International Journal of Advanced Multidisciplinary Research (IJAMR)
ISSN: 2393-8870
www.ijarm.com

Research Article

A Review on intelligence secure routing protocol in the Mobile AD HOC Networking

[1]Himadri Nath Saha, [2]Kuhelica Hazra, [3]Ishani Mondal, [4]Monalisa Chakraborty and [5]Sunrita Sarkar

[1,2,3,4,5]Department of Computer Science, Institute of Engineering and Management, Y-12, Block-EP, Salt Lake Electronics Complex, Kolkata-700091, West Bengal, India
Corresponding Author : *shani340@gmail.com/himadri@iemcal.com/kly.kuhelica@gmail.com/monaiou0@gmail.com*

Keywords

Ad hoc Network
Swarm Intelligence
Secure Routing Ant colony
Bee colony
Particle Swarm Optimization

Abstract

Mobile ad hoc networks (MANETs) are infrastructure-less, multi-hop, wireless mobile networks formed spontaneously. It refers to any set of networks where all devices have equal status on a network and are free to associate with any other ad hoc network device in link range. The ad hoc network does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data to other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity. Ad hoc networks are useful when infrastructure is not available, is impractical, or is expensive. It is mainly used for military applications, rescue and for home networking. Many conventional routing algorithms have been proposed for MANETs. An emerging area that has recently captured much attention in network routing researches is Swarm Intelligence (SI). Swarm intelligence deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization. In particular, it focuses on the collective behaviors that result from the local interactions of the individuals with each other and with their environment. Examples of systems studied by swarm intelligence are colonies of ants and termites, schools of fish, flocks of birds, herds of land animals. In this paper we explore the SI approach to address routing problem in ad hoc networks with the goal to reduce routing overheads.

Introduction

In Mobile Ad hoc Networks nodes are self-organized and use wireless links for communication between themselves. They dynamically form a temporary network without using any existing network infrastructure or centralized administration. These are often called infrastructure-less networking since the mobile nodes in the network dynamically establish routing paths between themselves. Examples are conference, battlefield, rescue scenarios, sensor networks placed in an area to monitor the environment, mesh networks for wireless Internet access etc. Routing solutions must address the nature of the network, and aim at minimizing control traffic, to preserve both bandwidth and energy at nodes.

Mobile Ad hoc Network (MANET) Routing Protocols

The specific challenges and possible applications of MANETs have made this a very popular research area, and a lot of routing algorithms have been proposed. People traditionally classify these algorithms as either proactive or reactive. In purely proactive protocols (e.g., DSDV) nodes try to maintain at all times routes to all other nodes. This means that they need to keep track of all topology changes, which can become difficult if there are a lot of nodes or if they are very mobile. Therefore, reactive protocols (e.g., AODV or DSR) are in general more scalable. In these protocols, nodes only gather routing information on demand: only when they have data for a certain destination they construct a path, and only when the path becomes infeasible they search a new path. In this way they greatly reduce the routing overhead, but they can suffer from oscillations in performance since they are never prepared for disruptive events. Hybrid algorithms like ZRP have both a proactive and a reactive component, in order to try to combine the best of both worlds. Most of the algorithms are single path: at any time, they use only one path between source and destination. Multipath routing offers an interesting alternative in terms of link failure robustness and load balancing. Some algorithms create multiple paths at path setup time, and use the best of these until it fails, after which they switch to the second best and so on (e.g., AODV-BR). A problem with this way of working is that alternative paths are often infeasible by the time they need to be used. Moreover, when only the best path is used, one loses the opportunity to spread data packets over the different paths, a practice which can improve the network throughput.

The first ant-based routing algorithms were ABC and AntNet. Both algorithms follow a similar general strategy. Nodes send ant agents out at regular intervals to randomly chosen destinations. The main aim of the ants is to sample the paths, assign a quality to them, and use this information to update the routing tables in the nodes they pass. These routing tables contain an entry for each destination and each neighbor, indicating the goodness of going over this neighbor on the way to the destination. This goodness value is called pheromone. This pheromone information is used for the routing of both ants and data packets: all packets are routed stochastically, choosing with a higher probability those links with higher pheromone values. If enough ants are sent to the different destinations, nodes keep up-to-date information about the best paths, and automatically adapt their data load spreading to this. Ant-based routing algorithms have a number of properties which are desirable in MANETs: they are highly adaptive to network changes, use active path sampling, are robust to agent failures, provide multipath routing, and take care of data load spreading. However, the fact that they crucially rely on repeated path sampling can cause significant overhead if not dealt with carefully. There have been a number of attempts to design ant-based routing algorithms for MANETs. Examples are ARA and PERA. However, these algorithms loose much of the proactive sampling and exploratory behavior of the original ant-based algorithms in their attempt to limit the overhead caused by the ants.

