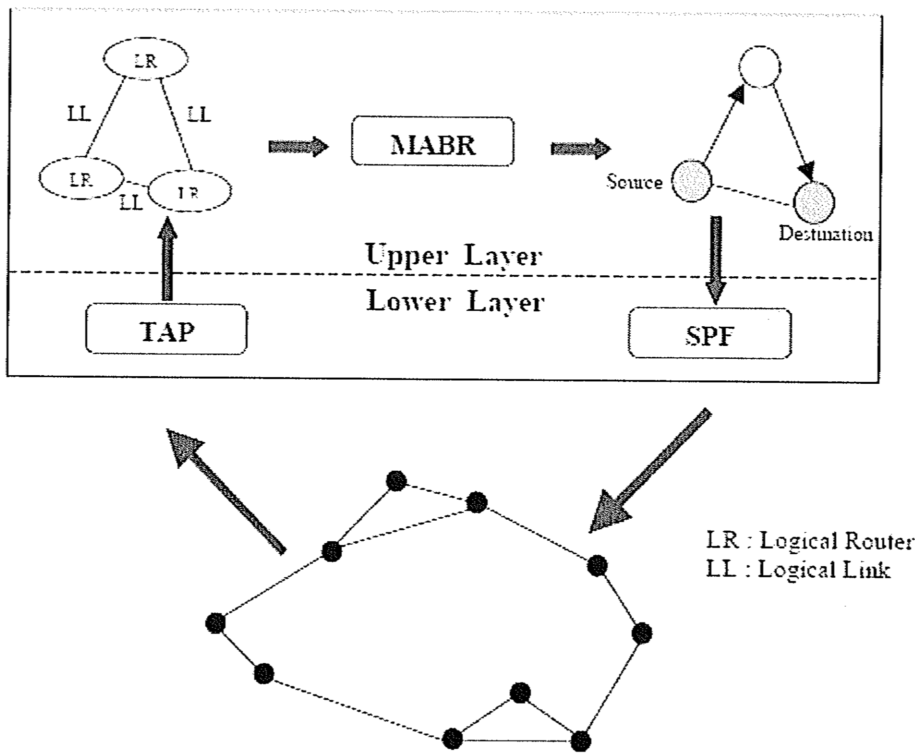

Figure 3.7: Overview of the architecture [21]

is completely set up when the backward ant returns to the source and data sending can commence. Where the backward ant fails to return within a period of time the process is restarted.

While communicating, through currently used paths, the search for new ones is done by sending out a proactive forward ant which updates routing tables like the reactive forward ant. The algorithm introduces a pheromone diffusion function which provides an alternative way to update pheromone information for existing paths and gives information to guide explanatory behavior of proactive forward ants. The process is carried out by the nodes broadcasting periodically in sync to all their neighbors. The information obtained is called bootstrapped pheromone which is used directly for maintenance of existing paths and indirectly for path exploration.

Nodes in AntHocNet achieve automatic load balancing of data in the network by forwarding the data stochastically according to pheromone values. The next hop is determined by a probability value when there are multiple hops towards the destination. Nodes detect failures in the network from failed unicast transmissions or periodic pheromone diffusion messages that do not arrive. Where there are no other paths towards the destination for a data packet, a route repair ant is sent out.

Other Protocols

Ahmed [1] studied the performance issues related to routing in MANETs. He develops a routing algorithm based on ACO and uses Kleinrock's delay analysis [20] technique. In particular, Ahmed focuses on the end-to-end delay in MANETs. He combines mobility modeling with queuing network analysis to evaluate end-to-end delay. Based on the experiments, Ahmed is able to determine that the algorithm provides a better performance by reducing the mean End-to-End delay than the AODV algorithm and that there is no significant effect on delay among the different mobility models considered.

The ACO routing algorithms in the literature insert ants (control packets) into the system to determine the shortest path from the source to the destination. Though this is done to select the best path, there is a drawback. If the network is large, then more ants have to be sent into the system. This decreases the scalability of the algorithm. In a recent paper [35], the authors propose a message migration scheme together with an efficient scheme for updating the probability of packet forwarding to achieve a scalable and efficient routing algorithm for MANETs.

In addition, Ant based algorithms have also been used to study quality of service issues related to routing in MANETs [44] and have been used to develop distributed topological control algorithms for MANETs [27].

The algorithm presented in the next chapter is inspired by the routing algorithm proposed by Islam[23], known as Source Update. The algorithm was proposed as a parallel implementation of ACO applied to an irregular problem such as MANETs. But in this thesis, I focus on the effectiveness of this approach in a simulated wireless environment.

3.4 Summary

In this chapter, I presented an overview of the ant colony optimization metaheuristic. I described the foraging behaviour of ants which is the foundation of the ACO approach. I also discussed with examples its applications in computation, particularly in telecommunication.

Chapter 4

An Improved ACO Algorithm for Routing in MANETs

In this chapter, we present an algorithm called PACONET (imProved Ant Colony Optimization algorithm for mobile ad hoc NETworks) as a solution to the challenges of routing in MANETs.

4.1 Paconet

PACONET is a routing protocol for mobile ad hoc networks inspired by the foraging behaviour of ants. It uses the principles of ACO routing to develop a suitable problem solution. This algorithm is inspired by the routing algorithm proposed by Islam[23]. It uses two kinds of agents: Forward ants (FANT) and Backward ants (BANT). The FANT explores the paths of the network in a restricted broadcast manner in search of routes from a source to a destination. The BANT establishes the path information acquired by the FANT. These agents create a bias at each node for its neighbours by leaving a pheromone amount from its source. Data packets are stochastically transmitted towards nodes with higher pheromone concentration along the path to the destination. FANTs also travel towards nodes of higher concentration but only if there exists no unvisited neighbour node in the routing table.

4.2 Routing Table

Each node in the network has a routing table (RT) whose size is the degree of the node times all the nodes in the network. That is, if the number of nodes is N and the degree of node v_i is d_i then the size of the RT is Nd_i. The rows of the RT represent the neighbours of node v and the columns represent all the nodes in the network. Each pair (row, column) in the RT has two values: (i) a binary value indicating if the node has been visited and (ii) the pheromone concentration.

