

RESEARCH

Open Access

# An ant colony-based multi objective quality of service routing for mobile *ad hoc* networks

Perumalsamy Deepalakshmi\* and Shanmugasundaram Radhakrishnan

## Abstract

Routing, the act of moving information from a source node to a destination node across any kind of network is one of the major issues in computer network literature. *Ad hoc* wireless networks are increasing in popularity, because of the spread of laptops, sensor devices, personal digital assistants, and other mobile electronic devices. These devices will eventually need to communicate with each other, without an adequate infrastructure to rely on. For mobile *ad hoc* networks, the complexity of routing increases because of its characteristics such as dynamic topology, absence of centralized authority, time varying quality of service (QoS) requirements, etc. The biggest challenge in this kind of networks is to find a path between communication end points satisfying user's QoS requirement in spite of frequent path failures because of node mobility. Recent advances in wireless technology and availability of mobile computing devices with networking capabilities have generated a lot of interest in wireless *ad hoc* networks for QoS-based real-time multimedia applications. In this article, we are proposing AMQR, an ant-based multiobjective on demand QoS routing algorithm for mobile *ad hoc* network which will be highly adaptive, efficient, scalable, and mainly reduces end-to-end delay in high mobility cases.

**Keywords:** MANET, quality of service (QoS) routing, ant colony optimization

## 1. Introduction

Mobile *Ad hoc* NETWORK (MANET) [1] is a collection of wireless mobile nodes, which dynamically form a temporary network, without using any existing network infrastructure or centralized administration making these mobile nodes to dynamically establish routing paths between themselves. In recent years, a large number of MANET routing algorithms have been proposed which deal with dynamic aspects of MANETs using reactive or proactive approach or a combination of both. However, features such as multi-hop relaying, dynamic topology because of the random movement of mobile nodes, absence of centralized authority which is unique to MANET lead to massive overheads for route discovery and maintenance. This fact is further provoked by resource constraints in energy, computational capacities, and restricted bandwidth with variable capacity links.

To address these routing challenges in MANETs, myriad number of approaches is provided in literatures which can be classified as proactive and reactive. The

proactive category [2,3] continuously maintains routing information about every node in the network. The main idea behind this behavior is to have always a path available on which to send an eventual flow of data packets. The reactive category otherwise referred as on-demand [4-6] will find routes whenever required, that is, they establish a multihop path between a pair of nodes only if there are packets to be transmitted. Thus, saving bandwidth and reducing overhead is achieved. Also, reactive algorithms are efficient if routes are used at irregular intervals. All these previous works provide a basic "best effort" routing functionality which is sufficient only for applications where data to be transmitted is just a text file or an email message.

Since the demand for real time and quality of services (QoS) has been increased as the Internet expands, provision of QoS is necessary in addition to basic routing functionality. The goal of QoS provisioning is to achieve a more deterministic network behavior, so that information carried by network can be better delivered and network resources can be better utilized. Lack of central coordination and imprecise state information about network links and flows makes the job tough. Link-specific

\* Correspondence: [deepa79mca@yahoo.co.in](mailto:deepa79mca@yahoo.co.in)

Department of Computer Science and Engineering, Kalasalingam University, Srivilliputtur 626126, Tamil Nadu, India

state information includes bandwidth, delay, delay jitter, loss rate, error rate, stability cost, and distance value of each link whereas flow-specific information includes maximum and minimum bandwidth requirements, maximum delay and maximum delay jitters.

With the growth in potential use of MANETs, a lot of research is being focused on providing QoS [7-15]. For the current Internet, there are two different models to obtain a QoS guarantee, namely, Integrated Services (IntServ) [9] and Differentiated Services (DiffServ) [16]. Using resource reservation protocol, RSVP [17,18] IntServ carries QoS parameters from senders to receivers to make resource reservations along a path and hence provides a rich end-to-end QoS solution, by the way of end-to-end signaling, state-maintenance for each RSVP-flow as well as reservation and admission control at each network element. As an alternate, DiffServ works on a service level agreement between providers and users. All packets from a user carry service levels and are treated accordingly. Multiple flows in DiffServ model are mapped to a single service level and state information about every flow need not be maintained along the paths. The IntServ-based model on per-flow resource reservation is not particularly suitable for MANETs because of its frequently changing topology and limited resources. This results more signaling overhead and unaffordable storage and computing power for mobile nodes. However, DiffServ-based approach is a lightweight model using a relative-priority scheme to soften the requirements of hard QoS models like IntServ. Since service differentiation is based on per-hop behaviors no flow states need to be maintained and making DiffServ as a potential QoS model in MANETs.

The role of a QoS routing strategy is to compute paths that are suitable for different types of traffics generated by various applications while maximizing the utilizations of network resources. Since QoS routing is a multiobjective problem with high computational complexity, heuristics algorithms can be used for QoS routing to reduce this complexity but at the expense of not attaining optimal solution finding just a feasible solution.

Recently swarm intelligence has been offered as a novel computational approach that replaces traditional emphasis on control, preprogramming and centralization, with designs featuring autonomy, emergence, and distributed functioning. Swarm Intelligence systems [19] are typically made up of a population of simple agents interacting locally with one another and also with their environment. Usually, there is no centralized control structure dictating how individual agents should behave, but local interactions between such agents often lead to emergence of a global behavior. Examples of systems like this can be found in nature, including ant colonies,

bird flocking, bee swarming, animal herding, bacteria molding, and fish schooling.

The ant colony optimization (ACO) metaheuristic [20] is based on generic problem representation and definition of ant's food foraging behavior. The ACO metaheuristic has successfully been applied to a number of different combinatorial optimization problems such as Vehicle Routing Problem [21], traveling salesman [22], and routing in communication networks such as AntNet [23], Mobile Ants Based Routing (MABR) algorithm [24], ARA [25], ARAMA [26], and AntHocNet [27].

In this article, we have attempted to design a QoS-based routing algorithm for mobile *ad hoc* networks using ACO metaheuristic. The rest of the article is organized as follows. In Section 2, we discuss about basics of ACO and its suitability to network routing. In Section 3, similar works related to proposed work are discussed. The proposed work, AMQR, is discussed in Section 4 and subsequently simulation and results are discussed in Section 5.

## 2. Ant colony optimization

A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. ACO has been formalized into a metaheuristic for combinatorial optimization problems by Dorigo et al. [28,29]. ACO takes inspiration from the foraging behavior of ant species which deposit pheromone on ground in order to mark some favorable paths that should be followed by other members of the colony. ACO exploits a similar mechanism for solving optimization problems. An individual ant is a simple insect with limited memory and capable of performing simple actions. Though a single ant has no global knowledge about the task it is performing and its actions are based on local decisions and are usually unpredictable, an ant colony expresses a complex collective behavior providing intelligent solutions to problems such as carrying large items, forming bridges and finding shortest routes from nest to a food source. This intelligent behavior naturally emerges as a consequence of self-organization and indirect communication among ants which can be termed as emergent behavior or swarm intelligence.

