Ashdin Publishing
International Journal of Sensor Networks and Data Communications
Vol. 1 (2011), Article ID X110203, 14 pages
doi:10.4303/ijsndc/X110203

Research Article

Ant Colony and Load Balancing Optimizations for AODV Routing Protcol

Ahmed M. Abd Elmoniem,[1] Hosny M. Ibrahim,[2] Marghny H. Mohamed,[1] and Abdel-Rahman Hedar[1]

[1]Computer Science Department, Faculty of Computer and Information Systems, Assiut University, Assiut 71111, Egypt
[2]Information Technology Department, Faculty of Computer and Information Systems, Assiut University, Assiut 71111, Egypt
Address correspondence to Ahmed M. Abd Elmoniem, ahmedcs982@hotmail.com

Received 22 February 2011; Revised 4 May 2011; Accepted 6 May 2011

Abstract In this paper, we propose two methods to improve the Ad-Hoc On-Demand Distance-Vector (AODV) protocol. The main goal in the design of the protocol was to reduce the routing overhead, buffer overflow, end-to-end delay and increase the performance. A multi-path routing protocol is proposed which is based on AODV and Ant Colony Optimization (ACO). This protocol is refereed to Multi-Route AODV Ant routing (MRAA). Also we propose a load balancing method that uses all discovered paths simultaneously for transmitting data. In this method, data packets are balanced over discovered paths and energy consumption is distributed across many nodes through network. This protocol is refereed to Load Balanced Multi-Route AODV Ant routing algorithm (LBMRAA).

Keywords mobile ad-hoc network (MANET); routing; ant colony optimization; load balancing

1 Introduction

The problem of mobile ad-hoc network (MANET) can be summarized in the answer of this question: how to find the route between the communicating end-points. One of the main reasons is that routing in MANETs is a particularly challenging task due to the fact that the topology of the network changes constantly and paths which were initially efficient can quickly become inefficient or even infeasible. Moreover, control information in the network is very restricted. This is because the bandwidth of the wireless medium is very limited, and the medium is shared. It is therefore important to design algorithms that are adaptive, robust and self-healing. Moreover, they should work in a localized way, due to the lack of central control or infrastructure in the network [5, 7].

In this paper, we present a modification to optimize the well-known MANET routing protocol AODV using swarm intelligence [1]. ACO algorithm is one of the most important algorithms in swarm intelligence algorithms category. It considers the ability of simple ants to solve complex problems by cooperation. The interesting point is that the ants do not need any direct communication for the solution process, instead they communicate by the indirect communication of individuals through modifying their environment. Several algorithms which are based on ant colony were introduced in recent years to solve different problems (e.g. optimization problems, including MANET [7, 11]). To show that the approach of ACO has the potential to be applied to optimize the existing ant-free MANET protocols, we have taken the existing implementation of AODV then applied the ACO algorithm to AODV without removing or affecting any of pre-existing AODV mechanisms. Results are based on simulations made with the current implementation in OPNET Modeler V.14 [19], which complies to the published IETF version of AODV routing protocol [21].

Also, we present a modification to optimize the proposed MRAA protocol using the concept of load balancing. In networking, load balancing is a technique to distribute workload evenly across two or more computers, network links, or other resources; in order to get optimal resource utilization, maximize throughput, minimize response time, and avoid overload. Using multiple routes with load balancing, instead of a single route, may increase reliability through redundancy. Several protocols which are based on load balancing were introduced in recent years (e.g. optimization problems, including MANET [10, 25, 28]). To show that the approach of load balancing has the potential to be applied to optimize the proposed MRAA MANET protocol, we have taken the implementation of MRAA then applied a load balancing technique to MRAA. Results are also based on simulations made with OPNET Modeler V.14 [19].

The remainder of this paper is organized as follows: In Section 2, we present the Mobile Ad-Hoc Network routing problem. In Section 3, we introduce the ant colony optimization meta heuristic for MANETs. In Section 4, we explain the new routing algorithm (MRAA) in detail. In Section 5, we show some simulation results to show the ability of the ACO to improve routing performance. In Section 6, we explain the new routing algorithm (LBMRAA) in detail. In Section 7, we show some simulation results to show the ability of the load-balancing to improve routing performance. Finally, a conclusion is given in Section 8 and a brief future work is given in Section 9.

2 Mobile Ad-Hoc Network routing problem

Mobile Ad-Hoc Networks (MANETs) nodes are typically distinguished by their limited power, processing, and memory resources as well as high degree of mobility. In such networks, the wireless mobile nodes may dynamically enter the network as well as leave the network. Due to the limited transmission range of wireless network nodes, multiple hops are usually needed for a node to exchange information with any other node in the network. Thus routing is a crucial issue to the design of a MANET [17].

Routing algorithms in wired networks are usually based upon either the well-known distance vector or link state routing algorithms. These algorithms need frequent routing advertisements messages to be broadcasted by each router, which allow the announcement of new routes to other routers in the network. In distance vector routing [14], each router broadcasts to only its neighboring routers the distance to all other nodes, then upon receiving this information the neighboring routers recompute the shortest path to each node. In link-state routing [16], each router broadcasts to all routers in the network the status of each of its adjacent links, then routers of the network recompute the shortest distance to each node based upon the complete topology of the network.

The above two ways of how routes are found are clearly not efficient for the type of dynamic changes which may occur in an ad-hoc network. In wired networks, routers can only leave or join the network by human intervention. In an environment with mobile nodes moving constantly, the changing topology will not only cause continuous re-computation of routes but the overall convergence to stable routes may be infeasible due to the high-level of mobility and will also cause the network become congested by routing information packets [17].

So, there are numerous issues to consider when deploying MANETs. The following are some of the main issues: unpredictable behavior, unreliable communication links, limited resources and changing topology [17].

Routing protocols for MANETs must deal with these issues to be effective. So our goal in this paper is to try to apply the ant-colony optimization technique, that has proved its effectiveness in wired networks routing protocols [6], to optimize the existing MANET routing protocols. In our case, we applied ACO to AODV routing protocol to overcome the addressed issues and to improve its performance.

2.1 Ad-Hoc On-Demand Distance Vector

Routing protocols for a MANET can be classified as proactive (table-driven) and reactive (on-demand), depending on how they react to topology changes [24]. A host running a proactive protocol propagates routing-related information to its neighbors whenever a change in its link state is detected. The information may trigger other mobile hosts to recompute their routing tables and further propagate more routing-related information. The amount of information propagated each time is typically proportional to the scale of the MANET. Examples of proactive protocols include Wireless Routing Protocol (WRP) [18] and Destination Sequenced Distance Vector (DSDV) [22].