4.2.1 Route Maintenance

Hello messages are used to detect and monitor links to neighbors. Each node periodically broadcasts a Hello message, that all its neighbors within its transmission range receive. Hello messages are transmitted at an interval of Hello_Interval seconds. Therefore, if a node fails to receive several Hello messages from a neighbor, a link breakage is detected and its routing table can be updated by deleting the entries in the routing table for that neighbor.

4.2.2 Route Discovery

When a source node S wishes to communicate with a destination node D for which it has no route information, it sends out a FANT to all its neighbours in search of the destination node. When a FANT from a source S traveling to a destination D, arrives at a node, v, the FANT determines its path or next hop neighbour by looking at the node's routing table. It considers node v's neighbours by looking at the rows against the column D in the RT. By considering column D, the FANT can select the best path from a neighbouring node to D rather than the best link between itself and its neighbour.

The FANT will only consider the pheromone concentration when all neighbours in column D have been visited. The purpose of this is to ensure that all possible paths are explored to find the best path towards the destination. The node with the highest pheromone is chosen as the next hop after the FANT has determined that it has not visited the node before. This is to avoid the ant traveling in cycles. The FANT maintains a list of all nodes visited on its journey to D for this purpose. The FANT keeps in memory the total time (T) it has travelled. When a next hop node, v_j is selected from v_i, the FANT moves there and updates the pheromone entry for (v_i, S) in v_j's RT using the following equation

$$\delta(v_i, v_s) = \delta(v_i, v_s) + \frac{\epsilon}{T(v_s,v_i)+W(v_i,v_j)}$$ [1]

where ϵ is a user defined runtime parameter; $\delta(v_i, v_j)$ and $w(v_i, v_j)$ represent the pheromone value on each edge and the time period for which the links are in connection respectively. For all the other nodes in the source column, the pheromone values are decremented by the following equation:

$$\delta(v_l, v_s) = (1 - \xi)\, \delta(v_l, v_s), \forall l \neq i$$

where ξ is evaporation rate of the pheromone. ξ is also determined by the user. The total time of the path just traversed is recorded as T(v_s, v_i)+w(v_i, v_j).

When the FANT reaches the destination, a corresponding BANT is created with the source of the FANT as its destination. The BANT travels towards its destination using the list of visited nodes acquired from the FANT while updating the pheromone concentration for the destination column. The purpose is to increase the concentration of pheromone closest to the destination and decrement the concentration for other nodes in column D. That is, to update an entry (v_b,v_D) for an ant at node v_k, traveling backwards from v_b, we look at the rows of v_b's neighboring nodes and column D:

$$\delta(v_b, v_D) = \delta(v_b, v_D) + \frac{\epsilon}{T'}$$

where T′ is T(v_s, v_d) - T (v_s, v_k).

The advantage of performing this update is that it makes it easy to determine the best available path reachable from a source and to find a path easily when another ant considers the source as its destination.

[1] All the equations used are the same as those used by Islam [23].

4.2.3 Routing data packets

Data packets are routed using highest pheremone concentration at each node towards the destination.

4.3 Flow Diagrams of algorithm

The algorithm executed by each node and ant are illustrated in the flow charts in Figures 4.1 and 4.2 below.

4.4 Summary

In this chapter, an improved algorithm, PACONET, for routing in MANETs based on the ACO technique was presented. Also included are flowcharts for more detailed description.

