Solving the MANET Routing Problem using Ant Colony Algorithm

Subodh M. Iyengar
Soumya N. Pattnaik

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Solving the MANET Routing Problem using Ant Colony Algorithm

Thesis submitted in

May 2010

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Bachelor of Technology

by

Subodh M. Iyengar

(Roll 10606002)

Soumya N. Pattnaik

(Roll 10606061)

under the supervision of

Prof. S.K. Rath

and co-supervision of

Prof. S. Chinara

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India. www.nitrkl.ac.in

Dr. S.K. Rath
Professor

May 7, 2010

Certificate

This is to certify that the work in the thesis entitled *Solving the MANET Routing Problem using Ant Colony Algorithm* by *Subodh M. Iyengar and Soumya N. Pattnaik*, bearing roll numbers 10606002 and 10606061 respectively, is a record of an original research work carried out by them under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr. S.K. Rath

Prof. S. Chinara

Acknowledgment

This thesis would not have been possible without the inimitable support and guidance of many individuals. Even though mere words cannot qualify the help that they have given us, we shall yet attempt to thank them in this feeble manner.

The invaluable advice and spur given by *Prof. S.K. Rath* has inspired us to broaden our horizons and accomplish tasks that we never thought were possible. He believed in us even when we did not believe in ourselves.

Our sincere thanks to *Prof. S. Chinara*, for her constant encouragement and her support in our endeavor to understand our area of research. She taught by example, and we have learnt a tremendous amount with our interactions with her.

It has also been a privelege to have known illustrious individuals such as *Prof. B. Majhi* who has always been there when we have needed help and it has been a wonderful experience working with him.

Last but not the least, we would also like to acknowledge the help of *Prof. P.K. Sa* without whom this thesis would not look as good as it does today.

Subodh M. Iyengar and Soumya N. Pattnaik

Abstract

Mobile ad-hoc networks (MANETs) are a collection of mobile nodes communicating wirelessly without a centralized infrastructure. The biggest challenge in MANETs is to find a path between communicating nodes, that is, the MANET routing problem. The considerations of the MANET environment and the nature of the mobile nodes create further complications which results in the need to develop special routing algorithms to meet these challenges. Swarm intelligence, a bio-inspired technique, which has proven to be very adaptable in other problem domains, has been applied to the MANET routing problem as it forms a good fit to the problem. In this thesis, a study of Ant Colony based routing algorithms is carried out taking into consideration two of the most popular algorithms Ant based algorithms, AntHocNet and the Ant Routing Algorithm (ARA). A thorough analyis of ARA is carried out based on the effect of its individual routing mechanisms on its routing efficacy. The original ARA algorithm, although finds the shortest path between source and destination, is observed to not be competitive against other MANET algorithms such as AODV in performance criteria. Based on the analysis performed, modifications are proposed to the ARA algorithm. Finally, a performance evaluation of the original ARA and the modified ARA is carried out with respect to each other, and with respect to AODV, a state of the art MANET routing algorithm vis-a-vis mobility criteria. The motivation behind the thesis is to realize application of MANETs in real world applications by solving the problem of routing.

Contents

List of Figures

Chapter 1

Introduction

Research into the field of Mobile Ad-Hoc Networks (MANETs) is still in its nascent stages. This is partly because it is not cost effective to realize a fully operational real-life MANET environment such as a MANET test bed for different testing scenarios. As a result, a majority of the developments in MANET technology over the years have happened as a consequence of good results obtained via simulation analyses. Over the years many simulators have been developed, and they have managed to incorporate features that not only simulate network parameters such as noisy channels, but also real world parameters such as weather conditions and the mobility of nodes. Suffice to say that simulator technology has developed a lot in response to the need of answering questions regarding MANET performance in a quick and a cost effective manner. This thesis focuses on the issues related to the adoption of MANET technology in real world situations drawing from the results from simulations. Biologically inspired techniques are employed to solve the problems of MANET routing.

1.1 MANETs

A Mobile Ad hoc network (MANET) is a set of mobile nodes which communicate wirelessly over radio frequencies with no centralized infrastructure. This is in stark contrast to the infrastructure of other networks such as Local Area Networks, or even peer to peer networks. The properties of mobility and wireless communication present huge problems to the creation of such networks and the maintenance of services on these networks. Since nodes keep moving away or towards each other, it is understandable that parameters like Quality of Service (QoS) will suffer unless special schemes are developed to sustain such networks. With the ubiquity of laptops and mobile devices, there is an ever increasing importance for the realization of such networks.

The important considerations [1] in the performance of MANETs are the nature of the participating nodes, and the mobility of the mobile nodes. MANETs are composed of power limited devices with a limited transmission range, so in most cases they will not be able to communicate directly with the destination device. Thus, communication must be relayed through intermediate devices resulting in multiple-hops to the destination. MANETs may also be composed of different types of devices, which have different transmission ranges, this heterogeneous situation results in various problems, for example, unidirectional links [2]. Unidirectional links cause problems during the search for a path from the source to destination. Due to differing transmission ranges of intermediate nodes, a path from the source to the destination might not be valid from the destination to the source since one node might not be able to transmit to its preceding node in the route. The motion of nodes in MANETs results in nodes frequently going out of the transmission range of other nodes, thus interfering with MANET routing. Also, since the MANET is a decentralized network, information about the state of the network is not known to any one node. Thus to support the routing function, nodes frequently exchange data to become “aware” of the state of the network.

Despite the problems of MANETs, MANETs have a tremendous potential to be used in various real-world situations such as battle field scenarios, rescue operations and vehicular networks, where setting up a traditional network infrastructure would be implausible.

Other kinds of wireless networks exist which are closely related to MANETs. Wireless mesh networks [3] are networks in which each node, can act as a router.

In this respect it is similar to MANETs, except that the nodes of a mesh network are not mobile. Wireless sensor networks [4] consist of devices that are power limited, however the devices are stationary and are often greatly removed from one another.

1.2 MANET Routing

Routing in MANETs occurs at the network layer. The objective of routing in MANETs is to find a path between the source and destination over which packets can be forwarded. Since the MANET is a mobile network and the topology of the MANET changes continuously, and due to the other considerations in a MANET environment, additional requirements are imposed on the Routing Algorithm. A MANET routing algorithm should not only find the shortest path between the source and destination, but it should also be adaptive, in terms of the changing state of the nodes, the changing load conditions of the network and the changing state of the environment. This is the reason, the majority of MANETs are connectionless in nature, since connections are less effective in delivering the QoS that is required in the rapidly changing MANET environment and impose additional overhead on the network. The MANETs are also multi-hop in nature, in that packets need to be relayed through other nodes to get to the destination. Thus MANETs require that traditional algorithms be redefined to accommodate these additional requirements.

Every MANET routing algorithm has three essential components: A route discovery mechanism, a route error correction mechanism, and a route maintenance mechanism. The route discovery mechanism finds initial routes between the source and destination nodes, the route maintenance mechanism maintains the routes discovered during the transmission of packets and the route error correction mechanism rebuilds routes when they fail. The mechanisms of routing are discussed in detail in Chapter 4.

1.3 Inspiration from Nature

The central idea of this thesis surrounds the application of Ant Colony Optimization to the problem of MANETs. Ant Colony optimization falls into a class of biologically inspired algorithms that have recently been developed. To name a few, the techniques of Particle Swarm optimization [5] and Bacterial Foraging [6] have been inspired by natural phenomenon. The Ant Algorithm mimics the behavior of ants in nature while they are searching for food. Particle swarm optimization is inspired by the behavior of flocks of birds as they fly in search of food. Bacterial foraging is yet another recent algorithm that simulates the behavior of bacteria searching for food. All these techniques are combinatorial in nature and when viewed in the perspective of optimization involve searching for the optimum solution in a given search space. It has been observed that when these patterns, that are observed in nature, are applied to complex engineering problems, they provide good solutions.

