International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 38855-38862

Bee-Inspired Routing the ultimate routing process for Energy Efficient MANET

Sasmita Mohapatra
Research Scholar, Dept. of Electronics and Communication Engineering,
Sri Siddhartha Academy of Higher Education, Tumkur, India, sasmitamohapatra0@gmail.com

Dr.M.Siddappa
Professor and Head of the Department, Dept. of Computer Science and Engineering,
Sri Siddhartha Academy of Higher Education, Tumkur, India, siddappa.p@gmail.com

Abstract
Mobile ad hoc network (MANET) is one of the most important and unique network in wireless network which has brought maximum mobility and scalability. High efficient routing is an important issue in the design of limited energy resource MANETs. In the last few decades many research work have been conducted by the researchers in the field of routing protocols for MANETs. Nowadays the main area of concern is based on routing protocols utilizing the concept of swarm intelligence in which bee inspired routing and ant inspired routing are suggested. But among these two Bee inspired routing has been accepted widely for energy efficient and scalable multipath routing protocol based on dynamic cluster and foraging behavior of a bee swarm. Here in this paper the advantages of Bee inspired routing have been discussed with respect to its architecture and working for choosing the intermediate nodes and different paths by comparing different parameters of all the algorithms from the ant colony optimization and bee colony optimization for energy efficient MANETs where the performance of Bee-AdHoc-C is found to be best.

Keywords— MANET;Energy Efficiency;Bee Inspired Protocols; Scouting; Foraging; Bee-AdHoc-C

I. INTRODUCTION

MANET is self-organizing, rapidly deployable which does not require any fixed infrastructure. Mobile nodes self-organize to form a network over radio links. The goal of MANETs is to broaden mobility into the area of autonomous, mobile and wireless domains, where a set of nodes form the network routing infrastructure in an ad-hoc manner. The main characteristics of a MANET are:

- Packets may need to be forwarded by several nodes to reach the destination.
- Dynamic topology due to the nodes' mobility or nodes leaving/joining the network, which causes packet loss and route change.
- Resource constrains: wireless medium bandwidth, device’s battery, processing speed and memory.

As the nodes in the MANET are battery operated so there are possibilities that some of the nodes may fail for communication in between for which care has been taken to make the MANETs energy efficient. In this purpose the swarm intelligence concept is considered as one of the best way. Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial. Ants, Bees, flock of birds or Termites show impressive collective problem-solving capabilities. Properties associated with their group behavior like self-organization, robustness and flexibility are best characteristics for optimization of artificial systems, control or task execution. Swarm Intelligence mainly consists on Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Honeybees paradigms. A swarm is defined as a set of (mobile) agents that collectively solve problems. In the nature animals form into swarms to search food, build nests, to hunt and avoid being hunted etc. Each individual of the swarm has simple rule of action and access to a limited amount of information via its immediate neighbors or local environment. Due to the nature, architecture, topology and functionality of ad hoc and wireless networks, Swarm Intelligence approaches are most suitable for the routing and energy resources optimization related issues in MANETs. Bio inspired, Swarm Intelligence approaches are more promising for ad hoc and wireless AdHoc networks due to

i) Locality of interactions
ii) Availability of multiple paths,
iii) Self-organizing behaviors
iv) Failure backup,
v) Ability to adapt in a quick and robust way to topological and traffic changes and component failures,
vi) Scalable performance robustness to failures,
vii) Losses internal to the protocol,
viii) Easiness of design and tuning.

In this paper we have discussed regarding different algorithms for ant colony and bee colony optimization required for energy efficient MANETs and have discussed the advantages of bee colony optimization with respect to its architecture and working principle for packet transfer between the nodes. Lastly we have done comparative analysis of different algorithms for ant and bee colony with respect to different parameters.

II. ANT COLONY OPTIMIZATION

The Ant Colony Optimization (ACO) metaheuristic has been inspired by operating principles of ants [1], which empower a colony of ants to perform complex tasks like nest building and foraging [2]. The ants are able to find the shortest path from their nest to a food source by sharing information through stigmergy [2, 3] which is a form of communication in which ants communicate indirectly through the environment [4, 3]. As ants lay pheromone while foraging the concentration of pheromone on the shortest path is more than on the other paths. Ants tend to prefer higher pheromone concentration paths, which results in a majority of ants using a shortest path for foraging in a steady state [3]. This method was mainly used for fixed network.

For circuit switched network the Ant-Based Control (ABC) method was used where an algorithm is used for packet-switched networks on the basis of the ideas of *ABC*. Where the circuit was designed according to two types of ants: *regular* and *uniform*. According to the algorithm each node has determined the cost information of the link to its neighbors.