Observing that a proactive protocol may pay costs to construct routes even if mobile hosts do not have such need, thus wasting the limited wireless bandwidth, many researchers have proposed the usage of reactive-style protocols, in which routes are only constructed on-demand. Many reactive protocols have been proposed based on such on-demand philosophy, such as Dynamic Source Routing (DSR) [4], Signal Stability-Based Adaptive routing (SSA) [9], Ad-Hoc On-Demand Distance Vector routing (AODV) [23], and Temporally-Ordered Routing Algorithm (TORA) [20]. Recently, a hybrid of proactive and reactive approaches, called the Zone Routing Protocol (ZRP) [12], has also been proposed.

An on-demand routing protocol only tries to discover/maintain routes when necessary. As a routing protocol for MANET, it is going to need to address the following three issues: route discovery, data forwarding, and route maintenance. When a node wants to send data to another node, it has to find a route first. Then data packets can be forwarded. Note that the topology of the MANET may change over the time due to some nodes that may go out of transmission range or even may fail, which may disconnect an existing route while data packets are being transmitted. Better routes may also be discovered during the forwarding process. This is referred to as route maintenance. In the following, we review AODV as one of the reactive (on-demand) routing protocols according to these issues [26].

When a node *S* needs a route to some destination *D*, it broadcasts a ROUTE-REQUEST message to its neighbors, including the last known sequence number for that destination. The ROUTE-REQUEST is flooded in a controlled manner through the network until it reaches a node that has a route to the destination. Each node that forwards the ROUTE-REQUEST creates a reverse route for itself back to node *S*.

When the ROUTE-REQUEST reaches a node with a route to *D*, that node generates a ROUTE-REPLY that contains the number of hops necessary to reach *D* and the sequence number for *D* most recently seen by the node generating the ROUTE-REPLY. Each node that participates in forwarding this reply back toward the originator of the ROUTE-REQUEST (node *S*) creates a forward route to *D*. The state created in each node along the path from *S* to *D* is hop-by-hop state that is, each node remembers only the next hop and not the entire route, as would be done in source routing.

In order to maintain routes, AODV normally requires that each node periodically transmit a HELLO message. Failure to receive three consecutive HELLO messages from a neighbor is taken as an indication that the link to the neighbor is down. Alternatively, the AODV specification briefly suggests that a node may use physical layer or link layer methods to detect LINK-BREAKAGES to nodes that it considers neighbors as described in [5,23].

3 Ant colony optimization for MANET routing problem

The ant colony optimization meta-heuristics is a particular class of ant algorithms. Ant algorithms are multi-agent systems, which consist of agents with the behavior of individual ants [3,8].

3.1 Real ants mechanism

The basic idea of the ant colony optimization meta heuristic is taken from the food searching behavior of real ants. When ants are on they way to search for food, they start from their nest and walk toward the food. When an ant reaches an intersection, it has to decide which branch to take next. While walking, ants deposit pheromone, which marks the route taken. The concentration of pheromone on a certain path is an indication of its usage. With time the concentration of pheromone decreases due to diffusion effects. This property is important because it is integrating dynamic aspect into the path searching process.

Figures 1, 2, 3, 4 show a scenario with a route from the nest to the food place in which all ants follow the pheromone trail. Suddenly, an obstacle gets in their way so the first ants randomly select the next branch between the two branches: the upper and lower branches. Since the upper route is shorter than the lower one, the ants which take this shorter path will reach the food place first. On their way back to the nest, the ants again have to select a path. After a short time the pheromone concentration on the shorter path will be higher than on the longer path, because the ants using the shorter path will increase the pheromone concentration faster. The shortest path will thus be identified and eventually all ants will only use this one. This behavior of the ants can be used to find the shortest path in networks. Especially, the dynamic component of this method allows a high adaptation to changes in mobile ad-hoc network topology, since in these networks the existence of links are not guaranteed and link changes occur very often [11].

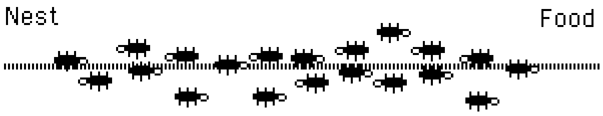


Figure 1: Ants trail from nest to food.

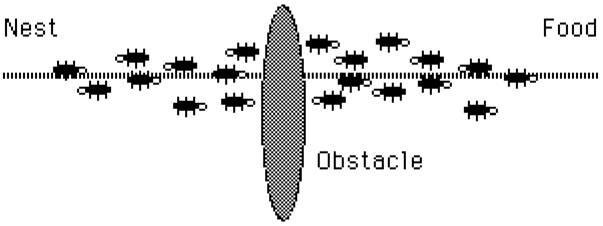


Figure 2: Ants facing an obstacle.

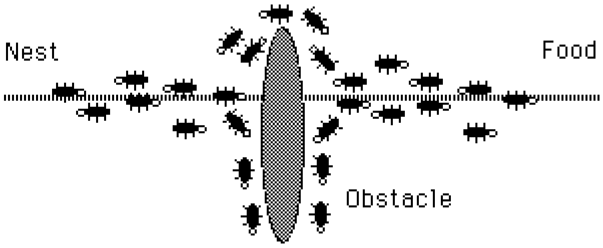


Figure 3: Ants try to found alternative routes.

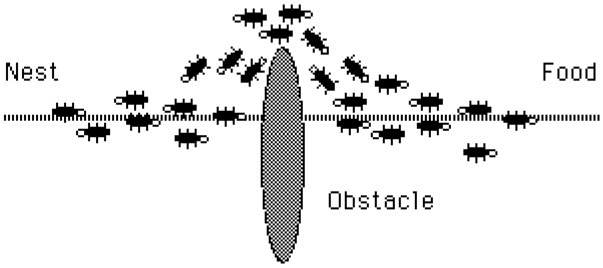


Figure 4: Ants go through the shortest route.

3.2 Why ant colony optimization meta-heuristic suits to ad-hoc networks

The simple ant colony optimization meta-heuristic shown in the previous section illustrates different reasons why this kind of algorithms could perform well in mobile multi-hop ad-hoc networks. We will discuss various reasons by considering important properties of mobile ad-hoc networks [11]:

- ACO is based on agent systems and works with individual ants. This allows a high adaptation to the current dynamic topology of the network;

- ACO is based only on local information, that is, no routing tables or other information blocks have to be transmitted to neighbors or to all nodes of the network;
- it is possible to integrate the connection/link quality into the computation of the pheromone concentration;
- each node has a routing table with entries for all its neighbors, which contains also the pheromone concentration. Thus, the approach supports multi-path routing.

4 The multi-route AODV ant (MRAA) algorithm elements

Hereby, we describe the different procedures used to handle the different events of the proposed protocol. The events to be handled are neighbor connectivity, route establishment request, route establishment reply, route expiry, connection loss and local repair. More formal details of these components are given below.

