Comparison of different Ant Colony Based Routing Algorithms

Vasundhara Uchhula
Dharamsinh Desai University
Nadiad, Gujarat
India
vasundhara.uchhula@gmail.com

Brijesh Bhatt
Dharamsinh Desai University
Nadiad, Gujarat
India
brij_bhatt@hotmail.com

ABSTRACT
A Mobile Ad-Hoc Network (MANET) [1] is a collection of wireless mobile nodes forming a temporary network without using centralized access points, infrastructure, or centralized administration.

Many researchers have developed proactive and reactive algorithms for MANETs. DSR and AODV are reactive algorithms establishing paths only when they are needed. When a node has some data to send to another node, it searches for a path by flooding the network with control messages. Their dissemination introduces some delay before data packets can be sent and reactive routing algorithms are inefficient when there is much continuous but intermittent traffic in the network. DSDV on the other hand, is a typical proactive routing algorithm. They prepare paths to all destination nodes beforehand and maintain them by exchanging control messages periodically. They require a network to carry a lot of control traffic into a network.

Swarm Intelligence (SI) [1] is an artificial intelligence technique based around on the study of collective behavior in decentralized, self-organized systems. Ant Colony Optimization is popular among other Swarm Intelligent Techniques. Ants-based routing algorithms have attracted the attention of researchers because they are more robust, reliable, and scalable than other conventional routing algorithms. Since they do not involve extra message exchanges to maintain paths when network topology changes, they are suitable for mobile ad-hoc networks where nodes move dynamically and topology changes frequently.

In this paper a detailed comparison of different Ant based algorithms is presented. The algorithms discussed here are Ant Based Control Routing, Ant Colony based Routing Algorithm Routing, Probabilistic Emergent Routing Algorithm, AntHocNet, AntNet.

General Terms
Ant Colony Optimization, Swarm Intelligence.

Keywords
Adhoc Network, MANET, Routing.

1. INTRODUCTION

1.1 Routing in MANET
Routing in MANET is a Dynamic Optimization Problem as the search space changes over time. The routing policy is defined as the rule that specifies what node to take next at each decision node to reach the destination node. Due to the time varying nature of the topology of the networks, traditional routing techniques such as distance-vector and link-state algorithms that are used in fixed networks, cannot be directly applied to mobile ad hoc networks. The constraints of MANETs demand the need of specialized routing algorithms that can work in a decentralized and self-organizing way. The routing protocol of a MANET must dynamically adapt to the variations in the network topology.

The routing scheme in a MANET can be classified into two major categories – Proactive and Reactive. The proactive or table driven routing protocols maintain routes between all node pairs all the time. It uses periodic broadcast advertisements to keep routing table up-to-date.

This approach suffers from problems like increased overhead, reduced scalability and lack of flexibility to respond to dynamic changes. The reactive or on-demand approach is event driven and the routing information is exchanged only when the demand arises. The route discovery is initiated by the source. Hybrid approaches combines the features of both the approaches. Destination Sequence Distance Vector (DSDV) is a flat proactive routing protocol whereas Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distance Vector (AODV) routing are examples of flat reactive or on demand protocols. The increased latency may render the reactive protocols unsuitable for time-critical traffic.

1.2 Ant Colony Optimization
The Ant colony optimization is based on the foraging behavior of ants [5]. When ants search for food, they wander randomly and upon finding food return to their colony while laying a chemical substance called pheromone. Many ants may travel through different routes to the same food source. The ants, which travel the shortest path, reinforce the path with more pheromone that aids other ants to follow. Subsequently more ants are attracted by this pheromone trail, which reinforces the path even more. This autocatalytic behavior quickly identifies the shortest path.

Ants are simple autonomous agents that interact via indirect communication known as stigmergy. Stigmergy is an indirect form of communication where individual agents leave signals in the environment and other agents sense them to drive their own behavior. This form of communication is local wherein simple agents interact locally without having any global information.

