

A Multipath Lifetime-Prolonging Routing Algorithm for Wireless Ad Hoc Networks

Mohamed Amine RIAHLA

University of Boumerdès
LIMOSE Laboratory
South Campus, 35000 Boumerdès, ALGERIA

Karim TAMINE

University of Limoges
XLIM Laboratory UMR CNRS 6172
83, rue d'Isle, 87000 Limoges, France

Abstract—Dynamic networks can be tremendously challenging when deploying distributed applications on autonomous machines. Further, implementing services like routing and security for such networks is generally difficult and problematic. Consequently, multi-agent systems are well suited for designing distributed systems where several autonomous agents interact or work together to perform a set of tasks or satisfy a set of goals, hence moving the problem of analyzing from a global level to a local level therefore reducing the design complexity. In our previous paper, we presented a Multi Agent system model that has been adapted to develop a routing protocol for ad hoc networks. Wireless ad hoc networks are infrastructureless networks that comprise wireless mobile nodes which are able to communicate with each other outside the wireless transmission range. Due to frequent network topology changes, the limited energy and underlying bandwidth, routing becomes a challenging task. In this paper, we present a new version of routing algorithm devoted for mobile ad hoc networks. Our new algorithm helps controlling the network congestion and increasing the network lifetime by effectively managing nodes energy and link cost. The performance of our new version is validated through simulation. The Simulation results show the effectiveness and efficiency of our new algorithm compared to state-of-the-art solutions in terms of various performance metrics.

Keywords—*Mobile Multi Agent System; Ad hoc Network Lifetime; Ant Routing Protocol; Distributed Algorithm; Network Congestion*

I. INTRODUCTION

Dynamic networks can be tremendously challenging when deploying distributed applications on autonomous machines. But, these networks can meet problems when implementing services such as routing and security in general.

We have chosen ad hoc networks as a case study. For this, we focused on multi-agent approach which has a particular interest in the distributed problems in general, and for which it is difficult to prevent all situations. The multi-agent systems [1–3] are well suited for the design of distributed systems. Several autonomous agents interact or work together to perform a set of tasks or satisfy a set of goals, and moving the problem of analyzing from a global level to a local level and reducing the design complexity. In this paper, we present a generic mobile Multi Agent system that has been adapted to develop a new routing protocol version for ad hoc networks (The first version is published in [4]). The first goal is improving the previous version in terms of packet loss and end to end delay by adding new features to it. The second goal is increasing the network

lifetime by effectively managing the nodes energy and controlling the network congestion during route discovery process and data transmission.

In multi-hop wireless ad hoc networks (MANETs) [5], mobile nodes cooperate with each other to form a network without a fixed infrastructure such as access point or base stations. The network nodes perform routing discovery and routing maintenance in a self-organized manner. The routing is particularly a challenging task in MANETs because of the frequent changes in the network topology triggered by nodes displacements, the establishment of new nodes connections and nodes disconnections and the unstable routes discovery process. Practically speaking, efficient routes may quickly become inefficient or even unusable ones. To tackle this problem and ensuring a suitable routing through reliable algorithms, one important way is to update routing information more regularly than in wired networks. However, this requires more routing control packets which is specifically an issue in MANETs since the bandwidth of the wireless medium is very limited and the medium is shared.

Beyond the routing overhead problem, our proposed protocol also attempts to solve several problems such as: packet delivery ratio, end-to-end delay, network lifetime and energy consumption. For this purpose, we propose a hybrid method that entails both proactive and reactive processes. The routes are established and periodically maintained with a constant number of mobile agents. The Agent is periodically created by each node, thus the number of agents in the network can continually be controlled. However, when a connection is planned to be established by a node with another one in the absence of a route in its routing table, the latter makes a route request by setting a local variable available for Agents passing through it.

Our model is based on the ant behavior. A number of ant-based routing algorithms exist either in wired or in wireless [6–13] networks (Ad hoc and sensor networks). They are based on the pheromone trail laying-following behavior of real ants and the related framework of ant colony optimization ACO [14]. In all of these approaches, a source node broadcasts an Agent whenever it plans to build a route to a fixed destination. One of Agent roles is to deposit amounts of pheromone in order to mark optimal paths between a couple of nodes namely source and destination nodes. Unlike these methods, we do not exploit a broadcasting technique that exponentially increases the routing overhead, rather we introduce a new idea via an ant-based algorithm that consists in setting a local route request

whenever a node plans to send a data packet. It is, therefore, the Agent's role moving within the network during the proactive phase to disseminate this information and provide routes towards the requested destination. Note that our protocol doesn't necessarily establish the best route, since the agents are not broadcasted. However, the agents attempt to get as close as possible to the best route.

The remainder of this paper is organized as follows. In section 2, we present some definitions on self-organized multi-agent systems and their analogy with dynamic networks. In section 3, we present the principle of ad hoc networks and then describe the routing problem in this class of networks; afterwards, we give an overview of some representative related works. In section 4, we present the general architecture and mode of operation of a multi-agent system adapted to a generic case of dynamic systems. We detail the instantiations of the generic model: the previous version of our routing protocol for ad networks. In section 5, we present a new version of our routing approach. Before concluding in section 7, we discuss in section 6 results and tests of the new version of routing protocol using the NS2 simulator.

II. MULTI-AGENT SYSTEMS AND DYNAMIC NETWORKS: ANALOGIES AND ISSUES

Over the last years, a new topic of research has emerged: dynamic networks (also called autonomous systems). A dynamic and distributed network consists of a set of auto-configurable nodes that are constantly changing (the number of nodes and links change over time). The topology change is also one of the properties of these networks, because the network nodes can join and/or leave the network spontaneously. The main advantage of this type of networks is the fast and inexpensive deployment and installation.

A multi-agent system (MAS) is a set of agents operating in a common environment. This set of agents, not necessarily smart, is a complex system that shows a collective intelligence. This collective intelligence comes from the emergence of a global behavior of all agents. An example of such collective action concerns the behavior of a colony of ants that act like an entities (ants) with no cognitive capacity, but achieving a high degree of organization and adaptation quote.

An agent is a software entity (program) reactive, proactive and with social skills, able to act autonomously in its environment. Responsiveness refers to maintaining a constant link with the environment when a change occurs. The proactivity means that the system allows agents to generate and satisfy its goals. Social skills indicate that the system allows the agent to interact or cooperate with its environment and / or with other agents.

It is, therefore, clear that there is an analogy between dynamic networks and multi-agent systems. In fact, each node of a dynamic network is autonomous because 1) it is not controllable by any other node on the network, 2) reactive because it can act as a server for other nodes, 3) proactive in the case of its client node status, and, lastly, 4) social because it communicates and cooperates with other nodes in the network.

Accordingly, an inherent issue in the management of dynamic networks is: the definition of a data routing protocol in the case of ad hoc networks.