The following data structures have been used by all elements of the algorithm:

(1) a pheromone matrix **Ak**: a matrix organized as in vector-distance routing table algorithms, but with probabilistic entries which is the pheromone value of each possible route. **Ak** defines the probabilistic routing policy currently adopted at node k: for each possible destination d and for each neighbor node n, **Ak** stores a pheromone value **Pnd** expressing the goodness (desirability), under the current network-wide routing policy, of choosing n as next node when the destination node is d;

(2) a routing table **Tk**: a routing table organized as in vector-distance algorithms, but with a pheromone entry added. **Tk** defines the pheromone value besides the usual routing entries at node k: for each possible destination d, **Tk** stores a pheromone value **Pnd** expressing the best (most desirable) of choosing n as the next node when the destination node is d.

4.1 Neighbor connectivity

When a packet from a new neighbor has just arrived, perform the following procedure.

Procedure 1 *Neighbor_Connectivity*

(1) If the arriving packet is from a new neighbor then:
- *add a route to the new neighbor in the routing table **Tk** with initial pheromone value of 0.0,*
- *update the pheromone matrix **Ak** to reflect the new neighbor as a candidate next hop for all other nodes in the network but with initial value of pheromone of 0.0, and*
- *increase the number of available active neighbors for this node by 1.*

(2) Else if the neighbor already exists:
- *update the expiry time of the connection for this neighbor.*

4.2 Route establishment request

Send forward ant agents to find a route to the requested destination because no route exists, by applying the following procedure.

Procedure 2 *Route_Establishment_Request*

(1) If this node is the source node then:
- *initialize the memory of the forward ant,*
- *if a route to destination does not exist and there exists no highest pheromone neighbor then:*
  Broadcast the forward ant to all available active neighbors, and
- *else if a route to destination does not exist but an active neighbor with highest pheromone exists then:*
  Send the forward ant to that neighbor.

(2) Else if it is an intermediate node then:
- *update the memory of the forward ant,*
- *if a route to destination does not exist and there exists no highest pheromone neighbor then:*
  Broadcast the forward ant to all available active neighbors, and
- *else if a route to destination does not exist but an active neighbor with highest pheromone exists then:*
  Send the forward ant to that neighbor.

(3) Else if it is the destination node:
- *send a backward ant to the source to give it the feedback of the route.*

4.3 Route establishment reply

Send backward ant agent to inform the source with a route to destination and apply pheromone as it returns back, by acting up on the following procedure.

Procedure 3 *Route_Establishment_Reply*

(1) If this node is the source node then:
- *calculate the pheromone value of the arriving backward ants using their memory with the following equation:*

$$P = \frac{N_n}{N_h} + \frac{L_C}{100} + \frac{1}{T}, \tag{1}$$

where P is pheromone of a node Src to a node Dest, N_h is number of hops from Src to Dest, N_n is number of nodes in the network,

L_C is accumulated link capacity from Src To Dest, and T is trip time of the ant between Src and Dest,

- *for each neighbor in the pheromone matrix **Ak** that uses this node as next hop to destination do:*
  update the pheromone entry in matrix to pheromone value calculated by equation (1)*, and*
- *find the highest pheromone neighbor and update the routing table **Tk** for that destination to make the node forward packets to that neighbor.*

(2) Else if it is an intermediate node then:

- *update the reverse memory of the backward ant, and*
- *(2.1) If the route to destination does not exist then:*
  - *create a new route table entry for that destination node in routing table **Tk**,*
  - *for each neighbor in the pheromone matrix **Ak** that uses this node as next hop to destination do:*
    *update the pheromone entry in matrix **Ak** to pheromone value calculated by* (1)*, and*
  - *find the highest pheromone neighbor and update the routing table **Tk** for that destination to make the node forward packets to that neighbor, and*
  - *increase the number of active destinations by 1.*
- *(2.2) Else if the route to destination exists then:*
  - *increase the pheromone value of that neighbor to that destination using the following equation:*

    $$Pn = \alpha * Po + \beta * P, \tag{2}$$

    where Pn is the new pheromone value, Po is the old historic pheromone value, P is current pheromone value calculated using equation (1)*, $\boldsymbol{\alpha}$ is weight of the historic pheromone value, and $\boldsymbol{\beta}$ is weight of the current pheromone value.*[1]
  - *decrease the pheromone value of all other neighbors to that destination using the following equation*

    $$Pn = \alpha * Po + \beta * P, \tag{3}$$

    where Pn is the new pheromone value, Po is the old pheromone value, P is pheromone value calculated using equation (1)*, $\boldsymbol{\alpha}$ is weight of the historic pheromone value, and $\boldsymbol{\beta}$ is weight of the current pheromone value.*
  - *find the highest pheromone valued neighbor and update the route entry for that destination to make the node forward packets to that highest valued neighbor.*

(3) Else if it is the destination node:

- *initialize the reverse memory of the backward ant,*
- *copy the memory of arriving forward ant to the memory of backward ant, and*
- *make the backward ant follow the same path which is stored in its memory to the source leaving behind a pheromone value accordingly on each hop in the back path to source node.*

4.4 Route expiry

After some time, route expiry timer timeouts. Then the route entry must be removed from routing table and pheromone array, by going through the following procedure.

Procedure 4 *Route_Expiry*

(1) If the route expiry timeout event has occurred:

- *remove that destination from the pheromone matrix **Ak**,*
- *remove the routing table entry from routing table **Tk**, and*
- *decrease number of active destinations.*

4.5 Connection loss

After some time, if no packets arrive whether data, ants or hello from a neighbor, then we do the following actions.

Procedure 5 *Connection_Loss*

(1) If the connection timeout event has occurred:

- *reset the pheromone value of that neighbor to 0.0 in the pheromone matrix **Ak**,*
- *remove the routing table **Tk** entries that go through that neighbor,*
- *if any other available neighbor has a nonzero pheromone value to that destination, update the routing table **Tk** to use the other neighbor (Automatic Repair using Multi-Ant Routes), and*
- *decrement number of active neighbors.*

[1] We have used 0.75 as a value for $\boldsymbol{\alpha}$ and 0.25 as a value for $\boldsymbol{\beta}$, which gives more weight to previous accumulated historic value of pheromone

4.6 Local repair

If some request packet arrives and no routing entry exists for it, then the algorithm try to do local repair as follows.