These Nature inspired techniques share a common characteristic, the whole information about the state of the system is contained not in a single entity, but rather some part of the information is stored in many of the entities. This constitutes a swarm intelligence, in which decisions are made by an individual by processing information from not only itself but also other entities. The mechanism of communication of the information and the processing of that information is where the techniques differ. This paradigm forms a great fit for the MANET environment, since it is a decentralized network, in which each node contains only a part of the information about the state of the network.

1.4 Motivation

To realize the real world applications of MANETs, we need efficient routing algorithms for MANETs which can adapt to the dynamic topology and which can handle delivery of packets to the destination while delivering high performance in terms of scalability, and adaptability to the changing topology. The study of Ant

Colony Optimization applied to solving the MANET Routing problem, comes as a result of cognizance of this fact. The Ant Algorithm has been shown to be very adaptive and responsive to changing environmental conditions in other problem domains and hence are a good fit for the MANET routing problem.

1.5 Organization

This thesis is organized as follows. In Chapter 2, an overview of the Ant Colony Algorithm is presented and an understanding of the algorithm is provided through the double bridge experiment and the description of two variants of the ant colony algorithm. In Chapter 3, principles of MANET routing are explained and current routing algorithms in literature are described. In Chapter 4 an analysis of the effect of different routing mechanisms on routing performance is carried out. In Chapter 5 the modifications to the Ant Routing Algorithm are described and results are presented. Chapter 6 concludes the thesis.

Chapter 2

Ant Colony Optimization

2.1 The Ant Colony Meta-heuristic

Ants are creatures of limited intelligence, yet in nature they manage amazing feats such as building nests and finding food. They do this through an organized collaborative behavior that exploits the intelligence of the swarm of individuals in the ant colony. The French entomologist, Pierre Paul Grassé, investigated the social behavior of insects and discovered that ants are capable to react to what he referred to as "significant stimuli", which are signals that activate a genetically encoded reaction. He observed that the effects of these reactions can act as new significant stimuli for both the insect that produced them and for the other insects in the colony. Grassé coined the term *Stigmergy* to describe this particular type of indirect communication in which the workers are stimulated by the performance they have achieved.

> Stigmergy is defined as a method of indirect communication in a self-organizing emergent system where its individual parts communicate with one another by modifying their local environment.

Ant colony optimization (ACO) [7] is an optimization technique inspired by the exploratory behavior of ants while finding food. Ants start from their nest and find different paths to the food. In this context, the local information available to the ant is the path that it took to the destination. However a single ant is not aware of the complete topology of the environment. Ants thus communicate with each other by depositing traces of pheromone as they walk along their path. Subsequent ants that arrive in search of food, base their decisions of which path to take on the pheromone traces left in that locality by the previous ants. This form of communication is indirect, i.e., one ant releases the pheromone information into the environment, and another ant senses that pheromone information from the environment just as Grassé had defined. As more ants travel over a particular path, the concentration of pheromone increases along that path. Pheromones along a path also gradually evaporate decreasing their concentration on that path. The pheromone acts acts significant stimuli since other ants are able to sense the pheromones deposited by each other, and they generally take the path of maximum pheromone concentration. This is how the ants progressively converge on a single optimum path between their nest and the food.

Meta heuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic: either a constructive heuristic starting from a *null* solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. The meta heuristic part permits the low-level heuristic to obtain solutions better than those it could have achieved alone, even if iterated.

2.1.1 The Double Bridge Experiment

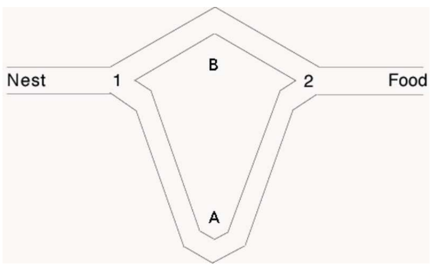

Figure 2.1: Double bridge experiment

A better idea of the Ant Colony Algorithm can be obtained from the double bridge experiment. In Figure 2.1, lower path A is longer than upper path B. In the beginning when there are no pheromones, ants must make a random choice between A and B. However, since the path B is shorter, ants traveling along that path will reach the destination faster and turn back towards the source. Thus pheromone evaporation in path B gets offset by more ants reaching the destination faster. Consequently, this results in other ants favoring path B, increasing the pheromone concentration further. Thus the ant colony optimization technique is capable of finding a shortest path between source and destination through this stigmeric process and is capable of adapting to changes in the links quickly.

2.2 Formulation of ACO

In the context of a computer algorithm, the natural ants are modeled as artificial ants, looking for the optimum path in a graph G. The formulation of ACO [8] as a combinatorial optimization problem can be done as follows:

C = $c1, ..., cn$ is a set of basic components. A subset S of components represents a solution of the problem; $F \subseteq 2^C$ is the subset of feasible solutions, thus a solution S is feasible if and only if $S \in F$. A cost function f is defined over the solution domain, $f : 2^C \rightarrow R$, the objective being to find a minimum cost feasible solution S^*, i.e., to find S^*: $S^* \in F$ and $f(S^*) \leq f(S), \forall S \in F$.

The search space S is defined as follows. A set of discrete variables, X_i , $i = 1, ..., n$, with values $v_i^j \in D_i = \{v_i^1, ..., v_i^{|D_i|}\}$, is given. Elements of S are full assignments, i.e., assignments in which each variable X_i has a value v_i^j assigned from its domain . The set of feasible solutions F is given by the elements of S that satisfy all the constraints.

In the Ant Colony Optimization paradigm, problems are usually modeled as a graph. Let $G(V, E)$ be a connected graph with $n = V$ nodes. Thus the components c_{ij} are represented by either the edges or the vertices of the graph. The objective of the problem is to find a shortest path between the source node V_s and destination V_d. Each edge of G maintains a value τ which denotes an artificial pheromone concentration value over that node which is modified whenever an ant transitions over it. To simulate the natural ant foraging process, three equations are used: Pheromone evaporation, Pheromone increase, and Path selection.

If an ant currently at node i transitions to node j:

$$\tau_{ij} = \tau_{ij} + \Delta\tau \tag{2.1}$$

τ_{ij} is the artificial pheromone concentration over link j at i. The artificial Pheromones gradually evaporate over time, which is modeled by:

$$\tau_{i,j} = (1 - \lambda) * \tau_{ij} \tag{2.2}$$

Where $(1-\lambda)$ is called the pheromone decrease constant. At each node the ant has to make a decision about the next hop over which to travel. To simulate the exploratory behavior of ants the artificial ant makes a stochastic decision based on probabilities of the next hop. The probability of an ant transitioning to node j from node i at node d, where N_i represents a set of neighbors, is calculated by the equation:

$$p_{ij}^{d} = \begin{cases} \frac{\tau_{ij}^{k}}{\sum\limits_{j \in N_i} \tau_{ij}^{k}} & \text{if } j \in N_i \\ d0 & \text{otherwise} \end{cases} \tag{2.3}$$

Where k is called the route select exponent and determines the sensitivity of the ant algorithm to pheromone changes.

2.3 The Ant Algorithm

In ACO, artificial ants build a solution to a combinatorial optimization problem by traversing a fully connected construction graph, defined as follows. First, each instantiated decision variable $X_i = v_i^j$ is called a solution component and denoted by c_{ij}. The solution is constructed by incrementally choosing the components from the Graph $G(V, E)$. As mentioned before, the components can be associated with either the vertices or the edges of the graph.

Each component has a pheromone value associated with it τ_{ij}. The ants move through the graph, and at each node probabilistically choosing the next component to add to the solution determined by the pheromone value of the components. The ant also deposits an amount of pheromone on the component depending on the quality of solution found. The ACO algorithm as described by [9] is shown in Algorithm 1.