III. ROUTING ISSUES IN AD HOC NETWORK AND RELATED WORK

In multi-hop "wireless" ad hoc networks (MANETs), "mobile nodes" cooperate with each other to form a network "without a fixed infrastructure" such as access point or base stations, the network nodes perform a routing discovery and a routing maintenance in a self-organized way. In other word, Wireless ad hoc networks are infrastructureless networks that comprise wireless mobile nodes that are able to communicate with each other outside the wireless transmission range. Due to frequent network topology changes and the limited underlying bandwidth, routing becomes a challenging task.

Several types of routing protocols have been specifically designed for ad hoc networks and classified into two main categories: reactive and proactive protocols. In reactive routing protocols such as AODV [15] (Ad Hoc On demand Distance Vector) and DSR [16] (Dynamic Source Routing), the routes are only discovered when required in order to save node and network resources, while in proactive routing protocols such as OLSR [17] (Optimized Link State Routing Protocol) and DSDV [18] (Destination Sequenced Distance Vector) the routes are established in advance, avoiding consequently the delays that occur during the discovery of new routes.

The problem raised by proactive protocols consists in the routing overhead, especially when there are frequent topology changes. This is highly inefficient when updating routes that rarely carry traffic. A reactive protocol is, in contrast, much more appropriate for such situations, since it generates lower overhead in terms of used bandwidth.

There is another kind of protocol that combines both reactive and proactive approaches called hybrid routing protocol. In our work, we focus on a particular class of hybrid routing protocols based on an optimization technique known as ant colony optimization (ACO) which are inspired from the foraging general behavior of some ant species. The ant underlying behavior can be summarized as follows: ants deposit pheromone on the ground in order to mark some favorable paths that should be followed by other members of the colony, for instance, ants walking to and from a food source deposit on the ground a substance called pheromone. Other ants perceive the presence of pheromone and tend to follow paths where the pheromone concentration is higher. Through this mechanism, ants are able to transport food to their nest in a significant effective way.

Several properties belonging to ant-based routing algorithms are strongly appropriate to address the problems inherent in MANETs: they are highly adaptive to network changes, robust to agent failures, and provide multipath routing [19]. However, since they mainly rely on repeated path sampling, a significant overhead can be induced within native routing algorithms. Several ant-based routing algorithms for MANET have been proposed in state of the art previous work.

However, in order to limit the overhead caused by the ants, these algorithms considerably lose the inherent proactive sampling and exploratory properties belonging to the ants behavior adopted in the original ant-based algorithms.

The classic ant methods inspired from Ant-Colony-Optimization algorithm like ARA [20] and AntHocNet [21] use during the reactive phase, a broadcasting technique that exponentially increases the routing overhead and energy consumption: 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 broadcasts out ant-like agents, called forward ants (FANTs), to look for paths to d . At their arrival in d , they become backward ants (BANTs) which trace back the path and update routing tables.

Some more recent algorithms attempt to limit the number of broadcasts like SARA for MANETs [22] and ESARA for Wireless Sensor Networks [23]; they adopt a controlled neighbor broadcast mechanism in route discovery to avoid flooding the network with FANTs. Each node broadcasts the FANT to its direct neighbors, and only one of them further rebroadcasts the FANT to its neighbors. However, there always still broadcasts messages.

Our protocol is based on ant general behavior, but differs from the other algorithms. We detail below the route discovery process of the previous version of our protocol in section 4 and the new version is section 5.

IV. A MULTI AGENT SYSTEM FOR GENERIC AND DYNAMIC NETWORKS

A. Abbreviations and Acronyms

The Generic multi-agent system that we propose to manage services in dynamic networks is composed of two types of agent communities (see figure 1). Both agent communities interact through different types of communication which we will present later on.

The first community of agents is SMA_Node, it will manage the functions of mobile nodes in the dynamic network. Each agent will be called node, then represent a network node whose features will be explained below. Each node agent will be able to provide a set of network data packets that are called packet. This set of data packets will be the second community of agents that will call SMA_Packet. Packet agents will travel along the dynamic network in a completely random way according to some metric contained in the node agents and packet agents at the interaction between these two types of agents. Moving a packet agent to a node agent network is defined by a behavior of packet agent which will be defined later.

Each node agent SMA_Node from community will be defined generically by the following behaviours:

- Detect_neighbors () return Liste_node: This function allows the node agent to be able to see all the node agents that are in its "scope" and with which it can directly communicate.

- Connect (v: node): This function allows the node agent to start a connection with its neighbor node V in order to establish a communication with it.

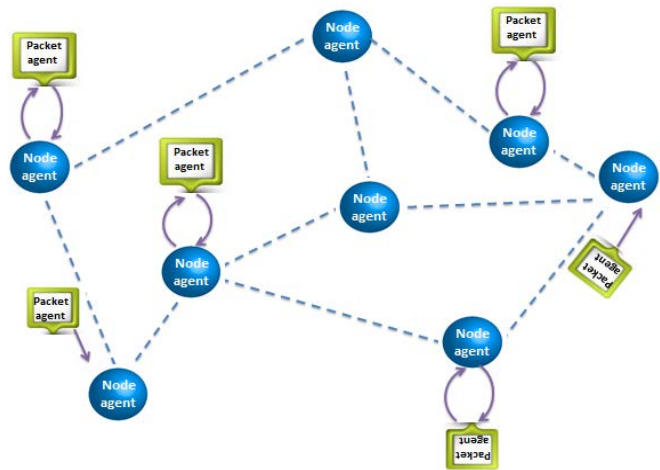


Fig. 1. General scheme of our model

- Disconnect (v: node): This function allows the node agent to delete a connection with a neighbor node. This function also has the effect of removing the node agent v from its list of neighbor nodes.
- Connect (): This function allows a node agent to join a network of node agents.
- Disconnect (): This function disconnects the agent from network.
- Generate () return packet: This function allows the agent to create and distribute a new packet agent in the network.
- Read_info (p: packet, info: information): This feature allows the agent to read and get information broadcasted and carried over the network by the packet agent p while moving from node to another.
- Write_info (p: packet, info: information): This feature allows the agent to write information into the packet agent p in order to be distributed in the network.
- Move (p: packet, n: node): This function allows the node agent to send the packet agent p to another node neighbor agent (node agent n). This the feature that allows the distribution of packet agents in the network.

Each agent from community SMA_packet will be generically defined by the following behavior:

- Create (): This feature allows the packet agent to be created by the creator node agent;
- Delete (): This feature allows the packet agent to be destroyed by its creator node agent;
- Transfert (n: node, info: information): This feature enables the packet agent to transfer the information carried to the current node where it is located. This

feature will be used by the current node agent to perform a read of information in the network;

- Update_packet (n: node, info: information): This function allows the packet agent to update its data with information provided by the current node agent. This feature will be used by the current node agent in order to broadcast information in the network;
- Choice_displacement (n: node) return node: This feature allows us to know randomly according to some metrics contained in the node agent n and in the current packet agent the next node where the current packet agent will move to;
- Move (init: node, final: node): This function allows the packet agent moving from the node agent « init » to the node agent « final ». The move action will be achieved through the move () function of init node agent.

