

AMTR: THE ANT BASED QoS AWARE MULTIPATH TEMPORALLY ORDERED ROUTING ALGORITHM FOR MANET'S

Debajit Sensarma¹ and Koushik Majumder¹

¹ Department of Computer Science & Engineering, West Bengal University of Technology, Kolkata, INDIA

debajit.sensarma2008@gmail.com,
koushik@ieee.org

ABSTRACT

Mobile Ad hoc networks (MANETs) are self organized by a collection of mobile nodes, which are interconnected by multi hop wireless paths. Providing QoS support in MANETs is an active research area, basically for supporting real time applications that are based upon interaction between the routing scheme and QoS provisioning. The goal of QoS aware routing is to achieve more deterministic behavior of the network i.e. a stable path, so that information carried by the network can be delivered in an uninterrupted fashion and for better resource utilization. In this paper we have developed AMTR, a new ant based QoS aware on-demand multipath routing algorithm for MANETs with better route failure management and which will be highly adaptive and energy efficient.

KEYWORDS

MANET, Ant Colony Optimization, TORA, QoS Routing.

1. INTRODUCTION

With the growth of internet, the demand for real time and quality of services (QoS) of mobile ad hoc network has increased. So, demand of QoS-aware routing in case of mobile ad hoc networks has also increased. The major objectives of QoS-aware routing are:-i) finding path from source to destination satisfying user's requirement. ii) optimizing network resource usage and iii) repairing or re-computing the path quickly in case of path break or link failure or unwanted things like congestion, without degrading the level of QoS. Again, centralized algorithms have scalability problems, static algorithms have trouble keeping up-to-date with network changes, and other distributed and dynamic algorithms have oscillations and stability problems. Ant based routing [1-4] provides a promising alternative to these approaches. Ant colony provides a number of advantages [5] due to the use of mobile agents and stigmergy (a form of indirect communication used by ants in nature to coordinate their problem-solving activities). It provides scalability, i.e. according to the network size the population of the agents can be adapted. It is fault tolerant, i.e. it does not rely on a centralized control mechanism. Therefore, node mobility or link breakage does not result in catastrophic failure. Besides this, it provides adaptation, where according to the network changes agents can change, die or reproduced. Speed is achieved, because network topology change can be propagated very fast. Agent acts individually, independent of other network layers, so it gives modularity. Also, autonomy is provided because little or no human

supervision is required. It also provides parallelism, i.e. agents operate in parallel. However, besides these advantages, one of the biggest difficulties with ant colony algorithms applied in network routing area is that multiple constraints often make the routing problem intractable [6].

The proposed algorithm is a QoS aware multipath on-demand routing algorithm. The advantage of AMTR is that, it supports energy efficient multipath routing and as well as takes care of the QoS constraints-delay, bandwidth, energy and drain rate which is very essential for real time and multimedia applications.

The paper is organized as follows: Section 2 describes the mathematical model. In section 3 the proposed algorithm combining the idea of ACO and TORA is illustrated. Section 4 explains the performance analysis. Finally section 5 concludes the paper.

2. MATHEMATICAL MODEL

For mathematical analysis MANET is represented by a connected undirected graph. Let $G(V, E)$ represents the mobile ad hoc network. Here V denotes the set of network nodes and E denotes the set of bidirectional links. QoS metrics with respect to each link $e \in E$ is delay (e) and bandwidth (e). With respect to node $n \in V$, it is delay (n), energy (n) and drain rate (n) which is the energy dissipation rate of node 'n'. Another QoS metric considered here is hop count. It is important because multiple hops are used for data transmission in MANET. So, it is necessary to find paths with minimum hops. The main motivation of this proposed algorithm is to find path from source to destination which will satisfy the QoS requirements such as delay, bandwidth, energy, drain rate and hop count.

Let, path (i, j) or R is entire path from node i to j where QoS constraints have to be satisfied.