AntNet was proposed in [5, 6, 7, 8]. The algorithm is designed for asymmetric packet-switched networks, and the primary objective of the algorithm is to maximize the performance of a complete network by balancing the load by probabilistically distributing packets on multiple paths.

Ant Colony Routing (ACR) is a general framework for designing fully distributed and adaptive systems for network control and management, in [9].This can be viewed as a distributed society of static agents, which are known as node managers, and mobile agents, which are proactively or reactively launched in the network.

According to Flooded Forward Ant Routing (FF) [10] ants even can be misguided due to the obstacles or moving destinations. The protocol is based on flooding of ants from source to the sink. In the case where the specific destination is not known at the beginning by the ants, or cost cannot be estimated (e.g., address-based destination), the protocol reduces to basic ant routing, and the problem of wandering around the network to find the destination exist. This is the case where FF exploits the MANET.

According to Flooded Piggyback ant routing (FP) a new ant species is brought to forward ants; namely data ants whose function is to carry the forward list. The control of the flooded forward ants is the same as in FF. The protocol succeeded in combining forward ants and data ants using constrained flooding to route data and to discover optimal paths at the same time so as to minimize energy consumption of the network with the data ants carrying the forward list. In the case of control of the flooded forward ant, the data do not only pass the data to the destination, but also remember the paths which can be used by the backward ants to reinforce the probability on the links. As compared to FF and basic ant routing it was found to outperform with high success rate, but incurred relatively high energy consumption.

Energy Efficient Ant Based Routing (EEABR) [11] is an improved version of the Ant based routing where the protocol does not only consider the nodes in terms of distance but also in terms of energy level of the path traversed by the ants. According to previous algorithms the forward ants are sent to no specific destination node, which means that AdHoc Network nodes must communicate with each other and the routing tables of each node must contain the identification of all the AdHoc Network nodes in the neighborhood and the correspondent levels of pheromone trail. This could be a problem since nodes would need to have a large amount of memory to save all the information about the neighborhood. In the work, the memory of the forward ant is reduced by saving only the last two visited nodes. Also the quality of a given path is measured based on the number of nodes on the path and the level of energy. Much improvement was observed as regards to the energy saving of the network.

In Improved Energy-Efficient Ant-Based Routing Algorithm (IEEABR) available power of nodes and the energy consumption of each path is considered. It improves on memory usage, utilizes the self organization, self adaptability and dynamic optimization capability of ant colony system to find the optimal path and multiple candidate paths from source nodes to sink nodes. The algorithm avoids using up the energy of nodes on the optimal path and prolongs the network lifetime while preserving network connectivity. According to [12], for forward ants sent directly to the sink-node, the routing tables only need to save the neighbor nodes that are in the direction of the sink-node, which considerably reduces the size of the routing tables and, in consequence, the memory needed by the nodes. As adopted in [13], the memory of each ant is reduced to just two records, the last two visited nodes. Since the path followed by the ants is no more in their memories, a memory must be created at each node that keeps record of each ant that was received and sent. Each memory record saves the previous node, the forward node, the ant identification and a timeout value. Whenever a forward ant is received at any node, it searches for any possible loop with the aid of its identification (ID). For the situation where no record is found, the necessary information is retrieved and the timer is restarted, hence forwarding the ant to the next node, else, the ant is eliminated if a record containing the ant identification is found. When a backward ant is received, the source ID is searched so as to know where to send it to. In this section, some modifications are done on EEABR to improve the Energy consumption in the nodes of MANETs and also to in turn improve the performance. The improvements are based on a new scheme to intelligently initialize the routing tables, giving priority to neighboring nodes that simultaneously could be the destination, intelligent update of routing tables in case of node or link failure, and reducing the flooding ability of ants for congestion control. The algorithm also reduces the flooding ability of ants in the network for congestion control. Many evolutionary algorithms are done for ant colony routing processes but there were certain disadvantages for them as according to [1] the success of ants in collectively locating shortest paths is only statistical.

According to Ant Colony Optimization if many ants initially happen to choose a non optimal shortest path, other ants will follow this path, which will result in pheromone reinforcement along this path. Consequently, ants will travel on a stagnating non optimal path in a steady state. However, if we assume that ants do find the shortest path in a steady state, then even this stagnation is not helpful because if all packets follow the shortest path, this will lead to congestion on it. Consequently, the path becomes non optimal, and other non optimal paths may become optimal due to changes in network conditions, or discovery of new paths after changes in the topology [14]. There are other problems like evaporation of pheromone, aging of pheromone and aging of the bee to secret pheromone etc.