B. Application of the model in the case of routing in ad hoc networks: previous version of AgentRouting protocol

The main idea of the protocol AgentRouting is to build a multi-agent based system where each node provides several kinds of agents (packet agents). Regarding the purposes of the routing task, we design two main types of packet agents. A first mobile agent, called Ant-Agent, is responsible of establishing routes. A second mobile agent, called Rectifier-Ant, is issued by a node whenever a change in the network topology is detected (the features of this agent are detailed in the previous version of the protocol). Our protocol is a complete multi-agent based system, where an agent works independently from the others. This fits spontaneous networks such as wireless ad hoc networks very well, because of the very high mobility and self-organization properties of this type of networks. Our protocol inherits from the advantages of this kind of model: autonomous work, distributed intelligence and robustness. Furthermore, the use of mobile agents allows to easily extend the functionalities of the protocol by simply adding other agents or by assigning other functionalities to the existing ones.

AgentRouting protocol is based on a hybrid algorithm. In the proactive phase, the protocol uses mobile Agent as follows: each node (Origin-Node) periodically creates one Ant-Agent (see function Generate () in section IV.A) that moves across the network from one node to another (see function Move (p: packet, n: node) in section IV.A) and builds paths from the current node to its Origin-Node and paths from the current node to the last visited one when the Ant-Agent returns back to its Origin-Node (the Ant-Agent has two phases: a Go-phase when it is sent by its Origin-Node and a Back-Phase when it returns to its Origin-Node).

When a data session is started between a source node s and a destination node d, s checks whether it has up-to-date routing information about d. If not, the node s makes a local route request. In our case, the route request is not broadcasted to every node as it is the case in a classic ant routing protocol [3,7,8], but it is stored on node s. The broadcast task is assigned to the Agent (packet agents created during the proactive phase and moving within the network) that have the responsibility to "intelligently" disseminate the route request throughout the network.

These Agents gather information about the quality of paths they followed, and at their arrival to node s (which contains the route request to destination d), they return back to their source node by tracing back the path and updating the routing tables (see function Read_info (p: packet, info: information) in section IV.A).

Before detailing the previous version of our routing protocol, let's consider some of these assumptions. Each node must be able to broadcast hello messages to its one hop neighbors. We also consider that the links between the nodes of the networks are always bi-directional. Moreover, as the protocol must operate in an ad hoc environment where nodes are highly mobile, the routing protocol must take into account this constraint and be responsive to these frequent changes of the topology. Therefore, our previous version of routing protocol is consisted of three modules: neighborhood discovery module (see function Detect_neighbors () in section IV.A), path discovery module and managing broken links module. In this paper, we focus our contribution on the path discovery module and data transmission.

1) Route discovery in the previous version of our protocol

a) Proactive phase: AgentRouting Protocol is a routing protocol for wireless ad hoc networks based on mobile Agent. In order to establish routes between nodes, our protocol uses mobile agents which are periodically created by each network node (see function Generate () in section IV.A). An Ant-Agent belongs to a single node called Origin-Node. An Ant-Agent moves across the network from one node to another (see function Move (init: node, final: node) in section IV.A). When it reaches a node, the Agent establishes and builds in its memory (see function Update_packet (n: node, info: information) in section IV.A) and in the routing table of the node, a path between this node and its origin node (Figure 2). Thereafter, the Agent chooses a next hop among its neighbors in a stochastic manner and proportionally to the amount of pheromone deposited by the other Agent during their Back-Phase (see function Choice_displacement (n: node) in section IV.A).

To avoid routing loops, we assign a unique identifier <node_ID, Ant_ID> to agents (see function Write_info (p: packet, info: information) in section IV.A), that is incremented at each creation of a new Ant-Agent. If a node receives several times the same Ant-Agent, it accepts the information given by the first one and ignores the others. In order to monitor its Ant-Agent, a node assigns a configurable Time To Live (TTL) to agents (see function Write_info (p: packet, info: information) in section IV.A) whose value is proportional to the network dimension and decremented on each hop. This means that an Ant-Agent will have two phases during its life cycle: Go-Phase (Figure 2) where the agent builds a path from the current node to its Origin-Node and a Back-Phase (Figure 3) where the agent follows a reverse path from the one followed during its first phase (the Ant-Agent saves in its memory a reverse path during its Go-Phase). At each visited node during the Back-Phase, the Ant-Agent builds and stores in the routing table of this node a path from this current node to the last node visited during the first phase (when TTL=0). This is the first step in the routing discovery process which is proactive.

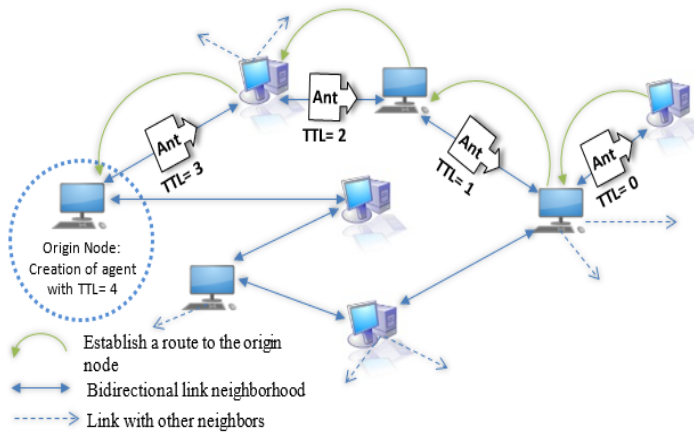


Fig. 2. Go phase

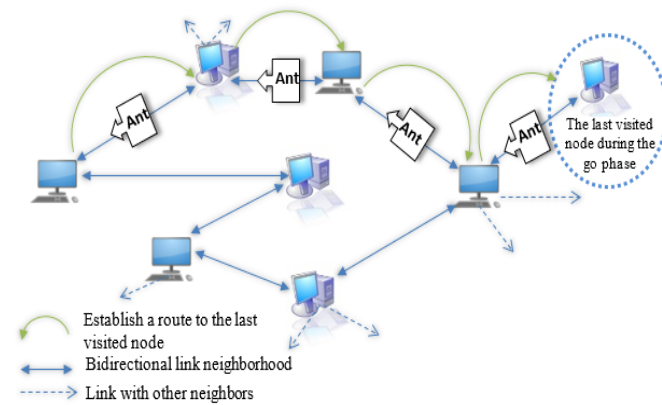


Fig. 3. Back-Phase

b) *Reactive phase:* During the proactive phase, a large number of paths are built. However, when a node d plans to send or forward data packets to an unknown destination node t (see figure 4), it triggers a route request process where the route request is locally saved. When an Ant-Agent, during its Go-Phase, visits a node which has made a route request (a node can have several route requests), the Ant-Agent switches to its Back-phase and deposits an amount of pheromone on each node of the reverse path towards its origin node. This mechanism is used to mark the paths towards the node d and hence informing the other nodes (and Agents) about this route request. The amount of pheromone deposited by the Ant-Agent is defined by the following equation (1):

$$\psi_{it} = \psi_{i(t-1)} + \Delta q \quad (1)$$

Where ψ_{it} is the pheromone level in the node i at time t and Δq is a positive constant.