Algorithm 1 ACO Meta heuristic

Require: parameters

```
1: while Iterations not complete do
2:    Construct Solutions;
3:    Update Pheromones;
4:    Daemon Actions ; {optional}
5: end while
```

Construct Solutions, chooses a subset of the set of components C. The solution begins with an empty partial solution $s^p = \phi$ and then at each construction step a feasible component is added to s^p. The choice of the next feasible component is made by the path selection equation which depends on the ant algorithm system being used. *Daemon Actions* are usually used to perform centralized actions that cannot be performed by a single ant and that may be problem specific. *Update Pheromones* serves two tasks: To increase the pheromone values of the components which are good, and to decrease the pheromone values of the components which are bad. The pheromone decrease is achieved through evaporation. Many different algorithms have been proposed with different pheromone update equations.

2.4 Ant Algorithm Systems

The path selection equations and the pheromone update equations are dependent on the variant of the Ant Algorithm System being used. Some of the popular systems are described here:

2.4.1 Ant System

Ant system (AS) was the first ACO algorithm to be proposed in the literature [10]. The pheromone values are updated by all the ants that have completed the tour as follows:

$$\tau_{ij} = (1-\lambda).\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

Where m is the number of ants that completed the tour and $\lambda \in (0,1]$ and determines the evaporation rate of the pheromone. $\Delta\tau_{ij}^{k}$ is the quantity of pheromone deposited by ant k on edge (i,j) and is given by:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{L_k} & \text{if (i,j) in tour of k} \\ 0 & \text{otherwise} \end{cases}$$

Where L_k is the length of the tour of ant k. The path selection equation for the AS is governed by $p(c_{ij}|s_k^p)$ which determines the probability of selecting a component c_{ij} or edge (i,j) given the partial solution till now s^p.

$$p(c_{ij}|s_k^p) = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{c_{ik}\in N(s^p)} \tau_{ik}^{\alpha}.\eta_{ik}^{\beta}}$$

$N(s^p)$ represents the neighbors of the components or edges in the partial solution to enable the solution to be constructed further. η_{ij} represents the heuristic cost function of traveling path (i,j) which could be as simple as a greedy cost function $\eta_{ij} = 1/d_{ij}$ where d_{ij} is the distance between vertices i and j.

2.4.2 MAX-MIN Ant System

The MAX-MIN ant system (MMAS) is an improvement over the original Ant system [11]. In MMAS, only the best ant causes the increment of pheromone trails.Also, minimum and maximum values of the pheromone values are explicitly defined.

The pheromone update equation is given by:

$$\tau_{ij} = (1-\lambda).\tau_{ij} + \Delta\tau_{ij}^{best}$$

Where $\Delta\tau_{ij}^{best}$ is defined as:

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}} & \textit{if (i,j) in tour of best ant} \\ 0 & \textit{otherwise} \end{cases}$$

L_{best} is the length of the tour of the best ant. The pheromone values τ_{ij} are also bounded explicitly within a range $[\tau_{min}, \tau_{max}]$. The path selection equation of the MMAS is similar to the one used by AS.

Chapter 3

Routing Algorithms in MANETs

To realize the real world applications of MANETs, we need efficient routing algorithms for MANETs which can adapt to the dynamic topology and which can handle delivery of packets to the destination while delivering high performance in terms of scalability, and adaptability to the changing topology. There are currently various proposed routing algorithms for MANETs in literature. They are classified [12] into reactive [13,14], proactive [15,16] and hybrid [17] algorithms. In reactive algorithms, routes are created only when required between a source and destination. In contrast proactive algorithms maintain the route information throughout the life of the MANET. In proactive algorithms each node broadcasts control information (called HELLO packets) about route information that it has to other nodes periodically, and the nodes which receive that information update their routing tables. Generally, reactive schemes have a high initial connection time, but perform well in high mobility scenarios since the broadcasting of control packets in a proactive algorithm incurs an overhead. Hybrid schemes strive to merge the best of proactive and reactive schemes to produce an algorithm that performs better than each of the classes.

Ant Colony optimization, a bio-inspired meta-heuristic [7] has been applied to the MANET routing problem resulting in algorithms [18–21] to improve MANET performance. The technique is based on ant-agents modifying their environment to guide other ant-agents through the shortest path, a process called stigmergy.

These algorithms have been shown to be very adaptive and responsive to changing environmental conditions in other domains and hence are a good fit for the MANET routing problem.

In this thesis, two popular Ant Colony algorithms in literature are described, the AntHocNet algorithm and the Ant Routing Algorithm (ARA). Both these form a good basis for testing since they encompass the spectrum of routing protocol types, AntHocNet being hybrid in nature, that is, having a proactive component and ARA being purely reactive.

3.1 AODV

Ad-Hoc On-Demand Distance Vector Routing (AODV) [13] is one of the pioneering routing algorithms for MANETs and has become very popular in literature with many newer algorithms comparing their efficacy with respect to AODV. AODV is a purely reactive routing algorithm and establishes a route to the destination only on demand. AODV also avoids the counting-to-infinity problem of other distance-vector protocols by using sequence numbers on route updates.

As mentioned previously, each MANET routing algorithm consists of three major mechanisms: Route discovery, Route maintenance, and Route error correction.

3.1.1 Route Discovery

In AODV, there is no network communication until and unless a connection is needed to send a packet from a source node to a destination node. The source node then broadcasts a Route Request packet (RREQ) which contains information about the source identifier (SrcID), the destination identifier (DestID), the source sequence number (SrcSeqNum), the destination sequence number (DestSeqNum), the broadcast identifier (BcastID), and the time to live (TTL). DestSeqNum indicates the freshness of the route that is accepted by the source. Each node is given a unique SrcSeqNum to identify it and prevent the packets from looping.

Other AODV nodes receiving the RREQ forward this message, and keep a record of node that forwarded the RREQ to it, i.e., the preceding node in the route. When a node receives an RREQ and already has a route to the desired node, it sends a message called the Route Reply (RREP) backwards through a temporary route to the previous node, and subsequently down the chain till the source. The source then begins using the route that has the least number of hops through other nodes. The freshness of the node is determined by the DestSeqNum. A node updates its path information only if the DestSeqNum of the current packet received is greater than the last DestSeqNum stored at the node.

If a RREQ is received multiple times, which is indicated by the SrcSeqNum-SrcID pair, the duplicate copies are discarded. All intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send RREP packets to the source. Every intermediate node, while forwarding a RREQ, enters the previous node address, its SrcSeqNum and its SrcID. Due to the route discovery process, a node can obtain multiple route to the destination. Thus AODV is also a multi-path routing scheme.

3.1.2 Route Maintenance and Error Correction

In AODV there is no explicit route maintenance mechanism. The route maintenance process maintains the freshness of the routes by recycling unused entries in the routing tables after a period of time. The route error correction mechanism prevents the routes created by the Route Discovery process from failing due to link failures during the transmission of packets.

When a link fails, a Route Error message (RERR) is passed to the preceding node in the forwarding chain in the direction of the source node. The previous node then initiates a route discovery process to create a new route to the destination.

AODV attempts to lower the number of messages to conserve the capacity of the network. Each RREQ has a sequence number. Nodes use this sequence number so that they do not RREQs that they have already forwarded. Also, RREQs have a TTL field that limits the number of times they can be forwarded to another node. Each time they are forwarded the TTL is decremented by one. When the TTL goes to zero, the packet is discarded. Also, if an RREQ fails, another RREQ may not be sent until twice as much time has passed as the timeout of the previous route request.

The advantage of AODV is in its reactive nature, that it incurs no network overhead due to transmission of control packets on links that are not being used. AODV is also a very simple algorithm, and does is not high time or memory complexity. However AODV requires more time to establish a connection, and the initial communication to establish a route is heavier than some other approaches.

3.2 The AntHocNet Algorithm

The Anthocnet algorithm is a hybrid routing algorithm which is based on the Ant colony algorithm and contains both reactive and proactive elements [22]. The algorithm in reactive in that it keeps routing information of only those nodes involved in communication. It is proactive because nodes exchange routing information while the communication is going on. The algorithm of AntHocNet is shown in Algorithm 2, and Functions 3 and 4.