Because of all the problems the Bee Hive network was used for data transfer in the network.

III. WORKING PRINCIPLE OF BEE HIVE NETWORK

The BeeHive is a dynamic, simple, efficient, robust, flexible, and scalable multi-path routing algorithm, inspired by the foraging principles of honeybees. The communicative model of bees was instrumental in designing intelligent bee agents, which are suited for large and complex topologies. The results obtained from extensive simulation experiments conclude that bee agents occupy smaller bandwidth and require significantly less processor time compared to the agents of existing state-of-the-art algorithms. However, even with such simple agents, Bee-Hive achieves similar or better performance compared to state-of-the-ant routing algorithms like Ant Net.

The Bee Agent Model uses mainly the foraging principles of a honeybee colony. Bee agent model consists of four types of agents: packers, scouts, foragers, and bee swarms.

Packers mimic the task of a food-storer bee. Packers reside inside a network node, and receive data packets from and store them in 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 agent is removed from the system.

The task of scouts is to discover new routes from their launching node to their destination node. A scout is broadcast to all neighbors in range using an expanding time-to-live (TTL) timer. At the start of the route search, a scout is generated, its TTL is set to a small value and it is broadcast. If, after a certain amount of time, the scout is not back with a route, the strategy consists of the generation of a new scout and the assignment of a TTL higher than in the previous attempt. In this way, the search radius of the generated scouts is incrementally enlarged, increasing the probability of a scout's 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 at its source node, it recruits foragers for its route by utilizing a mechanism derived from the waggle dance of scout bees in nature. A dance is abstracted into the number of clones that could be made of the same scout, which is encoded in their dance number (corresponding to recruiting forager bees in nature).

Foragers are the main workers in the **BeeAdHoc** algorithm. They 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 so that the lifetime of the network is maximized. 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.. This reduces the overhead generated by control packets, saving energy at the same time.

Bee swarms are the agents used to explicitly transport foragers back to their source node when the applications are using an unreliable transport protocol like UDP, such that no acknowledgments are sent for the received data packets. To optimize forager transport, one bee swarm agent can carry multiple foragers: one forager is put in the header of the bee swarm while the others are put in the agent's payload. The bee swarm is launched once the difference between the incoming foragers from a certain node *i* and the outgoing foragers to the same node *i* exceeds a threshold value at a destination node *j*. Once the bee swarm arrives at the node *i*, the foragers are extracted from the payload and stored on the dance floor.

IV.ARCHITECTURE OF BEE ADHOC NETWORK

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, and the dance floor. The structure of the hive is shown in Fig. 1. The entrance floor is an interface to the lower MAC layer, while the packing floor is an interface to the upper transport layer. The dance floor contains the foragers and the routing information to route locally generated data packets. The functional characteristics of each floor composing the hive are explained in the following.

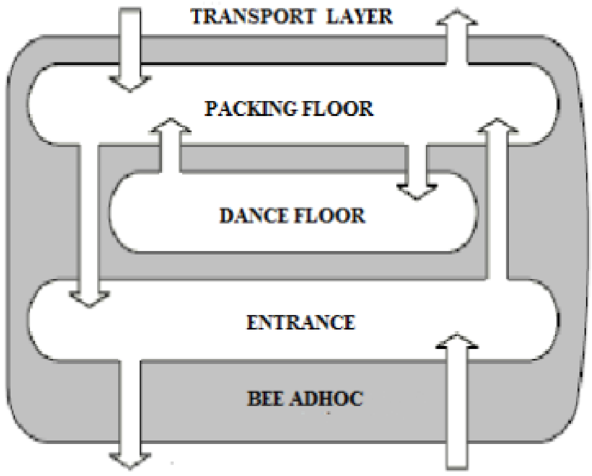

Fig 1. Architecture of BeeAdHoc Network

Packing Floor-

The packing floor is an interface to the upper transport layer to the dance floor. Once a data packet arrives from the transport layer, a matching forager for it is looked up on the dance floor. If a forager is found then the data packet is encapsulated in its payload. Otherwise, the data packet is temporary buffered waiting for a returning forager. If no forager comes back within a certain predefined time, a scout is launched which is responsible for discovering new routes to the packet's destination.