There are four multi-hop wireless ad hoc network routing protocols that cover a range of design choices : Destination-Sequenced Distance-Vector (DSDV),Dynamic Source Routing (DSR),Ad Hoc On-Demand Distance Vector Routing (AODV).While DSDV is a table-driven routing protocol, DSR, AODV, fall under the On-demand routing protocols category. Our New Approach is a hybrid multipath algorithm, designed along the principles of ACO routing. It consists of both reactive and proactive components. It gains the advantages of both proactive and reactive routing algorithms.i) Route Discovery– Reactive Low routing overheads, ii) Route Maintenance -- Proactive Dynamic path maintenance, iii) Ant Colony Optimization- Small Ant , converge to best path quickly.iv) Multi Path Routing-- Congestion control and load balancing.It does not maintain paths to all destinations at all times (like the ACO algorithms for wired networks), but sets up paths when they are needed at the start of a session. This is done in a reactive path setup phase, where Ant agents called reactive forward Ants are launched by the source in order to find multiple paths to the destination, and backward Ants return to set up the paths. The paths are represented in pheromone tables indicating their respective quality. After path setup, data packets are routed stochastically as datagram's over the different paths using these pheromone tables. While a data session is going on, the paths are probed, maintained and improved proactively using different agents, called proactive forward Ants. The algorithm reacts to link failures with either local path repair or by warning preceding nodes on the paths.

Security

A lot of research has been done in the past but the most significant contributions have been the PGP (Pretty Good Privacy) and trust based security. None of the protocols have made a decent tradeoff between security and performance. In an attempt to enhance security in MANETs many researchers have suggested and implemented new improvements to the protocols and some of them have suggested new protocols.

Attack classifications

These attacks on MANETs challenge the mobile infrastructure in which nodes can join and leave easily with dynamics requests without a static path of routing. Schematics of various attacks as described on individual layer are as under:

- Application Layer: Malicious code, Repudiation
- Transport Layer: Session hijacking, Flooding
- Network Layer: Sybil, Flooding, Black Hole, Grey Hole. Worm Hole, Link Spoofing, Link Withholding, Location disclosure etc
- Data Link/MAC: Malicious Behavior, Selfish Behavior, Active, Passive, Internal External
- Physical: Interference, Traffic Jamming, Eavesdropping.

Swarm-based Network Management

The first swarm-based approaches to network management were proposed in 1996 by Schoonderwoerd et al., and in 1998 by Di Caro and Dorigo. Schoonderwoerd et al. proposed *Ant-based Control* (ABC), an algorithm for routing and load balancing in circuit-switched networks; Di Caro and Dorigo proposed AntNet, an algorithm for routing in packet-switched networks. While ABC was a proof-of-concept, AntNet, which is an ACO algorithm, was compared to many state-of-the-art algorithms and its performance was found to be competitive especially in situation of highly dynamic and stochastic data traffic as can be observed in Internet-like networks. An extension of AntNet has been successfully applied to ad-hoc networks (Di Caro, Ducatelle and Gambardella 2005). These algorithms are another example of successful artificial/engineering swarm intelligence system.

Cooperative Behavior in Swarms of Robots

There are a number of swarm behaviors observed in natural systems that have inspired innovative ways of solving problems by using swarms of robots. This is what is called swarm robotics. In other words, swarm robotics is the application of swarm intelligence principles to the control of swarms of robots. As with swarm intelligence systems in general, swarm robotics systems can have either a scientific or an engineering flavour. Clustering in a swarm of robots was mentioned above as an example of artificial/scientific system. An example of artificial/engineering swarm intelligence system is the collective transport of an item too heavy for a single robot, a behavior also often observed in ant colonies.

SWARM INTELLIGENCE*:*

Swarm intelligence is the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization. In particular, the discipline focuses on the collective behaviors that result from the local interactions of the individuals with each other \

and with their environment. Examples of systems studied by swarm intelligence are colonies of ants and termites, schools of fish, flocks of birds, herds of land animals. Some human artifacts also fall into the domain of swarm intelligence, notably some multi-robot systems, and also certain computer programs that are written to tackle optimization and data analysis problems.

In nature, several animals tend to live in large swarms because in the swarm each animal is more effective for evolution than single animals. They exhibit an astonishingly well-developed social behavior and are able to self-organize even in the absence of central leader as like honey bees. Honey bees communicate the locations of the food sources by the language of bee-dance, some other insects use an indirect form called stigmergyi.e. leaving traces in the environment that can be understood by other insects. This way groups of insects manage to synchronize so that they all work on the same spot.

This concept is employed in work on artificial intelligence. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local, and to a certain degree random, interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents. Examples in natural systems of SI include ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling. The definition of swarm intelligence is still not quite clear. In principle, it should be a multi-agent system that has self-organized behavior that shows some intelligent behavior.

The application of swarm principles to robots is called swarm robotics, while 'swarm intelligence' refers to the more general set of algorithms. 'Swarm prediction' has been used in the context of forecasting problems.

Natural vs. Artificial: It is customary to divide swarm intelligence research into two areas according to the nature of the systems under analysis. We speak therefore of *natural* swarm intelligence research, where biological systems are studied; and of *artificial* swarm intelligence, where human artifacts are studied.

Scientific vs. Engineering: An alternative and somehow more informative classification of swarm intelligence research can be given based on the goals that are pursued: we can identify a *scientific* and an *engineering* stream. The goal of the scientific stream is to model swarm intelligence systems and to single out and understand the mechanisms that allow a system as a whole to behave in a coordinated way as a result of local individual-individual and individual-environment interactions. On the other hand, the goal of the engineering stream is to exploit the understanding developed by the scientific stream in order to design systems that are able to solve problems of practical relevance.