During its Go-Phase, a Ant-Agent chooses the next hop in a stochastic manner and proportionally to the amount of pheromones; this process increases the likelihood to select a path towards a claimant node d without penalizing the other paths. Choosing the next node randomly and proportionally to the amount of pheromones allows us to increase the number of agents towards the claimant node d . On the one hand, this approach increases the chances of having an Ant-Agent issued from the destination node t (i.e., the Agent's source node is the

destination t). On the other hand, it allows to quickly reaching nodes (like node k in figure 4) that have already established paths towards this destination t .

c) *Stochastic data routing:* In our protocol, the nodes stochastically forward the Agent. When a node has several neighbors concerned by nodes that made a route request, it randomly selects one of them with the probability p (see function Choice_displacement (n : node) in section IV.A).

Each neighbor can have a quantity of pheromone related to nodes which made a route request. Let's consider $N(i)$ the set of i 's neighbors and ψ_{jt} the amount of pheromone associated to a neighbor j stored in the routing table of the node i at time t .

The expression that gives the probability p to select a next hop j from node i is defined in equation (2).

$$P = \psi_{jt} / \sum \psi_{kt} \quad (2)$$

In order to consider route requests in an equitable manner leading to a self-organizing system and a better management of frequent changes in the network topology, we propose to set up an evaporation process. The latter allows to no longer take into account the old route requests already satisfied. At each time interval, the amount of pheromone corresponding to each route request is decreased as defined in equation (3):

$$\psi_{it} = (1-\alpha) * \psi_{i(t-1)} \quad (3)$$

Where ψ_{it} is the amount of pheromone related to a claimant node d , stored in the node i at time t and α is a real ($0 < \alpha < 1$) (we choose $\alpha = 0.1$ for our simulations).

V. NEW FEATURES FOR OUR PROTOCOL

The first version of our routing protocol is published in [4]. Additionally considering our routing protocol as a completely mobile multi agent system, we implemented other functionalities in order to improve the network lifetime and the routes discovery.

Our new version strives to avoid low energy routes during route discovery phase and data transmission, so that the network lifetime can be prolonged. Thereby, we added other features in order to improve the previous version in terms of packet loss and end-to-end delay.

A. Adapting pheromone quantity to the link cost and the energy

In the previous version of our protocol, when an Ant-Agent visits a node d (see figure 4) which has made a route request, this agent switches to its Back-phase and deposits an amount of pheromone on each node of the reverse path to its origin node, the goal is to mark the paths towards the node d , and therefore inform the other nodes and Agents about this route request. The amount of pheromone deposited by the Ant-Agent in the previous version of our protocol is constant (Δq in equation 1). In the new version, we adapt the quantity of pheromone deposited by ant agents according to the limited energy of nodes and the link cost.

Each node continuously updates its energy and exchanges 'hello' messages with neighbors to establish and maintain a neighbor list containing updated information such as ID and energy level [24].

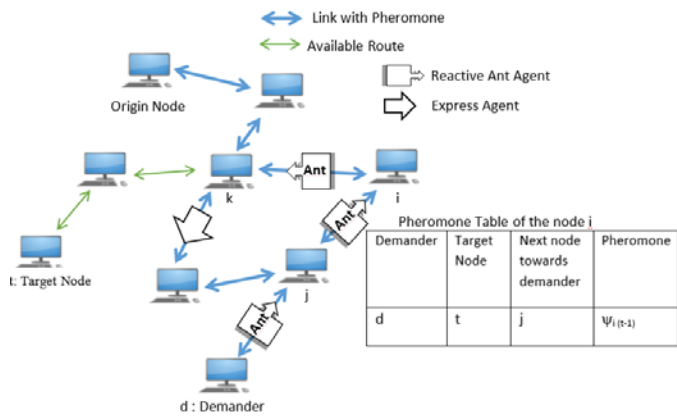


Fig. 4. Reactive phase

In figure 4, Ant-Agent “Ant” is going towards its origin node after visiting a demander node (node d); it deposits an amount of pheromone on each node of the path. When this Ant moves from node j to node i, it deposits a pheromone quantity Δq on the node i. The new value of Δq is:

$$\Delta q = C(i,j,d) * E_j + q, C(i,j,d) = 1/(1+n) \quad (4)$$

Where $C(i,j,d)$ represents the link cost between node i to its neighbor node j towards destination d (node which has made a route request). This cost is based on the number of times (n) the link was previously used. E_j is the energy of node j. q is a positive constant (we have chosen $q=0.1$ for our simulations).

The Ant-Agent continues in a similar fashion until it reaches the origin node.

We have previously seen that an Ant-Agent during its Go-Phase chooses the next hop in a stochastic manner and proportionally to the amount of pheromones (see equation 2). With the new version of protocol, the probability to choose a neighbor with more energy and lower link cost is higher. The advantages of this new feature are:

Network lifetime[25-29] is one of the most prominent barriers in deploying ad hoc networks for large-scale applications. Our new algorithm helps to enhance the lifetime of the network by taking care of nodes energy. This feature allows to increase network lifetime and to control the network congestion through load balancing.

The link cost parameter helps the Ant-Agents during their Go-Phase to use different paths towards the claimant node; this approach increases the probability of having an Ant-Agent issued from the destination node, or to quickly reaching nodes that have already established paths towards this destination.

B. A new type of agent: Express agents

This simple new feature must use another type of Agents called Express-Agent. When an Ant-Agent switches to its Back-phase after visiting a claimant node, it can pass through a node which has requested the route (like node k in figure 4). This node generates a special Agent called Express-Agent which will be immediately sent to the claimant node. This approach allows improving the end-to-end delay and the packet delivery fraction, because the data packets are sending immediately towards their destinations before the topology

changes. An Express-Agent use the same method detailed in section V.A when it goes towards the claimant node.

The rest of the route discovery procedure is similar to previous version of our protocol

C. A new data routing procedure

Once the routes are discovered, they are selected on the basis of number of hops, energy of the node, and link cost. When multiple routes are discovered using a reactive and proactive phases then the route is selected on the basis of probability procedure. The route selection in our new version of routing protocol differs from the classic ant methods inspired from Ant-Colony-Optimization algorithm. Our protocol introduces Utility of route metric in route selection criteria (Utility (n, n_j, d)).

$$Utility(n, n_j, d) = (1-k) * \frac{1}{1+NH} + k * Cost(n, n_j, d) \quad (5)$$

Where NH is the number of hops between node n and destination d, n_j is the neighbor node of n. The parameter k is used for providing greater importance to the number of hops or the cost of route “Cost”. Cost depends on the cost link C (n, n_j, d) (see equation 4) and the energy e of node n_j. The metric Cost can be calculated as

$$Cost(n, n_j, d) = \frac{C(n, n_j, d) * e}{\sum_{k=0}^{k=M} C(n, n_k, d) * e} \quad (6)$$

M is the number of adjacencies of node n.