Entrance-
The entrance is an interface to the lower-level MAC layer. The entrance handles all incoming and outgoing packets. Actions on the dance floor depend on the type of packet that entered the floor from the MAC layer. If the packet is a forager and the current node is its destination, then the forager is forwarded to the packing floor; otherwise, it is directly routed to the MAC interface of the next hop node. If the packet is a scout, it is broadcast to the neighbor nodes if its TTL timer has not expired yet or if the current node is not its destination. The information about the ID of the scout and its source node is stored in a local list. If a replica of a previously received scout arrives at the entrance floor, it is removed from the system. If a forager with the same destination as the scout already exists on the dance floor, then the forager's route to the destination is given to the scout by appending it to the route held so far by the scout.

Dance Floor-
The dance floor is the heart of the hive because it maintains the *routing information* in the form of *foragers*. The dance floor is populated with routing information by means of a mechanism reminiscent of the waggle dance recruitment in natural bee hives: once a forager returns after its journey, it recruits new foragers by "dancing" according to the quality of the path it traversed. A lifetime forager evaluates the quality of its route based on the average remaining battery capacity of the nodes along its route. Mimicking forager bees in nature, it dances enthusiastically when it finds a route worth exploiting, recruiting in this way a number of foragers; a lifetime forager can be cloned many times in two distinct cases. In the first case, the nodes on the discovered route have a good amount of spare battery capacity, which means that this is a good route that can be well exploited. In the second case, a large number of data packets are waiting for the forager, so that the route needs to be exploited even though it might have nodes with little battery capacity. On the other and, if no data packets are waiting to be transported, then a forager with a very good route might even abstain from dancing because the other foragers are fully satisfying traffic requests. This concept is directly borrowed from the behavior of scout/forager bees in nature, and it helps automatically regulate the number of foragers for each route. The central activity of the dance floor module consists of sending a matching forager to the packing floor in response to a request from a packer. The foragers whose lifetime has expired are not considered for matching. If multiple foragers can be identified for matching, then a forager is selected in a random way. This helps in distributing the packets over *multiple paths*, which in turn serves two purposes: avoiding congestion under high loads and depleting batteries of different nodes at a comparable rate. A clone of the selected forager is sent to the packing floor and the original forager is stored on the dance floor after reducing its dance number, that is, the number of permitted clones. If the dance number is 0, then the original forager is sent to the packing floor, removing it in this way from the dance floor. This strategy aims at favoring young over old foragers, since the former represent fresher routes, which are expected to remain valid in the near future with higher chances than the older ones because they represent recent state of the network. If the last forager for a destination leaves a hive, then the hive does not have any more a route to the destination. Nevertheless, if a route to the destination still exists, then soon a forager will be returning to the hive; if no forager comes back within a reasonable amount of time, then the node has probably lost its connection to the destination node. This mechanism eliminates the need for explicitly monitoring the validity of the routes by using special Hello packets and informing other nodes through route error messages, as is done in several state-of-the-art algorithms such as AODV, as well as in several ACO implementations for MANETs. In this way, fewer control packets are transmitted, resulting in less energy expenditure.
The Ad hoc On Demand Distance Vector (AODV) routing algorithm is a routing protocol designed for ad hoc mobile networks which is capable of both unicast and multicast routing. It is an on demand algorithm, meaning that it builds routes between nodes only as desired by source nodes. It maintains these routes as long as they are needed by the sources.
In this paper we have considered BeeAdHoc C protocol which is having certain advantages over all the swarm inspired protocols discussed till now. This is mainly inspired by the cluster based working principle used by Bee-Sensor C protocol [15] used mainly for WSNs. Bee AdHoc C - In this paper, we have compared the performance of an energy-efficient and scalable multipath routing protocol for MANETs, Bee-AdHoc-C by integrating a dynamic clustering scheme and enhanced BeeAdHoc described before. In our protocol, bee agents were modeled to suit the energy resource constraints in MANETs for the purpose of constructing cluster near the event source and finding the multiple paths of better quality, by extensively borrowing from the principles behind the honey bee communication. It has four important features which are listed as follows.

- (i) *Bio inspired technique*: Bee-AdHoc-C is based on bee inspired mechanism.
- (ii) *Dynamic clustering scheme*: Bee-AdHoc-C adopts a dynamic clustering scheme to provide parallel data transmission near the event.
- (iii) *Multiple paths selection*: Bee-AdHoc-C adopts an enhanced multipath construction method to achieve the energy consumption balance.
- (iv) *Multi cluster scenario support*: Bee-AdHoc-C takes the multi cluster scenario into consideration based on the conclusion of our mathematical analysis.