3.2.1 Route Discovery

At a node, when a packet is received by the network layer from the higher layers, the node checks to see if routing information is available for destination d over any of its neighbors. If found, it forwards the packet over that node, if not, it broadcasts a forward ant to find a path to the destination. When a forward ant is received by an intermediate node, the node checks if it has routing information for the destination over its neighbors and if found, it unicasts the ant over that neighbor, else, it broadcasts the forward ant. Loops are prevented by a sequence id mechanism and endless flooding is restricted by enforcing maximum number of hops for the ant. Once the ant reaches the destination it becomes a backward ant and it follows the same path it came from, back to the source. Any duplicate

Algorithm 2 AntHocNet Algorithm

Require: Packet p

```
p = getPacketfromMAC();
if p is FANT then
    if FANT seen then
        end;
    end if
    n=nextHop(p);
    if n==false then
        floodFANT(p.destination);
    else
        sendToMAC(p,n);
    end if
end if
if p is BANT then
    if p.destination==this node then
        updatetable(p.destination,p.prevHop,p);
    else
        updatetable(p.destination,p.prevHop,p);
        n=getNextNodeBANT(p);
        sendToMAC(p,n);
    end if
end if
if p is error then
    bufferpacket(p);
    floodFANT(p.destination);
end if
if p is packet then
    n = nextHop(p);
    if n == false then
        bufferpacket(p);
        floodFANT(p.destination);
    else
        sendToMAC(p,n);
    end if
end if
if p is HELLO then
    i = p.source;
    for every destination d is Hello do
        $v_i^d$ = best value of $\tau_{ij-1}^d$ in Hello;
        $k_{ji}^d = (v_i^d)^{-1} + (j^{-1})^{-1}$;
        $w_{ji}^d = k_{ji}^d$;
        if ($\tau_{ji}^d ! = 0$) and $v_i^d$ is not virtual then
            $\tau_{ji}^d = w_{ji}^d$;
        end if
    end for
end if
```

Function 3 updatetable(destination d,prevHop j,packet p)

$\tau_{ij}^d = r * \tau_{ij}^d + (1-r) * (c_i^d)^{-1}$;
return;

Function 4 nextHop(p)

$d = p.destination$;
for all τ_{ij}^d **do**
  $p_j = \frac{(\tau_{ij}^d)^\beta}{\sum_{j \in N_i^d} (\tau_{ij}^d)^\beta}$;
end for
if p is FANT **then**
  $p_{j2} = \frac{(\omega_{ij}^d)^\beta}{\sum_{j \in N_i^d} (\tau_{ij}^d)^\beta}$;
end if
$n =$ choose j from all p_j and p_{j2};
if No n and No τ_{ij}^d **then**
  return false;
else
  return n;
end if

forward ants reaching the destination are discarded. Each node in the backward path updates it routing table with the information of the backward ant. When a node receives a backward ant it updates its pheromone information.

When the pheromone reaches the source it calculates the probability of its next hop on the basis of a probability function similar to Equation 2.3.

3.2.2 Route Maintenance

Anthocnet uses a proactive updates to improve route quality. Nodes periodically broadcast information about the best pheromone values to each destination at that node. The neighboring nodes on receiving the information, then adjust the value of their existing pheromone values of the routing table entries to every destination over the broadcasting node. This diffusion process is slow and could result in new paths being discovered to the destination. However, these paths are not reliable and are thus not used directly in packet forwarding, and are marked as virtual pheromones to be explored later during another route discovery phase. Neighborhood discovery is done through the periodic broadcast of HELLO messages.

3.2.3 Route Error Correction

A link error may be detected when a Hello message is not received from a neighbor for a timeout period, or if a packet fails to transmit though a link. The algorithm corrects the routing table to reflect the link failure. In the case of packet sending failure, Anthocnet checks for alternative routes and if not found initiates a route repair process. It also broadcasts a link failure notification to inform its neighbors about the change in routing information.

3.3 The ARA Algorithm

ARA was one of the first Ant Colony algorithms proposed and unlike AntHocNet, it is a purely reactive MANET routing algorithm. It does not use any HELLO packets [15] to explicitly find its neighbors. The algorithm for ARA is shown in Algorithm 5 and Functions 6 and 7.

3.3.1 Route Discovery

At a node when a packet is received by the network layer from the higher layers, the node checks to see if routing information is available for destination d in its routing table. If found, it forwards the packet over that node, if not, it broadcasts a forward ant to find a path to the destination. At every intermediate node, if the forward ant has been seen before, it is discarded, else it is flooded again. When the ant reaches the destination it is sent back along the path it came, as a backward ant. All the ants that reach the destination are sent back along their path. At every node in the backward path, if the backward ant is seen before, it is discarded, else it is forwarded. Nodes modify their routing table information when a backward ant is seen according to number of hops the ant has taken. When a

Algorithm 5 ARA

Require: Packet p

```
p=getPacketfromMAC();
if p is FANT then
    if FANT seen then
        end;
    end if
    if nodeIsDestination(p) then
        sendBANT();
    else
        floodFANT(p.destination);
    end if
end if
if p is BANT then
    if nodeIsDestination(p) then
        updatetable(p.destination,p.prevHop,p);
        debufferPackets(p.destination);
    else
        updatetable(p.destination,p.prevHop,p);
        n=getNextNodeBANT(p);
        sendToMAC(p,n);
    end if
end if
if p is error then
    Node n=getRoute(p.destination);
    if n==false then
        if n==p.packetSource then
            bufferpacket(p);
            floodFANT(p.destination);
        else
            n=getNextNodeBANT(p);
            sendToMAC(p.packet,n);
        end if
    else
        sendToMAC(p.packet,n);
    end if
end if
if p is packet then
    n = nextHop(p);
    if n == false then
        n = getNextNodeBANT(p);
        sendError(p.n);
    else
        sendToMAC(p,n);
    end if
end if
```

Function 6 Updatetable(destination d,nextHop j,Packet p)

```
τ_ij^d = (c_i^d)^-1;
return;
```

$\tau_{ij}^{d} = (c_i^d)^{-1}$;
return;

Function 7 nextHop(p)

d=p.destination;
for all τ_{ij}^{d} **do**
  $p_j = \frac{(\tau_{ij}^{d})}{\sum_{j \in N_i^d} (\tau_{ij}^{d})}$;
end for
$n =$ choose j from all p_j;
if no n and no τ_{ij}^{d} **then**
  return false;
else
  return n;
end if

route is found the packet is forwarded over the next hop stochastically according to equation 2.3.

3.3.2 Route Maintenance

ARA does not flood any proactive packets. The route is maintained through the adjustment of pheromone values of the links present in the node routing tables. Whenever a particular link is selected as the next hop, the pheromone value of that link for the destination of that packet is incremented by a constant value in the routing table, according to equation 2.1. Pheromone values also made to constantly decrease according to equation 2.2. ARA prevents loops by assigning each packet a sequence id. If the packet is seen again, a DUPLICATE_PACKET error is sent to the previous node. Consequently the previous node sets the pheromone value of the link to the origin of the DUPLICATE_PACKET to zero. ARA tries to find another path to the destination.

3.3.3 Route Error Correction

A link error may be detected if a packet fails to be sent over the next hop that is selected. In this case, ARA sets the pheromone value of the link to the next hop to 0 and tries to find an alternative next hop. If it fails, it sends a ROUTE_ERROR message to the previous node from which the packet came with a copy of the message, where the process is repeated. If the ROUTE_ERROR reaches the source, a new route discovery process is initiated.