In basic ACO, after initialization, the metaheuristic iterates over three phases, namely, ConstructAntSolutions, ApplyLocalSearch, and UpdatePheromones. A number of solutions are constructed by ants which are optionally improved through a local search and finally the pheromone is updated. Given this formulation, ants in ACO build solutions to the problem being solved by moving concurrently and asynchronously on an appropriately defined construction graph which represents any given problem domain. It defines the way of solution construction, pheromone updating, and possible

interactions in the solution process. ACO's constructive population based metaheuristic [28,29] which exploits an indirect form of memory of prior performance makes it a unique approach to solve optimization problems.

Mobile *ad hoc* network routing is a difficult problem because network characteristics such as traffic load and network topology may vary stochastically and in a time-varying nature. The distributed nature of network routing is well matched by the multiagent nature of ACO algorithms. The set of core properties [20] that characterizes ACO instances for routing problems provides traffic-adaptive and multipath routing, relying on both passive and active information monitoring and gathering, making use of stochastic components, not allowing local estimates to have global impact, setting up paths in a less selfish way than in pure shortest path schemes favoring load balancing and showing limited sensitivity to parameter settings.

### 3. Related works

A network or service provider can offer different kinds of services [30] to users which can be categorized a set of measurable prespecified service requirements such as minimum bandwidth, maximum delay, maximum delay variance (jitter), and maximum packet loss rate. After accepting a service request from user, network has to ensure that service requirements of user's flow are met, as per the agreement, throughout the duration of flow.

As discussed in [31], several researchers [32-37] addressed general problem of QoS in MANETs, providing overviews and insights on the work being done in this area. To achieve QoS provisioning, QoS routing algorithms normally integrate QoS provisioning into routing protocols. The QoS version of AODV (QoS-AODV) [37], Core-Extraction Distributed Ad Hoc Routing (CEDAR) protocol [38], and Multimedia Support for Mobile Wireless Networks (MMWN) protocol [39] are some of the examples of QoS routing algorithms proposed for MANETs. However, QoS signaling techniques are independent of underlying routing protocols. The In-band signaling for QoS in Ad-Hoc Mobile Networks (INSIGNIA) algorithm [40] is a typical signaling protocol designed exclusively for MANETs. The idea of CEDAR and MMWN is to broadcast link-state information across the network in order to enable other nodes to find routes that meet QoS criteria, like minimum bandwidth. On the other hand, INSIGNIA piggybacks resource reservations onto data packets, which can be modified by intermediate nodes to inform the communication endpoint nodes in case of lack of resources. These approaches are based on the idea that wireless links between mobile nodes have certain QoS-related properties, in particular a known amount of available

bandwidth, and that nodes are able to give guarantees for traffic traversing these links.

### 4. Proposed algorithm

The proposed algorithm ant-based multi-objective QoS routing, AMQR inspired by ant food foraging intelligence, is an on-demand QoS routing algorithm for MANETs to face requirements of various applications that may vary time to time. The proposed approach has two phases, namely, route exploration and route maintenance phases. When a source node has data for passing to a destination node with QoS requirements, it starts with route discovery phase. Data transfer will take place once the route is found. While data transmission is going on, it is also required to maintain paths during route maintenance phase which is desirable and required in mobile *ad hoc* networks.

The proposed algorithm incorporates positive feedback, negative feedback, and randomness into routing computation. Ant-like packets, analogous to ant foragers, are used to locally find new paths. Artificial pheromone is laid on communication links between adjacent nodes and route reply and data packets are inclined toward strong pheromone, whereas next hop is chosen probabilistically. Positive feedback is initiated from destination nodes to reinforce existing pheromone on recently learned good paths. To prevent old routing solutions from remaining in the current network status, exponential pheromone decay is adopted as negative feedback.

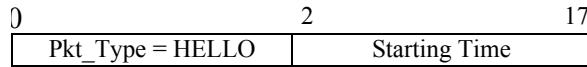
Each node running this algorithm contains three tables, namely, neighbor, path preference, and routing. In neighbor table, each neighbor is listed along with pheromone substance indicating goodness of outgoing link to various destinations and available bandwidth of outgoing link from that neighbor. In routing table, desired destinations for which source node has data will be listed with best next hop. While in path preference table, each entry for a destination is associated with a list of neighbor nodes. A probability value in the list expresses goodness of a neighbor node as next hop on the path to destination. The neighbor node which has a higher path preference value will be copied to routing table for the related destination on desire.

#### 4.1. Message formats

To implement proposed algorithm AMQR, following four control messages are used.

##### 4.1.1. Hello message

The hello packet shown in Figure 1 consists of its starting time at current node. When it is broadcast, neighbors who receive it will advertise hello packet receiving time back to the actual sender. Based on starting time



**Figure 1** Hello packet format.

and receiving time of hello packet and also size of hello packet, current node will calculate available bandwidth of outgoing links. For each node from which hello packet receiving time has been reported, an entry is created in neighbor table along with calculated available bandwidth and initial pheromone of the neighborhood links. For subsequent hello messages, available bandwidth as well as pheromone will be updated to indicate current status of outgoing links.

**4.1.2. Route request message**

The route request message, RREQ\_ANT of AMQR, is shown in Figure 2. In addition to exploring shortest path from source to destination, RREQ\_ANT additionally senses the network environment-related factors like nodes visited, end-to-end delay from source to destination, and available bandwidth of the path through which it is propagated.

**4.1.3. Route reply message**

The route reply message, RREP\_ANT of AMQR, is shown in Figure 3. From Req\_Starting\_Time and RREQ\_ANT's arrival time, end-to-end delay experienced by RREQ\_ANT is found and converted as the parameter, delay. RREP\_ANT will bring route availability information in addition to parameters sensed by RREQ\_ANT through the same path taken by RREQ\_ANT in reverse direction.

**4.1.4. Route error message**

Whenever a node learns that it cannot reach a particular destination for which it has an entry in its routing table, the node updates its routing table and sends route error message, RERR in the format as shown in Figure 4. Upon receiving this message, intermediate nodes will update their routing table and path preference probability table for the unreachable destination.

**4.2. Mathematical model**

The objective function of proposed work is to find a path from source to destination through a neighbor

with maximum path preference probability. As in [41], path preference probability from source  $i$  to destination  $d$  through  $i$ 's neighbor  $j$  is calculated as

$$P_{ijd} = \frac{[\tau_{ij}]^\alpha [D_{ijd}]^\beta [\eta_{ijd}]^\gamma [B_{ijd}]^\delta}{\sum_{l \in N_i} [\tau_{il}]^\alpha [D_{ild}]^\beta [\eta_{ild}]^\gamma [B_{ild}]^\delta} \quad (1)$$

where  $N_i$  is a set of neighbor nodes of  $i$ ,  $\tau_{ij}$  relative weight of pheromone trail on link  $(i, j)$ ,  $D_{ijd}$  is relative metric for delay on the path from  $i$  to  $d$  through  $j$ ,  $\eta_{ijd}$  Relative metric for number of nodes on the path from  $i$  to  $d$  through  $j$ ,  $B_{ijd}$  available bandwidth of the path from  $i$  to  $d$  through  $j$ ,  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  tunable parameters which represent the importance of pheromone decay, delay, number of hops and available bandwidth on the path from  $i$  to  $d$  through  $j$ .