By our experimental evaluation, Bee-AdHoc-C performs better in restricting routing overhead and energy efficiency than other SI based MANET routing protocols. Bee-AdHoc-C is scalable and can significantly balance energy consumption among the nodes.

V.ARCHITECTURE AND WORKING OF BEE-ADHOC-C

Bee-AdHoc-C is an event driven and on-demand multipath routing protocol for MANETs. As shown in Figure 2, Bee-AdHoc-C is mainly divided into three phases: cluster formation, multipath construction, and data transmission. The entire network has been initialized completely and each node has its own ID and relative sink node ID before the cluster formation. The first phase is to build cluster structure when an event happens. In the second phase, improvements are made to Bee-AdHoc for constructing multipath between CH and relative sink, followed by carrying data to sink through the stochastically selected path.

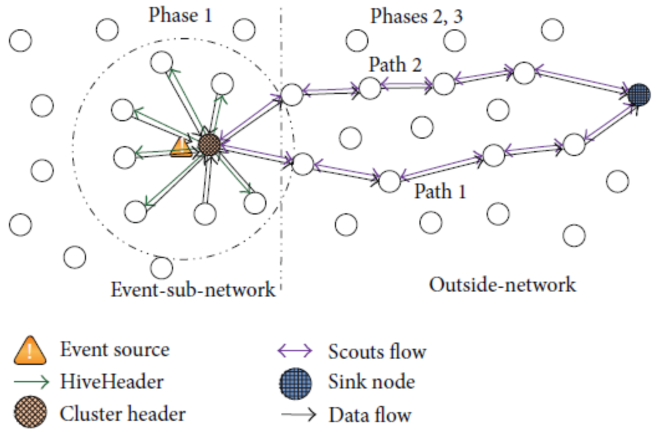

Fig-2: Workflow of Bee-AdHoc-C Network

Cluster Formation.
When an event occurs in the network, all nodes near the event need to send the perceived data in which BeeAdHoc is poor in the overall performance. Therefore Bee-AdHoc-C adopts an event-based dynamic clustering algorithm. Being different from Bee-AdHoc, Bee-AdHoc-C adds a new agent called Hive Header into bee-hive for each AdHoc Network. The major responsibility of Hive Header is to claim that the node wants to be a cluster header (CH) node in an event area. When the event in the network happens, nodes nearby will become activated and measure the specific perceived attribute then the nodes having information about the event will join cluster. The clustering region is called event sub-network and the region outside the event-sub-network is referred to as outside-network. The clustering algorithm obeys the following rule: node i can decide whether to join the cluster according to the received signal strength (RSS) of relative event. If RSSi ⍰ RSS Threshold, node i will be located in the event-sub-network. The size of cluster can be adjusted dynamically according to the threshold value of RSS (RSSThreshold). Once the cluster forms, Bee-AdHoc-C takes an efficient mechanism to select CH node. The principle can be stated in detail as follows. Each node has a Hive Header agent. When node i detects an event, it will wait T_i before broadcasting its Hive Header agent in the event-sub-network (T_i is the waiting time). Hive Header of node i will be replaced by that of node j when it receives Hive Header of node j. If node i receives different Hive Headers from the different nodes simultaneously, it will choose the node with higher residual energy. Finally, the node that firstly sends Hive Header will be the only CH node, and the other nodes become cluster members in the event-sub-network. T_i is calculated using

$$T_i = t\left(\frac{\alpha - (Rradio\ Range - d_i)}{\beta \times E_{r_i}}\right)$$

where t is a constant time to control the value of T_i and can be adjusted appropriately based on specific network conditions. Usually, the value of t is equal to the time that the Hive Header can travel once in the event area. Er_i is the current residual energy of node i. Radio Range is the maximum radio range of node. d_i is the distance from event source to node i. α and β are two user-defined constants which can be adjusted so that different values of T_i can differ approximately by t. Because the energy consumption is relative to the distance between two nodes, the sum of energy consumption is found to be least when the node located in the center of an area communicating with other nodes in the area.