Procedure 6 *Local_Repair*

(1) If local repair is enabled on this node:

- *if any other available neighbor has a non-zero pheromone value to that destination update the routing table **Tk** to use the other neighbor (Automatic Repair using Multi-Ant Routes). Then, send all packets in queue to the destination using this new neighbor as the next hop, and*
- *otherwise use default built-in AODV Local Repair strategy.*

5 Simulation and performance evaluation

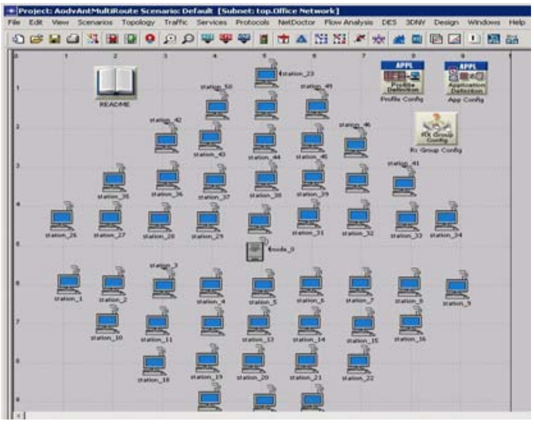
5.1 Simulation environments

In order to evaluate the performance of the proposed protocol, we use the network simulator OPNET Modeler V.14 and make modification to the AODV. We used more than one case to establish that MRAA was an improvement over AODV. The following is the explanation of the four chosen cases.

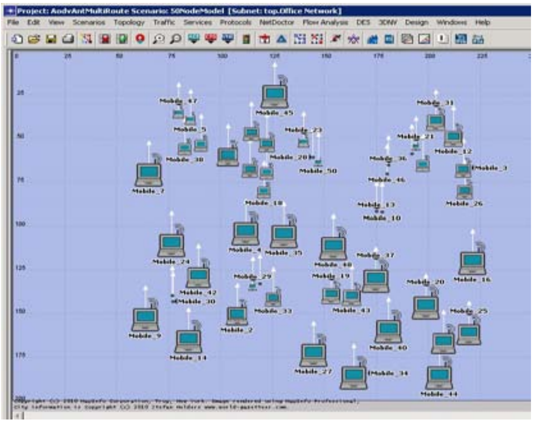
(1) Number of 50 Nodes and running the same MANET Routing Protocol with default parameters but the simulation was run for 1 hour, FTP is configured to generate medium traffic loads and is allowed to make only serial connections with 3 repeats during the whole simulation. For traffic source, FTP application is used generating data starting randomly using uniform distribution from the 100th second to the end of simulation and each time the application continues to send data for 10 seconds three times during the whole simulation to random chosen destinations. However, the wireless nodes are fixed rather than mobile as shown in Figure 5.[2]

(2) The same model exactly as above but with Number of 50 Nodes and running the same MANET Routing Protocol with default parameters but the simulation was run for 5 hours, FTP is configured to generate high traffic loads and is allowed to make concurrent connections with unlimited repeats during the whole simulation as shown in Figure 5.

(3) Number of 50 Nodes positioned randomly in an area of 300×200 meters and running the same MANET Routing Protocol with default parameters but the simulation was run for 15 minutes. For traffic source, data was generated with interarrival time of packets is exponentially distributed with mean outcome of 10 seconds; for packet size we use exponential distribution with mean outcome of 1024 byte and the nodes start sending from start to end of simulation to randomly chosen destinations. As a mobility model, we use the random waypoint model as shown in Figure 5. In the model, each node starts its journey from a random location to another random location with a uniformly chosen speed. Once the node reaches its destination, another new random destination is targeted starting from the previous random destination after a pause time.


Project: AodvAntMultiRoute Scenario: Default [Subnet: top.Office Network]
File Edit View Scenarios Topology Traffic Services Protocols NetDoctor Flow Analysis DES 3DNV Design Windows Help
README
Profile Config
App Config
Rx Group Config
node_0

(a)


Project: AodvAntMultiRoute Scenario: 50NodeModel [Subnet: top.Office Network]
File Edit View Scenarios Topology Traffic Services Protocols NetDoctor Flow Analysis DES 3DNV Design Windows Help

(b)

Figure 5: (a) A figure showing the 50 stationary nodes network topology and their positions. (b) A figure showing the 50 mobile nodes network topology and their initial positions, note that the upward arrow does not mean they all move upward but they move according to random way-point in random directions with changing speeds and making some stops during the simulation.

[2] Note that default parameters differ from protocol to another but for AODV the route expiry timeout is 3 seconds, allowed hello is 2 times, hello interval is uniform (1,1.1), net diameter is 35 hops, and local repair is enabled.

Protocol name	Simulation duration	Average delay	Routing traffic (S)	Routing traffic (R)
–	(s)	(s)	bits	bits
AODV	5	0.006848	1,693.27	4,995.96
MRAA	14	0.0007845	6,758.01	22,458.53

(a)

Protocol name	Loss	FTP download	FTP upload	Throughput	Network load
–	no.	(s)	(s)	(bits/s)	(bits/s)
AODV	0.0	0.7450	0.6122	20,354.80	18,606.08
MRAA	0.02694	1.970	1.867	23,483.96	7,783.63

(b)

Table 1: Simulation results for case 1. (a) A table showing the average values for average delay, throughput, network load, routing traffic sent and routing traffic received. (b) A table showing the number of dropped packets, FTP download response time, FTP upload response time and simulation duration.

(4) The same model exactly as above but with Number of 100 Nodes positioned randomly in an area of 600×400 meters as shown in Figure 5.

5.2 Performance metrics

Five key performance metrics are evaluated in our experiments:

- throughput: it represents the total number of bits (in bits/s) forwarded from wireless LAN layers to higher layer which also means “received” throughput at the destinations of the data sources;
- WLAN network load: it represents the total number of bits (in bits/s) submitted to wireless LAN from higher layer which also means the combined sending rate of all data sources;[3]
- WLAN end-to-end delay: this includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer time;
- normalized routing overhead: this metric has two variants: packet overhead is the number of routing packets “transmitted” per data packet “delivered” at the destination;
- average loss ratio: average number of data packet dropped not routing protocol packets dropped.

5.3 Simulation results

We can see from results of Table 1 that in the case of stationary stations with no load and concurrent connections in the network, and also the users being only allowed to make 3 FTP sessions during the simulation which lasts for only 1 hour, the traditional AODV outperforms MRAA in the routing overhead, no. of dropped packets, FTP up/download time and simulation time but MRAA outperforms the AODV in throughput, network load and the average end-end WLAN delay. But from the results of simulation (not shown in the table) the number of dropped AODV packets in case of traditional AODV was 3.5 and in case of MRAA was 11.3.

However, as we can see from the results of Table 2, in the case of stationary stations where they are making high load traffic and are allowed to make unlimited number of concurrent connections in the network, in addition to the users being allowed to make unlimited number of FTP sessions during the simulation which lasts for 5 hours, that MRAA outperforms AODV in the routing overhand, network load, FTP up/down-load time and simulation time but AODV outperforms the MRAA in throughput, no. of dropped packets and the average end-end WLAN delay. But note that the differences in the average delays of AODV and MRAA are not great at all and also note that from the results of simulation (not shown in the table) the number of dropped AODV packets in case of traditional AODV was 22,488.25 and in case of MRAA was 11,001.05 which means there is more wasted AODV packets in case of traditional AODV than MRAA.