The typical swarm intelligence system has the following **properties:**

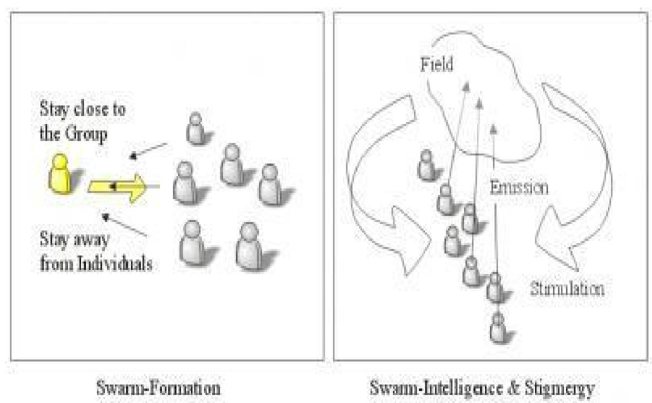

Figure: 1

- it is composed of many individuals; the individuals are relatively homogeneous (i.e., they are either all identical or they belong to a few typologies);
- the interactions among the individuals are based on simple behavioral rules that exploit only local information that the individuals exchange directly or via the environment;
- The overall behavior of the system results from the interactions of individuals with each other and with their environment, that is, the group behavior self-organizes.

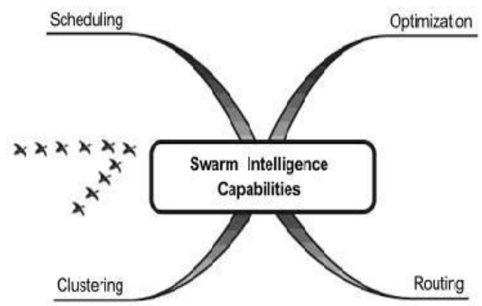

Fig. 6.11 Key capabilities of swarm intelligence

Because of the above properties, it is possible to design swarm intelligence system that are scalable, parallel, and fault tolerant.

APPLICATIONS OF SWARM INTELLIGENCE

Beckers et al. (1994) have programmed a group of robots to implement clustering behavior of ants. This is one of the first swarm intelligence scientific oriented studies in which artificial agents were used.

A number of swarm intelligence studies have been performed with swarms of robots for validating mathematical models of biological systems.

In a now classic experiment done in 1990, Deneubourg and his group showed that, when given the choice between two paths of different length joining the nest to a food source, a colony of ants has a high probability to collectively choose the shorter one. Deneubourg has shown that this behavior can be explained via a simple probabilistic model in which each ant decides where to go by taking random decisions based on the intensity of pheromone perceived on the ground, the pheromone being deposited by the ants while moving from the nest to the food source and back.

Some of the Swarm-Inspired Methods are
Ant colony optimization – ACO
Particle swarm optimization – PSO

METHODOLOGY

ANT COLONY OPTIMIZATION (ACO):

Ant Colony Optimization (ACO) is the primary optimization algorithm being used in Swarm Intelligence .The reason behind choosing ACO is its distributed nature and inherent randomness. Owing to its randomness, it prevents convergence to local maxima/minima but tend to search for global optimization technique.

Why **Ant Colony Optimization** is suitable for mobile ad hoc network?

- The algorithm is fully distributed ⇒there is no single point of failure;
- The operations to be performed in each node are very simple;
- The algorithm is based on an asynchronous and ⇒autonomous interaction of agents;
- It is self-organizing, thus robust and fault tolerant there is no need of defining path recovery algorithms;
- It is intrinsically traffic adaptive without any need for complex and yet inflexible metrics;
- It is inherently adaptive to all kinds of long-term variations in topology and traffic demand, which are difficult to be taken into account by deterministic approaches.

It is a branch of the newly evolving field of computer science known as Artificial Intelligence which deals with the natural phenomenon of biological species such as insects, bees etc., giving birth to an easy solution of finding the optimized routing path in a mobile ad hoc (for a specific purpose) network which is quite vulnerable due to wireless communication medium and the presence of mobile nodes which can establish connection at their own will. MANET is a self-organized wireless network, due to the fact it has vulnerable attacks that can easily damage the whole network; that's why there should be some solutions which works even some of the mobile nodes compromised in the network. One of the primary challenges of secure routing is to provide authentication (trustworthiness) of users in the network. In case of distributed communication environment in MANET, authentication is open and any un-authentic node may be use to compromise routing traffic in order to disrupt the communication. There are some of the major responsibilities of secure routing which are given below.

- It provides assurance that modified and replayed route replies should be rejected in order to avoid fabrication of attacks.
- Routing protocol responsiveness itself provide safety among different routing attacks.

Ant colony optimization (ACO) is an optimization technique inspired by the exploratory behavior of ants while finding food. Although not being aware

we almost daily see applied swarm intelligence when we go to a park or on the street. Everybody knows bees or ants, but we would not regard them as intelligent since they are not capable of executing a complex task on their own. Nevertheless a swarm of bees or ants is intelligent since the insects influence each other's behavior e.g. through pheromones and in the end intelligent behavior emerges through the strategy known as stigmergy. For instance are ants capable of finding the shortest way back to their nest through going in the direction of the highest pheromone concentration. Thus you can say they take the approved path.