Multipath Construction:
When CH needs sending data to sink node, it looks whether there are appropriate foragers of any nodes in the event sub-network for the existing multipath. If CH fails to find valid foragers, an improved BeeAdHoc algorithm is used to establish multipath between CH and a relative sink node. Bee-AdHoc-C utilizes forward scouts to explore the outside-network in search of an interested sink node using a broadcasting principle. Firstly, CH broadcasts a forward scout to its neighbors in outside-network. However, if there is no limit to the hop of forward scouts then this will result in excess communication overhead. Therefore, an improvement has been done that realizes the self-destruction of forward scouts. The method is to add an allowable maximum value of the hops between CH and sink node into forward scouts. The maximum value is defined as H_max which is a tradeoff between the number of paths and communication overhead and can be estimated. When the number of hops is higher than H_max, self-destruction of forward scouts occurs. In Bee-AdHoc-C, if the residual energy of node i is far less than the average residual energy of the path it will directly drop the forward scout and not do further processing in order to prolong the network lifetime. Secondly, when a lot of nodes are randomly distributed in a target area, there may be dense nodes in a small region. In this situation, it is possible that only one node can be the member of multiple paths even though other nodes are supposed to be potential member of paths.

Data Transmission.
In Bee-AdHoc-C, when the backward scout with unique path ID arrives at the CH, it recruits foragers using the “waggle dance” of honey bees. The dance number represents the quality of the path. According to the minimum energy of node and energy required to cover the path multi path link between the source and sink is decided.
There must be proper repair strategy for the situation of unreachable Sink when all the forward scouts could not reach the destination before their self destruction.
Route maintenance is very important where the path is validated by the swarms and also they collect and update the information about the quality of the path. Mainly this is done after the last forager has reached the sink and came back.

VI.RESULTS AND ANALYSIS

Certain parameters have been considered to find out the performance of all ant and bee adhoc routing protocols used for transfer of data in MANETs. These various parameters explain the efficiency of the MANET with the respective protocol. The comparison is done using NS-2 simulation.

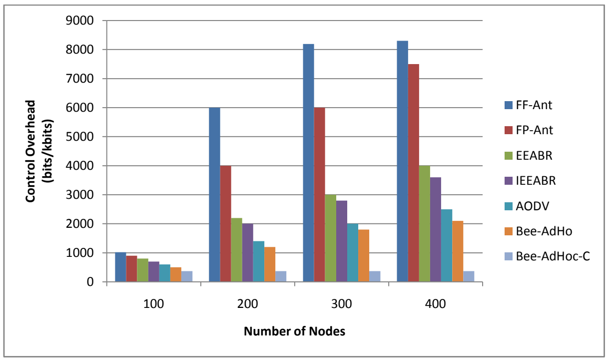

Figure-3 (Control Overhead)

Control Overhead.

Figure 3 shows the control overhead of all protocols. It is the total size of control packets needed by a routing protocol for one kbits data packets received successfully at the sink (bits/kbits). In any MANET routing the primary goal is to establish a correct and efficient route between any pair of nodes with minimum overhead. If the control overhead of a proposed method is very high, then that method cannot work well in MANET. As noticed from the graph Bee-AdHoc-C has the smallest value among all the protocols and remains consistently unchanged. This is mainly due to the technique of clustering and restricted flooding in the phase of multipath construction. The dynamic clustering method reduces the number of parallel event sources, which contributes to the high packet delivery rate because of the less traffic load. However, in Bee AdHoc, each event node needs to broadcast the scout agents to construct respective multipath, which directly leads to more collisions and rapidly increased control overhead with the increasing number of source nodes. Therefore, the performance of Bee-AdHoc is poor, especially with the increase of node density. For the same reason, the other protocols also have high control over head.

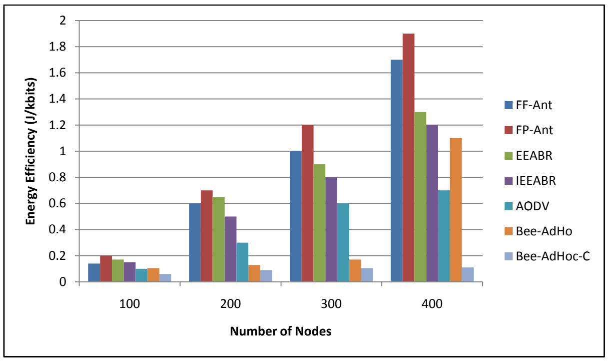

Figure-4 (Energy Efficiency)

Energy Efficiency.

Figure-4 describes the energy efficiency of the protocols. It is defined as the total energy consumed per 1,000 bits delivered at the destination (J/kbits). Bee-AdHoc-C has the best performance in terms of energy efficiency in this scenario. As the number of nodes increases, more traffic is generated in the network, and the total energy consumption rises unavoidably. Due to smoothly increased control traffic and high packet delivery rate, Bee-AdHoc-C performance is the best. Bee-AdHoc has also good performance but as the number of nodes incease the efficiency decreases. FP-Ant is the worst protocol in terms of total energy consumption it is not a pure flooding protocol but performs restrictive flooding of forward ants that carry the data ants.