From an arbitrary node i to an arbitrary node j , delay, bandwidth, energy, drain rate and hop count is calculated as-

$$\text{delay (path (i, j)) or } D(R) = \sum_{e \in P(i, j)} \text{delay}(e) + \sum_{n \in P(i, j)} \text{delay}(n)$$

where, delay (path (i, j)) is the transmission and propagation delay of the path(i, j) and delay (n) is the processing and queuing delay of node 'n' on path(i, j).

$$\text{bandwidth(path(i,j)) or } B(R) = \min_{e \in P(i, j)} \{ \text{bandwidth}(e) \}$$

where, bandwidth (e) is the available bandwidth of that link on path(i, j).

$$\text{energy (path (i, j)) or } E(R) = \min_{n \in P(i, j)} \{ \text{energy}(n) \}$$

where, energy (n) is the residual energy of node 'n' on path(i, j).

$$\text{drain rate (path (i, j)) or } DR(R) = \max_{n \in P(i, j)} \{ \text{drain rate}(n) \}$$

where, drain rate (n) is the rate of energy dissipation of node 'n' on path(i, j).

hop count (path (i, j)) or $HC(R) =$ Number of nodes in the path.

2.1 CALCULATION OF PHEROMONE

Ant deposits pheromone during traversal of the link for finding a route. The quantity of pheromone it deposited on each link (i, j) along the route R is noted by $\Delta\tau_{i, j}$ and it is a function of global quality of route R . It is expressed by the following equation-

$$\Delta\tau_{i, j} = \frac{B(R)^{\lambda_B} + E(R)^{\lambda_E}}{D(R)^{\lambda_D} + HC(R)^{\lambda_{HC}} + DR(R)^{\lambda_{DR}}} \quad (1)$$

Here $\lambda_B, \lambda_E, \lambda_D, \lambda_{HC}$ and λ_{DR} are the weight factors which indicate the relative significance of the QoS parameters during pheromone update on path (i, j). The quantity of the deposited pheromone is defined only after finding the route.

The pheromone quantity of the link (i, j) is updated according to the following equation:

$$\tau_{i,j} = \sigma \cdot \tau_{i,j} + \Delta \tau_{i,j} \quad (2)$$

2.2 CALCULATION OF PATH PREFERENCE PROBABILITY

Path Preference Probability is calculated in each intermediate node as well as source node upon receiving of QRY Reply_Ant.

Suppose current node i receives QRY Reply_Ant from node j for destination d, then the Path Preference Probability is calculated as-

$$P_{ij,d} = \frac{[\tau_{ij}]^{\alpha_1} \cdot [D_{j,d}]^{\alpha_2} \cdot [\eta_{j,d}]^{\alpha_3} \cdot [B_{j,d}]^{\alpha_4} \cdot [E_{j,d}]^{\alpha_5} \cdot [DR_{j,d}]^{\alpha_6}}{\sum_{k \in N_i} [\tau_{ik}]^{\alpha_1} \cdot [D_{k,d}]^{\alpha_2} \cdot [\eta_{k,d}]^{\alpha_3} \cdot [B_{k,d}]^{\alpha_4} \cdot [E_{k,d}]^{\alpha_5} \cdot [DR_{k,d}]^{\alpha_6}} \quad (3)$$

Here $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5$ and α_6 are the tunable parameters which control the relative weights of pheromone trails, hop count, bandwidth, energy and drain rate respectively.

N_i is the set of neighbors of i and k is the neighbor node of i through which a path to destination is known.

The relative metrics are calculated from source i to destination d via j as-

$$D_{ij,d} = \frac{1}{\text{delay}(\text{path}(i, d))}$$

$$\eta_{ij,d} = \frac{1}{\text{hopcount}(\text{path}(i, d))}$$

$$B_{ij,d} = \text{bandwidth}(\text{path}(i, d))$$

$$E_{ij,d} = \text{energy}(\text{path}(i, d))$$

$$DR_{ij,d} = \frac{1}{\text{drain rate}(\text{path}(i, d))}$$

Now, source as well as neighbors has multiple paths from source to destination. The path with higher Path Preference Probability is selected for the data transmission.

3. PROPOSED ALGORITHM

The proposed algorithm is a multipath on-demand routing algorithm. Here QoS constraints-delay, bandwidth, energy and drain rate are considered which are very essential for multimedia applications.

This algorithm has three phases namely route discovery phase, route maintenance phase and route erasure phase. In route discovery phase, multiple paths which satisfy the required QoS constraints are created and stored in the cache. The route with higher path preference probability is selected for the routing. In route maintenance phase, when a node fails to transmit packets, it will check for the alternate route to the desired destination with better path preference probability. If such path exists, it is chosen for the routing. In the next case, if the failure node has no outbound link to other nodes in the network, then it will send an error message to the source and

immediately start local link maintenance. All routes going through the failure node are deleted from the cache. In the next step, when source finds the error message, it will check for other alternate route in the cache with next better path preference probability. If it exists, that route gets selected for routing. Otherwise, a new route discovery phase is started. If network partition is detected during route maintenance phase, then route erasure phase is invoked and all invalid routes are deleted from cache.

