Scholars Journal of Engineering and Technology

Abbreviated Key Title: Sch J Eng Tech
ISSN 2347-9523 (Print) | ISSN 2321-435X (Online)
Journal homepage: https://saspublishers.com

OPEN ACCESS

Challenges on ACO Based and BCO Based Routing Protocols in MANETS

Dr. Arun Kumar Singh[1*], Dr. Sushil Kumar Garg[1], Dr. Firoz Parvez[2]

[1]Professor, CSE, BGIET Sangrur, Punjab, India
[2]Associate Professor, CSE, BGIET Sangrur, Punjab, India

DOI: 10.36347/sjet.2020.v08i12.004 | **Received:** 11.11.2020 | **Accepted:** 15.12.2020 | **Published:** 30.12.2020

***Corresponding author:** Dr. Arun Kumar Singh

Abstract **Review Article**

Networks have changed our life style drastically in last 10 years. We always want to stay connected to an external network like Internet or Intranet with the help of wireless communication. A lot of research has been in done this field to explore the best and the optimum protocols which can be used in WSN, WMN and Ad-hoc networks. Identification of optimal route for data communication, efficient utilization of energy, providing congestion free communication, offering scalability, maintaining the Quality of Service are few research issues in the Ad-hoc Networks. To overcome these issues swarm Intelligence (SI) inspired routing algorithms have become a research focus in recent years due to theirs self-organizing nature, which is very suitable to the routing problems in Mobile Ad hoc Networks (MANETS). In this paper, we focus on routing algorithms based on Ant Colony Optimization (ACO) and Bee Colony Optimization (BCO) for MANETS that have been designed according to the principles of swarm intelligence.

Keywords: MANETS, Ad-hoc, ACO, BCO, SI.

General Terms: Mobile adhoc networks, Swarm Intelligence, Ant Colony Optimization, Bee Colony optimization, Routing protocol.

1. INTRODUCTION

MANET [1] is one of those types of wireless networks which do not require any infrastructure to set up itself; we can set up this type of network anywhere with the help wireless facility of its nodes as shown in the figure below:

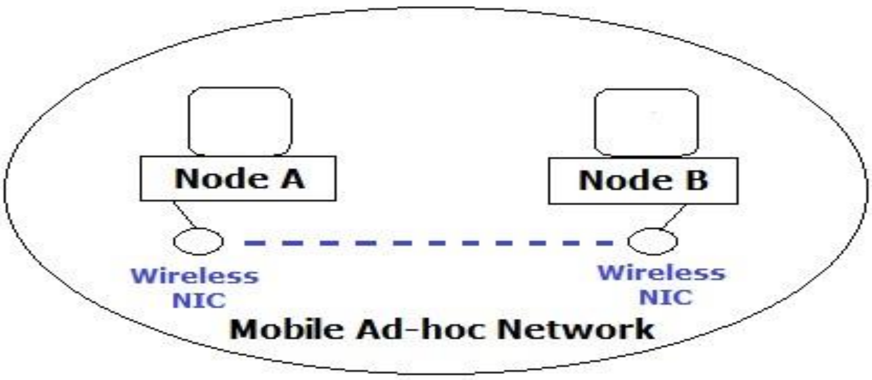

Figure 1: A MANET Scenario

This type of network which can be set up without any centralized control or infrastructure is also known as Infrastructure less network. Such a network can operate in standalone fashion or either connected to the Internet: after one of the nodes is configured as a gateway as shown in the figure below:

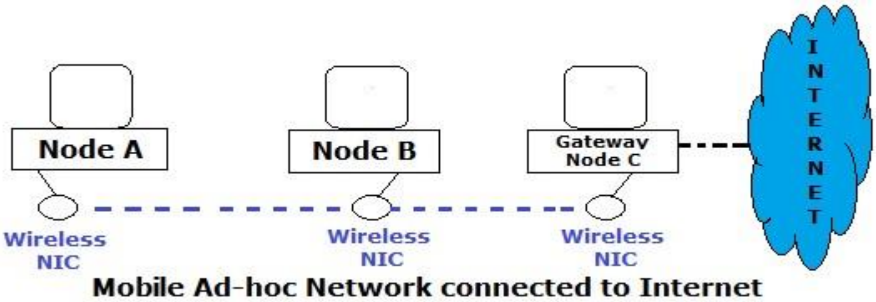

Figure 2: MANET connected to Internet

The nature of this kind of networks makes them suitable for deployment in places where network infrastructure is hard to build and maintain. Applications of ad hoc networks are emergency search and rescue operations, war like situations, VANETS or meetings in which persons wish to quickly share information.