1.2.1 Exploration
The success of any stochastic search method heavily depends on striking an optimal balance between exploration and exploitation. These two issues are conflicting but very crucial for all the meta-heuristic algorithms. Exploitation is to effectively use the good solutions found in the past search whereas exploration is expanding the search to the unexplored areas of the search space for promising solutions. The reinforcement of

the pheromone trail by the artificial ants exploits the good solution found in the past. However, excessive reinforcement may lead to premature convergence. To maintain the diversity in the search space the following methods are suggested:

(a) Control the exponent of the pheromone trail.

(b) Introduction additional randomization in the ants' decision process.

2. ANT BASED ALGORITHMS

2.1 Ant Based Control (ABC) Routing

In this Algorithm, every node in the network has a pheromone table entry for every possible destination in the network, and each table has an entry for every neighbor. Initially all are assumed to have 0.5 probabilities.

Ants are launched from any node in the network. Each node has random destination. Ants move from node to node, selecting the next node to move to according to the probabilities in the pheromone tables for their destination node. Arriving at a node, they update the probabilities of that node's pheromone table entries corresponding to their *source* node. They alter the table to increase the probability pointing to their previous node. When ants have reached their destination, they die. The increase in these probabilities is a decreasing function of the age of the ant, and of the original probability. The ants get delayed on parts of the system that are heavily used. Some noise can be added to avoid freezing of pheromone trails.

The method [1] used to update the probabilities is quite simple: when an ant arrives at a node, the entry in the pheromone table corresponding to the node from which the ant has just come is increased according to the formula:

$$p = (p\_old + \Delta p)/(1 + \Delta p) \qquad \text{(i)}$$

Here p is the new probability and ∇p is the probability increase. The other entries in the table of this node are decreased according to:

$$p = (p\_old + \Delta p)/(1 + \Delta p) \qquad \text{(ii)}$$

The probabilities are updated according to the following formula, where *age* stands for the number of time steps that passed since the launch of the ant:

$$\Delta p = ((0.08)/age) + 0.005) \qquad \text{(iii)}$$

The *delay* in time steps that is given to the ant is a function of the spare capacity *s* of the node:

$$delay = \lfloor 80.e^{-0.075.s} \rfloor \qquad \text{(iv)}$$

2.2 Ant Colony Based Routing Algorithm

Ant Colony Based Routing Algorithm (ARA) [2] works in an on-demand way, with ants setting up ***multiple*** paths between source and destination at the start of a data session FANTs are broadcasted by the sender to all its neighbors. Each FANT has a unique sequence number to avoid duplicates. A node receiving a FANT for the first time creates a record ***[destination address, next hop, pheromone value]*** in its routing table.

The node interprets the source address of the FANT as destination address, the address of the previous node as next hop, and computes the pheromone value depending on the number of hops the FANT needed to reach the node. Then the node relays the FANT to its neighbors. When the FANT reaches destination, it is processed in a special way. The destination node extracts the information and then destroys the FANT. A BANT is created and sent towards the source node. In that way, the path is established and data packets can be sent.

Data packets are used to maintain the path, so no overhead is introduced. Pheromone values are changing

Let $G = (V, E)$ be a connected graph with $n = |V|$ nodes. The pheromone concentration, $\Phi i,j$ is an indication of the usage of the edge i, j. An ant located in node *vi* uses pheromone $\Phi i,j$ of node *vj* and *Ni* to compute the probability of node *vj* as next hop. *Ni* is the set of one-step neighbors of node *vi*.

$$p_{i,j} = \begin{cases} \frac{\varphi_{i,j}}{\sum_{j \in N_i} \varphi_{i,j}} & if\ j \in N_i \\ 0 & if\ j \notin N_i \end{cases} \qquad \text{(v)}$$

The transition probabilities *pi,j* of a node *vi* fulfill the constraint:

$$\sum_{j \in N_i} p_{i,j} = 1, i \in [1, N] \qquad \text{(vi)}$$

An ant changes the amount of pheromone of the edge *e*(*vi, vj*) when moving from node *vi* to node *vj* as follows:

$$\varphi_{i,j} = \varphi_{i,j} + \Delta\varphi \qquad \text{(vi)}$$

Like real pheromone the artificial pheromone concentration decreases with time to inhibit a fast convergence of pheromone on the edges.