The probability to pick a neighbor node n_j for routing data packets towards destination d is given by (7), which is based on Utility of route (energy, link cost, and number of hops):

$$P(n, n_j, d) = \frac{Utility(n, n_j, d)}{\sum_{k=0}^{k=M} Utility(n, n_k, d)} \quad (7)$$

This new algorithm strives to avoid low energy routes and optimizes the routing process through selection of the shortest path (number of hops) with more energy and lower link cost.

VI. SIMULATIONS RESULTS AND DISCUSSION

For performance evaluation of our proposed algorithm with existing protocols: ARA and AntHocNet (a classic ant based routing protocol), SARA (a more recent ant based routing protocol) and the old version of our protocol, we used NS 2 simulator by considering simulation parameters as shown in Table 1. The traffic is automatically generated by a Tcl script which is developed for this purpose. This traffic is randomly generated (the communications are established by randomly choosing pairs of nodes).

The performance metrics used to evaluate the performance of each algorithm are the following:

- The packet loss: this metric measures the number of packets which are not delivered to their destinations;
- The end-to-end delay: this metric represents the average delay between the packet sending time and its reception time;
- The total size of control messages generated by a protocol ;

- Total energy consumed;
- The network lifetime.

TABLE I. SIMULATION PARAMETERS

Parameter	Value
Simulation time	500s
Maximum node speed	30m/s
Sending frequency of Ant-Agent	0.8s
The evaporation frequency	0.5s
The evaporation rate « α »	0.1
The updating value of pheromone « q »	0.1
Packet size	512 bytes
Number of nodes	from 60 to 100 nodes
Number of data packets	from 700 to 1000 packets
MAC Type	IEEE 802.11
Simulation area	600m*600m
Protocol	UDP
Mobility model	Random way point
Receiving range	100m
Propagation model	Two ray ground
Initial energy of node	5J

We have calculated these metrics by varying, during the simulation time:

- The number of nodes : the number of data packets is fixed at 1000 and the maximum speed of the nodes is 5 m/s;
- The number of data packets : the number of nodes is fixed at 100 and the maximum speed of the nodes is 5 m/s;
- The nodes speeds: the number of data packets is fixed at 1000 and the number of nodes is fixed at 100.

In case of the performance metrics, total energy consumed and network lifetime, the number of data packets is fixed at 1000, the number of nodes is fixed at 100 and the maximum speed of the nodes is 5 m/s. The performance metrics are obtained by averaging over 30 simulations run from one source to one randomly selected destination. We assume that a node consumes 0.001J while receiving and 0.003J while transmitting.

A. The packet loss

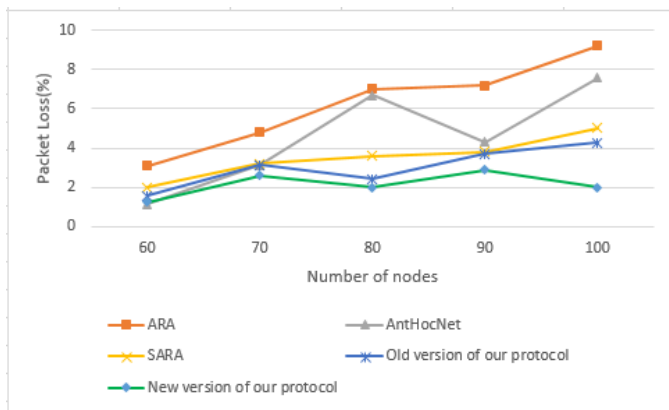


Fig. 5. Packet loss ratio according to the number of nodes

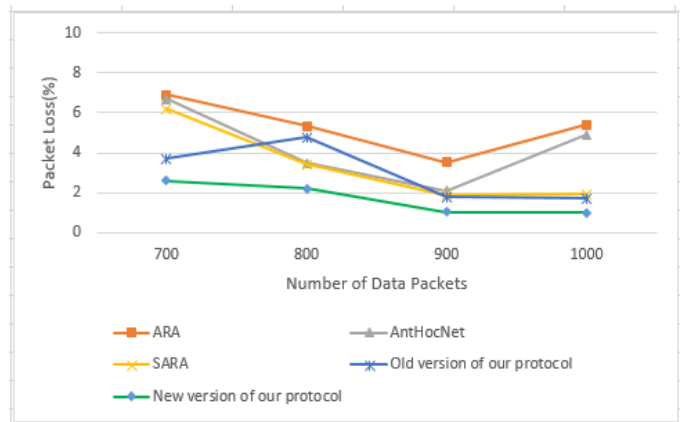


Fig. 6. Packet loss ratio according to the total number of data packets

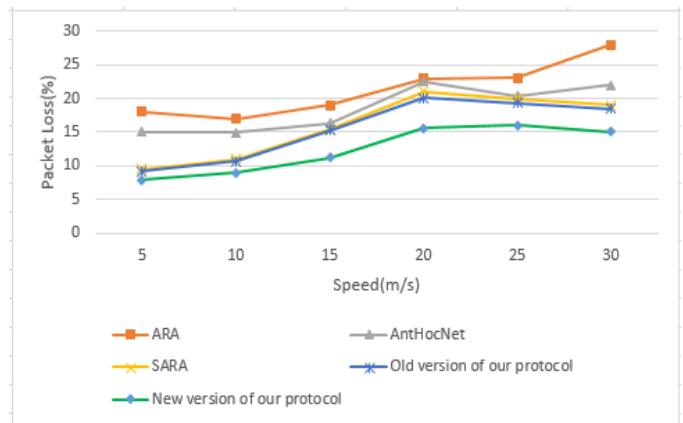


Fig. 7. Packet loss ratio/nodes speed

Figure 5 and 6 show the variation of % packet loss as function of the number of nodes and the total number of data packets. It is clearly shown in these figures that our scheme have less packet loss than the other protocols.

In ARA and AntHocNet, the number of lost packets increases very quickly in networks that have more than 90 nodes and in networks with high traffic (more than 900 data packets). We can explain these results by the fact that the ARA and AntHocNet protocols use a broadcast mechanism that generates a very important overhead and this fact is aggravated by the proliferation of collisions, which overloads the system and generates a very large number of non-accomplished transmissions. In SARA protocol, there is less packets loss because it uses less broadcast control messages than ARA and AntHocNet (see section Routing issues in ad hoc network and related work). In the case of our protocol, these results can be explained by the fact that we use an efficient reactive route discovery procedure instead of using a broadcast mechanism.

Simulation results showed that the new version of our routing protocol performed better than the previous version in terms of packet loss; our new algorithm favors nodes with high energy level and avoids the critical nodes to participating in the route discovery and data packet transmission. This produces fewer broken links and dramatically reduces the packet losses.