Challenges in MANETS Routing Protocols

- MANETS are usually made up of small or tiny nodes equipped with little memory, limited non-rechargeable battery, low-end processors, and small bandwidth links. As a result, MANET protocol designers face strict constraints on the use and the availability of node resources.
- The majority of target applications for MANETS require the deployment of the sensor nodes in large numbers, ranging from thousands to millions. Hence, the scalability of the used protocols is also a major concern.
- Individual nodes can potentially generate huge amounts of data. The transmission of every data bit to a common sink node would make use of a large amount of energy, bandwidth, and processing power. Therefore, possibly redundant information needs to be detected and filtered, in order to reduce the in-network traffic.

MANETs are dynamic, flexible and require few resources but at the same time they suffer from a variety of problems like: lack of centralized management, resource availability, Dynamic topologies, device discovery, security, reliability, quality of service and internetworking. Many different approaches dealing with these problems do exist, one of them is Swarm intelligence. The rest of the paper is organized as follows. In Section 2, we briefly discuss the Swarm Intelligence (SI) and SI based Ant Colony Optimization and Bee Colony Optimization techniques. The third and fourth sections are the main part of this paper gives description about the most commonly used ACO and BCO based routing protocols. In the fifth section we compare ACO and BCO based routing protocols.

2. SWARM INTELLIGENCE

Swarm intelligence (SI) [4-6] is a new discipline of study that contains a better optimal approach for problem solving which are inspired from the social behavior of insects and animals. Examples in natural systems of SI include ant colonies, bird flocking, animal herding, bacterial growth, fish schooling and Microbial intelligence. Different swarm intelligence fields are shown in figure3 below. This paper are lend itself to give brief information about swarm intelligence based ACO and BCO routing protocols. The nature of swarms largely resembles mobile ad-hoc networks (MANETs) and that is why ideas from swarm animals like ants and bees are used for creating suitable routing protocols for MANETs. Bio inspired, SI [4-6] approaches are more promising for MANETS due to the following reasons.

- Self-organizing behavior & Multiple path availability
- Failure backup & Dynamic topology

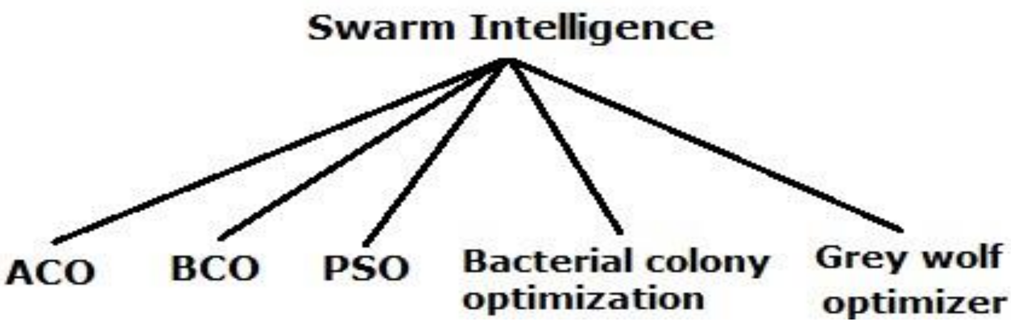

Figure 3: Subdivision of Swarm Intelligence

The following sub-sections will describe about ant colonies and bee colonies optimization techniques which can be used in conjunction with MANETS.

2.1 ANT COLONY OPTIMIZATION

Ant Colony Optimization [2, 3] is paradigm of Swarm Intelligence that is inspired by the collective behaviour of ants. The basic idea of ANT colony optimization is taken from the food searching behavior of real ants. When ant searches for food, they start from nest and walk towards the food. When ant reaches an intersection, ant has to decide which branch to take next while walking. Ants deposit a substance called pheromone. This marks the route taken. The concentration of pheromone decreases due to diffusion effects. The behavior of ants is used to find the shortest route in the networks. ACO meta-heuristic computational approach was proposed by Marco Dorigo in 1996. The basic principle of ACO is simulation of ability of ants to find the shortest path between their nest and a food source. Ants are capable to find the shortest path between their nest and food source, without any visible, central and active coordination mechanism.