**4.2.2. Calculation of relative metrics**

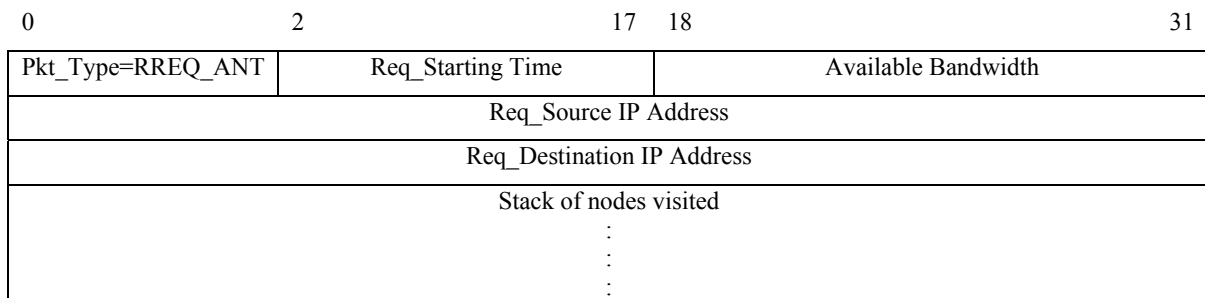
For calculating relative metrics, the additive metrics delay, hop count, and non-additive concave metric bandwidth are considered. Since additive metrics have to be minimized for shortest paths, reciprocal values are used while calculating relative metrics. Owing to the desire of maximizing bandwidth, it is considered as in (1).

**(i) Delay** The delay between source and destination is considered as

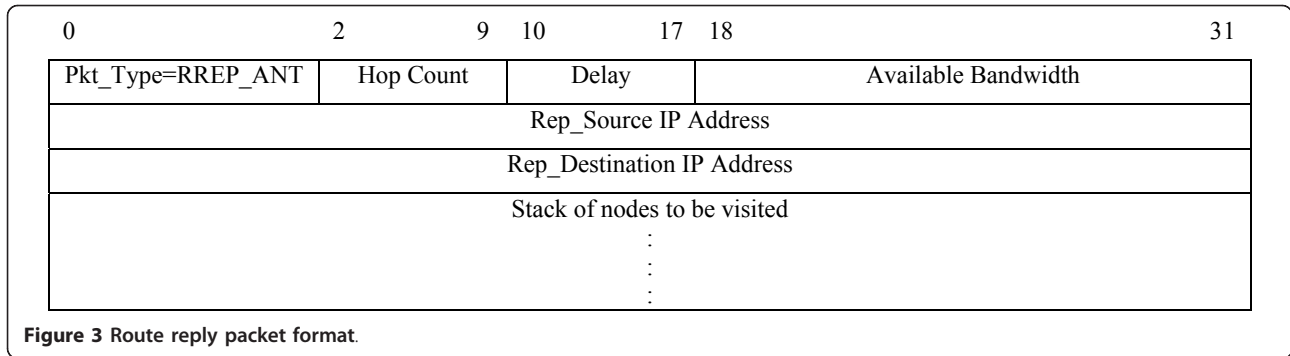
$$D_{ijd} = \frac{1}{\text{delay}(\text{path}_j(i, d))} \quad (2)$$

where  $\text{delay}(\text{path}_j(i, d))$  is experienced end-to-end delay from source  $i$  to destination  $d$  through neighbor  $j$  by route request message at the time of route exploration.

**(ii) Hop count** The measure hop count indicates the number of nodes visited by route request message from source to destination. A destination node finds this from route request message it receives in which first



**Figure 2** Route request packet format.



96 bits indicate source, destination identity, and QoS measure information whereas remaining bits contain list of nodes' IP addresses through which request message has traveled. Hop count can be measured as

$$\text{Hopcount} = \frac{\text{size of (Received RREQ\_ANT)} - 96}{32} \quad (3)$$

The relative metric for number of intermediate nodes between source and destination is found as

$$\eta_{ijd} = \frac{1}{\text{Hopcount}(\text{path}_j(i, d))} \quad (4)$$

where  $\text{Hopcount}(\text{path}_j(i, d))$  is the number of hops seen by route request message along the path  $i$  to  $d$  through  $j$  and found using (3).

**(iii) Bandwidth** The available bandwidth of the path from  $i$  to  $d$  is calculated as minimum of available bandwidth of all links along that path.

$$B_{ijd} = \min \{ \text{available\_bandwidth}(l) \} \quad \forall l \in \text{path}_j(i, d) \quad (5)$$

whereas available bandwidth of a link is calculated as

$$\text{available\_bandwidth}(\text{link}) = \frac{\text{HPS}}{\text{HPST} - \text{HPRT}} \quad (6)$$

where HPS is hello packet size, HPST is hello packet starting time, and HPRT is hello packet receiving time. Because hello messages are frequently transmitted to keep neighborhood alive connectivity, they can better reflect current available bandwidth of outgoing links rather than route exploration messages.

**(iv) Pheromone** Initially when there is no neighborhood relation between  $i$  and  $j$ , pheromone on link  $(i, j)$  is made as  $\tau_{ij} = 0.0$ . When  $j$  is detected as neighbor of  $i$  through hello message, an initial pheromone is deposited as  $\tau_{ij} = 0.1$ . Whenever a route reply message is received from  $j$  to  $i$ , it is considered that link  $(i, j)$  contributes to a possible path from source  $i$  to destination  $d$ . So it is positively reinforced as

$$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij} \quad (7)$$

where  $\Delta\tau_{ij} = 0.05$ .

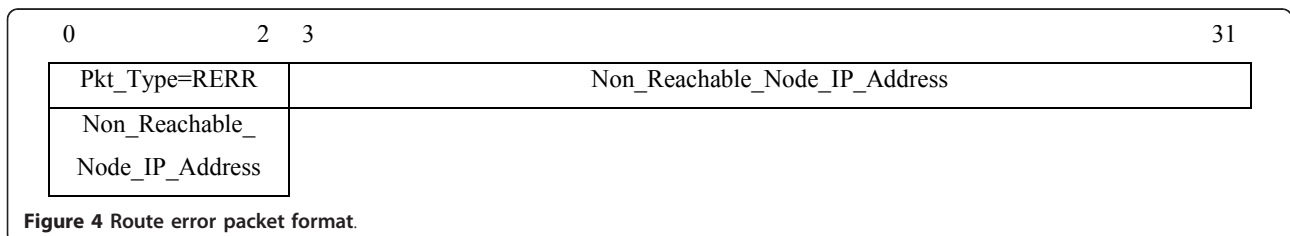
Though link  $(i, j)$  contributes for a possible path from  $i$  to  $d$ , if no data transmission is detected on that path, it is considered as a path with insufficient QoS requirements. In such case, pheromone on link  $(i, j)$  is decreased by a factor  $\rho$  as follows.

$$\tau_{ij} = \begin{cases} 1 & \text{if} \\ (1 - \rho)\tau_{ij} & \text{if} \\ 0.1 & \text{if } i \& j \text{ are still neighbors} \end{cases} \quad \forall i \in V, j \in N_i \quad (8)$$

where  $V$  is set of nodes in mobile network and  $N_i$  is set of neighbors of  $i$ .