Each node $k \in N$ has a height represented by as a quintuple: $H_k = (\tau_k, oid_k, r_k, \delta_k, k)$, where i) τ_k : A time tag indicating the time of link failure represents the reference level. ii) oid_k : Originator-id, id of node that defined the reference level. iii) r_k : 1 bit used to divide each reference level into 2 sub-levels. iv) δ_k : Integer used to order nodes with respect to a unique reference level and v) k : Unique identifier of the node. Each node k (other than the destination) maintains a link-state array with an entry $LS_{k,j}$ for each link $(k, j) \in L$, where j is a neighbor of k , L is total number of links. The height of node k , H_k and that of its neighbor j , $H_{N_{k,j}}$ determines the direction of the links, and is directed from the higher node to the lower node. RR_k is route-required flag of node k .

3.1. ROUTE DISCOVERY PHASE

Algorithm 1: Route Discovery

BEGIN

Step 1: Let, Source(S) wants to communicate with the destination (D) with the QoS constraints bandwidth, delay, drain rate, energy. No path exists in the cache previously.

Step 2: S sets its height as NULL initially i.e $(-, -, -, -, S)$.

Step 3: S sends a QRY Request_Ant packet to its neighbors and if the destination is not one hop away from the intermediate node then node also sets its height as NULL and broadcasts the packet.

Step 4: The QRY Request_Ant packet collects the information about bandwidth, delay, energy and drain rate during traversal.

Step 5: When an intermediate node k receives a QRY Request_Ant, it has four options:

5.1. If node k has no downstream links and RR_k is unset, it rebroadcasts the QRY Request_Ant packet and sets RR_k .

5.2. If node k has no downstream links and RR_k is set, it discards the QRY Request_Ant packet.

5.3. If node k has at least one downstream link and its height is NULL, it sets its height to $H_k = \min \{H_j \mid j \in N_k\} + \{0, 0, 0, 1, 0\}$ and broadcasts a QRY Reply_Ant packet.

5.4. If node k has at least one downstream link and its height is non-NULL, and if a QRY Reply_Ant packet has been broadcast since the link over which the QRY Request_Ant packet was received became active, it discards the QRY Request_Ant packet. Otherwise it broadcasts a QRY Reply_Ant packet. Also, if RR_k is set when a link becomes active, it broadcasts a QRY Request_Ant packet.

Step 6: QRY Reply_Ant follows the route corresponding to the QRY Request_Ant but in reverse order.

Step 7: When intermediate or source node k receives an QRY Reply_Ant from a neighbor j , k updates $HN_{k,j}$ to reflect the height of node j received in the QRY Reply_Ant and performs one activity from the two options:

7.1. If RR_k is set (implying the height of node k is NULL), node k sets $H_k = \min \{H_j \mid j \in N_k\} + \{0, 0, 0, 1, 0\}$, updates the link LS_k , unsets RR_k and broadcasts a QRY Reply_Ant packet with the new information.

7.2. If RR_k is unset, node k updates the links in LS_k .

Intermediate nodes also update the pheromone table according to equation (2) and calculate the path preference probability according to equation (3) and also update the probability table, and set the lower level node as downstream node.

Step 8: Source also calculates the Path Preference probability.

Step 9: If the calculated path preference probability is better than the requirements then the path is accepted and stored in the cache.

Step 10: The path with the better Path Preference Probability is selected for data transmission.

END

3.2. ROUTE MAINTENANCE PHASE

Algorithm 2: Route Maintenance

BEGIN

Here 2 cases can occur:

Case 1: When an intermediate node k detects the route failure, if it has an outbound link ($H_k > HN_{k,j}$) i.e. another unexpired route exists in the cache, the route with better Path Preference Probability is selected for routing.

Case 2: When an intermediate node detects the route failure, if it has no outbound link the following sequence of steps are performed:

Step 1: The node generates an Error packet and sends to the source node and all intermediate nodes delete routes from their cache which contains that node.

Step 2: As soon as source receives the error packet, it deletes routes from the cache which contain the node and if an unexpired route exists in the cache, the route with next better Path Preference Probability is selected for routing.