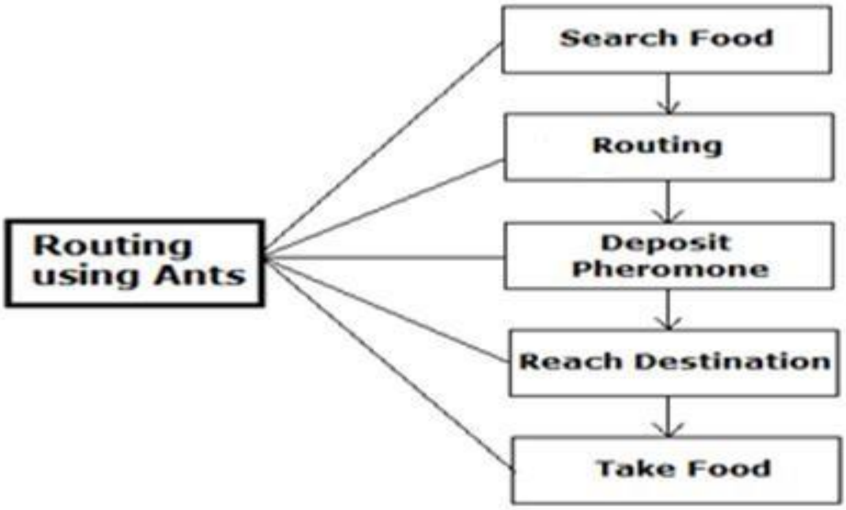

Figure 4: Steps in ACO

The real ants drop a pheromone, chemical from their bodies naturally, on the path which leads them for the various decisions. Rest of the ants reaches the food source by following this pheromone trail. General steps in ACO are shown in Figure 4. More the concentration of pheromone value on the path means path probability is greater than others. Here some important features of ad hoc networks which flavors the designing of swarm intelligence based protocols:

- ***Dynamic Topology:*** The dynamically change topology, causes bad performance of mostly routing algorithm in mobile multi-hop ad hoc networks. The working principle of ACO is based on agent systems and works individually and flavours high adaptation to the current topology of the network.
- ***Local Work:*** The ACO based algorithms are based only on local information, so it no needs the transmission of routing tables or other information to neighbour nodes in networks.
- ***Support for Multi-path:*** The selection decision is based on the pheromone value on the current node. It provides the multi-path selection choices.

2.2 BEE COLONY OPTIMIZATION

Similar to ACO, the Bee Colony Optimisation (BCO) [2, 3] is a nature-inspired meta-heuristic, which can be applied in order to find solutions for optimization problems that mimics the food foraging behaviour of swarms as in honey bees. BCO is the name given to the collective food foraging behavior of honey bee. The bee system is a standard example of organized team work, and simultaneous task performance with well-knit communication. In a bee colony there are different types of bees like a queen bee, many male drone bees and thousands of worker bees.

- **Queen Bee:** The main work of queen bee is to laying egg which is used to develop a new colony because there is only one queen bee in the hive.
- **Male Drone Bees:** In the hive there are two types of male drone bees first is Food Packers Bees the work of food packer bees is to serve the queen help it in laying the eggs. The second is Nurses Group Bees which is responsible for feeding the queen and the babies.
- **Worker Bees:** There are two types of worker bees, namely *scouts* and *foragers*. The scouts start from the hive in search of a food source randomly keeping on this exploration process until they are tired. The movement of scout bees in a typical bee hive is shown in figure 5. When they return back to the hive, they

convey to the foragers information about the odor of the food, its direction, and the distance with respect to the hive by performing dances. A *round dance* indicates that the food source is nearby whereas *a waggle dance* indicates that the food source is far away. Waggling is a form of dance made in eight-shaped circular direction and has two components: the first component is a straight run and its direction conveys information about the direction of the food; the second component is the speed at which the dance is repeated and indicates how far away the food is. Bees repeat the waggle dance again and again giving information about the food source quality. The better is the quality of food, the greater is the number of foragers recruited for harvesting. The Bee Colony Optimization (BCO) meta-heuristic has been derived from this behavior and satisfactorily tested on many combinatorial problems. Following is the flowchart which gives the basic structure of the working of the bee colony system.