### 4.3 Route exploration

When source node  $i$  has data to forward to a destination  $d$ , it searches its routing table for next best hop to reach destination. Since network follows on-demand routing approach, off the rack route will not be available in all cases. So node  $i$  initiates a route request message RREQ\_ANT through all its neighbors about which it has learned from periodic hello messages. While traveling to



destination, RREQ\_ANT checks available capacity of each link, identity of hops it has visited. If available capacity of link visited is lower than that of in RREQ\_ANT message, the available bandwidth field in RREQ\_ANT message is updated by recently visited link's capacity. This will make request message to carry minimum available bandwidth of a link along the path it has traveled. Finally, when RRREQ\_ANT reaches the destination, it will be converted as route reply message called RREP\_ANT. The RREP\_ANT will take same path of corresponding RREQ\_ANT, but in reverse direction. For this, RREP\_ANT replicates and converts stack of nodes visited by RREQ\_ANT as stack of nodes to be visited. At every node from RREP\_ANT's starting point, stack is popped to see next hop to forward RREP\_ANT. At intermediate nodes and at source  $i$ , information coming along with RREP\_ANT such as delay, bandwidth, and hop count are used to calculate path preference probability to reach destination. Source node  $i$  updates path preference probability to destination  $d$  through its entire neighbors provided, it has received RREP\_ANT through these neighbors. The neighbor node which contributes a higher path preference probability over all neighbors of node  $i$  is selected as best next hop to reach destination  $d$ . Once routing table is updated with best next hop for desired destination  $d$ , data transmission starts through that best next hop.

#### 4.4. Route maintenance

When data transmission is going on, paths are reinforced positively making it more desirable for further selection. Also when session is going on, load on selected paths may increase causing more delay and less available bandwidth which can be detected using periodic hello and route request messages. Also, nodes might have moved causing link failures. In such cases, path preference probability will automatically decrease and hence alternate routes can be used which are found during route discovery phase. In our proposed scheme, each node maintains an updated view of its immediate neighbors by periodically sensing neighbors. Hence, link failures can be detected as early as possible before they can lead to heavy transmission errors and subsequent packet loss. The presence of neighbor is envisioned by periodic hello message exchanges or sending and receiving unicast data messages. The desertion of neighbor node is detected when neighbor's presence is not sensed by hello messages for a predefined amount of period. When a neighbor desertion is detected, the node removes lost neighbor from its neighbor table, path preference probability table and also from routing table according to necessity. If data transmission was taking place through that failed link, alternate route with next

best updated path preference probability will be selected for further transmission. In worst case, all available routes to destination might be lost and reinitialization of route exploration phase might be required.

## 5. Simulation and result analysis

### 5.1. Experimental setup

Performance of proposed algorithm, AMQR, existing *ad hoc* on demand distance vector algorithm, AODV, and AntHocNet [29] are tested in a number of simulated scenarios. Diverse simulation scenarios have been created by varying number of nodes, number of constant bit rate (CBR) sessions in the entire network referred as flow count, and amount of pause time. Implementation of proposed algorithm, AMQR, is done in C++ and checked with NS-2.34 TCL simulation scripts [42]. In all scenarios, nodes are placed randomly in a 1500 m × 1500 m region. The maximum transmit range of each mobile node is 250 m. Normally, nodes move according to random way mobility model with a speed in the range [0,10] m/s and for certain scenarios, node mobility is varied from [10, 80] m/s to find the effect of node mobility in latency and throughput. All nodes move toward a new destination position in the region and stay there for a specified time called pause time and again proceed toward a new direction as previously described. The channel capacity is 2 Mbps and at MAC layer IEEE 802.11 DCF is used. Each simulation is run for 900 s each time. The data traffic was generated by a number of CBR sources with repeated sources and repeated destinations. Various timers used in AMQR for periodic control messages and tunable parameter values that can control the relative weight of pheromone trail are specified in Table 1.

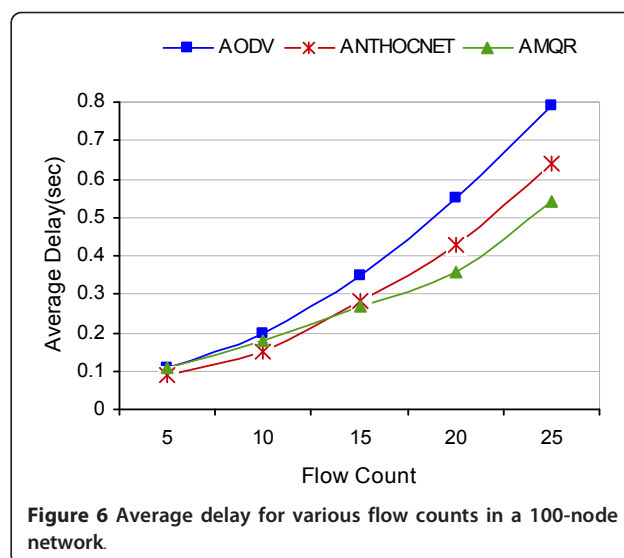
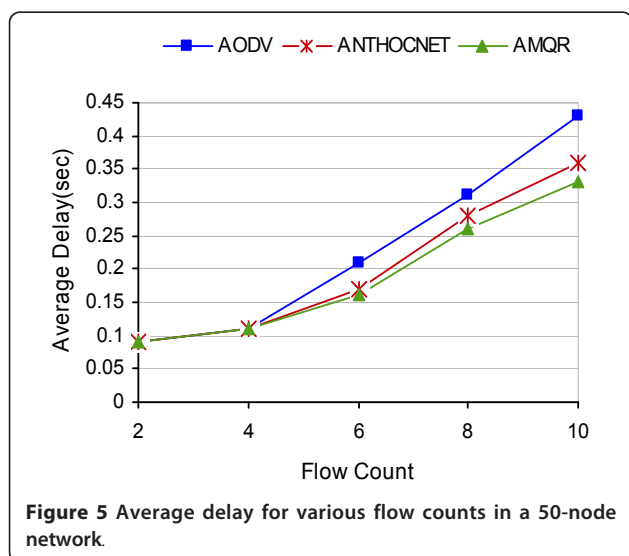
The value for relative weight parameters  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  is kept as 1.0 for the sake of initial study. Later values for these parameters are refined to check their influence on path preference probability.