$$\varphi_{i,j} = (1 - q).\varphi_{i,j},\ q \in [0, 1] \qquad \text{(vii)}$$

2.3 Probabilistic Emergent Routing Algorithm (PERA)

This algorithm works in an on-demand way, with ants being broadcast towards the destination at the start of a data session [8]. Multiple paths are set up, but only the one with the highest pheromone value is used by data and the other paths are available for backup [8]. The route discovery and maintenance is done by flooding the network with ants. Both forward and backward ants are used to fill the routing tables with probabilities. These probabilities reflect the likelihood that a neighbor will forward a packet to the given destination. Multiple paths between source and destination are created.

First of all, neighbors are discovered using HELLO messages, but entries are only inserted in the routing table after receiving a backward ant from the destination node. Each neighbor receives an equi-probable value for destination. This value is increased as a backward ant comes from that node, establishing a path towards destination. As ants are flooded, the algorithm uses sequence numbers to avoid duplicate packets. Only the greater sequence number from the same previous hop is taken into account. Forward ants with a lower sequence number are dropped. This approach is similar to AODV Route Request packets, but discovers a set of routes instead of one. Data packets can be routed according to the highest probability in the routing table for the next hop.

2.4 Ant Agents for Hybrid Multipath Routing [AntHocNet]

AntHocNet [7] is a multipath routing algorithm for mobile ad-hoc networks that combines both proactive and reactive components. It maintains routes only for the open data sessions. This is done in a *Reactive Route Setup phase,* where reactive forward ants are sent by the source node to find multiple paths towards the destination node. Backward ants are used to actually setup the route. While the data session is open, paths are monitored, maintained and improved proactively using different agents, called proactive forward ants.

2.5 Antnet – Ant Algorithm

In Antnet algorithm [2], at regular intervals Δt from every network node *s*, a forward ant $f_{s \rightarrow d}$ is launched toward a destination *d* to discover a feasible, low-cost path to that node and to investigate the load status of the network along the path. If f_{sd} is a measure (in bits or in number of packets) of the data flow $s \rightarrow d$, then the probability of creating at node *s* a forward ant with node *d* as destination is:

$$p_{sd} = \frac{f_{sd}}{\sum_{i=1}^{N} f_{si}} \qquad \text{(viii)}$$

While travelling toward their destination nodes, the forward ants keep memory of their paths and of the traffic conditions found. The identifier of every visited node *i* and the time elapsed since the launching time to arrive at this *i*-th node are stored in a memory stack.

At each node *i*, each forward ant headed toward a destination *d* selects the node *j* to move to, with a probability P_{ijd} computed as normalized sum of the pheromone τ_{ijd} with a heuristic value η_{ij} taking into account the length of the *j*-th link queue of the current node *i*:

$$P_{ijd} = \frac{\tau_{ijd} + \alpha\eta_{ij}}{1+\alpha(|N_i|-1)} \qquad \text{(ix)}$$

The heuristic value η_{ij} is a normalized value function of the length q_{ij} of the queue on the link connecting the node *i* with its neighbor *j*:

$$\eta_{ij} = 1 - \frac{q_{ij}}{\sum_{l=1}^{|N_i|} q_{il}} \qquad \text{(x)}$$

If a cycle is detected, the cycle's nodes are removed and all the memory about them is deleted. When an ant reaches a node that is already in its memory, a cycle is detected and all the nodes until this recurrent node are deleted from the ant's memory. When the destination node *d* is reached, the agent $F_{s \rightarrow d}$ generates backward ant $B_{d \rightarrow s}$, transfers to it all of its memory, and is deleted.

The backward ant takes the same path as that of its corresponding forward ant, but in the opposite direction. Arriving at a node *i* coming from a neighbor node, the backward ant updates the local model of the traffic M_i and the pheromone matrix T_i, for all the entries corresponding to the destination node *d*.