Step 3: After sending the Error packet, the node where link failure occur, starts local link maintenance according to the following conditions:

a. **Generate:** Node k has no downstream links due to link failure.

$(\tau_k, oid_k, r_k) = (t, k, 0)$, where t is the time of failure.

$$(\delta_k, k) = (0, k).$$

Node k defines a new reference level, if k has no upstream neighbors, it sets $H_k = \mathbf{NULL}$.

b. Propagate: Node k has no downstream links due to link reversal following receipt of an UPD and the ordered sets (τ_k, oid_k, r_k) are not equal for all $j \in N_k$.

$$(\tau_k, oid_k, r_k) = \max \{ (\tau_j, oid_j, r_j) \mid j \in N_k \}$$

$$(\delta_k, k) = \{ \min \{ \delta_j \mid j \in N_k \text{ with } (\tau_j, oid_j, r_j) = \max \{ (\tau_j, oid_j, r_j) \} - 1, k \}.$$

Node k propagates the reference level of its highest neighbors and chooses a reference lower than all neighbors of that reference level.

c. Reflect: Node k have no downstream links due to link reversal following receipt of an UPD packet and the sets (τ_j, oid_j, r_j) are equal with $r_j=0$ for all $j \in N_k$.

$$(\tau_k, oid_k, r_k) = (\tau_j, oid_j, 1)$$

$$(\delta_k, k) = (0, k).$$

Node k reflects back the reference level by setting the r bit.

d. Detect: Node i has no downstream links due to link reversal following receipt of an UPD packet and the sets (τ_j, oid_j, r_j) are equal with $r_j=1$ for all $j \in N_k$ and $oid_k=k$.

$$(\tau_k, oid_k, r_k) = (_, _, _)$$

$$(\delta_k, k) = (_, k).$$

Node k has detected a partition and route erasure phase invoked (described below).

e. Generate: Node k has no downstream links due to link reversal following receipt of an UPD packet and the sets (τ_j, oid_j, r_j) are equal with $r_j=1$ for all $j \in N_k$ and $oid_j \in k$.

$$(\tau_k, oid_k, r_k) = (t, k, 0), \text{ where } t \text{ is the time of failure.}$$

$$(\delta_k, k) = (0, k).$$

Node k experienced a link failure between the time it propagated a reference level and the reflected higher sub-level returned from all neighbors. This link failure required no reaction.

Step 4: Finally, after performing step 2, step 3, if the cache contains no unexpired routes, again a new route discovery phase is started.

END

3.3 ROUTE ERASURE PHASE

Route erasure is initiated upon the detection of a partition. Node k sets its height and the height entry for each $j \in N_k$ to \mathbf{NULL} . However, if the destination is a neighbor, the corresponding height array is set to \mathbf{ZERO} . Node k then updates its link-state array LS , and broadcasts a CLR packet that consists of a destination id, did and the reflected sub-level of node k, $(\tau_k, oid_k, 1)$. When a node k receives a CLR packet from a neighbor $j \in N_k$, it reacts according to the following steps of Algorithm 3:

Algorithm 3: Route Erasure

BEGIN

Step 1: If the reference level in the CLR packet matches that of node k , it sets its height and the height entry for each neighbor $j \in N_k$ to NULL (unless the destination is a neighbor, in which case the corresponding height array is set to ZERO), updates all the entries in its link-state array LS, and broadcasts a CLR packet.

Step 2: If the reference level in the CLR packet does not match that of node k , it sets the height entry for each neighbor $j \in N_k$ (with the same reference level in the CLR packet) to NULL and updates the corresponding link-state array entries.

Step 3: Lastly, the hop in error is removed from the node's route cache and all routes containing the hop are eliminated.

END

4. PERFORMANCE ANALYSIS

AMTR is an adaptive routing algorithm. So, it is suitable for the network where node mobility is higher and no centralized control exists. It takes the advantages of both TORA and Ant Colony Optimization technique.

This algorithm finds multiple paths between the source and the destination. Again, ACO always does not find the shortest path. Rather at the time of path set up, it also takes care of the QoS requirements and for this, link quality is improved. Besides this, if multiple paths exist and as ants store paths, which satisfies the QoS requirements, losing any one path cannot affect the communication. This is extremely necessary for the real time and multimedia communication. Use of multi path routing also increases the packet delivery ratio, decreases the packet loss rate. It also utilizes the bandwidth properly and for this, throughput and network stability or lifetime increase.