Chapter 4

Component Based Routing Algorithm Analysis

The efficacy of an algorithm is determined by the techniques or mechanisms used for each routing task. Since each algorithm uses a wide variety of techniques and there are many tasks involved in routing, no one algorithm can be said to be better than all others for every possible scenario, since the techniques itself perform differently in different scenarios. It thus becomes essential that instead of comparing the performance of algorithms as a whole, we instead start to compare and thus qualify the efficacy of the components of the algorithm, that is, the "choices" to be made or techniques used for each routing task.

If such a qualification is done, it would be possible to provide a set of choices, which act as components, for each MANET routing task, paving the way for a component based routing algorithm design. That is, the components can be mixed using a software tool with a simple graphical user interface and a MANET routing algorithm can be designed as per the requirements and environment of a specific real world application. This changes our view of routing algorithm design and comparison from a whole process to design from a set of building blocks. In this Chapter, the idea of component based routing algorithm design is presented and an initial attempt is made to qualify the efficacy of the components through an analysis of the three essential mechanisms of routing by taking various "choices" for each of them with respect to mobility of the nodes.

The analyses of the components of routing are carried out within the context of the ARA algorithm. Drawing from the analyses of the mechanisms of routing, modifications are proposed to the algorithm to improve its efficacy.

4.1 Routing Mechanisms

4.1.1 Route Discovery

The two choices adopted for route discovery are the FANT flood technique [7] and FANT forward technique [22]. In the flooding scheme, when a FANT is sent through the network, at each intermediate node, the FANT is flooded to all its neighbors. To limit flooding, a maximum hop count is imposed on the FANT and when the number of hops of the FANT exceeds the maximum hops, the FANT is dropped. In the FANT forwarding scheme, when a FANT reaches an intermediate node, the node checks it routing table to see whether it has a route to the destination over any of its neighbors. In ARA, a route is indicated by a positive pheromone value in the node's pheromone table over any of its neighbors to the FANT destination. If such a neighbor is found, the FANT is forwarded to only that neighbor, else, it is flooded to all its neighbors as in the flood scheme.

4.1.2 Route Maintenance

The pheromone updates are responsible for maintaining the routes during the motion of the nodes of the MANET. In Ant routing algorithms, pheromone updates thus play a critical role in the performance of the algorithm. The authors of [23] studied and classified the various pheromone update functions used in ant algorithms.

Three pheromone equations are considered to test route maintenance capability and are given below. The first and second pheromone equations contrast the implementation of the pheromone decay process. The third equation simulates an exponential decay. τ_{ij}^{d} denotes the pheromone concentration over link (i, j) for a destination d. t denotes the time interval between the sending of a forward ant and the receipt of the backward ant, and hops is the total number of hops made by the ant.

- Discrete Decay : The pheromone decay and increase functions are modeled by Equations (2.2) and (2.1). The initial pheromone is set using a function of number of hops as well as of time.

  $$\tau_{ij}^{d} = 2/(hops + t)$$

  The decay process is treated as a discrete one where every discrete interval of time the pheromone is decreased by Equation (2).

- Continuous decay [24]: The pheromone decay functions are similar to that of discrete decay. Even though the decay events occur discretely, it is simulated to be continuous by decreasing an amount of pheromone proportional to the time elapsed since the last decay (t_{obs}).

- Gamma pheromone function [23]: The pheromone update equation is given by:

  $$\tau_{ij}^{d} = \tau_{ij}^{d}.e^{(t-t_{obs}).q}$$

  It is a continuous pheromone decay process characterized as the gamma decay function.

4.1.3 Route Error Correction

When a link fails due to the mobility of the node, the MANET algorithm must correct it in order to ensure delivery of the packet to the destination. In MANET algorithms, two mechanisms are widely used for route error correction: local route repair [19], and route error back-propagation [18]. In local route repair, if a link at a particular node fails, the algorithm buffers the packet and sends out a new FANT to discover a route to the destination. Once a route is found, the packet is forwarded over that route.

In the route error back-propagation mechanism, if a link error occurs at a node and another route to the destination does not exist at that node, a ROUTE-ERROR packet is sent to the previous node in the forwarding chain. This is repeated until the ROUTE-ERROR packet reaches the source and then a route discovery process is initiated.

4.1.4 Simulation Parameters

ARA is implemented on Qualnet version 4.0 [25] and simulations for each "choice" are run on the Qualnet simulator. The simulations are conducted on an area of 1500m x 1500m. Node mobility is restricted to a maximum speed of 10 m/s and according to the random waypoint [16] mobility model. 802.11b is used as the underlying MAC layer protocol with a propagation limit of -111dB. Tcp-lite is used as the application layer protocol. Simulation time for each instance of an experiment is 300s. For each experiment the sample size is 5. 18 Constant bit rate connections are configured between the nodes. The random seed for the simulation is initially set to 300. Through each experiment various performance metrics of the algorithm are measured in terms of Packet Delivery Ratio, End-to-End delay, Jitter, and Throughput at receiver node.

Through experiential observations the maximum hops for the FANT is set to 10. Other parameters include a pheromone decrease constant of 0.4, a pheromone increase constant of 0.6, a decrease interval of 5s, and a route select exponent of 3. Measurements are taken by varying pause time, which is indicative of the mobility of the network. Pause times are taken to be 0s, 30s, 60s, 90s, 120s, and 150s. Lower pause times indicate a greater mobility of the nodes in the network.

Route Discovery

Figures (4.1, 4.2, 4.3, 4.4) show the variation of the Delivery ratio, Throughput, Delay and Jitter for the ARA algorithm in both forward and flood mode of route discovery. The route correction mechanism used in this case is the error back-propagation algorithm and for route maintenance the discrete pheromone

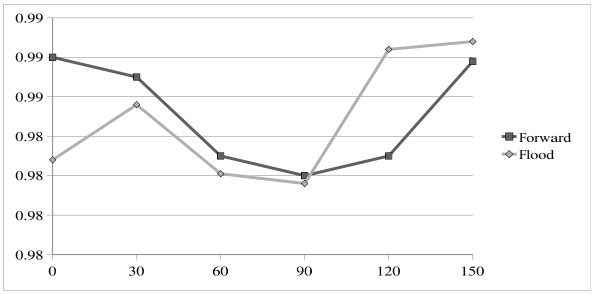

Figure 4.1: Delivery Ratio vs. Pause time

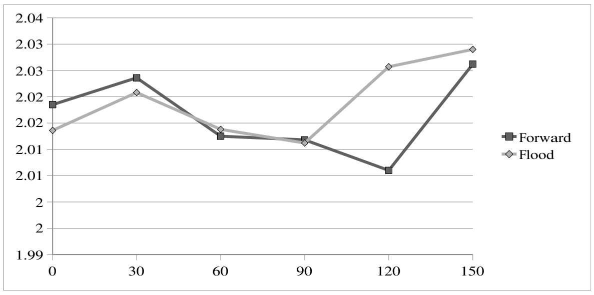

Figure 4.2: Throughput vs. Pause time

decay equation is used.

For the FANT flood technique, the delivery ratio decreases till about a pause time of 90, and then increases where its delivery ratio becomes better than that of the FANT forward technique. The throughput values of FANT flood and FANT forward are quite close to each other for lower values of pause time, but the FANT flood technique exceeds the throughput of FANT forward technique at a pause time of 90. The critical turning point for the FANT flood technique in the case of the delay metric is also a pause time of 90, where it shows less delay than the FANT forward technique. For jitter however, both FANT flood and forward techniques show similar values, with there being no critical point where one technique outperforms the other.