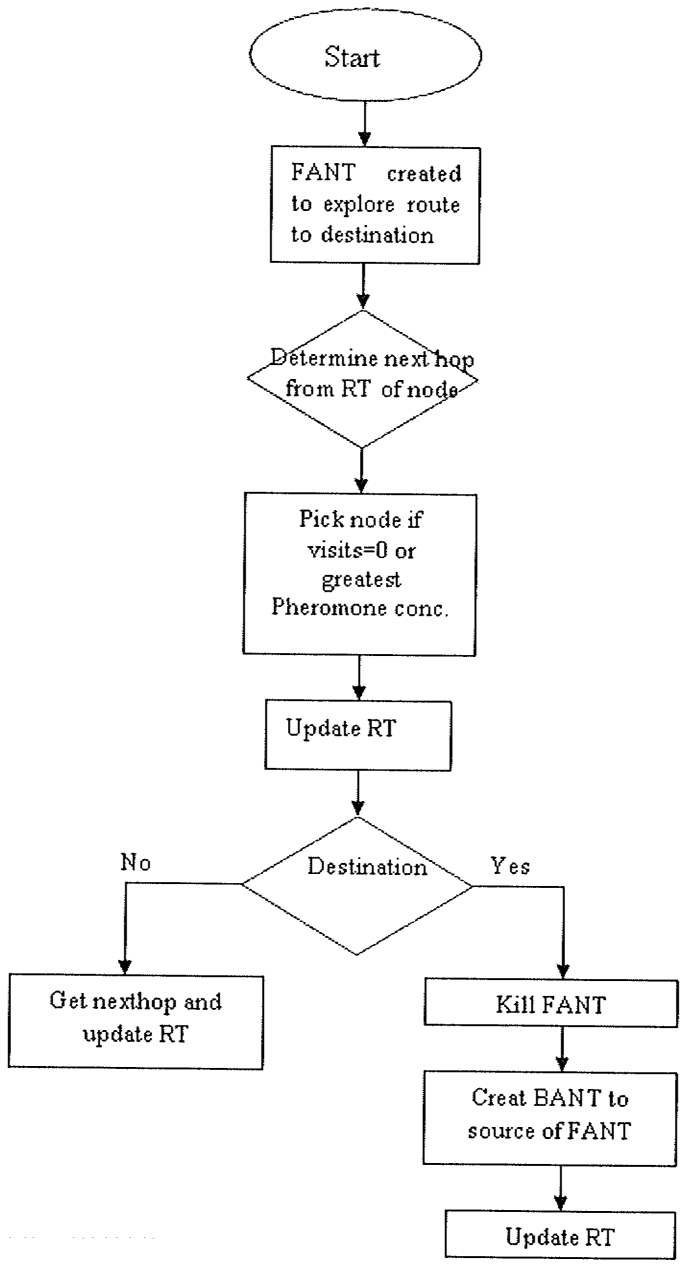

Figure 4.1: Algorithm executed by each Ant

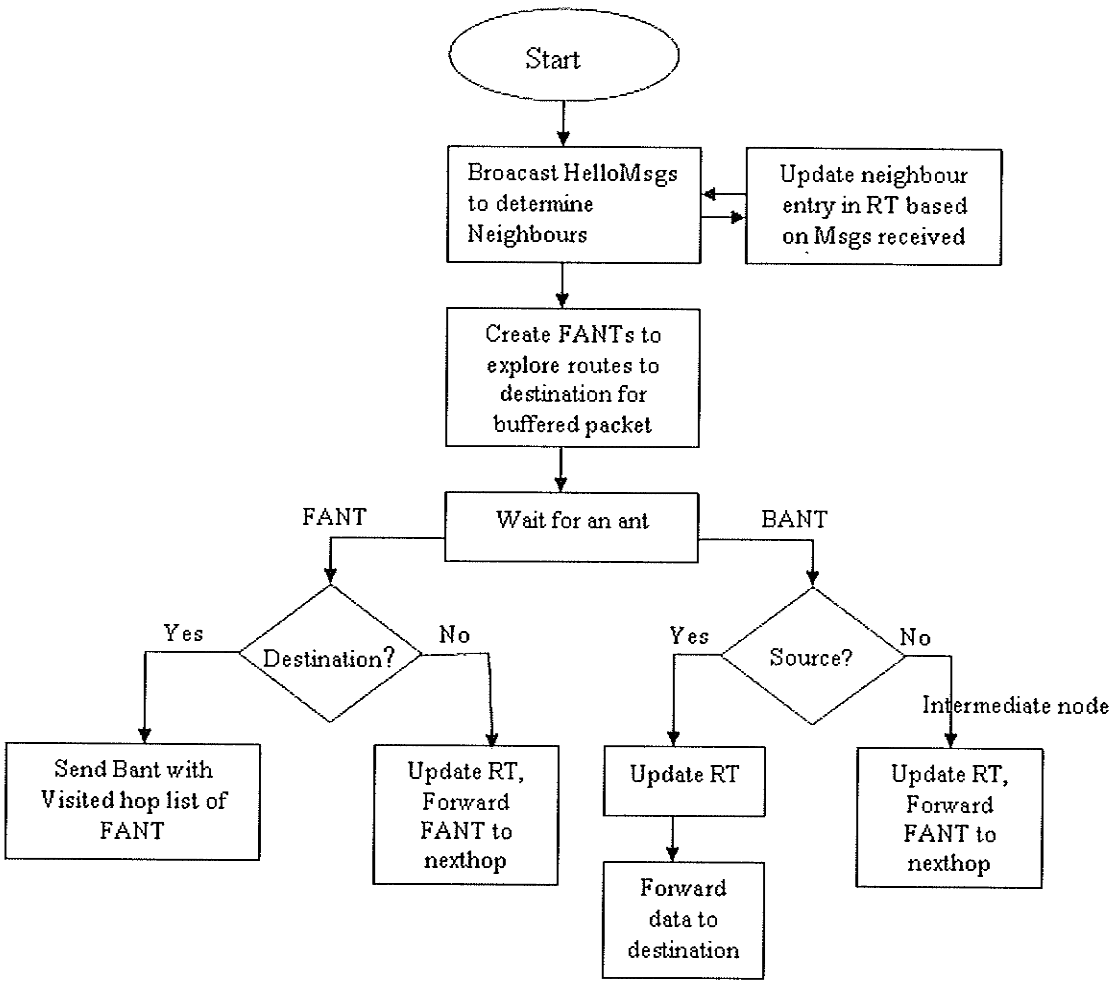

Figure 4.2: Algorithm executed by each Node

Chapter 5

Performance Evaluation and Results

5.1 Simulation Environment

My algorithm has been implemented using GloMoSim [50]. GloMoSim is a scalable simulation library for wireless network systems based on the parallel discrete-event simulation capability of PARSEC (PARallel Simulation Environment for Complex System) [2]. PARSEC is a C-based parallel simulation language. GloMosim can be modified to add new protocols and applications to the library.

GloMoSim was built and designed with the primary goal of simulating very large network models by using parallel execution to reduce the execution times of the simulation model. The implementation techniques GloMoSim adopts make it more scalable than some other nework simulators such as NS2 [16] and thus able to simulate larger networks. This is done by using node aggregation. As each entity needs to examine packets received only from the nodes located in the region it is simulating, many partitions are used to reduce the total search space for packet delivery. If a packet sent by a node located in Partition (0, 0) cannot reach the border of the partition, no message needs to be sent to the other partitions. Therefore, the other partitions do not have to process the reception of the packet. GloMoSim also has a more convenient programming environment when compared to similar tools and

Layer	Models
Physical (Radio Propagation)	Free space, Two-Ray
Data Link (MAC)	CSMA, MACA,TSMA, 802.11
Network (Routing)	-Bellman-Ford, FSR, OSPF, DSR, WRP, LAR, AODV
Transport	TCP, UDP
Application	Telnet, FTP

Table 5.1: Layers in Glomosim

supports several mobility models [3].

GloMosim Architecture

GloMosim is built using a layered approach, similar to the OSI seven layer network architecture. Table 5.1 shows a list of the models available at each layer. My algorithm was implemented in the network layer. GloMosim was installed with PARSEC on Windows XP for this thesis.

GloMosim Simulations

To execute simulations in GloMosim, the input file has to be configured and a command given within the BIN directory. The basic input file *config.in* is found in the Bin directory and contains the general simulation parameters for GloMosim. The simulation time, terrain, node mobility, routing protocol, etc. are determined from this file. It is structured according to each layer and allows a protocol to be chosen for each layer. In the file, anything following "♯", is treated as a comment. Therefore, in order to select a protocol, the "♯" is deleted. Figure 5.1 shows a snapshot of an input file.

The command *GloMosim config.in* in the BIN directory will run the simulation and produce an output file *GLOMO.stat*. This file contains the results for each layer as specified in the such as data packets transmitted and received etc.