AMTR's design is aimed at minimizing the aggregate bandwidth by minimizing the control packets. It also minimizes the communication overhead by localizing algorithmic reaction to topological changes. Especially in case of link failure, the number of nodes that must participate in the reaction is minimized in comparison with the other routing algorithms. In the first phase of route maintenance of AMTR, if link failure occurs and at that time a downstream link exists then only the route with valid QoS requirements is selected for routing. Thus, it minimizes the number of control packets and consequently, the available bandwidth utilized properly.

5. CONCLUSION

An ant based QoS aware multipath routing algorithm AMTR is proposed in this paper which supports real time and multimedia applications. It is more adaptive and provides energy efficient routing by considering node's remaining energy as well as drain rate (i.e. energy dissipation rate) as one of the QoS parameters. In high mobility cases it is very efficient in terms of quick route maintenance. This algorithm takes care of end to end delay, available bandwidth, and hop count as QoS parameters which increase the network throughput. It is a multipath routing which increases network stability. AMTR reduces the number of control packets and utilizes the bandwidth properly which is also very efficient especially in case of dense networks.

ACKNOWLEDGEMENTS

The authors would like to thank West Bengal University Technology, Kolkata, India for the supports and facilities provided to carry out this research. The authors also thank the reviewers for their constructive and helpful comments.

REFERENCES

- [1] P., Deepalakshmi, S., Radhakrishnan, (2009), "Ant Colony Based QoS Routing Algorithm for Mobile Ad Hoc Networks", *International Journal of Recent Trends in Engineering*, Vol. 1, No. 1, pp. 459-462.
- [2] S. B., Wankhade, M. S., Ali, (2011), "Route Failure Management Technique for Ant Based Routing in MANET", *International Journal of Scientific & Engineering Research*, Vol. 2, No. 9.
- [3] R., Asokan, A. M., Natarajan, C., Venkatesh, (2008), "Ant Based Dynamic Source Routing Protocol to Support Multiple Quality of Service (QoS) Metrics in Mobile Ad Hoc Networks", *International Journal of Computer Science and Security*, Vol. 2, No. 3.
- [4] S., Kannan, T., Kalaikumar, S., Karthik, V. P., Arunachalam, (2010), "Ant Colony Optimization for Routing in Mobile Ad-Hoc Networks", *International Journal of Soft Computing*, Vol. 5, No. 6, pp. 223-228.
- [5] I., Kassabalidis, M. A., El-Sharkawi, R. J., Marks, P., Arabshahi, A. A., Gray, (2002), "Swarm Intelligence for Routing in Communication Networks", *IEEE Globecom*.
- [6] S. J., Mirabedini, M., Teshnehlab, M. H., Shenasa, A., Movaghar, A. M., Rahmani, (2008), "AFAR: Adaptive fuzzy ant based routing for communication networks", *Journal of Zhejiang University Science A*, Vol. 9, No. 12, pp. 1666-1675.
- [7] D., Sensarma, K., Majumder, (2012), "A Comparative Analysis of the Ant Based Systems for QoS Routing in MANET," *Recent Trends in Computer Networks and Distributed Systems Security, Communications in Computer and Information Science*, Volume 335, pp 485-496.

Authors

Debaji Sensarma has received his B.Sc degree in Computer Science in the year 2009 from university of Calcutta, Kolkata, India and M.Sc degree in computer Science in the year 2011 from West Bengal State University, Kolkata, India. He obtained his M.Tech. degree in Computer Science and engineering from West He is now pursuing PhD from the department of Computer Science and Engineering, University of Calcutta, Kolkata. He has published several papers in International journals and conferences.



Koushik Majumder has received his B.Tech and M.Tech degrees in Computer Science and Engineering and Information Technology in the year 2003 and 2005 respectively from University of Calcutta, Kolkata, India. He obtained his PhD degree in the field of Mobile Ad Hoc Networking in 2012 from Jadavpur University, Kolkata, India. Before coming to the teaching profession he has worked in reputed international software organizations like Tata Consultancy Services and Cognizant Technology Solutions. He is presently working as an Assistant Professor in the Dept. of Computer Science & Engineering in West Bengal University of Technology, Kolkata, India He has published several papers in International and National level journals and conferences. He is a Senior Member, IEEE.