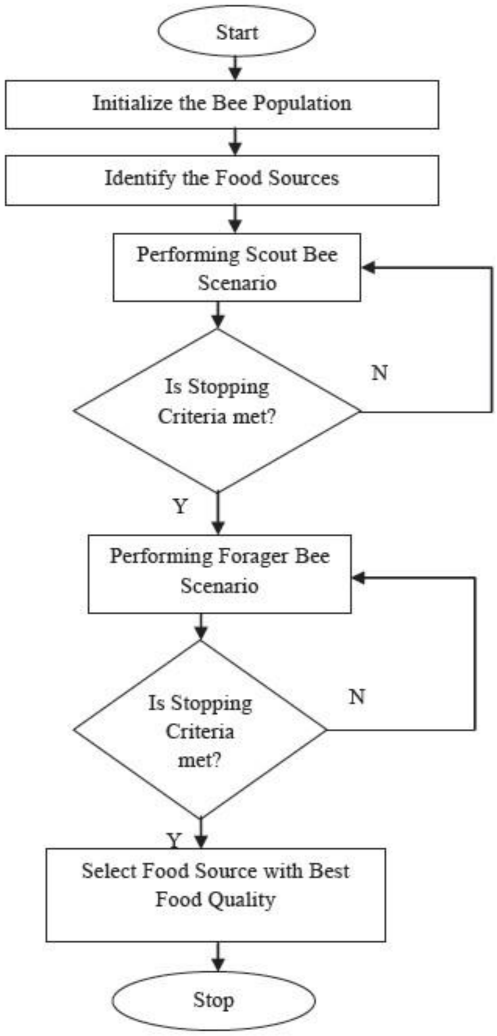

Figure 5: Flowchart for Basic BCO Steps

3. ACO Based Routing Protocols for MANETS

In this section we explained three Ant Colony Optimization based routing protocols [7, 8] namely: ARA, Antnet and AntHocnet. These three protocols are covered in the next subsections:

3.1 ARA

Gunes *et al.,* proposed Ant Colony Based Routing Algorithm [10] (ARA) which reduces overhead, because routing tables are not interchanged among nodes. ARA consists of three phases namely: Route Discovery, Route maintenance and Route failure handling. ARA is a purely reactive MANET routing algorithm. It does not use any HELLO packets to explicitly find its neighbours. When a packet arrives at a node, the node checks it to see if routing information is available for destination d in its routing table. In ARA, the route discovery is done either by the FANT (forward ant) flood technique [13] or FANT forward technique. In the FANT flooding scheme, when a FANT arrives to any intermediate node, the FANT is flooded to all its neighbours. If found, it forwards the packet over that node, if not, it broadcasts a forward ant (FANT) to find a path to the destination. By introducing a maximum hop count on the FANT, flooding can be reduced. In the FANT forwarding scheme, when a FANT reaches an intermediate node, the node checks its routing table to see whether it has a route to the destination over any of its neighbours. If such a neighbour is found, the FANT is forwarded to only that neighbour; else, it is flooded to all its neighbours as in the flood scheme.

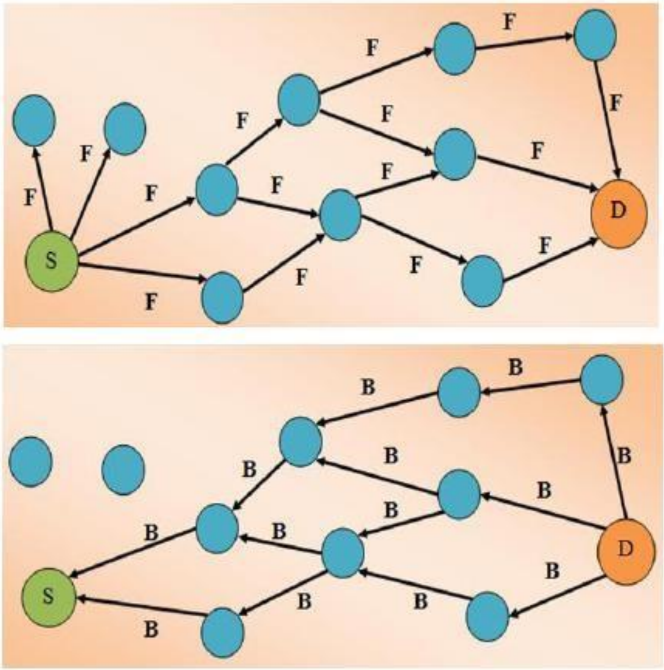

Figure 6: Route discovery phase in ARA showing a forward ant (F) and backward ant (B)

In ARA [10], a route is indicated by a positive pheromone value in the node's pheromone table over any of its neighbours to the FANT destination. 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. Nodes modify their routing table information when a backward ant is seen according to number of hops the ant has taken. In Route Maintenance phase, DUPLICATE ERROR flag is set for duplicate packets to prevent from looping problems. ARA [10] also allows for the evaporation of pheromone by decrementing factor in route table. In Route Failure Handling phase, node deactivates the path by reducing pheromone value to 0 in corresponding route table entry and go to the Route Discovery phase for selecting path and sending packets to the destination over that path [16].