The results for the route discovery mechanism reveal an interesting trend. The FANT forwarding technique does better in situation of high mobility, that is, in situations having a lower pause time. However in cases of lower mobility, the FANT flood technique does better in the metrics of packet delivery ratio, throughput,

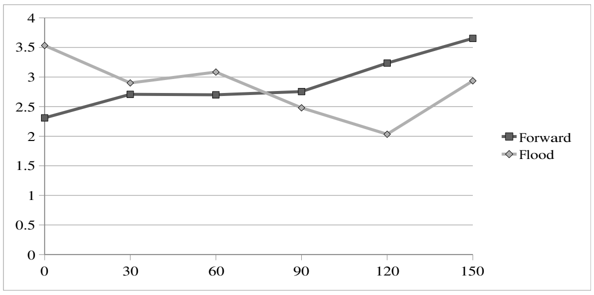

Figure 4.3: Delay vs. Pause time

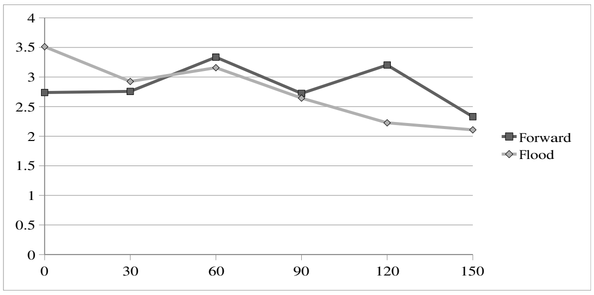

Figure 4.4: Jitter vs. Pause time

delay and jitter.

A possible explanation for this observation is that in lower mobility situations, the FANT flood technique causes a lot of overhead, and increases the time required to find a route to the destination. Also, the route that is found may expire quickly, and thus the overhead incurred to find multiple paths does not give the expected payoff. The FANT forward technique forwards packets as soon as a route is available and this mechanism does not incur much overhead. The route correction mechanism handles link failure cases quite well as shown in the comparison section. In situations of lower mobility, the fact that the FANT is flooded causes the flooding technique to discover multiple paths and the paths so discovered remain valid for a longer time thus providing a load balancing due to the stochastic nature of the ARA next hop selection.

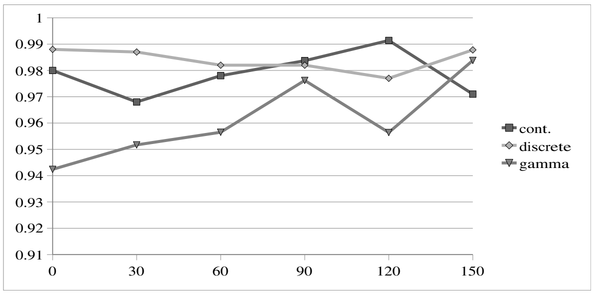

Figure 4.5: Delivery Ratio vs Pause time

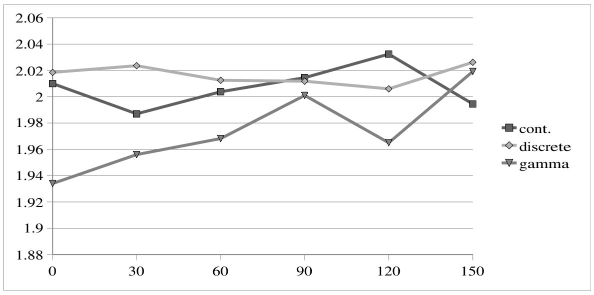

Figure 4.6: Throughput vs Pause time

Route Maintenance

Figures (4.5, 4.6, 4.7, 4.8) present the results for the different pheromone update equations in route maintenance. The route error correction mechanism used is the back-propagation technique and for route discovery, the FANT forward technique is used.

In the delivery ratio metric the gamma function shows a continuously increasing trend with increasing pause time, equaling the performance of the discrete and the continuous decay functions at a pause time of 150. The discrete decay function has a better value than the continuous decay function at lower values of pause time with the critical point being at a pause time of 90 where the continuous function performs better than the discrete function. In the throughput metric, the gamma function shows a similar increasing trend, while the discrete and continuous functions show the same trend as they did for the delivery ratio metric. For the delay metric, the discrete decay function performs consistently better than the continuous function and the gamma function shows a continuously

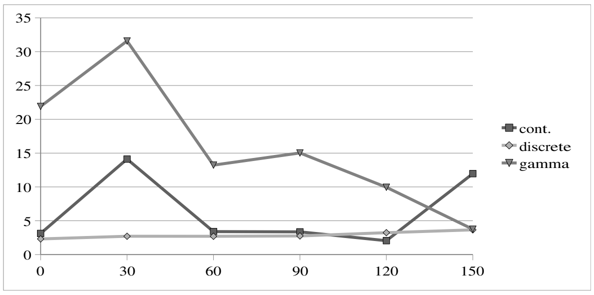

Figure 4.7: Delay vs Pause time

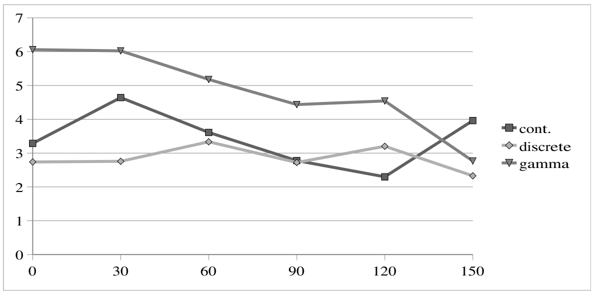

Figure 4.8: Jitter vs Pause time

decreasing trend. For the jitter metric, the gamma function decreases continuously with increasing pause time and the discrete function performs better for the lower values of pause time against the continuous function.

The pheromone update equation, as already stated forms the most important aspect of the ARA algorithm. Proof of this can be observed from the results obtained. The exponential decay gamma pheromone equation does not do as well as either of the other simple pheromone decay equations in the metrics presented. The gamma equations improve in performance as the mobility of the nodes decreases, that is, pause time increases. It is also observed that in cases of higher mobility, the discrete decay function performs better than the continuous decay function.

The reason could be attributed to the fact that in high mobility conditions, the discrete function, due to slower decay, maintains a larger amount of pheromone on the nodes, whereas the continuous function, due to the extra multiplier for time, and the exponential decay gamma equations, which cause decay to happen very fast, cause the pheromones at the nodes to approach the zero value quickly. In ARA, if a pheromone approaches zero value, then the route ceases to exist and a route correction mechanism needs to be initiated. The overhead of the route correction is more pronounced in the higher mobility scenarios where route failures happen often and have to compete with these zero pheromone generated, possibly spurious, route correction events. The route failures reduce with decreasing mobility and thus all three equations acquire similar values for the metrics as the mobility reduces. This reasoning is supported by another observation, which is not presented in this paper, that the continuous decay and gamma equations have a high route error packet generation to link failure ratio. This means that the algorithm was generating too many route correction packets for the number of link failures that had occurred. Thus there must be some other cause of the route error packets.