3. COMPARISON OF DIFFERENT ANT BASED ROUTING ALGORITHMS

ABC Routing was developed for wired telecommunication networks and it assumes symmetric path costs between nodes. Like in AntNet, each node *s* periodically sends out ants to randomly chosen destinations. Each ant has an associated age, which is increased proportionally to the load of each visited node. While traveling from its source *s* to its destination *d*, the ant updates the pheromone for the path backward to *s*, based on its age. This is an important difference with AntNet: ants update pheromone about the path to their source while going forward, and no backward ants are used. This is possible because of the assumption of symmetric path costs.

Another difference is that no path statistics are used to evaluate path quality measurements reported by the ants, and that no local heuristic is used to help guide the ants (in AntNet, the local queue lengths are used).

Finally, in ABC, it is not data packets that are routed according to the pheromone, but call setup messages. Moreover, these messages do not follow pheromone probabilistically, but greedily choose the directions with the highest pheromone level. Once a call has been set up successfully, data packets follow its circuit.

AntNet was developed for packet switched wired networks. HELLO messages are used initially to discover the neighbors. In PERA, HELLO messages are broadcasted each time any node moves to a different position so that node can discover its new neighbors.

In ARA entry in the routing table for each node is created when a forward ant arrives at that node. Pheromone value is the number of hops required by the forward ant to reach the current node from the destination.ARA is quite similar to PERA. One difference is that both forward and backward ants leave pheromone behind: forward ants update pheromone about the path to the source, while backward ants update pheromone about the path to the destination.

Another difference is that also data packets update pheromone, so that paths which are in use are also reinforced while the data session is going on. This comes down to repeated path sampling, so that ARA keeps more of the original ACO characteristics than PERA.

PERA uses routing table which has the following structure: [Destination, Next hop, Probability]. Initially probability value each node from source to destination is initialized with uniform probabilities (1/N) where N is the number of neighbors for each node. It is used for wireless networks. It works similar to Antnet. Each node periodically sends forward ant to randomly chosen destination, whereas source node is chosen according to some probability of data flow.

Anthocnet requires more number of resources as compared to other ant based algorithms. This is because there are two forward ants [Proactive and Reactive] and two backward ants [Proactive and Reactive]. Structure of ant is similar but number of ants generated varies with other ant based algorithms. Amount of control traffic generated due to ants is more than other ant based algorithms.

Anthocnet is most efficient in maintaining paths. It has greater chance of exploring new paths due to proactive nature with a hint of probability. This is due to the fact that proactive ants are normally unicast to sample the existing path found by reactive forward ants but also have a small probability at each node of being broadcast. Therefore even though Anthocnet maintains paths between nodes and explores new routes it is costly and requires more resources.

Table 1 gives the comparison of different ant based routing algorithms based on resources used by them.

4. SIMULATION PARAMETERS

We can now define the performance evaluation parameters which can be considered for the comparison of the ant based algorithms. There are two types of traffic which can be taken into consideration:

1. **Session oriented:** All packets have same destination for a given session.
2. **Session less:** Destination of each packet selected from uniform distribution

The parameters which can be considered are:

- **Average Throughput.** Throughput is a measure of how much traffic is successfully received at the intended destination in a unit interval of time. A routing protocol should try to maximize this value.
- **Packet Delay.** A good algorithm should be able to deliver packets with minimum delay.
- **Session Delay.** In case of session-oriented traffic, the most important parameter is time needed to complete a session. An application layer at the destination node only gets the packets after all the packets are received in the correct order. Packet delay factors out this waiting time and hence favors multi-path algorithms which deliver packets in an out-of-order manner but with smaller delays.
- **Sessions Completed.** The percentage of sessions which are completed without any support from transport layer protocols. For example if only one packet in a session is dropped due to congestion or TTL expiration, we report the session as an incomplete one. This parameter reports the way packets were deleted due to congestion.
- **Packet Delivery Ratio.** This measure tells us how many data packets are successfully delivered at their destinations. Under saturated loads a 1% improvement in packet delivery ratio at times means about few 100,000 more data packets delivered at their destination.
- **Packet Drop Ratio.** The percentage of data packets that are dropped because their time to live timer (TTL) value expired or the queue buffers were full.
- **Packet Loop Ratio.** The percentage of data packets that followed a cyclic path. A cyclic path is an error in an algorithm and should be reported.
- **Routing Overhead.** The ratio of the bandwidth occupied by the routing/control packets and the total available bandwidth in the network. This parameter shows the control overhead of the routing algorithm.
- **Suboptimal Overhead.** The difference between the bandwidth consumed when transmitting data packets from all the sources to destinations and the bandwidth that would have been consumed should the data packets have followed the shortest hop count path.