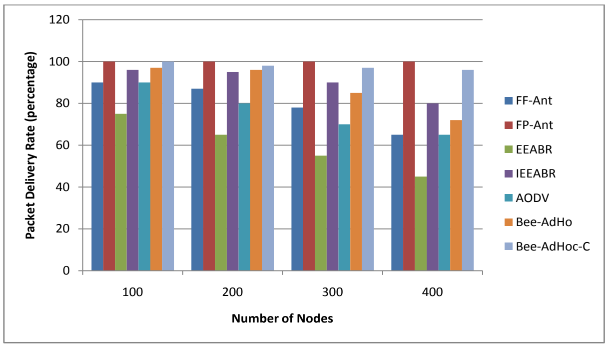

Figure-5(Packet Delivery Rate)

Packet Delivery Rate.

Figure 5 shows the delivery rate of the protocols under evaluation. It is the ratio of the total number of events received at a sink node to the total number of events generated by the nodes in the adhoc network (an event is dispatched in a packet). The loss ratio is:

$$\textit{Loss ratio} = 1 - \textit{Packet delivery ratio}$$

It is obvious that the delivery rate of Bee-AdHoc-C remains close to the maximum value with the increased number of nodes. Higher control traffic and data traffic cause more collisions and packet loss. Although the node density is increasing, the cluster structure in Bee-AdHoc-C significantly reduces the network traffic so that it can achieve the best performance in delivery rate. However, due to more collisions and congestion, the other protocols have worse delivery rate with the increment of node density. FP-Ant has the highest packet delivery ratio, followed by Bee-AdHoc. The results demonstrate that the packet delivery ratio of Bee-AdHoc-C is significantly higher than that of *EEABR* and *AODV*. The obvious reason for this superior performance is that it reduces the amount of control traffic in the network which results in quick convergence of the protocol.

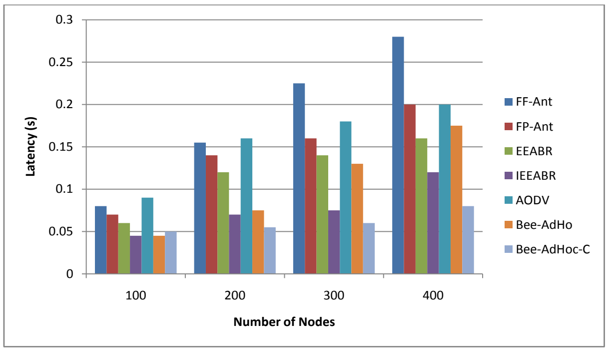

Figure-6 (Latency)

Latency.
The latency of protocols is plotted in Figure 6. It is defined as the difference in time when an event packet is generated at a source node and it is eventually received at the sink node. Bee-AdHoc-C has better performance compared with other protocols mainly due to its fewer collisions. In Bee-AdHoc-C, the paths between CH and sink are robust. Although constructing cluster and collecting the data in the cluster take some time in Bee-AdHoc-C, the time is relatively short against delay brought about by frequent collisions of data and control packets in other protocols. These reasons are supported by the fact that the result of Bee-AdHoc-C is the best in this scenario. Also the performance of Bee-AdHoc and IEEABR are good but they don't provide consistent performance. IEEABR launches the *forward ants* at regular intervals from all nodes in the network. As a result, the complete network becomes unstable due to a large amount of control traffic, and this behavior is more pronounced in larger topologies. In BeeAdHoc the higher delay is due to the dynamic nature of an application, which frequently initiates a new route discovery process.

VII.CONCLUSION AND FUTURE SCOPE

Thus from all the above discussions we conclude that Bee colony optimization is definitely better than ant colony optimization with respect to different parameters comparison done for different routing protocols for MANETs. Any ways the Bee-AdHoc-C is found to be the best protocol for MANETs which is a multipath routing protocol based on dynamic clustering and foraging behavior of bee swarm.

In future other parameters like Energy standard deviation, Routing building time, Lifetime, Algorithmic complexity, Control efficiency, and Control complexity can also be considered for the comparison of different protocols for MANETs.

As in the case of packet delivery rate the FP-Ant protocol has 100 percent efficiency so some future research can be done to get a hybrid protocol which is a combination of ant colony as well as bee colony protocols.

References

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz. Inspiration for optimization from socialinsect behaviour. *Nature*, 406:39–42, 2000.