The link cost parameter helps to control the network congestion through load balancing, so the number of buffer overflow in the queue of intermediate nodes is reduced (see section: Adapting pheromone quantity to the link cost and energy). As result, the claimant node can have different routes towards the destination node so the claimant can choose the best one in terms of node energy; link cost and number of hops (see equation 7). This concept also helps the claimant node to have alternative routes.

Figure 7, which represents the variation of % packet loss considering the variation of nodes speed. It shows that our protocol have less lost packets than ARA, AntHocNet and SARA protocols when we vary the nodes speed. The use of Express-Agents in the new version allows sending data packets immediately to the destination before topology changes, hence reducing the packet losses.

B. The end-to-end delay

Figures 8, 9 and 10 show the end-to-end delay in each protocol, considering respectively the number of nodes in the network, the total number of packets and the nodes speed. We can see that our protocol and the SARA protocol generate less important delays than those generated by the ARA and AntHocNet protocols.

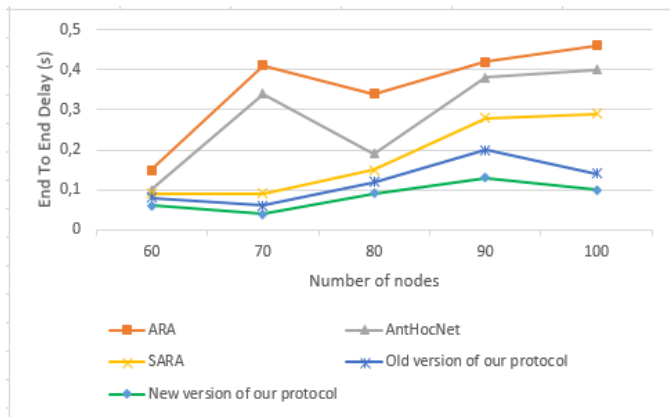


Fig. 8. End-to-end delay (s) according to the number of nodes in the network

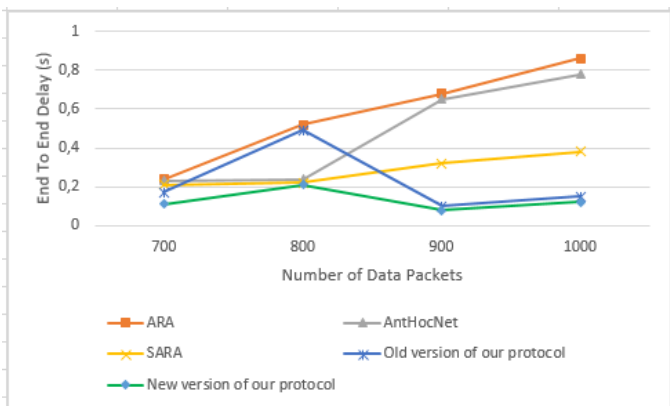


Fig. 9. End-to-end delay (s) according to the total number of data packets

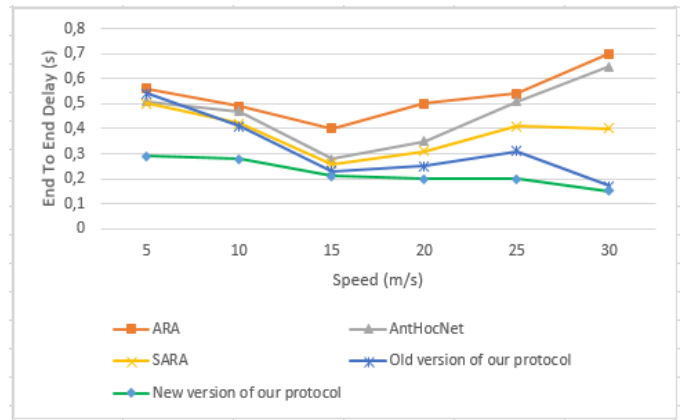


Fig. 10. End-to-end delay (s)/speed (m/s) of nodes

Our protocol and SARA protocol are both more efficient than ARA and AntHocNet protocols, especially in the case of a large number of nodes and in the case of high nodes speeds. Regarding our protocol, this may be due to five main reasons:

- Since our protocol is multipath, it allows each node to have several paths (towards the same destination) whenever it attempts to send a data packet, so it selects the best one (see equation 7 for the new version).
- The hybrid character of our route discovery scheme significantly reduces the transmission times. In fact, thanks to the proactive phase, it is not necessary to make a route request each time; the route may exist in the routing table;
- Unlike ARA, AntHocNet and SARA protocols, our protocol does not require, during the reactive phase, a broadcasting technique; in fact, the latter exponentially increases the routing overhead and thus overloads the network; consequently it delays the paths establishment and the data packets delivery process ;
- Once the size of the network increases, the new version of our protocol produces a better delay. The reason is that our protocol favors nodes having high energy level and avoids the critical nodes to participating in the route discovery and data packet transmission. This produces fewer broken links and greatly reduces the end-to-end delay ;
- The link cost parameter allows to improve the network congestion, so the number of data packets in the queue of intermediate nodes is reduced. This concept reduces the end to end delays.

C. The communication overhead

Control packets are the primary source of energy consumption. Less number of packets transmitted between intermediate nodes for route discovery procedure is an effective way to save energy.

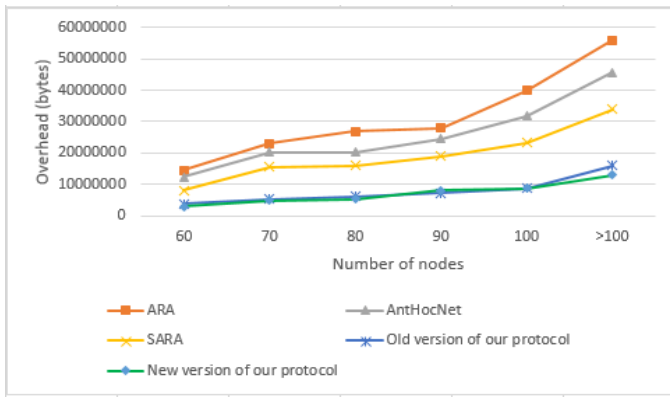


Fig. 11. The communication overhead (total size (bytes) of control packets) according to the number of nodes in the network

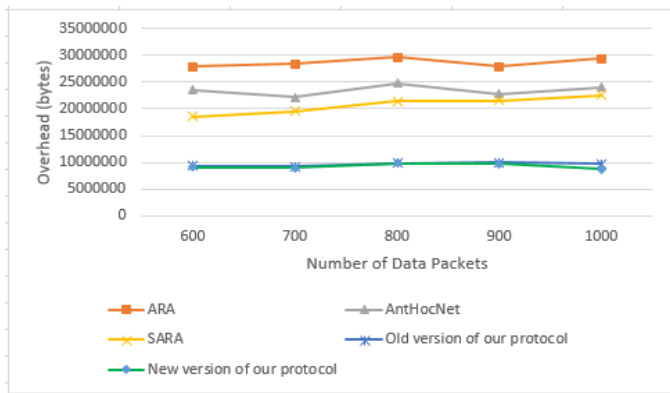


Fig. 12. The communication overhead (total size (bytes) of control packets) according to the total number of data packets