3.2 AntNet

Antnet [11] is proposed by G. Di Caro and M. Dorigo. It is an agent based routing algorithm that is influenced from the real ant's behaviour. In AntNet ants (nothing but software agents) explores the network to find the optimal paths from the randomly selected source destination pairs. Moreover, while exploring the network ants updates the probabilistic routing tables and builds a statistical model of the nodes local traffic. Ants use these tables to communicate with each other. The idea in AntNet is to use two different network exploration agents, i.e. forward ants (FANTs) and backward ants (BANTs), which collect information about delay, congestion status and the followed path in the network. FANTs are emitted at regular time intervals from each node to a randomly selected destination. This transmission occurs asynchronously and concurrently with the data traffic. As soon as a FANT arrives at the destination, a BANT moves back to the source node reverse the path taken by the FANT. The subdivision in forward and BANTs has the following reasons. The FANTs are just employed for data aggregation of trip times and node numbers of the path taken without performing any routing table updates at the nodes. The BANTs get their information from the FANTs and use it to achieve routing updates at the nodes. By using two different types of ant agents FANT and BANT; each node in the network has shortest path to all the destinations. However, they cause extra overhead but it reduces the available traffic capacity for actual data communication. AntNet is designed in such way that the forward ants carry the information about the status of the links it traverses. This status information can be captured and can be used to find the best path. AntNet [11] is one of the dynamic routing algorithms for learning new routes.

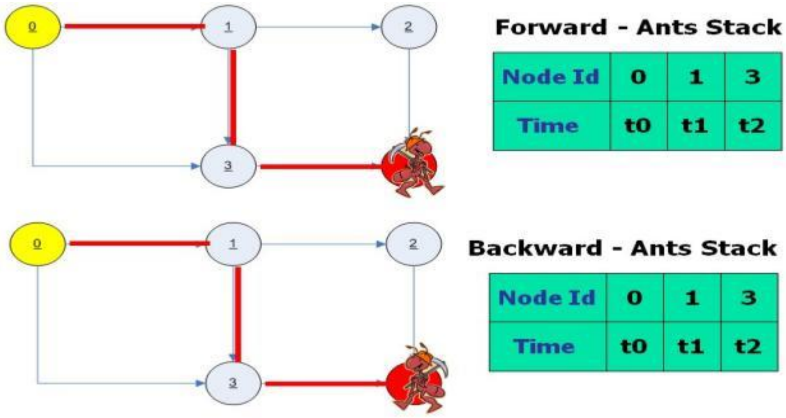

Figure 7: Trip times table in Antnet algorithm

An example of AntNet [11] trip time table is given in the figure 7. Routing table at each node stores the list of reachable nodes and their pheromone value. It is represented as structure consisting of following fields:

- Destination_id – This represents the address of the destination node
- Next_id – This represents the address of the adjacent node used to reach destination node.
- Pheromone – This represents the value used by the node to calculate the probability of each adjacent node to be the next hop in order to reach the destination. An example of such a routing table is given in the table below:

Table 1: Routing table based on Pheromone Values

Routing Table at node 1		
Destination	**Next Node**	**Pheromone Value**
0	2	0.5
1	1	1

3.3 AntHocNet (Ant Agents for Hybrid Multipath Routing)

AntHocNet [12] is a hybrid algorithm proposed by Di Caro, Ducatelle and Gambardella in 2004 consisting of reactive and proactive components. It does not maintain routes to all possible destinations at all times (like AntNet), but only sets up paths when they are needed at the start of a data session. It is reactive in the sense that a node only starts gathering routing information for a specific destination when a local traffic session needs to communicate with the destination and no routing information is available. It is proactive because as soon as the communication starts, and for the entire duration of the communication, the nodes proactively keep the routing information related to the ongoing flow up-to-date with network changes for both topology and traffic.

When a data session is started between source and destination, firstly source node *S* checks whether it has up-to- date routing information for destination *D*. If no such information is available, it reactively sends out ant-like agents, called reactive FANT, to look for paths to *D*. These ants gather information about the quality of the path they followed, and at their arrival in *D* they become BANT which trace back the path and update routing tables. The routing table Ti in node i contains for each destination *D* and each possible next hope on a pheromone value. In this way, pheromone tables in different nodes indicate multiple paths between *S* and *D*, and data packets can be routed from node to node as datagrams. Once paths are set up and the data session is running, *S* starts to send proactive FANT ants to *D*. These ants follow the pheromone values similarly to data packets. In this way they can monitor the quality of the paths in use. Moreover, they have a small probability of being broadcasted, so that they can also explore new paths. In case of link failures, nodes either try to locally repair paths, or send a warning to their neighbors such that these can update their routing tables.