**Table 1 AMQR Parameter settings**

Parameters	Values
Pheromone decay factor, $\rho$	0.05
Pheromone positive reinforcement, $\Delta\tau_{ij}$	0.05
Pheromone decay weight, $\alpha$	1.0
Delay weight, $\beta$	1.0
Shortest hop weight, $\gamma$	1.0
Bandwidth weight, $\delta$	1.0
Hello Interval	1 sec
Hello Retry times	3
Upper pheromone bound	1.0
Lower pheromone bound	0.1

### 5.2 Result analysis

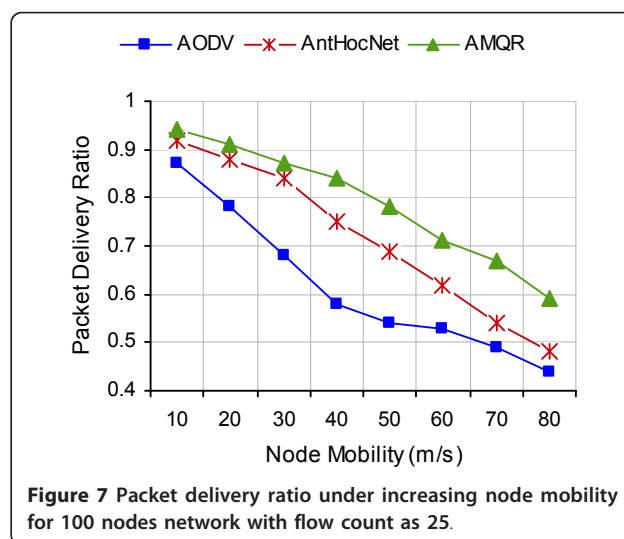
Figures 5 and 6 show the effect of end-to-end average delay with various flow counts for 50 and 100 nodes scenario. In case of AODV, there are possibilities that some links might be shared by more than one shortest path. This will increase traffic on selected links which may lead to congestion and hence data packets transmitted through these links may face additional delay. The situation is further aggravated if many traffic connections start sharing these links. In case of AMQR, a path is considered as good not only based on number of hops, but also on other parameters like available bandwidth along the links on that path. So, when a link contributes itself on a shortest path of a particular traffic, available bandwidth of that link will get reduced. Since bandwidth is a kind of property to be maximized, this leads to a reduction in path preference probability of the paths which have this link as one of their intermediate links. This will lead to the selection of other possible paths which may have higher path preference probability. Though alternate paths may not be shortest in number of hops, they may have less traffic. Hence, data may be transmitted as they are received and not making them to wait in the queues. This can be realized well when number of flow count is increasing. AntHocNet considers pheromone, prespecified delay bound at receiver and number of hops to determine path probability and randomly select a path for data transmission. This offers good load balance in the entire network and may not offer shorter delay and wider bandwidth paths for a specific data flow. In all our experiments, we have seen that AntHocNet has close performance with proposed work, AMQR. From Figures 5 and 6, it is seen that proposed the algorithm has lesser delay especially in



increased flow counts compared to AODV and AntHocNet.

Also, in case of high number of CBR connections, many numbers of links are used by multiple traffic flows and hence probability of occurrence of congestion is high. In AODV, local repair of link breaks in routes some times results in increased path lengths to destinations. This introduces heavy end-to-end delay for data packets. Since AMQR as well as AntHocNet maintains multiple paths to destinations based on current status of network, alternate paths with less traffic can be selected with better path preference. When such paths are selected for data transmission, delay faced will be less than AODV using shortest paths.

Figure 7 shows the packet delivery ratio for 100 nodes network with 25 CBR connections by varying node



mobility. Between AODV and AMQR, performance difference is very much visible at all mobility levels. Also, AMQR performs better than AntHocNet. Figure 8 indicates performance difference between AODV, AntHocNet, and AMQR for average delay with increasing node mobility for the same 100 nodes network with 25 as flow count. AMQR performs well compared to AODV in case of low node mobility because as mobility increases, paths between communication end points will be broken frequently. Owing to subsequent path repairing and new path exploration, average delay is getting increased for all three algorithms. However, AMQR depends upon alternate available paths and in worst case in which all identified paths are broken, it can go for new path exploration mechanism. Figure 9 compares average jitter, the inter packet arrival time for various node mobility. At high node mobility, jitter increases because of path breaks and subsequent alternate path selection or new path identification.

Normally at higher pause time, nodes may stick on to the terrain boundaries for longer time making itself to be invisible for others. This leads to path breaks for longer time causing a reduced amount of packet delivery ratio at higher pause times as shown in Figure 10. But, if number of nodes in network increases and different nodes start moving at different timings, network connectivity can be maintained well. In case of shorter pause time, nodes will not stay in one place and keep on moving frequently. This creates path breaks but connectivity among network nodes will be maintained well. Owing to multiple path maintenance, AMQR and AntHocNet can achieve good packet delivery ratio at shorter pause time. As pause time increases, delivery ratio decreases whereas jitter increases. Average delay variation, jitter, has been analyzed in Figure 11 for

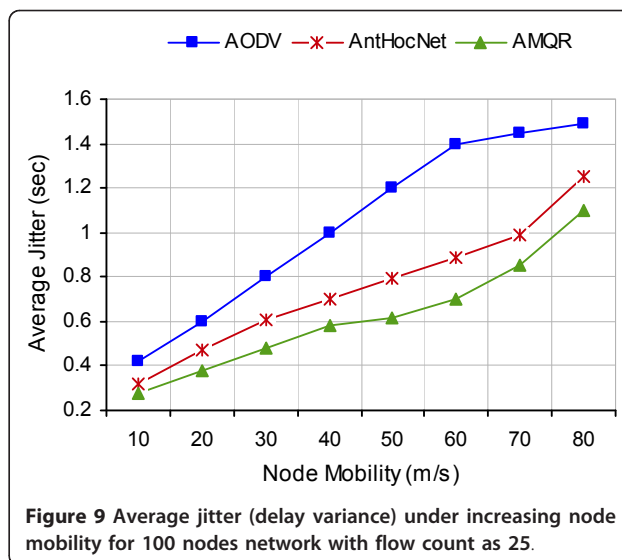


Figure 9 Average jitter (delay variance) under increasing node mobility for 100 nodes network with flow count as 25.

increasing pause time. In AODV, when a link breaks, if possible route repair is done or otherwise new paths are explored. So, in case of path breaks, a time gap is introduced while data are transmitted causing increase in jitter.

The improved performance we have achieved in AMQR is at a cost of routing control overhead compared to AODV and AntHocNet. In AMQR, four kinds of control messages are used to keep neighborhood connectivity, to search for routes using request messages, to reply for routes requests, and to inform route errors. The size of request message will dynamically increase as it moves through network toward destination whereas reply message's size will decrease as it move toward source node. AMQR satisfies QoS requirements and achieves better performance by maintaining multiple

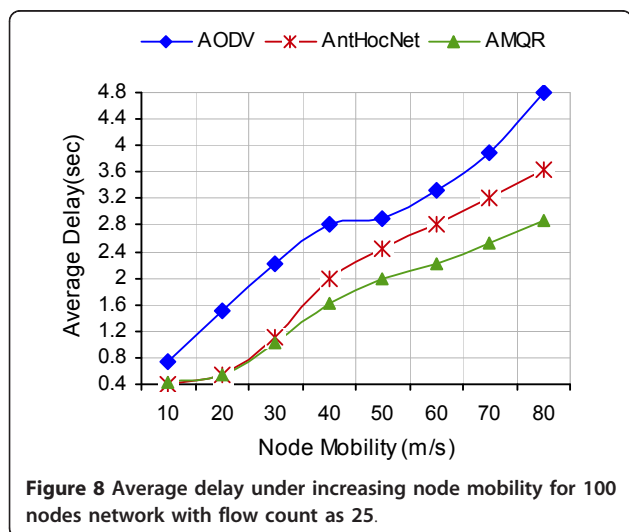


Figure 8 Average delay under increasing node mobility for 100 nodes network with flow count as 25.