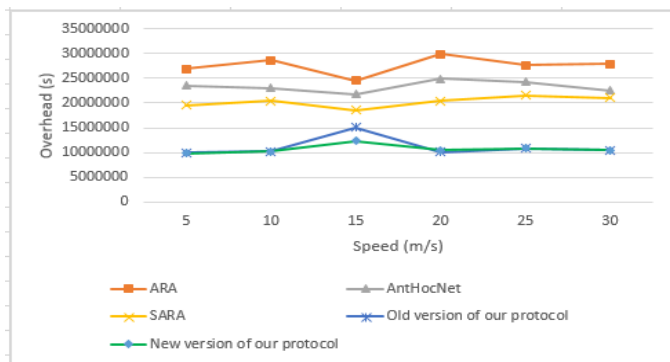


Fig. 13. The communication overhead (total size (bytes) of control packets) /speed (m/s) of nodes

The packets that consume much bandwidth and energy are those issued during the path discovery and maintenance phases. Therefore, we have measured the use of control messages by each studied protocol during these phases. Figures 11, 12 and 13 show the communication overhead in terms of, respectively, control packets size according to the number of nodes, the node activities and the speed of nodes.

We can see an important difference between our protocol and both ARA and AntHocNet protocols. The number of control packets increases slowly and linearly in our protocol, either by increasing the number of nodes or by increasing the network traffic, while it increases rapidly in the case of ARA

protocol. This could be explained by the fact that in the case of our protocol, the number of agents is managed and controlled by each node and this number is still proportional to the number of nodes in the network and to the transmission frequency of agents. Besides, we avoid the broadcast technique that generates a lot of overhead. Instead, we use a more accurate and a more intelligent technique which only makes use of the available agents in the network. In ARA protocol, the number of control packets depends on many factors, including the number of route requests that consume much bandwidth, since they require a significant number of broadcasts. Moreover, we can expect an important number of collisions due to the broadcast technique and network density. The collisions lead to an increasing number of retransmissions, which consequently induces the increasing of the total number of packets in the network.

Figures 11, 12 and 13 show also that our protocol generates less overhead than SARA protocol. This proves the effectiveness of our protocol, since the SARA protocol is known to be efficient in terms of communication overhead.

The total size of control messages in the previous and the new version of our protocol are approximately the same. Each node periodically creates an Agent, thus the number of agents in the network can continually be controlled, when a connection is planned to be established by a node with another one where there is a lack of a routes in its routing table, we only use the available agents in the network to satisfy this route request. We use the same process in the two versions of our protocol.

The adding of Express-agents has no influence on the communication overhead because the size of these agents is very small. Thus, we improved some parameters in the new version like the sending frequency of Ant-Agent (0.8s, see table 1, the previous value was 0.5s).

D. Total energy consumed

The control packets are the primary source of energy consumption. This is demonstrated in Figure 14 in a different way by figuring out the total energy consumption of the overall network. This figure justifies the facts that less number of control packets transmitted between nodes is an effective way to save energy. ARA, AntHocnet and SARA use more control packets, therefore more energy is consumed comparing to our protocol.

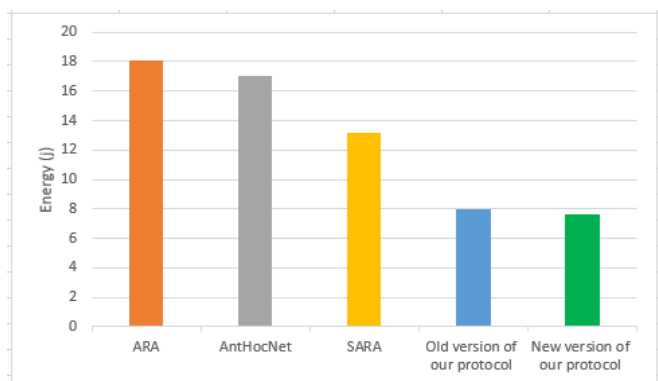


Fig. 14. Total energy consumed

E. The network lifetime

The network lifetime can be defined in three ways [30]: the time taken to exhaust the battery of the first network node, the time taken to exhaust the battery for N network nodes, and the time when the battery of the last network node is exhausted. We choose the second way.

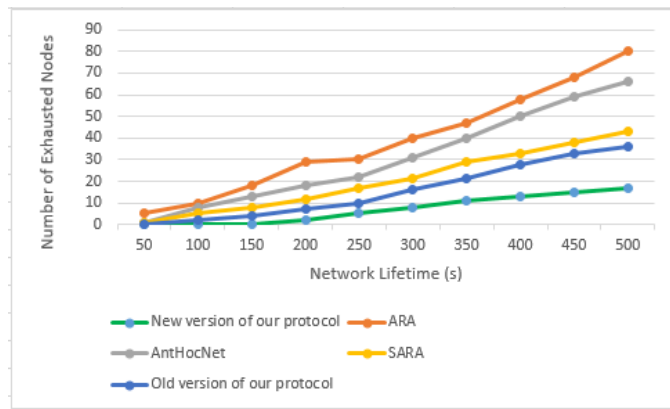


Fig. 15. Network lifetime versus number of exhausted nodes

The network lifetime metric is shown in figure 15 with the number of network nodes equal to 100. The network lifetime of our protocol is longer than those of ARA, AntHocNet and SARA. As result, the new version of our protocol exhausts fewer nodes compared to the previous version, which noticeably increases the network lifetime.

As demonstrated in Figure 15 and seen previously in the new protocol version description, our new algorithm balances the energy amongst all the nodes and expands the individual node lifetime, hence the entire network lifetime. Our protocol also avoids the broadcasting technique which consumes a lot of energy and increases the number of exhausted node.

VII. CONCLUSION AND FUTURE WORK

We have presented in this paper an improved version of our ant based routing protocol for Ad hoc networks, that aims to reduce the packet losses, the transmission delays and to maximise the network lifetime by using a new route discovery technique and data transmission. Considering the simulation results obtained using the new proposed algorithm, we can argue that the new version of our protocol significantly reduces the packet losses and transmission delays. Regarding routing overhead, our algorithm out performs SARA protocol which is assumed to generate fewer messages than ARA and AntHocNet. Our protocol doesn't require any broadcasting mechanism that leads to increase the number of control messages. This underlying concept extends the network lifetime and improves energy consumption when compared with other solutions known in the literature regarding network lifetime and network congestion. Also, our algorithm allows achieving better results than ARA, AntHocNet and SARA. It is capable to avoid nodes with low energy and high cost link in the multipath construction. In other words, paths with higher energy are identified and selected for transmission.

In our future work, we plan to use our generic multi-agent system within another dynamic environment such as Wireless

Sensors Networks (WSN) and P2P networks. In WSN, a new Simple Ant Routing Protocol for Wireless Sensor Networks (SARPWSN) inspired from ant colony optimization will be designed using our multi-agent system. In P2P networks, we will describe the design and experimentation of a new Anonymous and confidential P2P system, the protocol works on the principle of social networks with nodes identified by virtual addresses and uses a set of control messages that work like a Multi Agents Mobile System.