While searching the environment for food, the ants deposit pheromones on the ground. Other ants are attracted by pheromones and tend to follow trails of previous ants. This mechanism enables the ants to find shortest paths between the nest and a food source. The basic principle of an ant routing algorithm is mainly the depositing of pheromone on the path followed by the ant. They follow simple rule of following the path which has higher concentration of pheromone. The pheromone concentrations on a path allow the other ants to find their way to the food source. Thereby more ants follow the same path and more and more pheromone is deposited on the path which is the shortest route to the food source. It was found that the pheromone-trail-following behavior gives rise to the emergence of the shortest path which is followed by other ants of the colony. When a previously short route get blocked/ lengthened due to an obstacle in route, the alternate short route get strengthened with higher pheromone content due to shorter end-to-end travel time and more ants move to this route. Hence the path can also dynamically adapt to fast changes in the environment. The following flow chart can give us a clear idea about their strategy:-

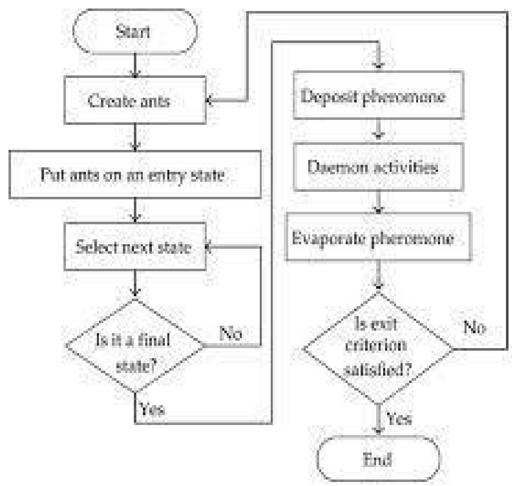

Figure: 2

Ant colony based routing algorithm(ARA):

This algorithm uses two mobile agents FANT and BANT. FANT agent having unique sequence number and source address is broadcasted by the sender and will be relayed by the neighbors of the sender. A node receiving a FANT for the first time creates a record (destination address, next hop, pheromone value). The node interprets the source address of the FANT as destination address, the address of the previous node as the next hop, and the number of hops the FANT needed to reach this node decides the pheromone value.

Thus FANT creates the pheromone track to the source node. The destination node extracts the information of the FANT, destroys it, and creates BANT which establishes the pheromone track to the destination node. The sender starts data transmission after receiving BANT. No special packets are needed for route maintenance. Subsequent date packets are used instead to maintain the route. For a node A sending a data packet to destination D through a neighbor B pheromone value of the entry (D, B, m) is increased by Δm, thus strengthening the path to destination. Also, the next hop B increases the pheromone value of the entry (S, A, m) by Δm, thus strengthening the path to source node S. On receiving a duplicate packet, node sets the DUPLICATE ERROR flag, sends the packet back to the previous node refraining that node from sending more data packets in this direction, and hence preventing loops. When route failure occurs, node deactivates that path by reducing pheromone value to 0 in corresponding route table entry.

Either the node sends the packet through alternate path if it exists, or the node try to transmit this packet through its neighbors. If packet still not reaches the destination, source initiates a new route.

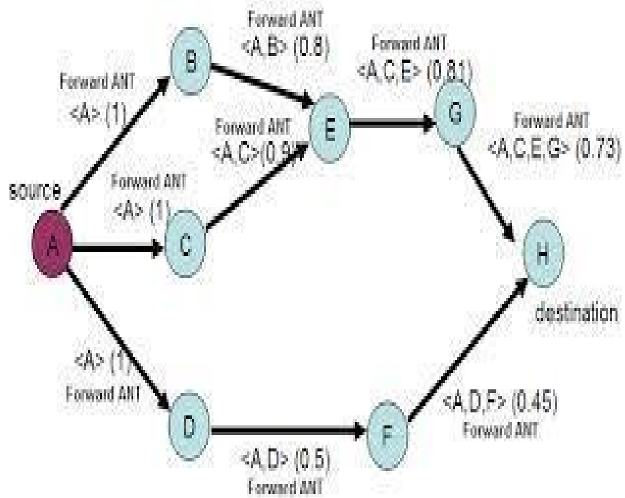

Figure: 3

The BANT will take the same path as the FANT but in the opposite direction. When the sender receives the BANT from the destination node, the path is established and data packets can be sent.

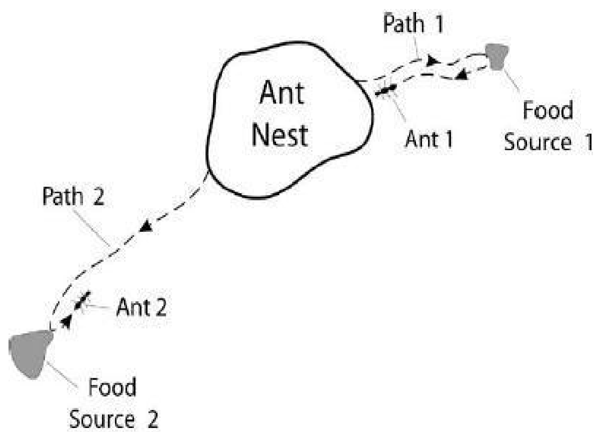

Figure: 4

Overhead of ARA:

The expected overhead of ARA is very small, because there are no routing tables which
Are interchanged between the nodes. Unlike other routing algorithms, the FANT and
BANT packets do not transmit much routing information. Only a unique sequence number is transmitted in the routing packets. Most route maintenance is performed through data packets, thus they do not have to transmit additional routing information. ARA only needs the information in the IP header of the data packets.