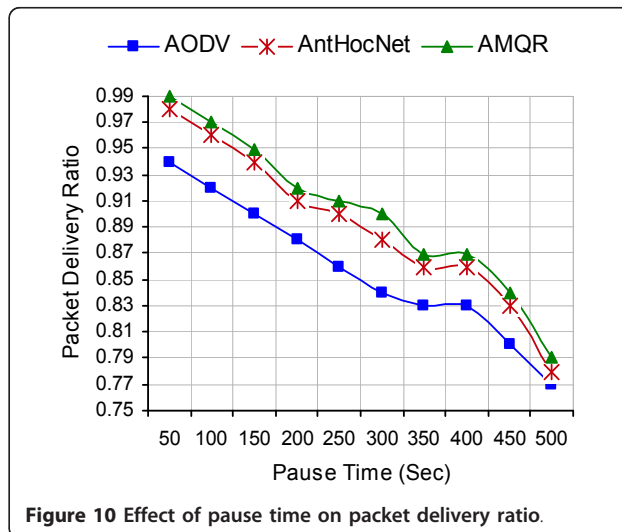
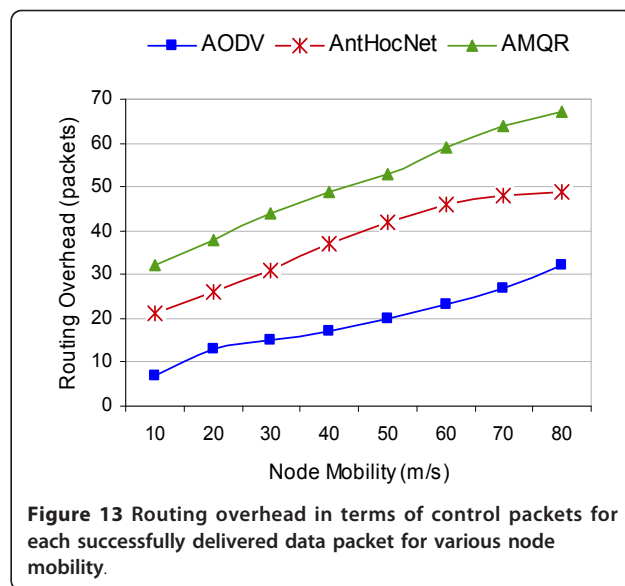
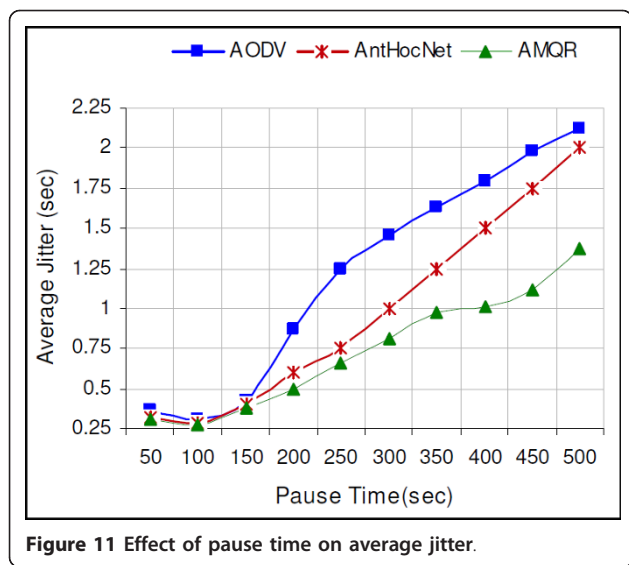


Figure 10 Effect of pause time on packet delivery ratio.



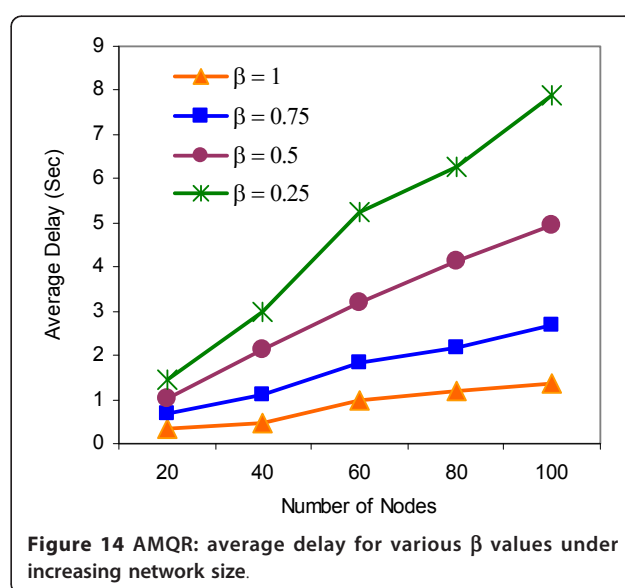
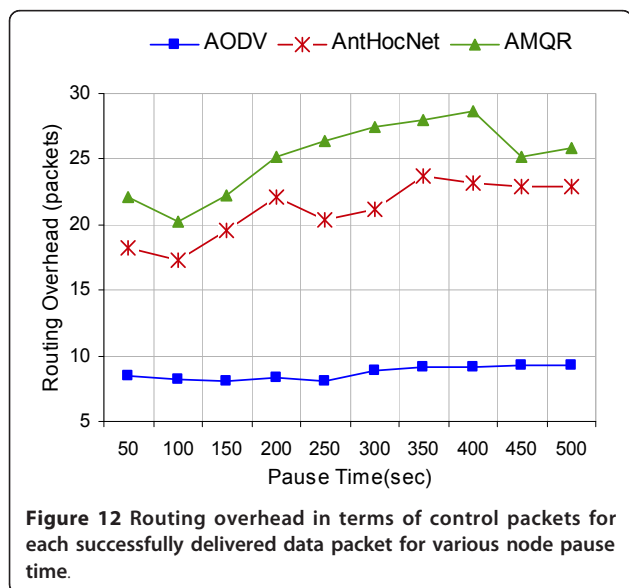


paths to destinations through these dynamically size varying ant messages. Figures 12 and 13 show routing overhead in terms of packets for increasing pause time and increasing node mobility. AODV has least control overhead and AntHocNet has moderate routing overhead. Since proposed algorithm, AMQR, shares nodes visited and delay information through request and reply messages periodically to maintain multiple paths, it has highest control overhead.

In all simulations carried out previously, equal weight is given to the parameters  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  which represent the importance of pheromone decay, delay, number of hops, and available bandwidth, respectively. We have carried additional simulations by varying  $\beta$  and  $\delta$  values to find their effect on path preference probability and

subsequently on delay and packet delivery ratio. To learn the importance of considering delay in case of selecting best paths, value of  $\beta$  alone is varied as 0.25, 0.5, 0.75, and 1 and results are expressed in Figure 14.

It is also observed that as  $\beta$  value is decreasing, average delay is increasing. The reason is, as  $\beta$  decreases, importance to paths with shorter delay are given less preference than paths with higher delays which may have least hop count, less available bandwidth. This may increase path preference probability of paths with higher delays. So when these paths are selected for data transmission, higher delays are introduced. Similarly, to know the importance of considering available bandwidth of paths while calculating path preference probability,



value of  $\delta$  is varied as 1, 0.75, 0.5, and 0.25, and the analysis is shown in Figure 15 in which packet delivery ratio is compared for various values of  $\delta$ . Clearly proposed algorithm performs well for higher values of  $\delta$  and the packet delivery ratio gets decreased as the value of  $\delta$  decreases.