We can see from the results of Tables 3 and 4 that in the case of mobile stations, traditional AODV outperforms MRAA in only the number of dropped data packets and throughput, but MRAA outperforms the AODV in network load, routing overhand, and the average end-end WLAN delay. But from the results of simulation (not shown in the table) the number of dropped AODV packets in case of traditional AODV was 30.8 (50 nodes) and 65.76 (100 nodes) and in case of MRAA was 27.68 (50 nodes) and 58.70 (100 nodes), and the greater throughput is due to the

[3] Note that without any retransmission, the ratio of throughput and offered load is simply the packet delivery ratio.

Protocol name	Simulation duration	Average delay	Routing traffic (S)	Routing traffic (R)
–	(s)	(s)	bits	bits
AODV	2700	0.20809	162,128.87	77,792.89
MRAA	1300	0.24189	17,488.72	48,654.78

(a)

Protocol name	Loss	FTP download	FTP upload	Throughput	Network load
–	no.	(s)	(s)	(bits/s)	(bits/s)
AODV	0.00322	522.97	350.48	901,054.14	1,035,794.03
MRAA	5.160	53.42	25.5557	266,315.41	263,84.1

(b)

Table 2: Simulation results for case 2. (a) A table showing average values for average delay, throughput, network load, routing traffic sent, routing traffic received. (b) A table showing the number of dropped packets, FTP download response time, FTP upload response time and simulation duration.

Protocol name	Average delay	Throughput	Network load	Routing traffic (S)	Routing traffic (R)	Loss
–	(sec)	(bits/s)	(bits/s)	bits	bits	no.
AODV	0.003082	2,451,903.38	77,287.38	75,077.51	2,449,693.51	0.0
MRAA	0.000669	744,958.77	64,478.37	15,359.42	695,840.87	1.3333
OSLR	0.00040425	2,475,661.72	56,414.14	44,762.75	2,194,792.82	0.0667
GRP	0.0003099	291,391.88	18,271.24	5,822.13	224,319.82	0.4169
DSR	0.00347	13,387.74	8,576.05	2,244.12	7,056.89	0.0

Table 3: Simulation results for case 3. A table showing the average values for average delay, throughput, network load, routing traffic sent, routing traffic received and number of dropped packets.

Protocol name	Average delay	Throughput	Network load	Routing traffic (S)	Routing traffic (R)	Loss
–	(s)	(bits/s)	(bits/s)	bits	bits	no.
AODV	0.00714	13,416,041.64	338,547.37	333,985,985.74	13,411,484.55	0.0
MRAA	0.00578	3,495,817.96	156,678.03	51,504.29	3,390,649.60	2.7578
OSLR	0.00056417	17,893,758.44	192,775.01	169,433.99	16,772,453.80	0.143
GRP	0.0003479	1,091,655.67	36,655.00	16,390.27	1,069,449.16	0.0
DSR	0.00694	58,201.28	17,759.09	5,233.99	26,682.38	0.88

Table 4: Simulation results for case 4. A table showing the average values for average delay, throughput, network load, routing traffic sent, routing traffic received and number of dropped packets.

high routing traffic that is generated, also the simulation time in case of AODV took 120 s (50 nodes) and 2078 s (100 nodes) to finish nearly twice longer than MRAA that took only 79.6 s (50 nodes) and 1015 s (100 nodes), which is due to that AODV makes more processing per node each time a route not found to find a fresh new route.

Also, as we an see from the results of Tables 3 and 4, that in the case of mobile stations when traditional AODV is compared to the other MANET routing protocols (OSLR, GRP, DSR) most of them outperform AODV. However, MRAA results could show that it can compete and with respect to some performance metrics it can outperform the other MANET routing protocols.

5.4 Results discussion

In order to compare the performance of the ant-colony optimized protocol with AODV, previous tables show the end-to-end delay, throughput, network load, sent routing traffic, received routing traffic, number of dropped data packets and simulation duration. As the more mobile nodes become rather than being stationary, the performance gain by optimization algorithm becomes more significant. The proposed protocol can deliver more packets to the destination in less time and with lower overhead than AODV because MRAA uses alternate routes for data delivery in case working routes are broken. As the number of alternate paths increases, the loss ratio of data packets slightly decreases. Also the performance of MRAA is superior in terms of the end-to-end delay and throughput with comparison with the network load in most cases that is due to the routing decisions based on more intelligent decision of pheromone value that is increased for more promising routes and decreased for less promising routes. In addition, we can observe that MRAA is superior in terms of both the amount of generated routing overhead and the time of processing overhead as less routing traffic is generated because ant agents during discovery phase and it reaches a node that previously traversed by another ant agent that has left a pheromone on that node then the ant agent does not rebroadcast itself instead it follow the path of the previous ant agent and it will reinforce that pheromone while it is returning back to the source node.

We can observe that the effectiveness of both protocols increases because of the decrease in packet mobility in comparison with other MANET routing protocols; but we can see from the results in tables that MRAA's performance and effectiveness do not degrade as much as AODV when compared to other routing protocols; for example see the following:

- MRAA versus OSLR: we can see that OSLR has a little bit lower delay and lower packet loss, but in contrast it has a great routing overhead compared to MRAA and places more load on the network from routing overhead than MRAA;
- MRAA versus GRP: we can see that GRP has a little bit lower delay, lower packet loss, lower routing overhead and of less load on the network than MRAA, that is because GRP is well suited for mobile nodes and use the GPS system to establish routes between the nodes;
- MRAA versus DSR: we can see that DSR has a little bit more delay, lower packet loss, much lower routing overhead compared to MRAA but it has a much more lower throughput when compared to MRAA and all other routing techniques.

6 Load balancing

Our just discussed ad-hoc routing protocol uses route pheromone as the routing metric. This protocol was based on the highest pheromone path routing; in which nodes on the high pheromone path will get more heavily loaded than others since they are frequently chosen as the routing path. Having a heavy load can exhaust a node's resources such as bandwidth, processing power, battery energy, and memory storage. Furthermore, if one of the heavily loaded nodes is congested, it can lead to packet loss and buffer overflow, resulting in longer end-to-end delay, decrease in throughput. So, it is important that some form of load balancing is present in the network.