Ant Hoc Net:

AntHocNet is a hybrid multipath algorithm. When a data session is started at node
S with destination d, s checks whether it has up-to-date routing information for d . If not, it reactively sends out ant-like agents, called reactive forward ants, 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 backward ants 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 hop n a value Ti n d IR Ti n d is an estimate of the goodness of the path over n to d , which we call pheromone.

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. They are stochastically spread over the paths: in each node they select the next hop with a probability proportional to its pheromone value.

Reactive path setup

Reactive forward ants looking for a destination d are either broadcasted or unicasted, according to whether or not the node they are currently in has routing information for d. Due to the broadcasting, ants can proliferate quickly over the network, following different paths to the destination. When a node receives several ants of the same generation (i.e., they started as the same original forward ant at the source), it will compare the path travelled by the ant to that of the previously received ants of this generation: only if its number of hops and travel time are both within a certain factor (a parameter which we empirically set to1.5) of that of the best ant of the generation, it will forward the ant. Using this policy, overhead is limited by removing ants which follow bad paths, while the possibility to find multiple good paths is not hindered.

The main task of the reactive forward ant is to find a path connecting s and d. It keeps a list P of the nodes [1;:::;n] it has visited. Upon arrival at the destination d, the forward ant is converted into a backward ant, which travels back to the source retracing P. The backward ant incrementally computes an estimate ^TP of the time it would take a data packet to travel over P towards the destination, which is used to update routing tables. ^TP is the sum of local estimates ^Ti!i+1in each node i2P of the time to reach the next hop i+ 1:^TP=Pn°1i=1^Ti!i+1. The value of^Ti!i+1is defined as (Qimac+ 1) ^ Tmac the product of the estimate of the average time to send one packet,^Timac, times the current number of packets in queue (plus one) to besent at the MAC layer.

Stochastic data routing

The path setup phase described above creates a number of good paths between source and destination, indicated in the routing tables of the nodes. Data can then be forwarded between nodes according to the values of the pheromone entries. Nodes in AntHocNet forward data stochastically. When a node has multiple next hops for the destination d of the data, it will randomly select one of them, with the probability P.

We take the square in order to be more greedy with respect to the better paths. According to this strategy, we do not have to choose a priori how many paths to use: their number will be automatically selected in function of their quality. The probabilistic routing strategy leads to data load spreading with consequent automatic load balancing. When a path is clearly worse than others, it will be avoided, and its congestion will be relieved. Other paths will get more traffic,
leading to higher congestion, which will make their end-to-end delay increase.

By continuously adapting the data traffic, the nodes try to spread the data load evenly over the network. This is quite important in MANETs, because the bandwidth of the wireless channel is very limited. Of course, to do this properly, it is important to frequently monitor the quality of the different paths. To this end we use the proactive ants.

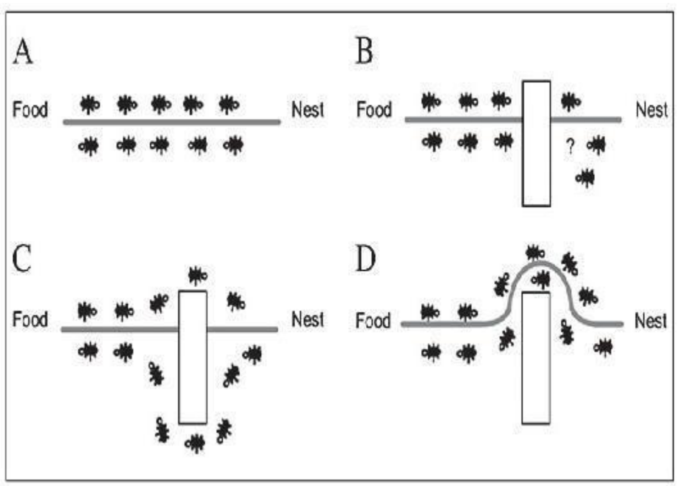

Figure 2. A. Ants in a pheromone trail between nest and food; **B.** an obstacle interrupts the trail; **C.** ants find two paths to go around the obstacle; **D.** a new pheromone trail is formed along the shorter path.

Figure: 5

PACONET:

PACONET is a routing protocol for mobile ad hoc networks inspired by the foraging behavior of ants. It uses the principles of ACO routing to develop a suitable problem solution. To find routes from source to destination FANT explores the network in a restricted broadcast manner. The BANT retraces the path of FANT and updating the pheromone value at same time. Data packets are stochastically transmitted towards nodes having higher pheromone concentration along the path to the destination. Each node in the network has a routing table whose size is the degree of the node times all the nodes in the network. The rows of the routing table of node N represent the neighbors of node N and the columns represent all the nodes in the network. Each pair (row, column) in the routing table has two values namely a binary value indicating if the node has been visited and the pheromone concentration. For discovering route from S to D, when FANT from a source S arrives at a node N, the FANT can select best path from neighboring node to D by looking at the rows against the column D in the routing table. For all unvisited neighbors in column D, FANT will choose the node with the highest pheromone as the next hop. At destination BANT created from FANT, travels towards source updating the pheromone concentration for the destination column.