There is a *nodes.input* file that allows you to specify the placement of nodes for a simulation run. The structure of each entry in the file is $< nodeaddress >$, $< 0 >$, $< (x, y, z) >$. I did make use of this file but instead chose the 'Random' placement option for the nodes in the config.in file.

```
SIMULATION-TIME  900S

TERRAIN-RANGE-X  1500
TERRAIN-RANGE-Y  300

NUMBER-OF-NODES  50

#NODE-PLACEMENT   FILE
#NODE-PLACEMENT-FILE ./nodes.input
#NODE-PLACEMENT   GRID
#GRID-UNIT      30
 NODE-PLACEMENT   RANDOM
#NODE-PLACEMENT   UNIFORM
#NODE-PLACEMENT   GROUP_RANDOM

#MOBILITY NONE

# Random Waypoint and its required parameters.

MOBILITY RANDOM-WAYPOINT
MOBILITY-WP-PAUSE    50S
MOBILITY-WP-MIN-SPEED 0
MOBILITY-WP-MAX-SPEED 10

#MOBILITY TRACE
#MOBILITY-TRACE-FILE ./mobility.in

CHANNEL-BANDWIDTH   2000000

MAC-PROTOCOL    802.11
#MAC-PROTOCOL    CSMA

NETWORK-PROTOCOL  IP

ROUTING-PROTOCOL  AODV
#ROUTING-PROTOCOL  DSR
#ROUTING-PROTOCOL  WRP
#ROUTING-PROTOCOL  ZRP
#ZONE-RADIUS    2
#ROUTING-PROTOCOL  PACONET

#ROUTING-PROTOCOL  STATIC
#STATIC-ROUTE-FILE  ROUTES.IN

APP-CONFIG-FILE  ./app.conf
```

Figure 5.1: Snapshot of an input file for Glomosim

The *app.conf* file is used to set traffic in the network. The structure of the file is $< sourcenodeaddress >$, $< destinationnodeaddress >$, $< no.ofitemstosend >$, $< itemsize >$, $< intervaltime >$, $< starttime >$, $< endtime >$. I used this file to set and vary the sending sources (traffic load) for my simulations.

Implementing a new protocol

Different steps have to be taken when adding a model or protocol to a layer in order to maintain consistency and proper functionality in GloMoSim. Three main functions have to be provided to add a new model:

1. Initialization Function; which must allocate and initialize the model specific data.

2. Finalization Function; which generates the output statistics from the simulation run for this model.

3. Simulation Event Handling Function; which performs simulation actions when scheduled with an event e.g sending out control packets at specified time intervals.

5.1.1 Simulation Setup

In GloMosim, a base simulation setting was created and several scenarios were obtained from it to run experiments. In the setting, 50 mobile nodes are randomly placed in a rectangular area of 1500m X 300m to form an ad hoc network. The nodes are made to move with a maximum velocity of 10m/s according to the Random Waypoint Model(RWM) [25]. In this model, each node at the start of the simulation remains at a particular location for a certain period of time (i.e pause time seconds). It will then select a random destination within the simulation area and a speed uniformly distributed between 0m/s and the maximum speed. The node travels towards the new destination and upon arrival it pauses for the specified time period and then proceeds as previously described. It repeats this process for the duration of the simulation. The RWM model has some drawbacks [49] but I have chosen to use it because it is the most commonly used. Each simulation is run for a total of 900 seconds. The data traffic is generated using 20 constant bit rate (CBR) sources, with sending rates of 4 packets of 64-bytes per second. The radio propagation range of the nodes is 250 meters and the data rate is 2Mbits/s. At the Mac layer we use the 802.11 protocol and at the physical layer, the free space signal propagation model is used.

To create different scenarios, I varied the movement pattern of the nodes by using different maximum velocities and different pause times for the nodes. I also varied the data traffic by using different numbers of sending sources.

5.1.2 Metrics

The following metrics, as suggested by the Mobile Ad-hoc Networks (MANET) working group for routing protocol evaluation have been used for studying the simulation results. The metrics are chosen to evaluate the efficiency and effectiveness of the protocol:

- **Packet Delivery Ratio** - is the ratio of the number of packets delivered to the destination to the number of packets generated by sources. This number presents the routing efficiency or the throughput of the protocol.
- **Average End-to-End Delay** This is the average time taken by a data packet to travel from the source to the destination. This includes all delays due to buffering during route discovery, queuing at the interface, and retransmission latency at the MAC layer, as well as propagation and transmission time.
- **Control Overhead** The control overhead is the total number of routing packets such as Fants, Bants, Hello Msgs transmitted for the entire simulation time.

5.2 Results

Many simulations were performed for the ad hoc protocol, the results are an average of the results obtained from different simulation runs with identical parameters. Each run has a different random number generator seed value. This seed is used to generate random numbers, such as the next destination of a mobile node. The results are presented in two scenarios. Scenario I focuses on the performance in terms of node mobility while scenario II focuses on performance with varing traffic loads.

5.2.1 Scenario I

In this scenario, I study the performance of PACONET in comparison to AODV in an increasingly dynamic environment. I vary the node mobility by either increasing the maximum node speed or decreasing the node pause time. With the RWM, the lower the pause time the higher the mobility. The following results (Figures 5.2 - 5.7) show the metrics considered as a function of node speed and node pause times. When the node pause time is varied, there is improved performance from PACONET compared to when the node speed was varied. The pause time happens to affect the network differently. A longer pause time means that nodes with the same transmission range remain connected for a longer time but on the other hand, certain nodes remain unreachable for a longer time.