4. BCO Based Routing Protocols for MANETS

In this section we explained three routing protocols based on Bee Colony Optimization [7-9] namely: BeeAdhoc, BeeSensor and BeeIp. These three protocols are covered in the next subsection:

4.1 BeeAdHoc

BeeAdHoc [13] is a nature inspired routing protocol for MANETs based on the foraging principles of honey bees. It is a reactive source routing algorithm based on the use of four different bee-inspired types of agents: *packers, scouts, foragers, and bee swarms*. Scouts are used to discover routes and foragers to transport data. Figure 8 will give an overview of the BeeAdHoc architecture. In this architecture each node maintains a hive with an Entrance, Packing Floor and a Dance Floor.

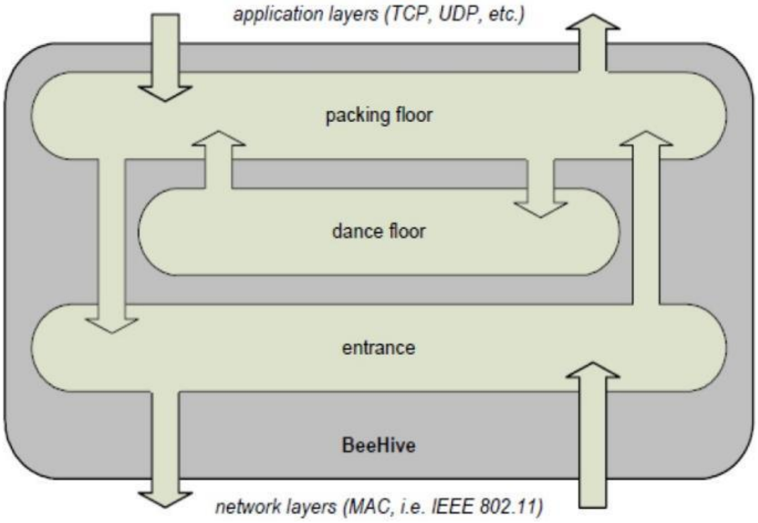

Figure 8: Overview of the BeeAdHoc architecture

- **Packers:** Packers mimic the task of a food-storekeeper bee, reside inside a network node, receive and store data packets from the upper transport layer. Their main task is to find a forager for the data packet at hand. Once the forager is found and the packet is handed over, the packer will be killed.
- **Scouts:** Scouts discover new routes from their launching node to their destination node. A scout is broadcasted to all neighbors in range using an expanding time to live (TTL). At the start of the route search, a scout is generated; if after a certain amount of time the scout is not back with a route, a new scout is generated with a higher TTL in order to incrementally enlarge the search radius and increase the probability of reaching the searched destination. When a scout reaches the destination, it starts a backward journey on the same route that it has followed while moving forward toward the destination. Once the scout is back to its source node, it recruits foragers for its route by dancing. A dance is abstracted into the number of clones that could be made of the same scout.
- **Foragers:** Foragers are bound to the bee hive of a node. They receive data packets from packers and deliver them to their destination in a source-routed modality. To attract data packets foragers use the same metaphor of a waggle dance as scouts do. Foragers are of two types: delay and lifetime. From the nodes they visit, delay foragers gather end-to-end delay information, while lifetime foragers gather information about the remaining battery power. Delay foragers try to route packets along a minimum-delay path, while lifetime foragers try to route packets in such a way that the lifetime of the network is maximized. A forager is transmitted from node to node using a unicast, point-to point modality. Once a forager reaches the searched destination and delivers the data packets, it waits there until it can be piggybacked on a packet bounded for its original source node. In particular, since TCP (Transport Control Protocol) acknowledges received packets, BeeAdHoc piggybacks the returning foragers in the TCP acknowledgments. This reduces the overhead generated by control packets, saving at the same time energy.
- **Bee swarms:** Bee swarms are the agents that are used to explicitly transport foragers back to their source node when the applications are using an unreliable transport protocol like UDP (User Datagram Protocol). The algorithm reacts to link failures by using special hello packets and informing other nodes through Route Error Messages (REM).

In BeeAdHoc, each MANET node contains at the network layer a software module called hive, which consists of three parts: the packing floor, the entrance floor, and the dance floor. The entrance floor is an interface to the lower MAC layer; the packing floor is an interface to the upper transport layer; the dance floor contains the foragers and the routing information.