HELLO messages are used to detect and monitor links to neighbors. In case of not receiving several HELLO messages from a neighbor, a link breakage is detected and its routing table can be updated by deleting the entries in the routing table for that neighbor. In PACONET, a node would wait several seconds to confirm a failed HELLO packet before modifying its routing table. Data packets going through this node towards the newly broken link will end up being lost due to the delay in updating the routing table.

PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a population-based stochastic approach for solving continuous and discrete optimization problems. In particle swarm optimization, simple software agents, called *particles*, move in the search space of an optimization problem. The position of a particle represents a candidate solution to the optimization problem at hand. Each particle searches for better positions in the search space by changing its velocity according to rules originally inspired by behavioral models of bird flocking. Particle swarm optimization belongs to the class of swarm intelligence techniques that are used to solve optimization problems.

BEE COLONY OPTIMIZATION

The Bees Algorithm is a population-based search algorithm which was developed in 2005. It mimics the food foraging behavior of honey bee colonies. The Bee Colony based algorithms are inspired from the food foraging behavior of honey bees whose colony can typically extend itself over 10 km to 14 km and in multiple directions simultaneously to exploit a large number of food sources. The colony tends to attain the optimal use of its members. Theoretically, the richer (more nectar or pollen) and closer a food source is, the more bees that would be sent to it. The food foraging process begins with scout bees' searching activities. They fly randomly to explore all the food sources in all directions. When they return to the hive, they report the searching results to the others by performing *waggle dance*, which communicates information about the food source, including its direction, distance and quality rating. The *waggle dance* is essential in the food source evaluation and it helps the colony to deploy flower bees to the food sources precisely.

A. Artificial Bee Colony Algorithm (ABC):

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga for optimizing numerical problems. The algorithm simulates the intelligent foraging behavior of honey bee swarms. It is a very simple, robust and population based stochastic optimization algorithm.

The performance of the ABC algorithm is compared with those of other well-known modern heuristic algorithms such as Genetic Algorithm (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO) on constrained and unconstrained problems. The performance of ABC algorithm on training neural networks is examined by tested on XOR, Decoder–Encoder and 3-Bit Parity benchmark problems and by [32] tested on pattern classification against widely used gradient-based and population based optimization algorithms.

The foraging bees are classified into three categories; employed bees, onlookers and scout bees. All bees that are currently exploiting a food source are known as employed. The Pseudo code for ABC algorithm

1. **Initial** food sources are produced for all worker bees.
2. **Do**
   1) Each worker bee goes to a food source and evaluates its nectar amount.
   2) Each onlooker bee watches the dance of worker bees and chooses one of their sources depending on the dances and evaluates its nectar amount.
   3) Determine abandoned food sources and replace with the new food sources discovered by scouts.
   4) Best food source determined so far is recorded.
3. **While** (requirements are met)

In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making a decision to choose a food source is called onlooker and one going to the food source visited by it before is named employed bee. The other kind of bee is scout bee that carries out random search for discovering new sources. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution, in the algorithm, the first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. The number of the employed bees or the onlooker bees is equal to the number of solutions (the cluster centers) in the population. At the first step, the ABC generates a randomly distributed initial population P(C = 0) of SN solutions (food source positions), where SN denotes the size of population. Each solution z_i where i= 1, 2, . . . , SN is a D-dimensional vector. Here, D is the number of product of input size and cluster size for each data set, i.e. the number of optimization parameters. After initialization, the population of the positions (solutions) is subjected to repeated cycles, C = 1, 2, . . . , MCN, of the search processes of the employed bees, the onlooker bees and scout bees. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). Provided that the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory. After all employed bees complete the search process, they share the nectar information of the food sources and their position information with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. As in the case of the employed bee, she produces modification on the position in her memory and checks the nectar amount of the candidate source. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

An artificial onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, calculated by the following expression ($p_i = fit_i \sum_{n=1}^{SN} fit_n$ where SN is the number of food sources equal to the number of employed bees, and fit_i is the fitness of the solution given in Eq. which is inversely proportional to the f_i given in Eq. where f_i is the cost function of the clustering problem. In order to produce a candidate food∈ position from the old∈one in memory, the ABC uses the following expression :$v_{ij} = z_{ij} + \_ij(z_{ij} - z_{kj})$ where k {1, 2, . . . , SN} and j {1, 2, . . . , D} are randomly chosen indexes. Although k is determined randomly, it has to be different from i. _i,jis a random number between [−1, 1]. It controls the production of neighbor food sources around $z_{i,j}$and represents the comparison of two food positions visible to a bee. As can be seen from , as the difference between the parameters of the zi, j and zk, j decreases, the perturbation on the position zi, j decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced. The food source of which the nectar is abandoned by the bees is replaced with a new food source by the scouts. In ABC, this is simulated by producing a position randomly and replacing it with the abandoned one. In ABC, providing that a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predetermined number of cycles is an important control ∈parameter of the ABC algorithm, which is called "limit" for abandonment. Assume that the abandoned source is z_i and j {1, 2, . . . , D}, then the scout discovers a new food source to be replaced with z_i. This operation can be defined as in