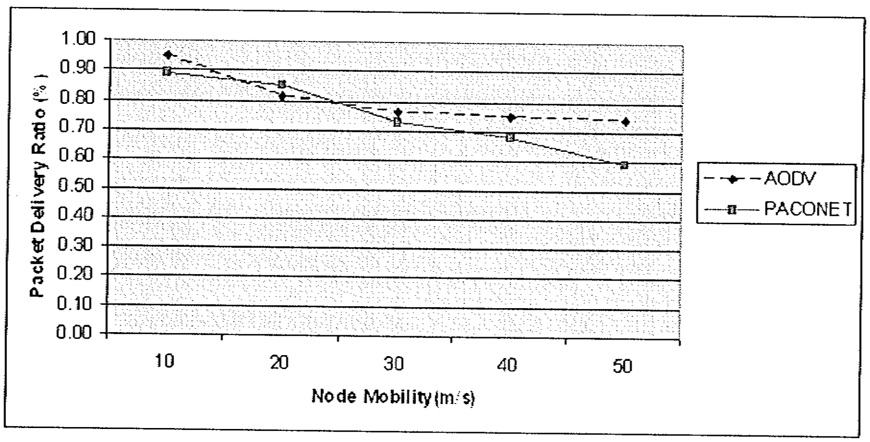

Figure 5.2: Packet Delivery Ratio measured against varying node speeds

In Figure 5.2, The packet delivery ratio of PACONET is close to that of AODV but unfortunately does not outperform it. This is probably due to the process of link failure handling. In PACONET, a node would wait several seconds to confirm a failed hello packet before modifying its routing table. Data packets going through this node towards the newly broken link will end up being lost due to the delay in updating the routing table. The lower packet delivery ratio of PACONET is not entirely due to dropped packets. A large number of packets in some cases were seen to be in motion at the end of the simulation run. The higher mobilty of nodes means more link failures, which results in affected nodes initiating route discoveries during which packets are buffered.

In Figure 5.3, where the packet delivery ratio is measured at varying node pause times, both algorithms show improved performance. Their performance actually alternates, AODV starts out better but PACONET finishes with higher delivery ratio for longer pause times.

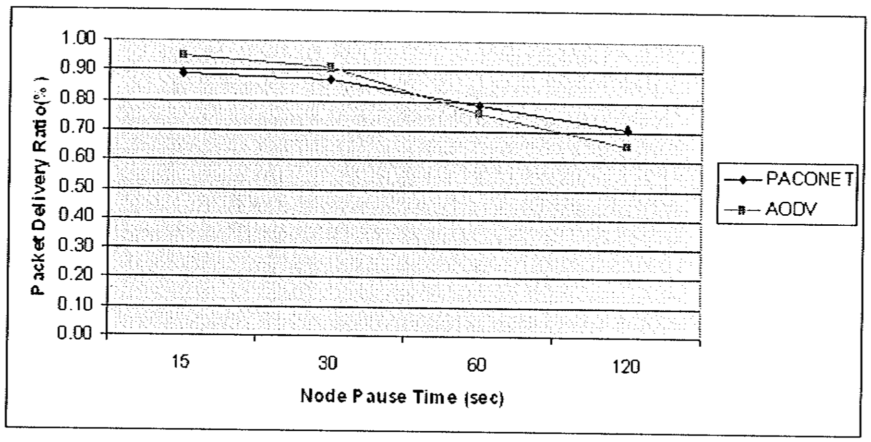

Figure 5.3: Packet Delivery Ratio measured against varying Pause times of mobile nodes

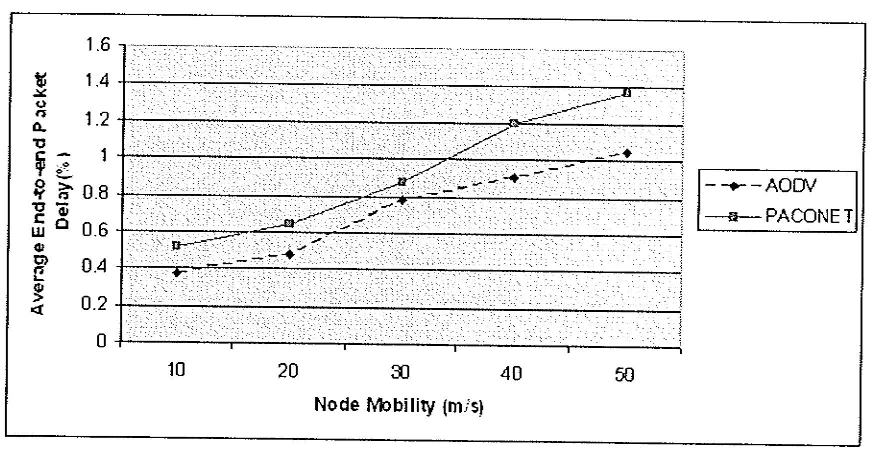

Figure 5.4: Average End-to- End Delay measured against varying node speeds

From Figure 5.4, It can be seen that PACONET experiences a higher end-to-end

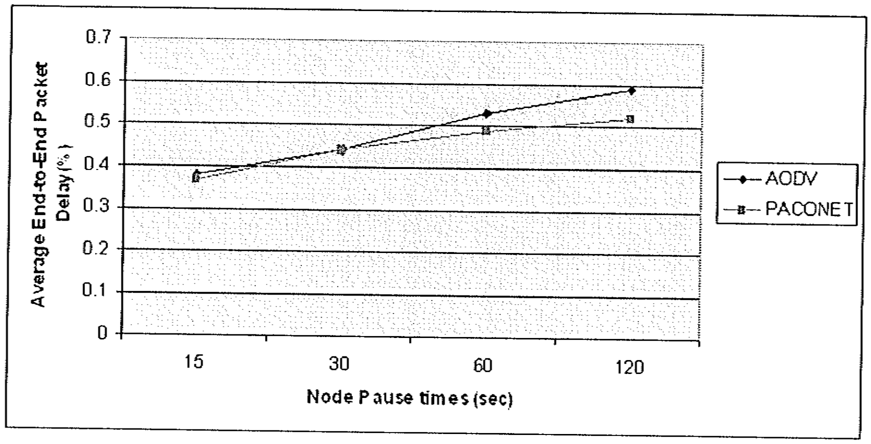

Figure 5.5: Average End-to-End Delay measured against varying Pause times of mobile nodes

delay than AODV. But in Figure 5.5, where node pause time is varied, AODV shows higher delay overall. The delay in PACONET is usually significant at the start of the simulation beacuse of the initial search for routes. Intermediate nodes in AODV are able to respond to route requests thus saving the time for path discovery. In PACONET, the source node has to wait till it gets a BANT sent by the destination. The initial peak in end-to-end delay is what causes AODV to outperform PACONET at higher node speeds. Also, unlike most protocols, PACONET does no broadcasting which limits the extent of path discovery

In Figure 5.6 and 5.7, PACONET experiences quite a bit overhead due to different control packets that have to be sent around the network for route maintenance and discovery. Hello messages in particular are continously sent out to help nodes keep track of there neighbours. Although AODV also makes use of Hello messages, it is not implemented in the simulator giving AODV an advantage over PACONET. Despite this advantage, PACONET still outperforms AODV in Figure 5.7. It can also be noted that the overhead appears to increase steadily afterwards at some point. This could be attributed to the way PACONET adjusts to network changes. Less control packets are sent out when the nodes are more stable, that is, the nodes remain in connection longer.