5. CONCLUSION

Among wireless networks Anthocnet is more efficient among all the considered ant based algorithms because it has greater chance of exploring new paths based on probability but it is more costly and requires more resources for implementing it. This is due to the fact that there is lot of ant traffic generated.

PERA is better in terms of less cost and also efficient in maintaining and exploring new paths. ARA is similar to PERA but in ARA both forward and backward ants update pheromone value.

For wired networks Antnet works best in maintaining the established paths as compared to ABC routing because ABC uses greedy approach and Antnet chooses best paths based on probability.

6. ACKNOWLEDGMENTS

I would like to sincerely thank Prof. C. K. Bhensdadia and Prof. Brijesh Bhatt for their sincere support and guidance in my research.

7. REFERENCES

[1] Anandamoy Sen 2006. Swarm Intelligence based optimization Of MANET cluster formation.

[2] Schoonderwoerd, Ruud; Holland, Owen; Bruten, Janet; Rothkrantz, Leon 1996. Ant-based load balancing in telecommunications networks. Hewlett-Packard Laboratories, Bristol-England, pp 162-207.

[3] Mesut G¨unes,, Udo Sorges, Imed Bouazizi 2002. ARA – The Ant-Colony Based Routing Algorithm for MANETs.

[4] V.Laxmi, Lavina Jain, M.S.Gaur 2006. Ant Colony Optimization based Routing on *ns*-2.

[5] Gianni Di Caro, Frederick Ducatelle and Luca Maria Gambardella 2005. AntHocNet: an Ant-Based Hybrid Routing Algorithm for Mobile Ad Hoc Networks.

[6] S. Prasad, Y.P.Singh, and C.S.Rai 2009. Swarm Based Intelligent Routing for MANETs.

[7] Gianni Di Caro*, Frederick Ducatelle and Luca Maria Gambardella 2005. AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks.

[8] J. Baras and H. Mehta 2003. A Probabilistic Emergent Routing Algorithm for Mobile Ad hoc Networks (PERA).

Table 1. Comparison of different Ant based Algorithms

Characterisitic	Resource and Cost Comparison				
	ABC Routing	AntNet	ARA	PERA	AntHocNet
Types of Ants	Forward Ant	Forward Ant, Backward Ant	Forward Ant, Backward Ant	Forward Ant, Backward Ant	Reactive Forward Ant and Backward Ant, Proactive Forward Ant and Backward Ant
Ant structure	Source IP address, Dest IP address, Age of Ant [Time since last launched]	Source IP address, Dest IP address, Sequence num, field to identify as FA or BA, memory [node addresses and trip time]	Source IP address, Dest IP address, Sequence num, Hop count	Source IP address, Dest IP address, Stack [Node id, Node traverse time], Hop count, Sequence No	Source IP address, Dest IP address, Next Hop IP address,Stack [Node id, Node traverse time], Hop count, Sequence No
Routing Table Structure	Destination address, Next hop, Pheromone value	Destination address, Each neighbour, Pheromone value	Destination address, Next hop, Pheromone value	Destination address, Next hop, Probability	Goodness of next hop, Destination address, Next hop
Traffic Statistics Structure [Mean variance, Best value of trip time]	Not used	Used	Not Used	Used	Used

*FA- Forward Ant, BA-Backward Ant