$$\mathbf{Z_{ji} = Z_{jmin} + rand(0, 1)(Z_{jmax} - Z_{imin})}$$

After each candidate source position $v_{i,j}$is produced and then evaluated by the artificial bee, its performance is compared with that of its old one. If the new food source has an equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained in the memory. In other words, a greedy selection mechanism is employed as the selection operation between the old and the candidate one. There are three control parameters in the ABC: the number of food sources which is equal to the number of employed or onlooker bees (SN), the value of limit, the maximum cycle number (MCN).In a robust search process, exploration and exploitation processes must be carried out together. In the ABC algorithm, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process. The local search performance of ABC algorithm depends on neighborhood search and greedy selection mechanisms performed by employed and onlooker bees. The global search performance of the algorithm depends on random search process performed by scouts and neighbor solution production mechanism performed by employed and onlooker bees.

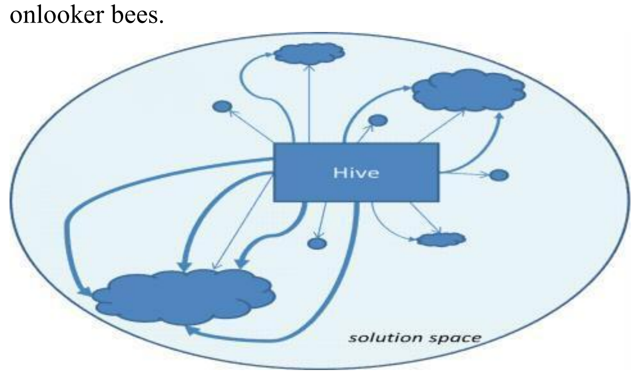

Figure: 6

B. Bee Ad-Hoc

Wedde, Farooq, et al. (2004) designed BeeAdHoc [4] with the aim of defining a MANET routing algorithm. Honey bee behavior served as the major inspiration in designing different types of agents at work in the system and its interaction.

Architecture of Bee Ad-Hoc analogus to BEEHIVE

Each node in MANET has a hive, which consists of three parts: packing floor, entrance and dance floor. The structure of the hive is shown in Figure below. The entrance is an interface to MAC (Medium Access Control) layer while packing floor is an interface to transport layer. All packets depart/enter the hive through the entrance. The dance floor contains the foragers (routing information) for routing of data packets originated at the node.

Packing Floor:

The packing floor is an interface to higher level transport layer like TCP or UDP. Once a data packet arrives from the transport layer, a packer is created in the packing floor which stores the data packet. After that the packer tries to locate a suitable forager for the data packet from dance floor. If it finds one then it hand over's the data packet to the forager and dies. Otherwise, it waits for a time (may be a returning forager is on its way toward the current hive) and if no forager arrives within this time, then it launches a scout which is responsible for discovering new routes to the destination of the data packet.

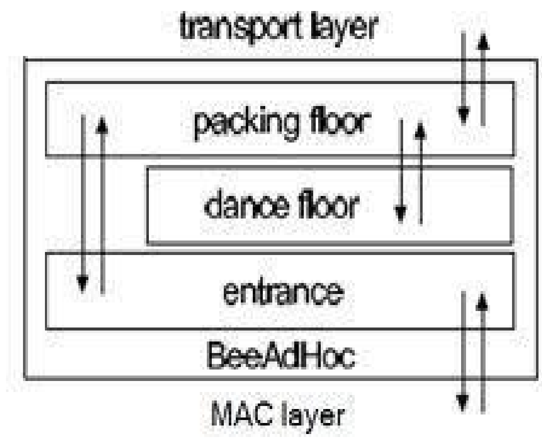

Figure: 7

Entrance Floors

The entrance is an interface to lower level MAC layer. The entrance handles all incoming/outgoing packets. A scout received at the entrance is broadcasted further if its time to live (TTL) timer has not expired or if it has not arrived at the destination. The information about the id of the scout and its source node is stored in a table. If another replica of an already received scout arrives at an entrance of a hive then the new replica is killed here. If a forager with a same destination as that of the scout already exists in the dance floor then the route to the destination is given to the scout by appending the route in the forager to its current route. If the current node is the destination of a forager then it is forwarded to the packing floor else it is directly forwarded to the MAC interface of the next hop node.

The Dance Floor*:*

The dance floor is the heart of the hive because it takes important routing decisions. Once a forager returns after its journey it recruits new foragers by dancing according to the quality of path that it traversed. However, the quality metric for each forager is different. As mentioned before, a lifetime forager evaluates the quality of its route based on the average remaining battery capacity of the nodes on its route. A lifetime forager might allow itself to be cloned many times (forager bees in Nature dance enthusiastically and consequently recruit more foragers) in two scenarios: one, the nodes on the route have enough remaining battery capacity (good route), two, if large number of packers are waiting for it even though its route might be having nodes with little battery capacity. In second case, it is sensible to send the packets through less good routes as well. On the other hand, if none of the packers are waiting then a forager with a very good route might not dance because its colleagues are doing a nice job in transporting the data packets.