## 6. Conclusion and future work

Mobile *ad hoc* networks grant a wide range of challenges in routing and network management because of their dynamic and distributed natures. Several protocols have been proposed to address the challenges. In this article, we have proposed AMQR, an ant-based routing protocol to meet the QoS requirement of mobile *ad hoc* networks to support real-time traffic. In addition to considering only presence of links or connectivity between nodes, network-related status of links like delay experienced through that links, available bandwidth in links, and a link's suitability for data transmission are also collected and measured. These measures provide an indication of how much a link is suitable while deciding about paths for providing QoS routing. From a given source to destination, multiple paths have been found and their path preference probability is calculated based on the above said measures. Apparently, the path with higher path preference probability offers an optimized consideration of multiple QoS metrics delay, bandwidth, and shortest hop count. Initially, data are sent over the path with higher path preference probability which can provide lesser delay, higher bandwidth, and shorter path in terms of number of hops. In case of any link failure leading to path breaks, alternate possible paths with

next higher path preference probability are immediately considered and data transmission will be continued. The proposed algorithm, AMQR, has been compared with AODV and AntHocNet in terms of delay, throughput, jitter with various flow counts, node mobility, and various pause times. It is also observed that proposed algorithm provides good packet delivery ratio and shorter delay and jitter compared to AODV and AntHocNet in all cases but at cost of increased routing overhead.

Future work may include testing the performance of our algorithm, AMQR for various MANET applications with variety of topologies and varying load situations. Also, values for tunable parameters  $\alpha$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  can be experimented for various applications of *ad hoc* network and also can be optimized.

## Pseudo code

**Algorithm: *recv\_Data\_Packet* (pkt)**

*Input: A data packet 'pkt' is received*

for each mobile node 'S' in the network

{

    route = find\_Route(Route\_Table, pkt->dest);

    if (route is available)

        forward(pkt, route->nexthop);

    else//if route not available

        if (pkt->src == S)//S is sending data

            insert data packet in queue;

            prepare\_Route\_Request (&RREQ\_ANT, pkt->dest);

            forward(RREQ\_ANT, IP\_BROADCAST);

        else//intermediate nodes

            drop(pkt);

            prepare\_Route\_Error\_Msg (RERR, pkt->dest);

            forward\_Route\_Error\_Msg (IP\_BROADCAST);

}

**Algorithm: *recv\_Route\_Request*(RREQ\_ANT)**

*Input: A RREQ\_ANT is received*

for each mobile node 'S' in the network

{

    if(RREQ\_ANT->dest == S)//S is the destination

        prepare\_Route\_Reply(&RREP\_ANT);

        prev\_node = stack\_Top(RREQ\_ANT->nodes\_visited);

        forward(RREP\_ANT, prev\_node);

    else//intermediate node

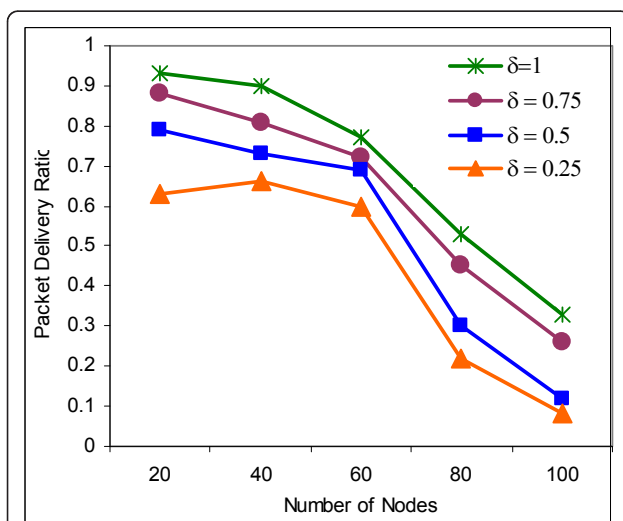
        if (stack\_Find(RREQ\_ANT->nodes\_visited, S))

            drop(RREQ\_ANT);//already seen by S

        else//request received first time

            store\_Reverse\_Route(RREQ\_ANT->src);

            stack\_Push(RREQ\_ANT->nodes\_visited, S);



**Figure 15** Packet delivery ratio for various  $\delta$  values under increasing network size.

```
update_Bandwidth(RREQ_ANT- > bandwidth);//  
forward(RREQ_ANT, IP_BROADCAST);  
  
}
```

**Algorithm: rcv\_Route\_Reply(RREP\_ANT)**

**Input: A RREP\_ANT is received**

for each mobile node 'S' in the network

```
{  
  
neighbor_hop = immediate node from which reply is  
received  
if(RREP_ANT- > dest == S)//reply comes to source  
update_pppTable(RREP_ANT- > src, neighbor_hop);  
update_Rtable(RREP_ANT- > src);  
while ((pkt = queue_Find(RREP_ANT- > src)) !=  
NULL)  
//data packets are waiting in queue to reply source  
forward(pkt, RREP_ANT- > src);  
else//intermediate nodes  
next_hop = pop_Stack(RREP_ANT- >  
nodes_to_be_visited)  
forward(RREP_ANT, next_hop);  
update_pppTable(RREP- > src, neighbor_hop);  
update_Rtable(RREP_ANT- > src);  
  
}
```

**Algorithm: rcv\_Hello\_notification ()**

**Input: A Hello notification is received form a neighbor**

for each mobile node 'S' in the network

```
{  
  
neighbor_hop = node from which notification is  
received  
hello_pkt_duration = Hello_rcv_time-  
hello_starting_time);  
available_bandwidth = sizeof(hello_Pkt)/  
hello_pkt_duration;  
update_Neighbor_Table(neighbor_hop,  
available_bandwidth);  
update_PPPtable_Neighbor(neighbor_hop);  
update_Rtable(neighbor_hop);  
  
}
```

**Algorithm: Hello\_Waiting\_Timer is expired for Hello\_Retry Times**

**Input: A neighbor, n's move is detected**

for each mobile node 'S' in the network

```
{  
  
remove_Neighbor_Table(n);  
update_PPPtable_Neighbor_Move(n);
```

```
update_Rtable(n);  
  
}
```

**Competing interests**

The authors declare that they have no competing interests.