REFERENCES

- [1] J. Dean, J. Macker, and W. Chao, "A Study of Multiagent System Operation within Dynamic AD HOC Networks," In IEEE Military Communications Conference, pp. 1-7, 2008.
- [2] Mitrović, D.; Budimac, Z.; Ivanović, M.; Vidaković, M., "Improving fault-tolerance of distributed multi-agent systems with mobile network-management agents," in Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on , vol., no., pp.217-222, 18-20 Oct. 2010.
- [3] Shenfang Yuan, Dong Liang, Lei Qiu, and Menglong Liu, "Mobile Multi-Agent Evaluation Method for Wireless Sensor Networks-Based Large-Scale Structural Health Monitoring," International Journal of Distributed Sensor Networks, vol. 2012.
- [4] Boussad Ait Salem, Mohamed Amine Riahlia, Karim Tamine, "A hybrid multiagent routing approach for wireless ad hoc networks," Wireless Networks 18(7): 837-845, 2012.
- [5] J. Garcia-Macias and J. Gomez, "MANET versus WSN," in Sensor Networks and Configuration, N. P. Mahalik, Ed., pp. 369-388, Springer, Berlin, Germany, 2007.
- [6] Caro, G. D., Ducatelle, F., & Gambardella, L. M., "An adaptive nature-inspired algorithm for routing in mobile ad hoc networks," European Transactions on Telecommunications, 16, 443-455, 2005.
- [7] Correia, F., Vazão, T., & Lobo, V. J., "Models for Pheromone Evaluation in Ant Systems for Mobile Ad-hoc Networks," In Proceedings of the 2009 First International Conference on Emerging Network Intelligence (EMERGING '09), IEEE Computer Society, Washington, DC, pp. 85-90, 2009.
- [8] Laxmi, V., Jain, L., & Gaur, M. S., "Ant colony optimisation based routing on NS-2," International Conference on Wireless Communication and Sensor Networks (WCSN), India, 2006.
- [9] Y. Kambayashi, "A review of routing protocols based on ant-like mobile agents," Algorithms, vol. 6, no. 3, pp. 442-456, 2013.
- [10] B. Talwar and A. K. Gupta, "Ant colony based mobile ad hoc networks routing protocols: a review," International Journal of Computer Applications, vol. 49, no. 21, pp. 36-42, 2012.
- [11] S. Mehruz and M. N. Doja, "Swarm intelligent power-aware detection of unauthorized and compromised nodes in MANETs," Journal of Artificial Evolution and Applications, vol. 2008, Article ID 236803, 16 pages, 2008.
- [12] H. Xie, Z.-G. Zhang, and X.-G. Zhou, "A novel routing protocol in wireless sensor networks based on ant colony optimization," in Proceedings of the International Conference on Environmental Science and Information Application Technology (ESIAT '09), vol. 2, pp. 646-649, Wuhan, China, July 2009.
- [13] J. Yang, M. Xu, W. Zhao, and B. Xu, "A multipath routing protocol based on clustering and ant colony optimization for wireless sensor networks," Sensors, vol. 10, no. 5, pp. 4521-4540, 2010.
- [14] Dorigo, M., Birattari, M., & Stutzle, T., "Ant colony optimization," IEEE Computational Intelligence Magazine, 1(4), 28-39, 2006.
- [15] Perkins, C. E., & Royer, E. M., "Ad hoc on-demand distance vector routing," In Proceedings of the second IEEE workshop on mobile computing systems and applications, pp. 90-100, 1997.
- [16] Johnson, D. B., Maltz, D. A., & Broch, J., "DSR: The dynamic source routing protocol for multipoint wireless ad hoc networks," In Ad hoc networking, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, pp. 139-172, 2001.

- [17] Clausen, T., & Jacquet, P., "Optimized Link State Routing Protocol (OLSR)," IETF RFC 3626, 2003.
- [18] CE Perkins, P Bhagwat, Highly dynamic destination sequenced distance vector routing (DSDV) for Mobile Computers, in Proc. of ACM SIGCOMM'94, ACM, London-UK, pp. 234–244, 1994.
- [19] Y Zuo, Z Ling, Y Yuan, A hybrid multi-path routing algorithm for industrial wireless mesh networks. EURASIP J. Wirel. Commun. Netw. 82(2013), 2013.
- [20] M. Günes, U. Sorges, and I. Bouazizi, "ARA—the ant-colony based routing algorithm for MANETs," in Proceedings of the International Conference on Parallel Processing Workshops, pp. 79–85, British Columbia, Canada, August 2002.
- [21] F. Ducatelle, G. A. Di Caro, and L. M. Gambardella, "Using ant agents to combine reactive and proactive strategies for routing in mobile Ad Hoc networks," Tech. Rep. IDSIA-28-04-2004, 2004.
- [22] F. Correia and T. Vazão, "Simple ant routing algorithm strategies for a (Multipurpose) MANET model," Ad Hoc Networks, vol. 8, no. 8, pp. 810–823, 2010.
- [23] Sohail Jabbar, Rabia Iram, Muhammad Imran, et al., "Energy Aware Simple Ant Routing Algorithm for Wireless Sensor Networks," Mathematical Problems in Engineering, vol. 2015, Article ID 194532, 11 pages, 2015. doi:10.1155/2015/194532
- [24] M. Imran, M. A. Alnuem, M. S. Fayed, and A. Alamri, "Localized algorithm for segregation of critical/non-critical nodes in mobile ad hoc and sensor networks," Procedia Computer Science, vol. 19, pp. 1167–1172, 2013.
- [25] A. Misra and S. Banerjee, "MRPC: Maximizing Network Lifetime for reliable routing in Wireless Environments", IEEE Wireless Communications and Networking Conference, 2002.
- [26] Maleki, M.; Dantu, K.; Pedram, M., "Lifetime prediction routing in mobile ad hoc networks," in Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE , vol.2, no., pp.1185-1190 vol.2, 20-20 March 2003.
- [27] Sankar, A.; Zhen Liu, "Maximum lifetime routing in wireless ad-hoc networks," in INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies , vol.2, no., pp.1089-1097 vol.2, 7-11 March 2004.
- [28] J Liu, J Chen, Y Kuo, "Multipath routing protocol for networks lifetime maximization in ad-hoc networks," in Proc. ed. by . of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom '09) (IEEE, Beijing-China), 2009.
- [29] Omar Smail, Bernard Cousin, Rachida Mekki and Zoulikha Mekkakia, "A multipath energy-conserving routing protocol for wireless ad hoc networks lifetime improvement," EURASIP Journal on Wireless Communications and Networking 2014, 2014 .
- [30] M Bheemalingaiah, MM Naidu, D Sreenivasa Rao, G Varaprasad, "Energy aware node disjoint multipath routing in mobile ad hoc network," J. Theor. Appl. Inf. Technol. 2, 416–431, 2009.