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

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

[4] P.P. Grass´e. La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. la th´eorie de la stigmergie: essai d'interpr´etation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–81, 1959.

[5] G. Di Caro and M. Dorigo. Ant colonies for adaptive routing in packet-switched communications networks. In *Parallel Problem Solving from Nature – PPSN V, LNCS 1498*,pages 673–682, Sept 1998.

[6] G. Di Caro and M. Dorigo. Mobile agents for adaptive routing. In *31st Hawaii InternationalConference on System Science*, pages 74–83, Big Island of Hawaii, 1998. IEEE Computer Society Press.

[7] G. Di Caro and M. Dorigo. An adaptive multi-agent routing algorithm inspired by ants behavior. In *Proceedings of 5th Annual Australasian Conference on Parallel Real Time Systems*, pages 261–272, 1998.

[8] G. Di Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communication networks. *Journal of Artificial Intelligence Research*, 9:317–365, December 1998.

[9] G. Di Caro. Ant colony optimization and its application to adaptive routing in telecommunication networks. Ph.D. thesis, Universit´e Libre de Bruxelles, Belgium, 2004.

[10] Y. Zhang, L.D. Kuhn, M.P.J. Fromherz, "Improvements on Ant Routing for Sensor Networks", in: M. Dorigo et al.(Eds.), ANTS 2004, LNCS 3172, pp. 289-313, 2004.

[11] A.M. Zungeru, L.-M. Ang, K.P. Seng, "Classical and Swarm Intelligence Routing Protocols for Wireless Sensor Networks: A Survey and Comparison", *Journal of Networks and Computer Applications (ELSEVIER)*, 2012, article in press, doi:10.1016/j.jnca.2012.03.004.

[12] M. Saleem, G.A. Di Caro, and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions", *Information Sciences*, vol. 181(20), pp. 4597-4624, 2010.

[13] K. Li, C.E. Torres, K. Thomas, L.F. Rossi, and C.-C. Shen, "Slime mold inspired routing protocols for wireless sensor networks", *Swarm Intelligence*, vol. 5, pp. 183-223, 2011.

[14] K.M. Sim and W.H. Sun. Ant colony optimization for routing and load-balancing: survey and new

directions. *IEEE Transactions on Systems, Man and Cybernetics–Part A*,33(5):560–572, 2003.

[15] Xuelian Cai, Yulong Duan, Ying He, Jin Yang, and Changle Li, Bee-Sensor-C: An Energy-Efficient and Scalable Multipath Routing Protocol for Wireless Sensor Networks, International Journal of Distributed Sensor Networks Volume 2015,pp.224-238, August 2014.

[16] T.C. Camilo, Carreto, J.S. Silva, and F. Boavida, "An Energy-Efficient Ant Based Routing Algorithm for Wireless Sensor Networks", in: Proceedings of 5th International Workshop on Ant Colony Optimization and Swarm Intelligence, Brussels, Belgium, pp.49-59, 2006.

[17] M. Saleem, S. A. Khayam, and M. Farooq, "A formal performance modeling framework for bio-inspired Ad Hoc routing protocols," in *Proceeding of the 10th Annual Genetic and Evolutionary Computation Conference (GECCO '08)*, pp. 103–110, New York, NY, USA, July 2008.

[18] D. Karaboga, S. Okdem, and C. Ozturk, "Cluster based wireless sensor network routing using artificial bee colony algorithm," *Wireless Networks*, vol. 18, no. 7, pp. 847–860, 2012.

[19] J. Kennedy and R. C. Eberhart, "A new optimizer using particle swarmtheory," in *Proceedings of 6th International Symposium on Micro Machine and Human Science*, pp. 39–43, Nagoya, Japan, 1995.

[20] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '98)*, pp. 69–73, Anchorage, Alaska, USA, May 1998.

[21] R. Ahuja, S. Keshav, and H. Saran. Design, implementation, and performance measurement of a native-mode ATM transport layer (extended version). *IEEE/ACM Transaction on Networks*, 4(4):502–515, 1996.

[22] U. Aickelin, P. Bentley, S. Cayzer, J. Kim, and J. McLeod. Danger theory: the link between AIS and IDS. In *Proceedings of the 2nd International Conference on Artificial Immune Systems*, volume 2728 of *LNCS*, pages 147–155. Springer, 2003.

[23] U. Aickelin and S. Cayzer. The danger theory and its applications to artificial immune systems. In *Proceedings of the 1st International Conference on Artificial Immune Systems*, volume 2728 of *LNCS*, pages 141–148. Springer, 2002.