Load balancing deals with increasing the performance of the system by distributing the packets from highly loaded nodes to lower loaded or idle nodes. By doing so, the total time to process all packets may reduce considerably and also makes it sure that no node sits idle while some packets are waiting to be processed. Routing protocols are important for the proper functioning of ad-hoc networks. A major problem of all existing ad-hoc routing protocols is that they do not have methods for evaluating the load and/or quality of a path during route discovery. So they cannot balance the load on different routes. Also, both proactive and reactive protocols chose a route based on the metric, the smallest number of hops to the destination. But it may not be the most significant route when there is congestion or bottleneck in the network. It may cause the drop rate, end-to-end delay, or routing overhead to be increased [28].

6.1 Load balancing routing protocol categorization

Over the years, several load balanced ad-hoc routing protocols have been proposed. Most of the approaches are on-demand-based protocols which combine load balancing strategies with route discovery. A route with the least load among multiple possible routes from source to destination is usually chosen. These routing protocols can generally be categorized into three types (based on their load balancing techniques):

- delay-based: where load balancing is achieved by attempting to avoid nodes with high link delay. An example protocol using this approach is Load-Aware On-Demand Routing (LAOR) [25];
- traffic-based: where load balancing is achieved by evenly distributing traffic load among network nodes. Examples of traffic based load balanced routing protocols are Associativity Based Routing (ABR) [27], Load Balanced Ad-Hoc Routing (LBAR) [13] and Traffic-Size Aware (TSA) scheme [2];
- Hybrid-based: where load balancing is achieved by combining the features of traffic and delay-based techniques. Examples are Content Sensitive Load Aware Routing (CSLAR) [15] and Load Aware Routing in Ad-Hoc (LARA) [2].

These are various proposed algorithms for load balancing that consider traffic load as a route selector, but these algorithms neither reflect burst traffic nor transient congestion. In order to ensure uninterrupted communication and in order to make routing protocols more efficient in the presence of node movement, two issues, route maintenance and bandwidth reservation, need due mention. A very good solution to these issues is multi-path routing. Due to such a multi-path routing, even if one path fails, data can still be routed to the destination using the other routes. Thus, the cost of rediscovering a new path can be salvaged [20].

6.2 Load balanced multi-route AODV ant routing

Similar to MRAA, LBMRAA is an on-demand routing protocol that consists of three main phases as follows.

- *Route establishment:* when a node has data to send and no route information is known, it initiates path discovery process by sending forward ant (FANT) to its neighbors. The FANT packet identifies the target host for which the route is requested, in addition to the address of the initiator of the request and the target of the request. Also each FANT contains a route record which is an accumulated record of hops during this route discovery as well as an accumulated value of the pheromone that get calculated over the route. Each FANT packet also contains a unique request ID, set by the initiator. To prevent the possibility of forming routing loops, each intermediate node that receives a FANT propagates it if their address is not already included in FANT's current Route Record then appends its own address to the FANT's Route Record before rebroadcasting it. If a node receives a FANT with a new request ID it stores the pheromone of this request in the pheromone array, appends its address to the Route Record of RREQ, recalculate the new value of pheromone of FANT and rebroadcast it [10].
  Rebroadcasting the FANTs is continued until they reach the destination. By using this method for propagation of the FANTs, many FANTs from different routes will be received to the destination. When a destination receives FANTs, it reverses the route in the route record from the received FANTs, and uses this route to send back the backward ant (BANT) packet to the source. As the BANTs travel back to the source, each node along the path sets up a forward pointer to the node from which the BANT came, NextHop, and updates its pheromone information for route entries to the source. When a node receives multiple BANTs from a node, it stores both the NextHop and pheromone entry along each route that could be found. This process is continued until the BANTs reach the source. These two phases create multiple routes from source to the destination [10].
- *Data transmission:* When the source receives BANTs, it can transmit data packets through the discovered routes. Our routing protocol uses hop-by-hop method for forwarding data. Each node that receives data packets sends them to the next hops according to their pheromone values. Each next hop that has greater pheromone receives more data than the next hops that have less pheromone. This process causes that all of the discovered routes are used and data packets are distributed across all of the paths simultaneously [10].
- *Route repair:* if a route is not used for some period of time, a node cannot be sure whether the route is still valid. Consequently, the node removes the route from its routing table. If data is flowing and a link break is detected, firstly, we try to distribute the data packets over any existing routes that are available and have non zero value of pheromone. However, if no route exists at all, a Route Error (RERR) packet is sent to the source of the data in a hop-by-hop fashion. If no entry for a destination exists in Route Table of source, it invalidates the route and reinstates route discovery process if necessary. But, if local repair is enabled in the settings of the routing protocol, then the intermediate node can in behalf of the source node to initiate route discovery for the destination [10].

7 Load balancing simulation and performance evaluation

7.1 Simulation environments

In order to evaluate the performance of the proposed protocol, we use the network simulator OPNET Modeler V.14 and make modification to the MRAA. We used more than one case to establish that LBMRAA was an improvement over MRAA. The following is the explanation of the four chosen cases.

(1) Number of 50 mobile nodes positioned randomly in an area of 2500×2500 meters and running the same MANET routing protocol with default parameters but the simulation was run for 30 minutes, FTP server is only configured on 5 nodes chosen randomly and all stations configured as clients to request large file of size 10 MB and is allowed to make concurrent connections unlimitedly during the whole simulation and the inter-request time of requests is exponential with mean value of 100 s. FTP clients are starting requests randomly from the start of simulation to the end of simulation with unlimited repeats during the whole simulation with interpretation time of uniform distribution between 10 and 20 s. As a mobility model, we use the random waypoint.

(2) The same model exactly as above but with Number of 100 nodes and in an area of 5000×5000 meters and running the same MANET Routing Protocol with default parameters but the simulation was run for 10 minutes. FTP server is only configured on 10 nodes chosen randomly.

(3) Number of 50 mobile nodes positioned randomly in an area of 2500×2500 meters and running the same MANET routing protocol with default parameters but the simulation was run for 30 minutes. The stations are configured to send 8 Kbyte request and receive response of 8 Kbyte with inter-request time of requests being 3600 req/h. Stations start requests randomly from the start of simulation to the end of simulation in a full mesh

Protocol name	Route errors	Route requests	Routing traffic (S)	Routing traffic (R)
–	no.	no.	(bits/s)	(bits/s)
MRAA	1407	1830	249,503	843,413
LBMRAA	1241	1790	229,840	828,371

(a)

Protocol name	Retransmission	FTP download	FTP upload	Throughput	Network load
–	pkts	(s)	(s)	(bits/s)	(bits/s)
MRAA	2.1820	16.564	16.231	1,355,120.52	1,538,341.61
LBMRAA	2.1716	18.948	8.334	1,336,367.07	1,504,241.79

(b)

Table 5: Simulation results for case 1. (a) A table showing the average values for route errors, route request, routing traffic sent and routing traffic received. (b) A table showing the number of retransmission attempts, FTP download response time, FTP upload response time, throughput and network load.