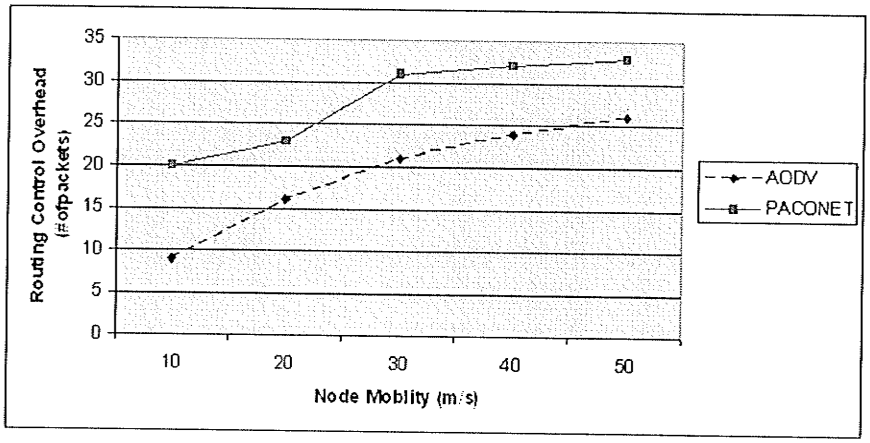

Figure 5.6: Routing Control Overhead measured against varying node speeds

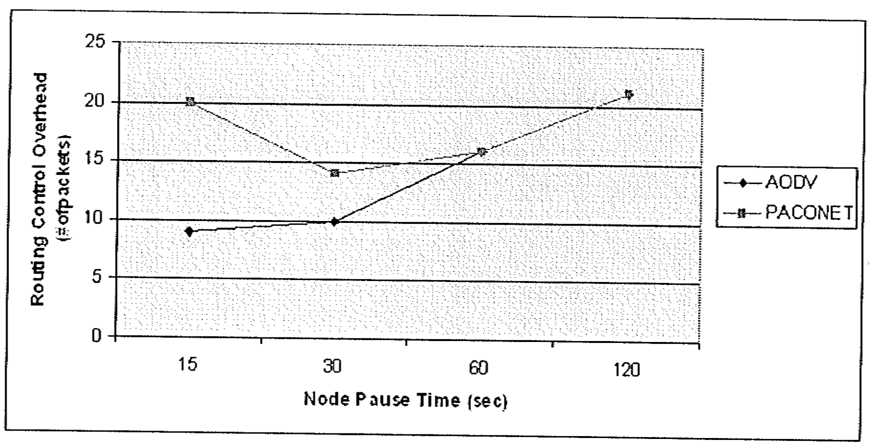

Figure 5.7: Routing Control Overhead measured against varying Pause times of mobile nodes

5.2.2 Scenario II

In this scenario, the performance is studied in terms of varying network traffic load.

The network load is varied using a number of CBR sources varying between 5 and 20 at a fixed maximum velocity. In Figure 5.8 and 5.9, PACONET appears to be able to manage traffic load better than AODV. This could be attributed to maintenance of a certain amount of routing information for all nodes in the network as opposed

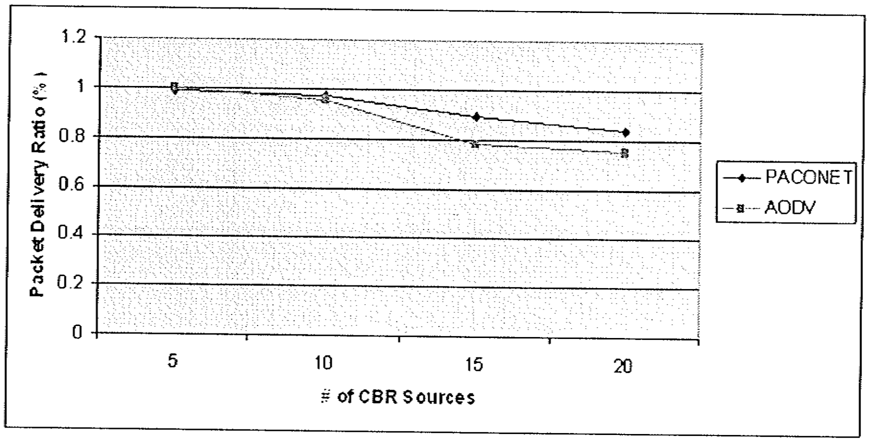

Figure 5.8: Packet Delivery Ratio measured against varying number of Constant bit rate sources

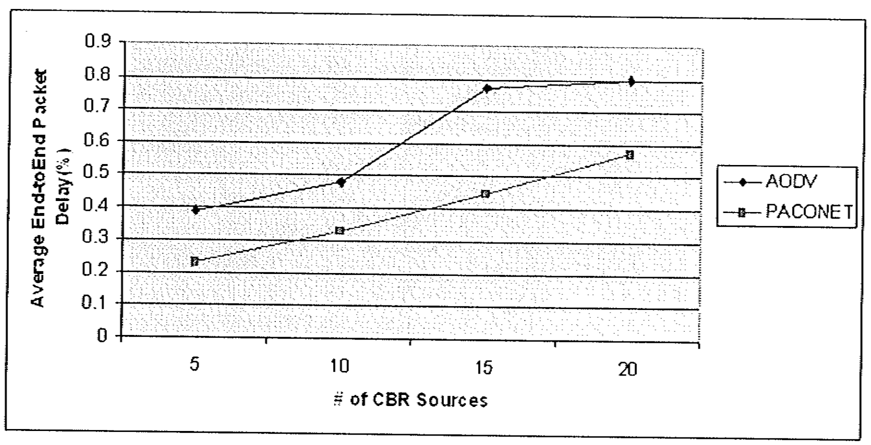

Figure 5.9: Average End-to-End Delay measured against varying number of Constant bit rate sources

to just certain routes. These routes remain valid for a longer time and are readily available making the delivery of more data packets successful. Although, AODV reacts extensively to packet loss and searches for new routes, that of PACONET is not as elaborate which is why it is unable to deliver more packets than it does where there is high mobility in the network.