This concept is directly borrowed from the behavior of scout/forager bees in Nature, and it helps in regulating the number of foragers for each route. The dance floor also sends a matching forager to the packing floor in response to a request from a packer. The foragers whose life time has expired are not considered in the matching function. If multiple foragers match the criteria then a forager is stochastically chosen among them. This helps in distributing the packets over multiple paths that serves two purposes: avoid congestion under high loads and battery of different nodes are depleted at an equal rate. A clone of the selected forager is sent to the packing floor and the original forager is stored in the dance floor after reducing its dance number. If the dance number is zero then the original forager is sent to the packing floor and its entry is deleted from the dance floor. Using the above-mentioned principle, young foragers, which represent latest routes and which are likely to remain valid in future, are favored over the older ones. If the last forager for a destination leaves a hive then the hive does not have a route to the destination. We believe that if a route to the destination exists then soon a forager would be returning toward the hive and if no forager comes within a certain time then the node has probably lost route to the destination node. This mechanism eliminates the need for explicitly monitoring the validity of the routes by using special hello packets and then informing other nodes through Route Error Messages (RERR). This result in transmitting less control packets, as a result, the algorithm has less energy expenditure.

CONCLUSION

An extensive and vigorous research has been done in the field of swarm intelligence applications on the distributed, infrastructure-less ad hoc network. A lot of research work had been done in the field of mobile ad-hoc network security features. None of the protocols invented till date could provide a decent trade-off between security and the performance issues of MANET, but the best service has been provided by PGP(Pretty Good Service). In an attempt to enhance the security of this wireless mobile medium, many researchers have proposed a lot of routing protocols. The protocols based on the intelligence secure routing techniques discussed so far have provided a vivid improvement over the conventional routing protocols in terms of security, battery life, scalability, maintainability, adaptability and so on. Moreover, in this connection we must mention the various shortcoming of a MANET out of which security is the most burning topic of discussion for the researchers. MANET is an insecure network due to many reasons: resource shortage, dynamic topology, mobile nodes, and wireless communication medium leads to the interference of external intruders inside the network. In this paper we have briefly introduced the nature-based routing protocols for MANET derived from the ant colonies and the bee hubs which can extensively enhance the performance of the network and also increases the security of MANET.

The agents in Ant Colony inspired routing algorithms communicate indirectly through the 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. From our study it was observed that the important advantage of BeeAdHoc is the distribution of the traffic to different routes proportional to their quality and capacity. This has the absence of many control packets compared to other algorithms and are limited to the scouts. The most important disadvantage of Bee Ad Hoc is the use of source routing. This also appears quite unnatural compared to real bees, because they use vector guidance. The disadvantage in computer networks comes from the control overload per packet and the limitation of the maximal route length. It uses the higher memory for storing every forager. It can be concluded that Swarm Intelligence based approach offers to be a powerful means to solve routing problems in Mobile Ad hoc Networks.

References

[1]E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence –From Natural to Artificial Systems. New York: Oxford University Press, 1991

[2]G. Serugendo, N. Foukia, S. Hassas, A. Karageorgos, S. K. Mostefaoui, O. F. Rana, M. Ulieru, P. Valckenaers, C. Van Aart, "Self-organisation: Paradigms and applications", Engineering Self-Organizing Systems, Vol. 2977, 2004

[3]Dorigo, M., & Caro, G.D 1999. Ant Algorithms for Discrete Optimization. Artificial Life.

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

[5]Dr.Arvinder Kaur, Shivangi Goyal.A Survey on the Applications of Bee Colony Optimization Techniques in International Journal on Computer Science &Engineering 2011.

[6]G.A. Di Caro, F. Ducatelle, and L.M. Gambardella. Theory and practice of Ant Colony Optimization for routing in dynamic telecommunications networks. In N. Sala and F. Orsucci, editors, Reflecting interfaces: the complex co evolution of information technology ecosystems. Idea Group, Hershey, PA, USA, 2008.

[7]Ahmed. A. A. Radwan, Tarek. M. Mahmoud, Essam. H. Hussein AntNet-RSLR: A Proposed Ant Routing Protocol for MANETs 2011.

[8]Ehsan Khosrowshahi-Asl, Majid Noorhosseini And Atieh Saberi Pirouz A Dynamic Ant Colony Based Routing Algorithm for Mobile Ad-hoc Networks Journal Of Information Science And Engineering, 1581-1596 2011.

[9]S. S. Dhillon, X. Arbona, and P. V. Mieghem, "Ant routing in mobile ad hoc networks," in Proceedings of International Conference on Networking and Services, 2007, pp. 67.

[10]E. KhosrowshahiAsl, M. Damanafshan, M. Abbaspour, M. Noorhosseini, and K.Shekoufandeh, "EMP-DSR: An enhanced multi-path dynamic source routing algorithm for MANETs based on ant colony optimization," in Proceedings of the 3rd Asia International Conference on Modeling and Simulation, 2009,pp. 692-697.

[11]R.Rameshkumar, Dr. A.Damodaram, ODASARA: A Novel on Demand Ant Based Security Alert Routing Algorithm for MANET in Grid Environment, IJCSNS International Journal of Computer Science and Network Security, April 2010.

[12]Jianping Wang, Eseosa Osagie, Parimala Thulasiraman, Ruppa K. Thulasiram HOPNET: A hybrid ant colony optimization routing algorithm HongKong,pp.1-20.