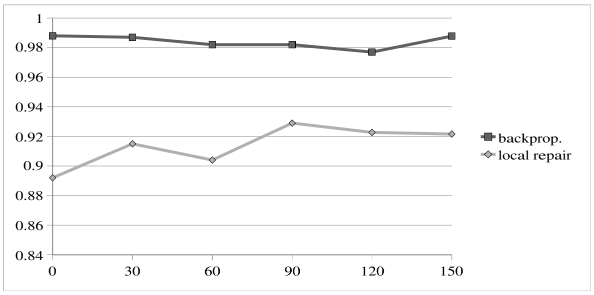

Figure 4.9: Delivery Ratio vs. Pause time

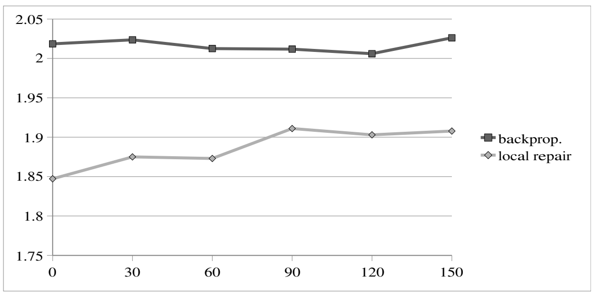

Figure 4.10: Throughput vs. Pause time

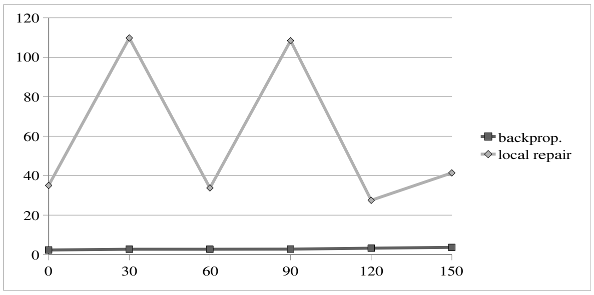

Figure 4.11: Delay vs. Pause time

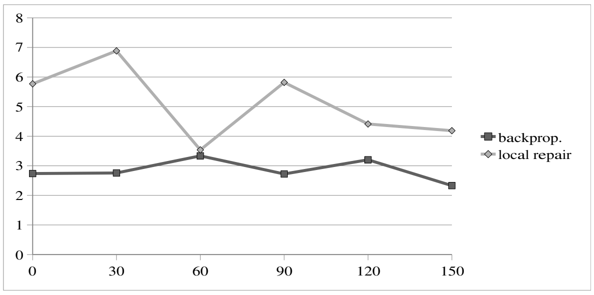

Figure 4.12: Jitter vs. Pause time

Route Error Correction

Figures (4.9, 4.10, 4.11, 4.12) respectively analyze the Delivery ratio, Throughput, Delay and Jitter of route back-propagation and local route repair with respect to changing pause times. Route discovery is done through the FANT forward technique and the discrete pheromone equation is used for route maintenance.

The results for the route error correction are clearly defined. The route error back-propagation technique performs better than the local route repair mechanism for all the metrics. The local route repair mechanism improves in performance though, as the mobility of the nodes decreases. The performance of the error back-propagation algorithm remains almost constant for all the values of pause time.

The local route repair algorithm used in this paper initiates a route discovery process all the way to the destination. It might be possible to design a better route repair strategy that returns a valid route from an intermediate node which becomes a sort of a bootstrapping process, however the same problems as that in flooding hamper the performance of the local route repair technique.

Chapter 5

Proposed modifications to ARA

In [26] the authors compare ARA with the AODV algorithm. Even though the ARA algorithm performs comparably to the AODV algorithm, the performance of the ARA algorithm leaves much to be desired. The adaptive nature of ant colony algorithms should in theory enable ARA to do much better than its purely reactive AODV counterpart. There is tremendous potential in this technique, since other algorithm such as AntHocNet [19], a hybrid protocol, have been compared to AODV and shown to have better performance. ARA being a reactive protocol has the potential to be used in scenarios of high mobility rather than proactive and hybrid algorithms. The improvements proposed by the authors of ARA to the algorithm in [26] over the original ARA algorithm [18] take into consideration some factors, however some other important factors affecting the performance of the ant routing algorithm are neglected. In this Chapter modifications are proposed to the algorithm to help ARA realize its potential for use in high mobility scenarios.

Since pheromone updates are responsible for the route maintenance of the ARA algorithm, they play a critical role in the performance of the ant algorithm. The authors of [23] study and classify the various pheromone update functions used in ant algorithms. The original ARA algorithm uses a function of only the number hops to compute the initial pheromone value during the route discovery, that is, it uses a classic pheromone filter. This metric might be suitable for static routing, but it is not robust for highly mobile nodes. Pheromone equations used in different algorithms can be classified under different categories. Two important classes are the Classic pheromone filter, where route quality is not taken into consideration, for example the original ARA pheromone equation, and the Gamma pheromone filter, which takes time and route quality into consideration.

5.1 The Time Metric

Taking path quality into consideration we devise a type of Gamma Pheromone filter for ARA to update pheromone values. A similar metric is used by AntHocNet during its routing [19]. It is shown in [23] that the Gamma Pheromone filters show a better performance over the classic pheromone filters. The modified pheromone update equation sets the initial value of the pheromone as:

$$\tau_{ij}^{d} = 2/(hops + t) \tag{5.1}$$

The pheromone update is done as per Equations (2.1) and (2.2). Where τ_{ij}^{d} denotes the pheromone concentration over link (i, j) for a destination d. t denotes the time interval between the sending of a forward ant and the receipt of the backward ant, and hops is the total number of hops made by the ant.

The inclusion of time in the equation creates a pheromone gradient from source to the destination point depending on the time it takes for the backward ant to reach the node that forwarded it. In the case of only FANT hops being taken into consideration, there form many paths with a similar gradient, however the time metric creates a marked difference in the path gradient and thus the packet would be stochastically forwarded over the path with the greatest pheromone gradient. This metric is thus expected to produce better results than if only number of hops is considered.

5.2 Pheromone decay process

Also, unlike the usual pheromone decay processes, the pheromone decay of the modified ARA is modeled as a discrete process rather than a continuous one.

As shown in section 3.3.2 in a discrete pheromone decay, the pheromone decreases asynchronously after a particular time interval and the discrete process allows more pheromone to be available on the routes so that routes live longer. The continuous pheromone update process is actually discrete in nature and it is emulated as a continuous process by decreasing the pheromone by a factor proportional to the time interval between consequent packets that are observed at a node.

5.3 Route Selection Exponent

In the route selection equation, that is equation 2.3, the original ARA algorithm uses a route select exponent of $k = 1$. This thesis introduces the route select exponent to the ARA algorithm and works with a $k = 3$. This increases the sensitivity of the algorithm to changes in pheromone values, making it more adaptive in nature.

5.4 Routing mechanisms

The flooding procedure in ARA during route discovery is replaced by a forwarding technique as shown in section 3.3.1 and similar to that found in Anthocnet [19], that is, if a route exists from a node to the destination, the FANT is forwarded over that route instead of flooding. This reduces the overhead during the route discovery process.

A maximum number of hops is imposed on the forward ant as well as the packet to prevent excessive flooding of control packets. The maximum number of hops is set through experiential observation of a reasonable amount of time it takes for the FANT to reach the destination from the source.

Through the observations in section 3.3.3 it is decided to use the error back-propagation method in the modified ARA algorithm. The resulting algorithm outperforms AODV and the original ARA in various parameters as described in the next section.

5.5 Simulation Results and Discussion

The simulation parameters for the testing of ARA are the same as in Section 4.1.4.

Fig. 5.1, 5.2, 5.3, 5.4 show the performance measured in terms of Packet Delivery Ratio, Throughput, End-to-End delay and Jitter for various values of pause time respectively.

The observations show the efficacy of the modifications to the ARA algorithm. For the delivery ratio, throughput, and jitter metrics, the modified ARA algorithm performs better then the ARA and the AODV algorithm for all the measured values of pause time. For the value of delay, the performance of the ARA algorithm, while remaining better than the original ARA algorithm, is not as good as AODV.

The differences between AODV and Modified ARA are pronounced. While AODV uses no route maintenance mechanism other than a timeout to delete stale routes, ARA uses a route maintenance mechanism to gradually modify the "freshness" of the routes. Modified ARA shows an advantage over the original ARA algorithm due to the inclusion of the time metric and this advantage is clearly shown in Figure 5.3. Additionally, the route selection exponent makes the ant route selection equations more sensitive to changes in pheromone values. These changes in pheromone values are indirectly indicative of the topology of the MANET, and causes the ant route selection equations to select varied routes. Thus a form of load balancing is set up during the route maintenance process which causes an improvement in both the Jitter as well as Throughput values.

Also during route discovery, modified ARA uses the FANT forwarding scheme, which causes routes to be found faster than ARA which uses FANT flooding in situations of high mobility. Modified ARA performs well in comparison to AODV in the Delay metric in situations of high mobility. In lower mobility scenarios, AODV delivers a better end to end delay. However the objective of using ARA for routing in high mobility scenarios has been realized.