Protocol name	Route errors	Route requests	Routing traffic (S)	Routing traffic (R)
–	no.	no.	(bits/s)	(bits/s)
MRAA	504.96	560.45	299,262.60	1,827,778.54
LBMRAA	313.49	432.12	234,486.41	1,639,654.27

(a)

Protocol name	Retransmissions	FTP download	FTP upload	Throughput	Network load
–	pkts	(s)	(s)	(bits/s)	(bits/s)
MRAA	0.00322	28.121	21.82	2,339,836.41	1,063,218.71
LBMRAA	5.160	8.977	0.64720	2,076,483.98	874,158.04

(b)

Table 6: Simulation results for case 2. (a) A table showing the average values for route errors, route request, routing traffic sent and routing traffic received. (b) A table showing the number of retransmission attempts, FTP download response time, FTP upload response time, throughput and network load.

pattern that each station sends to all other stations. As a mobility model, we use the random waypoint.

(4) The same model exactly as above but with Number of 100 nodes and in an area of 5000×5000 meters and running the same MANET Routing Protocol with default parameters but the simulation was run for 15 minutes. The stations are configured to send 16 Kbyte request and receive response of 16 Kbyte with inter-request time of requests being 3600 req/h.

7.2 Performance metrics

Five key performance metrics are evaluated in our experiments:

- throughput: it represents the total number of bits (in bits/s) forwarded from wireless LAN layers to a higher layer which also means “received” throughput at the destinations of the data sources;
- network load: it represents the total number of bits (in bits/s) submitted to wireless LAN from a higher layer which also means the combined sending rate of all data sources;[4]
- normalized routing overhead: packet overhead is the number of routing packets “transmitted” per data packet “delivered” at the destination;
- route errors: this statistic represents the total number of route error packets sent by all nodes in the network. If next hop teachability cannot be confirmed, the node sends back a route error message to all nodes that use that next hop to reach various destinations.

7.3 Simulation results

We can see from the results of Tables 5 and 6 that in the case of 50 mobile stations with very high traffic

[4] Note that without any retransmission, the ratio of throughput and offered load is simply the packet delivery ratio.

Protocol name	Route errors	Route requests	Routing traffic (S)	Routing traffic (R)
–	no.	no.	(bits/s)	(bits/s)
MRAA	605.04	447.49	254,047.64	952,786.17
LBMRAA	63.45	369.64	43,951.51	214,021.92

(a)

Protocol name	Retransmissions	Packet drop	TCP delay	Throughput	Network load
–	pkts	(bits/s)	(s)	(bits/s)	(bits/s)
MRAA	0.9683	139,170.76	73.48	2,100,309.47	1,539,024.08
LBMRAA	1.0948	37,059.72	65.01	349,397.34	216,048.45

(b)

Table 7: Simulation results for case 3. (a) A table showing the average values for route errors, route request, routing traffic sent and routing traffic received. (b) A table showing the number of retransmission attempts, packet drop rate, TCP delay, throughput and network load.

Protocol name	Route errors	Route requests	Routing traffic (S)	Routing traffic (R)
–	no.	no.	(bits/s)	(bits/s)
MRAA	875.38	1,645.30	723,664.64	2,676,637.69
LBMRAA	61.13	1,436.95	164,437.32	689,200.05

(a)

Protocol name	Retransmissions	Buffer overflow	TCP delay	Throughput	Network load
–	pkts	(bits/s)	(s)	(bits/s)	(bits/s)
MRAA	0.9426	284,950.85	78.01	5,424,502.00	3,751,460.25
LBMRAA	1.0554	65,534.16	40.61	1,062,665.34	603,010.68

(b)

Table 8: Simulation results for case 4. (a) A table showing average values for route errors, route request, routing traffic sent and routing traffic received. (b) A table showing number of retransmission attempts, Packet Drop Rate, TCP Delay, throughput and network load.

load and concurrent connections in the network where the stations were allowed to open unlimited sessions during the 30-minute long simulation, or in the other case of 100 mobile stations with very high traffic load and concurrent connections in the network where the stations were allowed to open unlimited sessions during the 10-minute long simulation, LBMRAA outperforms MRAA in the routing overhand, no. of dropped packets, FTP up/download time, throughput, network load and the average delay.

We can see from the results of Tables 7 and 8, that in the case of mobile stations traditional LBMRAA outperforms MRAA in network load, routing overhand, number of packets drop, route errors and the average delay, which leads us to the conclusion of that distributing the requested load and traffic over many routes is of a great advantage and improvement over sending the whole load and traffic over a single route; however this route is the best suitable route between source and destination.

7.4 Results discussion

In order to compare the performance of the load balanced MRAA with MRAA, previous tables show the end-to-end delay, throughput, network load, sent routing traffic, received routing traffic, number of dropped data packets and simulation duration. We noted that the performance gain by load balancing the discovered routes using ant colony optimization becomes more significant. The proposed load balancing technique (LBMRAA) can deliver more packets to the destination in less time and with lower overhead than MRAA because LBMRAA uses all found routes for data delivery which distributes the load over any possible route to destination where the more reliable and suitable routes get more traffic than less reliable and suitable routes. As the number of discovered paths increases, the loss ratio of data packets slightly decreases. Also the performance of LBMRAA is superior in terms of the end-to-end delay and throughput with comparison with the network load in most cases which is due to that the routing decisions are less made because always data is sent over more than one route which decreases possibility of waiting for routing protocol to find another route when the used one gets broken. In addition, we can observe that LBMRAA is superior in terms of both the generated routing overhead and the processing overhead as less routing traffic is generated because of less need to broadcast unnecessary RREQ packets.

8 Conclusions

In this paper, we proposed an alternative ad-hoc routing protocol to use the well-known ant colony optimization technique and also to find multiple disjoint routes between a source node and a destination. We compared the performance of the proposed protocol with that of the AODV routing protocol in terms of end-to-end delay, throughput, network load, sent routing traffic, received routing traffic, number of dropped data packets, simulation duration for both cases when the nodes are stationary and when they are moving. In the conventional AODV, a source node performs a route discovery procedure whenever an existing route is disconnected. In the proposed protocol, however, a source node can send data packets to its corresponding destination through one of backup routes pre-established. Our simulation results show that the proposed protocol yields a better performance than the conventional AODV protocol. In addition, it was able to achieve results near to the results of other more sophisticated protocols such as (OSLR, DSR, GRP) and even outperform them in some performance metrics.

Also, in this paper we proposed a load balancing technique to use the routes found by the well-known ant colony optimization technique which finds multiple disjoint routes between a source node and a destination node. We compared the performance of the proposed load distribution technique with that of the proposed MRAA routing protocol in terms of end-to-end delay, throughput, network load, sent routing traffic, received routing traffic, number of dropped data packets and route errors for different cases and scenarios. In the proposed MRAA, a source node performs a route discovery procedure whenever an existing route is disconnected. In the modified MRAA using load balancing, however, a source node can send data packets to its corresponding destination through all routes pre-established and collected by ant agents. Our simulation results show that the applied load balancing yields better performance than the ordinary MRAA protocol.