4.2 BeeSensor

Beesensor [14] (Saleem and Farooq, 2007) is an algorithm based on the foraging principles of honey bees with an on-demand route discovery (AODV). BeeSensor focuses on minimising the energy costs using bee agents. The algorithm works with four types of agents: packers, scouts, foragers and swarms. A brief description of each type of agent is as follows:

- **Packers:** Packers behave like the food-storer bees in a hive. Their major responsibility is to receive packets coming from the upper layer and locate an appropriate forager (route) for them. Once a forager is found, packet is encapsulated in its payload and the packer starts waiting for the next packet. Failure in locating a forager is an indication to the packer that no route exists to a sink.
- **Scouts:** Like their natural counterparts, scouts explore the network in search of a potential sink

node. Scouts are of two sorts: forward scouts and backward scouts. A scout is uniquely identified by its agent ID and the source node ID. Forward scouts propagate in the network using the broadcasting principle. During the exploration of the network, they do not construct a source routing header. The result – that their size becomes independent of the path length – helps BeeSensor to scale to large networks. Once a forward scout reaches a sink node, it delivers the event to the upper layer and starts its return journey as a backward scout. Its task is to build a path leading from the sink to the source node (or vice versa) and report the quality of the discovered path once it reaches at the source node.

- **Foragers:** As in BeeAdHoc [13], foragers are the main workers in BeeSensor. Their major role is to carry events to the sink nodes through a predetermined path that is selected stochastically at the source node. Foragers that follow the same path are grouped together in BeeSensor. Foragers traverse using point-to-point mode by utilizing the forwarding information stored at intermediate nodes. They index the table using their path identifier (PID). Foragers also evaluate the quality of their path and report it back to fellow foragers at the source node.
- **Swarms:** To save energy, foragers are implicitly piggy-backed in the link layer acknowledgment packets to the source node. However, sometimes they need to be explicitly transported back to their source nodes. A swarm agent exactly serves this purpose. Foragers wait for a certain amount of time at the sink node and then take the initiative to build a swarm of waiting foragers. A swarm can transport multiple foragers in its payload back to the source node. A swarm, like an individual forager, is also routed on the reverse links.

In BeeSensor [14], route discovery is achieved by using forward scouts and backward scouts. Results show that the honeybee- inspired protocol is able to transmit more packets than energy optimised version of AODV, achieving less control overhead and lower energy consumption.

4.3 BeeIP

BeeIP [15] is a new honeybee-inspired adaptive routing protocol based on the collaborative behaviours of honeybee foragers. Following a reactive approach, BeeIP’s honeybee agents explore the topology only when data are required to be transmitted between nodes. The model uses three types of agents in the form of data packets. The scout, the ack scout, and the forager.

- **Scout:** They are sent when a scouting process is initialized in order to discover new paths towards a given destination. A scout is transmitted using broadcast to all neighbouring nodes. This technique benefits not only the propagation of the initial request, but also the introduction of the transmitting node to its neighbourhood. Apart from the details of the scouting process, scouts also carry important information about their sender’s state. A node’s state is a group of attributes that describe the situation in which the node is at the time of broadcasting the scout packet.
- **Ack scout:** Once the scout reaches its destination the scouting is considered successful and an ack scout packet is created. Ack scouts use a source routing fashion to travel back to the source, using unicast transmission. Therefore, the route that was followed towards the destination is used in reverse. On their way back, ack scouts acknowledge the success of the scouting to both the intermediate nodes and the source node.
- **Forager:** When BeeIP [15] is unable to transmit a data packet, it stores it into a local queue and starts a new scouting process for its destination. This decreases the packet loss due to incomplete routing information. Once an ack scout returns back and acknowledges the existence of a path, all packets for the corresponding destination in the queue are being transmitted.

The way they do this, is by using the most important agent type of BeeIP [15], the forager. Foragers are specially crafted packets that have three important roles. Firstly, they carry (in form of payload) the data packets from the source to the destination. Secondly, they are used to update neighbouring nodes’ states and links’ information, just like scouts did in the first place. Thirdly, foragers are constantly monitoring the path they traverse for any improvements.