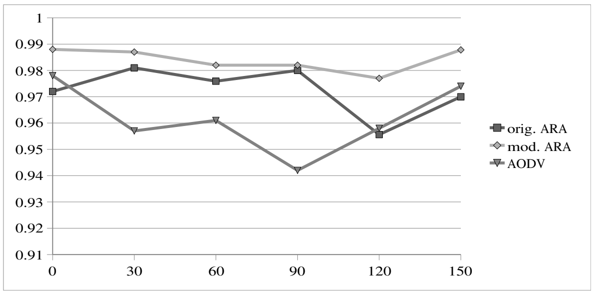

Figure 5.1: Delivery Ratio vs. Pause time

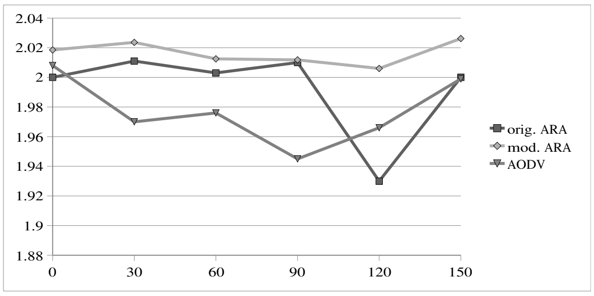

Figure 5.2: Throughput vs. Pause time

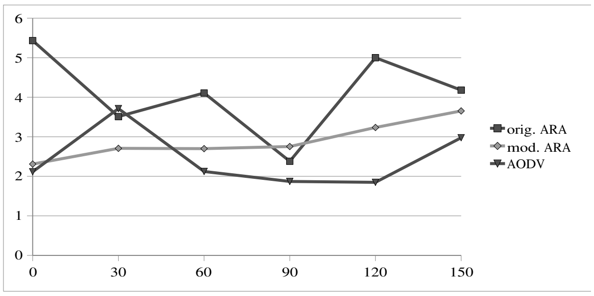

Figure 5.3: Delay vs. Pause time

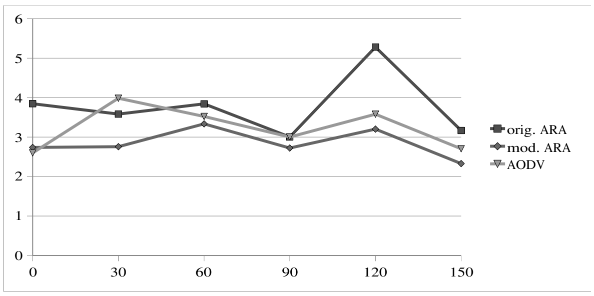

Figure 5.4: Jitter vs. Pause time

Chapter 6

Conclusion

There is an increasing relevance of MANET infrastructure with the proliferation of mobile devices and the need to support services on these devices in both remote and heterogeneous environments. But as mentioned in Chapter 1, overcoming the problems associated with MANET routing is of paramount importance to the real world realization of these networks. Again, due to the diverse conditions in which MANETs are expected to operate, a “one size fits all” solution is infeasible. Thus there is a need to develop algorithms which work well under certain constrained conditions.The component driven MANET routing algorithm design proposed in this thesis is a step towards that goal. Analyses carried out on the mechanisms of routing, enabled the selection of a set of techniques that would perform well in high mobility scenarios.

A modified version of the ARA algorithm is proposed based on the findings of the analyses carried out on the components of routing. It is observed how modified ARA performs in comparison to a state of art algorithm of MANET routing, namely, AODV. The routing algorithms were analyzed through various simulation based experiments and it was seen that modified ARA performed better in comparison to the other two algorithms in some metrics in terms of varying mobility. The modifications to ARA were drawn from not only this analysis but also inspired by the functioning of other ant colony routing algorithms.

Ant Colony algorithms show a lot of promise, and have been shown to be very adaptive to changing environments. They show comparable performance to existing state of the art MANET routing algorithms. However, their performance must be improved further and they must be able to solve the problems of heterogeneous networks before being recognized as state of the art solutions to the MANET routing problem.

Bibliography

[1] S.Corson. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. http://www.ietf.org/rfc/rfc2501.txt.

[2] Al Huda Amri and et. al. Scalability of manet routing protocols for heterogeneous and homogenous networks. *Computers and Electrical Engineering*, 2008.

[3] Ian F. Akyildiz, Xudong Wang, and Weilin Wang. Wireless mesh networks: a survey. *Computer Networks and ISDN Systems*, 47(4):445–487, 15 March 2005.

[4] Röme, Kay, and Friedemann Mattern. The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6):54–61, December 2004.

[5] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks, IV*, pages 1942–1948, 1995.

[6] K. Passino. Biomimicry of bacterial foraging for distributed imization and control. *Control Systems Magazine, IEEE*, 22(3):52 –67, Jun 2002.

[7] Dorigo M. and G. Di Caro. Ant colony optimization: a new meta-heuristic. In *Proceedings of the Congress on Evolutionary Computation*, 1999.

[8] V. Maniezzo. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. *INFORMS Journal of Computing*, 11(4):358–369, 1999.

[9] Dorigo M., G. Di Caro, and L. M. Gambardella. Ant algorithms for discrete optimization. *Artificial Life*, 5(2):137–172, 1999.

[10] A. Colorni, M. Dorigo, and V. Maniezzo. Distributed imization by ant colonies. In *Proceedings of ECAL'91, European Conference on Artificial Life*. Elsevier Publishing,Amsterdam, 1991.

[11] Thomas Stützle and Holger H. Hoos. Max-min ant system. *Journal of Future Generation Computer Systems*, 16:889–914, 2000.

[12] E.M. Royer and C.K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. In *IEEE Personal Communications*, volume 6, April 1999.

[13] C. Perkins. Ad hoc on-demand distance vector routing. Internet-Draft, draft-ietf-manet-aodv-00.txt, November 1997.

[14] *DSR: The Dynamic Source Routing protocol for multi-hop wireless ad hoc networks*, chapter 5, pages 139–172. Addison-Wesley, 2001.

[15] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance vector (dsdv) for mobile computers proc. of the sigcomm 1994 conference on communications architectures, protocols and applications. pages 234–244, Aug 1994.

[16] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). RFC 3626: Optimized link state routing protocol (OLSR), Oct 2003.

[17] Z.J. Haas and M.R. Pearlman. The zone routing protocol (zrp) for ad-hoc networks. IETF MANET working group, Internet Draft, June 1999.

[18] M. Günes, Sorges U, and I. Bouazizi. Ara - the ant-colony based routing algorithm for manets. In *In proceedings of the 2002 ICPP Workshop on Ad Hoc Networks (IWAHN 2002)*, pages 79–85. IEEE Computer Society Press, August 2002.

[19] G.A. Di Caro, F. Ducatelle, and Gambardella L.M. 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(5), October 2005.

[20] M. Roth and S. Wicker. Termite: A swarm intelligence routing algorithm for mobile wireless ad-hoc networks, stigmergic imization. *Studies in Computational Intelligence*, 34:155–185, 2006.

[21] O. Hussein and T. Saadawi. Ant routing algorithm for mobile ad-hoc networks. In *The International Performance Computing, and Communications Conference (IPCCC), Phonix, Arizona*, April 03.

[22] G. A. 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(5):443–455, September-October 2005.

[23] M. Roth and S. Wicker. Asymptotic pheromone behavior in swarm intelligent manets. In *Proceedings of the Conference on Mobile and Wireless Communication Networks (MWCN 2004)*, pages 335–336, 2004.

[24] J. Broch, D. A. Maltz, D. B. Johnson, Y.C. Hu, and J. Jetcheva. A performance comparison of multihop wireless ad hoc network routing protocols. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 85–97, 1998.

[25] The qualnet 4.0 programming manual.

[26] M. Gunes, M. Kahmer, , and I. Bouazizi. Ant-routing-algorithm(ara) for mobile multi-hop ad-hoc networks - new features and results. In *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net'2003), Mahdia,Tunisia*, 25-27,June 2003.