5.3 Summary

In this chapter the Glomosim simulation tool, which was used for this work, is introduced. The implementation details of the algorithm and the test scenarios are discussed. The performance evaluation of the algorithm and its comparison to AODV was discussed. The results show that PACONET algorithm has great potential but it does not outperform AODV in every situation.

Chapter 6

Conclusion

An efficient routing scheme is very important for a network to ensure that the nodes that form the network are able to communicate. There are also certain features that aid a protocol to give the best performance such as flexibility, energy conservation etc.

This thesis presented an improved routing protocol for routing in MANETs using the Ant Colony Optimization (ACO) metaheuristic. Unlike the previous work done, that is the Source Update, this algorithm is very dynamic. It incoporates mobility which the other does not. This key feature makes it adaptable to an ad hoc environment. The algorithm is also able to perform route maintenance and handle link failures in the network. The work done in this thesis studies the efficiency and effectiveness of the approach as a solution to the routing problem in a simulated ad hoc environment using the wireless network simulator, known as Glomosim. The initial work, on the other hand, studied the parallel implementation of the ACO algorithm as a solution to an irregular application such as MANETs. The performance of the algorithm was also studied in comparison to the AODV routing protocol.

The performance was measured using three metrics, packet delivery ratio, routing control overhead and end-to-end delay. A base simulation setup was created and certain parameters such as traffic load and node mobility was varied for the different experiments performed.

By varing either the speed of mobile nodes or the pause times of the nodes, the effect of mobility on the algorithm could be determined. Although Paconet did not outperform AODV in most of the results presented, its performance was not greatly different from that of AODV. This is a good indication of great potential of the algorithm if more improvements are made as described in the next chapter.

In terms of varying traffic load, Paconet showed better performance. The results show no unusual performance degradation for Paconet but rather consistent results.

Chapter 7

Future Work

Routing in ad hoc networks is more challenging than routing in other types of network. This is a fairly new area with much work still being done to produce better routing schemes.

From the results obtained from the evaluation of the algorithm, I have learned the strenght and weaknesses of the algorithm. This gives direction on how best to approach MANET routing for improved performance and added features to make the algorithm more suitable for a wireless environment.

The algorithm although not too greatly challenged by mobility is unable to outperform a state of the art routing protocol, AODV, in every scenario created. This might be improved in future work by doing the following things:

- Incorporating broadcasting and re-broadcasting or even multicasting of Fants in the network, which is consistent with most protocols. This should improve the search time for routes because more ants can be propagated through the network in a shorter period of time.

- An approach of informing the source or other neighbouring nodes about a failed link, so their routing tables can be updated to avoid packet loss due to link failures. Also attempting to repair failed links could improve the packet delivery ratio.

- Improving the behaviour of the ants and the routing information in control packets sent within the network should help reduce the overhead created by the algorithm. The improvement could be in terms of more control over ants generated and more detail in routing information exchange to reduce the need for continous search.

With such improvements listed above, one could then consider evaluating the algorithm's performance in terms of varying simulation areas sizes, more nodes in the network and the use of different mobility models. This work can be futher extended by doing a comparison to other ant algorithms, considering Quality of Service, energy conservation and other issues relating to MANETs.

Bibliography

[1] T. H. Ahmed. Simulation of Mobility and Routing in Ad Hoc Networks using Ant Colony Algorithms. In *International Conference on Information Technology: Coding and Computing*, volume II, pages 698–703, Las Vegas, Nevada, USA, April 2005. IEEE Computer Society.

[2] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song. PARSEC: A Parallel Simulation Environment for Complex System. In *IEEE Computer Magazine*, volume 31, pages 77–85, October 1998.

[3] L. Bajaj, M. Gerla, M. Takai, R. Ahuja, and R. Bagrodia. GloMoSim: A Scalable Network Simulation Environment. Technical Report 990027, University of California, 1999.

[4] J. Baras and H. Mehta. A Probabilistic Emergent Routing Algorithm for Mobile Ad hoc Networks (PERA). In *WiOpt'03: Modeling and Optimization in Mobile Ad Hoc and Wireless Networks*, pages 52–62, INRIA, Sophia Antipolis, France, March 2003.

[5] R. Bellman. On a routing problem. *Quart. Appl. Math*, 16:87–90, 1958.

[6] B. Bullnheimer, C. Strauss, and R.F. Hartl. An improved ant system algorithm for the vehicle routing problem. In *Annals of Operations Research*, volume 89, pages 319–328, 1999.

[7] D. Camara and A.A.F Loureiro. A GPS/ant–like routing algorithm for ad hoc

networks. In *Proceedings of Conference on Wireless Communications and Networking*, volume 3, pages 1232–1236, Chicago, USA, Septemeber 2000.

[8] G. Di Caro and M. Dorigo. AntNet: Distributed Stigmergetic Control for Communication Networks. *Journal of Artificial Intelligence Research*, 9:317–365, December 1998.

[9] G. Di Caro, F. Ducatelle, and L.M. Gambardella. AntHocNet: An adaptive nature inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications (Special Issue on Self-Organization in Mobile Networking)*, 16(2), 2005.

[10] T. Chen and M. Gerla. Global state routing: A new routing scheme for ad-hoc wireless networks. In *Proceedings of the IEEE international conference on communications (ICC)*, pages 171–175, Atlanta, GA, June 1998.

[11] C. Chiang, H. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks. In *Proceedings of IEEE Singapore International Conference on Networks*, pages 197–211, April 1997.

[12] M. Dorigo, G. Di Caro, and L. Gambardella. Ant colony optimization: A new meta-heuristic. In *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1470–1477, Washington D.C, July 1999. IEEE Press.

[13] M. Dorigo, Maniezzo, and A. V. Colorni. The ant system–optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics*, 26:29–41, February 1996.

[14] M. Dorigo and T. Stutzle. *Ant Colony Optimization.* MIT Press, US, 2004.