9 Future work

In this paper, we have presented an optimization on routing in ad-hoc networks mostly in terms of the network layer based on ant colony optimization algorithm. We have made modifications on a single existing MANET routing protocol, in particular, AODV, and it showed a great performance improvement which could encourage future work to try to apply the same ant colony optimization algorithm on other existing MANET routing protocols.

In our discussion of using ant colony optimization, we have been able to support multiple-route MANET routing; most of the protocols proposed only provide a single route to the destination. However, our modification developed mechanisms to be able to establish more than a source-destination route. So we may find a multi-route to destination which may encourage future work to do some form of load balancing by using all exiting routes and load-balance data packets among them in a round ribbon fashion. which may be required by some real-time applications. Using that form of multi-route load balancing routing is also a promising technique that can be expanded upon for applications other than video.

References

[1] A. M. Abdel-Moniem, M. H. Mohamed, and A. Hedar, *An ant colony optimization algorithm for the mobile ad hoc network routing problem based on AODV protocol*, in IEEE 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010), 2010, 1332–1337.

[2] A. H. Altalhi and G. Richard III, *Load-balanced routing through virtual paths: highly adaptive and efficient routing scheme for ad hoc wireless networks*, in IEEE 23rd International Conference on Performance, Computing, and Communications, 2004, 407–413.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: from Natural to Artificial Intelligence*, Oxford University Press, New York, NY, USA, 1999.

[4] J. Broch, D. B. Johnson, and D. A. Maltz, *The dynamic source routing protocol for mobile ad hoc networks*, http://www.monarch.cs.rice.edu/internet-drafts/draft-ietf-manet-dsr-00.txt, Internet draft, December 1998.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, *A performance comparison of multi-hop wireless ad hoc network routing protocols*, in Proc. 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98), October 1998, 85–97.

[6] G. Di Caro and M. Dorigo, *AntNet: distributed stigmergetic control for communications networks*, Journal of Artificial Intelligence Research, 9 (1998), 317–365.

[7] G. Di Caro, F. Ducatelle, and L. M. Gambardella, *AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks*, Tech. Report IDSIA-27-04-2004, Dalle Molle Institute for Artificial Intelligence (IDSIA), Manno-Lugano, Switzerland, September 2004.

[8] M. Dorigo and G. Di Caro, *The ant colony optimization meta-heuristic*, in New Ideas in Optimization, D. Corne, M. Dorigo, and F. Glover, eds., McGraw-Hill, London, UK, 1999, 11–32.

[9] R. Dube, C. D. Rais, K. Wang, and S. K. Tripathi, *Signal stability-based adaptive routing (SSA) for ad-hoc mobile networks*, IEEE Personal Communications, 4 (1997), 36–45.

[10] M. EffatParvar, M. R. EffatParvar, A. Darehshoorzadeh, M. Zarei, and N. Yazdani, *Load balancing and route stability in mobile ad hoc networks base on AODV protocol*, in International Conference on Electronic Devices, Systems & Applications (ICEDSA 2010), April 2010, 258.

[11] M. Gunes, U. Sorges, and I. Bouazizi, *Ara—the ant-colony based routing algorithm for MANETs*, in Proc. International Conference on Parallel Processing Workshops (ICPPW'02), August 2002, 79–85.

[12] Z. J. Haas and M. R. Pearlman, *The zone routing protocol (ZRP) for ad-hoc networks*, http://tools.ietf.org/html/draft-ietf-manet-zone-zrp-01, Internet draft, August 1998.

[13] H. Hassanein and A. Zhou, *Routing with load balancing in wireless ad hoc networks*, in Proc. 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM'01), 2001, 89–96.

[14] C. Hedrick, *Routing information protocol*, Request for Comments: 1058, http://www.rfc-editor.org/rfc/rfc1058.txt.

[15] Y. Li and H. Man, *Three load metrics for routing in ad hoc networks*, in IEEE 60th Vehicular Technology Conference, vol. 4, September 2004, 2764–2768.

[16] J. Moy, *OSPF Version 2*, Request for comments: 1247, http://www.faqs.org/rfc/rfc1247.txt.

[17] S. Mueller, R. P. Tsang, and D. Ghosal, *Multipath routing in mobile ad hoc networks: issues and challenges*, in Performance Tools and Applications to Networked Systems: Revised Tutorial Lectures, M. C. Calzarossa and E. Gelenbe, eds., vol. 2965 of Lecture Notes in Computer Science: Tutorial, Springer, Berlin, Germany, 2004, 209–234.

[18] S. Murthy and J. J. Garcia-Luna-Aceves, *An efficient routing protocol for wireless networks*, Mobile Networks and Applications, 1 (1996), 183–197.

[19] OPNET Modeler, Modeler Website.

[20] V. D. Park and M. S. Corson, *A highly adaptive distributed routing algorithm for mobile wireless networks*, in Proc. IEEE 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'97), vol. 3, April 1997, 1405–1413.

[21] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, *Ad hoc on-demand distance vector (AODV) routing*, Request for Comments: 3561, http://www.ietf.org/rfc/rfc3561.txt.

[22] C. E. Perkins and P. Bhagwat, *Highly dynamic destination-sequenced distance-vector (DSDV) routing for mobile computers*, in Proc. Conference on Communications Architectures, Protocols and Applications (SIGCOMM'94), September 1994, 234–244.

[23] C. E. Perkins and E. M. Royer, *Ad hoc on demand distance vector (AODV) routing (internet draft)*, http://tools.ietf.org/id/draft-ietf-manet-aodv-01.txt, August 1998.

[24] E. M. Royer and C.-K. Toh, *A review of current routing protocols for ad hoc mobile wireless networks*, IEEE Personal Communications, 6 (1999), 46–55.

[25] J.-H. Song, V. Wong, and V. Leung, *Load-aware on-demand routing (LAOR) protocol for mobile ad hoc networks*, in The 57th IEEE Semiannual Vehicular Technology Conference (VTC'03), vol. 3, April 2003, 1753–1757.

[26] I. Stojmenovic, ed., *Handbook of Wireless Networks and Mobile Computing*, John Wiley & Sons, New York, NY, USA, 2002.

[27] C.-K. Toh, *Associativity-based routing for ad hoc mobile networks*, Wireless Pers Commun, 4 (1997), 103–139.

[28] C.-K. Toh, A. N. Le, and Y. Z. Cho, *Load balanced routing protocols for ad hoc mobile wireless network*, IEEE Communications Magazine, 47 (2009), 78–84.