Received: 13 January 2011 Accepted: 2 November 2011

Published: 2 November 2011

**References**

1. J Macker, S Corson, Mobile Ad Hoc Networks (MANET). *IETF WG Charter*. <http://www.ietf.org/dyn/wg/charter/manet-charter.html> (1997)
2. P Jacquet, P Muhlethaler, T Clausen, A Laouiti, A Qayyum, L Viennot, Optimized link state routing protocol for ad hoc networks, in *Proceedings of the 5th IEEE International Multi Topic Conference (INMTC 2001)*, Le Chesnay, France (2001)
3. CE Perkins, P Bhagwat, Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers. *Comput Commun Rev.* **10**(4), 234–244 (1994)
4. CE Perkins, EM Royer, SR Das, Ad hoc on demand distance vector (AODV) routing, in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications(WMCSA'99)*, New Orleans, LA, USA (1999)
5. DB Johnson, DA Maltz, Dynamic source routing in Adhoc wireless networks, in *Mobile Computing*, ed. by Imelinsky T, Korth H (Kluwer Academic Publishers, 1996), pp. 153–181
6. VD Park, MS Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Kobe, Japan (1997)
7. S Chakrabarti, A Mishra, QoS issues in ad hoc wireless networks. *IEEE Commun Mag.* **39**(2), 142–148 (2001). doi:10.1109/35.900643
8. P Mohapatra, J Li, C Gui, QoS in mobile ad hoc networks. *IEEE Wirel Commun Mag.* **10**(3), 44–52 (2003). doi:10.1109/MWC.2003.1209595
9. R Braden, D Clark, S Shenker, Integrated services in the internet architecture—an overview, *Request For Comments 1663*, Internet Engineering Task Force (1994)
10. DD Perkins, HD Hughes, A survey on quality-of-service routing in ad hoc networks. *Wirel Commun Mobile Comput.* **17**(8), 503–513 (1999)
11. D Chalmers, M Sloman, A survey of QoS in mobile computing environments. *IEEE Commun Surv.* **2**(2), 2–10 (1999)
12. MS Corson, AT Campbell, Toward supporting quality of service in mobile ad hoc networks, in *Proceedings of the 1st Conference on Open Architecture and Network Programming*, San Francisco, USA (1998)
13. N Nikaein, C Bonnet, A glance at quality of services in mobile ad hoc networks, in *Proceedings of 16th conference of New Architectures for Communications (DNAC'02)*, Paris, France (2002)
14. Y Ge, T Kunz, L Lamont, Quality of service routing in ad hoc networks using OLSR, in *Proceedings of 36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, Hawaii, USA (2003)
15. S Chen, K Nahrstedt, Distributed quality-of-service routing in ad hoc networks. *IEEE J Sel Areas Commun.* **17**(8), 1488–1505 (1999). doi:10.1109/49.780354
16. S Blake, An architecture for differentiated services, *Request for Comments 2475*, Internet Engineering Task Force (1998)
17. R Braden, L Zhang, S Berson, S Herzog, S Jamin, Resource reservation protocol (RSVP)—version 1 functional specification, *Request For Comments 2205*, Internet Engineering Task Force (1997)
18. J Wroclawski, The use of RSVP with IETF integrated services, *Request for Comments (Standard Track) 2210*, Internet Engineering Task Force (1997)
19. E Bonabeau, M Dorigo, G Theraulaz, *Swarm Intelligence: from Natural to Artificial Systems*, (Oxford University Press, Oxford, USA, 1999)
20. M Dorigo, M Birattari, T Stutzle, Ant colony optimization: artificial ants as a computational intelligence technique. *IEEE Comput Intell.* **1**(4), 28–39 (2006)
21. JE Bell, PR McMullen, Ant colony optimization techniques for the vehicle routing problem. *Adv Eng Inf.* **18**(1), 41–48 (2004). doi:10.1016/j.aei.2004.07.001
22. M Dorigo, LM Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput.* **1**(1), 53–66 (1997). doi:10.1109/4235.585892

23. G Di Caro, M Dorigo, AntNet: distributed stigmergetic control for communications networks. *J Artif Intell Res.* **9**, 317–365 (1998)
24. M Heissenbu, T Braun, Ants-based routing in large scale mobile ad hoc networks, in *Proceedings of Kommunikation in Verteilten Systemen(KiVS03)*, Leipzig (2003)
25. M Gunes, U Sorges, I Bouazzi, ARA—the ant-colony based routing algorithm for MANETs, in *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW'02)*, Vancouver, BC (2002)
26. O Hussein, T Saadawi, Ant routing algorithm for mobile ad hoc networks (ARAMA), in *Proceedings of the International Performance Computing and Communications Conference(IPCCC 2004)*, Phoenix, Arizona (2004)
27. G Di Caro, F Ducatelle, LM Gambardella, AntHocNet: an adaptive nature inspired algorithm for routing in mobile ad hoc networks. *Eur Trans Telecommun.* **16**(2), 443–455 (2005)
28. M Dorigo, G Di Caro, The ant colony meta-heuristic, in *New Ideas in Optimization*, ed. by Corne D, Dorigo M, Golver F (McGraw-Hill, 1999), pp. 11–32
29. M Dorigo, G Di Caro, LM Gambardella, Ant algorithms for discrete optimization. *Artif Life* **5**(2), 137–172 (1999). doi:10.1162/106454699568728
30. C Siva Ram Murthy, BS Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols* (Prentice Hall: Upper Saddle River: NJ, 2004)
31. Z Liu, MZ Kwiatkowska, C Constantinou, A biologically inspired QoS routing algorithm for mobile ad hoc networks. *Int J Wirel Mobile Comput.* **4**(2), 64–75 (2010). doi:10.1504/IJWMC.2010.033054
32. P Mohapatra, J Li, C Gui, QoS in mobile ad hoc networks. *IEEE Wirel Commun.* **10**(3), 44–52 (2003). doi:10.1109/MWC.2003.1209595
33. H Zhu, M Li, I Chlamtac, B Prabhakaran, A survey of quality of service in IEEE 802.11 networks. *IEEE Wirel Commun.* **11**(4), 6–14 (2004). doi:10.1109/MWC.2004.1325887
34. T Bheemarjuna Reddy, I Karthigeyan, BS Manoj, C Siva Ram Murthy, Quality of service provisioning in ad hoc wireless networks: a survey of issue and solutions. *Ad Hoc Netw.* **4**(1), 83–124 (2006)
35. L Chen, WB Heinzelman, A survey of routing protocols that support QoS in mobile ad hoc networks. *IEEE Netw.* **21**(6), 30–38 (2007)
36. L Hanzo II, R Tafazolli, A survey of QoS routing solutions for mobile ad hoc networks. *IEEE Commun Surv Tutor.* **9**(2), 50–70 (2007)
37. EM Royer, CE Perkins, Quality of service for ad hoc on demand distance vector (AODV) routing. IETF Internet Draft, draft-ietf-manet-aodvqos-00.txt (2000)
38. R Sivakumar, P Sinha, V Bharghavan, CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE J Sel Areas Commun.* **17**(8), 1454–1465 (1999). doi:10.1109/49.779926
39. R Ramanathan, M Steenstrup, Hierarchically-organized, multihop mobile wireless networks for quality-of-service support. *ACM/Baltzer Mobile Netw Appl.* **3**(1), 101–119 (1998). doi:10.1023/A:1019148009641
40. SJ Lee, GS Ahn, AT Campbell, Improving UDP and TCP performance in mobile ad hoc networks with insignia. *IEEE Commun Mag.* **36**(6), 156–165 (2001)
41. P Deepalakshmi, S Radhakrishnan, Ant colony based QoS routing algorithm for mobile ad hoc networks. *Int J Recent Trends Eng.* **1**(1), 459–462 (2009)
42. FJ Ros, PM Ruiz, Implementing a new MANET unicast routing protocol in ns2, (Technical Report, University of Murcia, 2004)

doi:10.1186/1687-1499-2011-153

**Cite this article as:** Deepalakshmi and Radhakrishnan: An ant colony-based multi objective quality of service routing for mobile ad hoc networks. *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:153.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---