[15] M. Dorigo E. Bonabeau and G. Theraulaz. *Swarm Intelligence: From Natural to Artifical Systems.* Oxford University Press, New York, NY, 1999.

[16] K. Fall and K. Varadhan. The ns manual (formerly ns Notes and Documentation), 2002. http://www.isi.edu/nsnam/ns/doc/.

[17] S. Goss, S. Aron, J.-L. Deneubourg, and J. M. Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76:579–581, 1989.

[18] M. Gunes, U. Sorges, and I. Bouazzi. ARA – the ant-colony based routing algorithm for MANETs. In *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'02)*, pages 79–85, Vancouver, B.C., August 2002.

[19] Z. J. Haas. A new Routing Protocol for Reconfigurable Wireless Networks. In *Proceedings of IEEE International Conference on Universal Personal Communication*, pages 1–11, Atlanta, GA, October 1997.

[20] B. R. Haverkort. *Performance of Computer Communication Systems: A Model-Based Approach.* Wiley and Sons, Canada, 1998.

[21] M. Heissenbuttel and T. Braun. Ants-based routing in large scale mobile ad-hoc networks. *Proceedings of the 13. ITG/GI-Fachtagung Kommunikation in verteilten Systemen (KiVS 2003)*, pages 91–99, February 2003.

[22] O. Hossein and T. Saadawi. Ant routing algorithm for mobile ad hoc networks (arama). In *Proceedings of the 22nd IEEE International Performance, Computing, and Communications Conference*, pages 281–290, Phoenix, Arizona, USA, April 2003.

[23] M.T. Islam. *Design, Implementation and Performance Analysis of the Ant Colony Optimization Algorithm for Routing in Ad hoc Networks.* University of Manitoba, 2004.

[24] D. B Johnson and D. A Maltz. Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski and Hank Korth, editors, *Mobile Computing*, volume 353, chapter 5, pages 153–181. Kluwer Academic Publishers, Boston, 1996.

[25] D.B Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*, volume 353. Kluwer Academic Publishers, Boston, 1996.

[26] B. Liang and Z.J. Haas. Hybrid routing in ad hoc networks with a dynamic virtual backbone. In *IEEE Transactions on Wireless Communications.*

[27] L. Liu and G. Feng. A Novel Ant Colony Based QoS-Aware Routing Algorithm for MANETs. *Lecture Notes in Computer Science*, 3612:457–466, July 2005.

[28] V. Maniezzo and A. Colorni. The Ant System Applied to the Quadratic Assignment Problem. *Knowledge and Data Engineering*, 11(5):769–778, 1999.

[29] M. Manner. Table-Driven Routing Protocols in Ad-hoc Mobile Wireless Networks. http://www.cs.helsinki.fi/u/mjmanner/adhoctabledrivenmanner.pdf, 2006.

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

[31] S. Marwaha, C. Tham, and D. Srinivasan. Mobile agents based routing protocol for mobile ad hoc networks. In *Proceedings of the Global Telecommunicationations Conference*, volume 1, pages 163–167, Taipei, Taiwan, November 2002.

[32] H. Matsuo and K. Mori. Accelerated Ants Routing in Dynamic Networks. In *Proceedings of Intl. Conf. On Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 333–339, Nagoya Institute of Technology, Japan, August 2001.

[33] J. Mingliang, L. Jinyang, and Y.C. Tay. Cluster based routing protocol. Technical Report (Internet draft, draft-ietf-manet-cbrp-spec-00.txt), Mobile Computing Group, National University of Singapore, 2000.

[34] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing rotocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[35] Y. Ohtaki, N. Wakamiya, M. Murata, and M. Imase. Scalable and efficient ant-based routing algorithm for ad-hoc networks. *IEICE Transactions on Communications*, E89-B(4):1231–1238, 2006.

[36] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the IEEE INFOCOM - The Conference on Computer Communications*, pages 7–11, Kobe, Japan, April 1997.

[37] G. Pei, M. Gerla, and T. Chen. Fisheye state routing in mobile ad hoc networks. In *Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (ICDCS) Workshop on Wireless Networks and Mobile Computing*, pages 71–78, Taipei,Taiwan, April 2000.

[38] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, volume 24, pages 234–244, New York, NY, USA, 1994.

[39] C. Perkins and E. M. Royer. Ad–hoc on-demand distance vector routing. In *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.

[40] M. Roth and S. Wicker. Termite:Ad-Hoc Networking with Stigmergy. In *Proceedings of IEEE Global Telecommunications Conference*, volume 5, pages 2937–2941, San Francisco, USA, December 2003.

[41] E. Royer and C. Toh. A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, pages 46–55, April 1999.

[42] R. Schoonderwoerd, O. E. Holland, J. L. Bruten, and L. J. M. Rothkrantz. Ant-Based Load Balancing in Telecommunications Networks. *Journal of Adaptive Behavior*, 5(2):169–207, 1996.

[43] Sandip Sen and Partha Sarathi Dutta. The evolution and stability of cooperative traits. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*. ACM Press, 2002.

[44] C. Shen, Z. Huang, and C. Jaikaeo. Ant-Based Distributed Topology Control Algorithms for Mobile Ad hoc Networks. *Wireless Networks*, 11(3):299–317, 2005.

[45] T. Stutzle and U Hoos. Max min ant system. *Journal of Future Generation Computer Systems*, pages 889–914, 2000.

[46] D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning : A case study in routing in dynamic network. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 832–839, Nagoya, Aichi, Japan, 1997.

[47] C. K. Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems.* Prentice Hall, December 2001.

[48] Horst F. Wedde, Muddassar Farooq, Thorsten Pannenbaecker, Bjoern Vogel, Christian Mueller, Johannes Meth, and Rene Jeruschkat. BeeAdHOC: An Energy Efficient Routing Algorithm for Mobile Ad Hoc Networks Inspired by Bee Behavior. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 153–160, Washington, DC, June 2005.

[49] J. Yoon, M. Liu, and B. Noble. Random Waypoint Considered Harmful. In *Proceedings of INFOCOM*, San Francisco, CA, USA.

[50] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: A Library for Parallel Simulation of Large-Scale Wireless Networks. In *12th Workshop on Parallel and Distributed Simulation (PADS'98)*, pages 154–161, Banff, Alberta, CA, May 1998.