5. Summary of ACO and BCO based Routing Protocols

Routing Protocol	Year	Authors	SI Tech nique	Routing Type	Strength
Antnet	1998	Gianni Di Caro, Marco Dorigo	ACO	Proactive	Scalable
ARA	2002	Mesut Gunes, Udo Sorges, Imed Bouazizi	ACO	Reactive	Less overhead
AntHocNet	2004	Di Caro, G.A., Ducatelle F., Gambardella	ACO	Hybrid	PDR, delay
BeeAdhoc	2005	Horst. F. Wedde, Muddassar Farooq	BCO	Reactive	Energy efficient
BeeSensor	2007	Muhammad Saleem, Muddassar Farooq	BCO	Reactive	Energy efficient

BeeIP	2010	Alexandros Giagkos, Myra S. Wilson	BCO	Reactive	PDR, delay

6. CONCLUSION AND FUTURE SCOPE

The objective of this study is to briefly review the major contribution of swarm intelligence based MANET routing protocols. For this purpose, swarm intelligence based ACO and BCO protocols are reviewed and put to partial comparison. The agents in Ant Colony inspired routing algorithms communicate indirectly through the environment (stigmergy) and the agents provide positive feedback to a solution by laying pheromone on the links. Moreover, they have negative feedback through evaporation and aging mechanisms, which avoids stagnation. Finally, Bee Colony algorithms allow for direct agent-to-agent communication which makes them more responsive to changes in the network. It is shown that by using ideas taken from the simple behavior of ants and bees optimization and innovations in routing protocols can be done, that help outperform the standard MANET routing protocols because SI based protocols ensure higher packet delivery ratio, lesser overhead, lower energy consumption and low delay. These algorithms have lot of scope for future improvements. The future work lies in incorporating factors like signal strength into these algorithms so that the routing algorithms will adjust in dynamic topology changing environment.

REFERENCES

1. Boukerche, A., Ahmad, M., Turgut, B., & Turgut, D. (2008). A taxonomy of routing protocols in sensor networks, in: A. Boukerche (Ed.), Algorithms and Protocols for Wireless Sensor Networks, Wiley, pp. 129–160 (Chapter 6).
2. Dorigo, M., & Stutzle, T. (2004). Ant Colony Optimization, MIT Press, Cambridge.
3. Teodorovic, D., Davidovic, T., & Selmic, M. (2011). Bee colony optimization: The applications survey. *ACM Transactions on Computational Logic*, pp. 1–20.
4. Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). Swarm Intelligence.From Natural to Artificial Systems. Oxford University Press, 0-19-513159-2, Oxford.
5. Pradhan, D., & Priyanka, K. C. (2021). SDR network & network function virtualization for 5G Green communication (5G-GC). In *Future Trends in 5G and 6G* (pp. 183-203). CRC Press.
6. Bonabeau, E., Dorigo, M., & Theraulaz, G. (2000). Inspiration for optimization from social insect behaviour. *Nature*, 406(6), (July 2000) 39-42, 0028-0836.
7. Arvinder, K., & Shivangi, G. (2011). A Survey on the Applications of Bee Colony Optimization Techniques. *International Journal on Computer Science and Engineering (IJCSE*), 3(8). ISSN: 0975-3397.
8. Xun-bing, W., Yong-zhao, Z., Liang-min, W., & Liping, J. (2008). Ant Colony Optimization and Ad-hoc On-demand Multipath Distance Vector Based Routing Protocol. *IEEE Fourth International Conference on Natural Computation*, pp. 589-593.
9. Di Caro, G. A., Ducatelle, F., & Gambardella, L. M. (2005). Swarm intelligence for routing in mobile ad hoc networks. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 76–83, Pasadena, USA, IEEE Press.
10. Dorigo, M., Di Caro, G. A., & Gambardella, L. M. (1999). Ant algorithms for discrete optimization. *Artificial Life*, 5(2), 137– 172.
11. Di Caro, G. A., & Dorigo, M. (1998). AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research* (JAIR), 9, 317–365.
12. Gunes, M., Sorges, U., & Bouazizi, I. (2002). Ara-the ant-colony based routing algorithm for manets," in Parallel Processing Workshops, 2002. Proceedings. *International Conference on*, pp. 79 – 85.
13. Di Caro, G. A., Ducatelle, F., & Gambardella, L. M. (2004). AntHocNet: an ant-based hybrid routing algorithm for mobile ad hoc networks. In *Proceedings of PPSNVIII*, volume 3242 of LNCS, pages 461–470. Springer (Best paper award).
14. Wedde, H. F., Farooq, M., Pannenbaecker, T., Vogel, B., Mueller, C., Meth, J., & Jeruschkat, R. (2005). BeeAdHoc: an energy efficient routing algorithm for mobile ad-hoc networks inspired by bee behavior. In *Proceedings of GECCO*, pages 153–161.
15. Saleem, M., & Farooq, M. (2007). BeeSensor: A Bee-Inspired Power Aware Routing Protocol for Wireless Sensor Networks, Springer-Verlag.
16. Giagkos, A., & Wilson, M. (2014). BeeIP – A Swarm Intelligence based routing for wireless ad hocnetworks, Information Sciences, – Elsevier.