International Journal of Computational Intelligence Techniques
ISSN: 0976-0466 & E-ISSN: 0976-0474, Volume 4, Issue 1, 2013, pp.-122-126.
Available online at http://www.bioinfopublication.org/jouarchive.php?opt=&jouid=BPJ0000221

PROPOSAL TO ENHANCE ARA USING CONGESTION CONTROL AND SIMULATING ANNEALING

HASHEM S.H.* AND KUDHAIR E.Z.

Computer Sciences, University of Technology, Alsina'a St, Baghdad, Iraq.
*Corresponding Author: Email- soukaena_hassan@yahoo.com

Received: October 04, 2013; Accepted: October 28, 2013

Abstract- This research aims to enhance Ant-Colony-Based Routing Algorithm (ARA), which used in MANET, to be most efficient and strong to face the most important routing problem, congestion and stagnation. That by build proposed routing algorithm called Congestion Control-Simulating Annealing-ARA (CC-SA-ARA), which guide to the optimal shortest path with efficient deal for congestion and stagnation. Where congestion control CC is a modest mathematical equation tended to examine the congestion state for all neighbors' nodes to select the best one not just by it is highest pheromone (less cost) but with considering minimum congestion. Simulating annealing SA exploited as a solution to solve the stagnation problem by applying it after completing the first phase of ARA which ended by finding best paths from source to destination, save these paths then stimulate discovery phase using SA to discover if there is a best path better than these are saved. Finally by comparing the proposed CC-SA-ARA algorithm with traditional ARA algorithm according to the intrinsic routing criteria (jitter and message delivery ratio) related to congestion and stagnation, the result was very reasonable and encouraging.

Keywords- ARA, SA, CC, MANET, Routing Algorithms

Citation: Hashem S.H. and Kudhair E.Z. (2013) Proposal to Enhance ARA Using Congestion Control and Simulating Annealing. International Journal of Computational Intelligence Techniques, ISSN: 0976-0466 & E-ISSN: 0976-0474, Volume 4, Issue 1, pp.-122-126.

Introduction

A Mobile Ad hoc Network (MANET) is a wireless network without any fixed infrastructure or centralized control; it contains mobile nodes that are connected dynamically in an arbitrary manner. The nodes are the main components of the network, these nodes are mobile can move freely at any time, so the network structure changes dynamically due to node mobility. Each node behaves as a router; it takes part in discovery and maintenances the routes to other nodes in the network. The main characteristics of the MANET are dynamic topology, bandwidth constrained, variable capacity links, energy constrained operation. The nodes can move freely at any time and can leave or join the network. The research challenges in MANET are related to routing, security, reliability, scalability, quality of services, internetworking, energy consumption and multimedia applications. The MANET is essential in natural disasters like earthquake, flood, tsunami, fire and emergency services [1-3].

In recent years a large number of MANET routing algorithms have been proposed, these algorithms deal with the dynamic aspects of MANETs in their own way, using proactive or reactive behavior, or a combination of both. Proactive behavior means nodes try to maintain paths to all other nodes at all times. This means that they need to keep track of all topology changes, which can become difficult if there are a lot of nodes or if they are very mobile (e.g. Destination Sequenced Distance Vector (DSDV) Protocol), Reactive behavior means nodes only gather routing information on demand: when a data session to a new destination starts, or when a path which is in use fails. Reactive algorithms are in general more scalable, since they greatly reduce the routing overhead, but they can suffer from oscillations in performance because they are never prepared for disruptive events (e.g. Ad-hoc On-Demand Distance Vector (AODV) protocol and Dynamic Source Routing (DSR) protocol), In practice, many algorithms are hybrid algorithms (e.g. Zone Routing Protocol (ZRP)), using both proactive and reactive components in order to try to combine the best of both worlds. The interest in MANETs has been growing, and especially the design of MANET routing protocols has received a lot of attention. One of the 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 flow in the network is very restricted [4-6].

Ant Colony Optimization is the general name of the algorithm which is inspired by a behavior of feeding of ant which was proposed by Dorigo. The ACO meta-heuristic is based on generic problem representation and the definition of the ant's behavior. ACO adopts the foraging behavior of real ants. When multiple paths are available from nest to food, ants do random walk initially. During their trip to food as well as their return trip to nest, they lay a chemical substance called pheromone, which serves as a route mark that the ants have taken [7,8]. The first ACO routing algorithms were designed for wired networks (e.g. AntNet for packet switched networks and ABC for circuit-switched networks). These algorithms exhibit interesting properties which are also desirable for MANET routing: they work in a fully distributed way, are highly adaptive, use mobile agents for active path sampling, are robust to agent failures, provide multipath routing, and automatically take care of data load spreading. However, the fact that they crucially rely on repeated path sampling can cause significant overhead if not dealt with carefully. There have already been some attempts to design ACO routing algorithms for MANETs. Examples are Ant Routing Algorithm (ARA) [4,9].

Ant-Colony-Based Routing Algorithm (ARA) is a reactive routing protocol proposed by Gunes and Spaniol [10] for mobile ad hoc networks. The routing table entries in ARA contain pheromone values for choosing a neighbor as the next hop for each destination. The routing algorithm is very similar constructed as many other routing approaches and consists of three phases there are Route Discovery Phase, Route Maintenance, Route Failure Handling [11]. Route discovery phase in ARA is performed by a set of two mobile agents - forward ants and backward ants. The forward and backward ants update the pheromone tables at the nodes along the path for the source and destination nodes respectively, the second phase of the routing algorithm is called route maintenance, which is responsible for the improvement of the routes during the communication. ARA does not need any special packets for route maintenance, The third and last phase of ARA handles routing failures, which are caused especially through node mobility and thus very common in mobile ad-hoc networks [12].

Related Works

Gunes, et al [13] introduced an Ant- Colony-Based Routing Algorithm (ARA) is based on ant algorithms. The main properties of the algorithm are high adaptive, efficiency. They presents some extensions to the basic idea and shows through simulation results the performance gain and compares it with AODV and DSR, the network simulator is used to implement the result in the network simulator-2. Cauvery, et al [14] presents the enhancements of ARA to improve the performance of routing algorithm. ARA finds route between nodes in mobile ad-hoc network. The algorithm is on-demand source initiated routing algorithm. The algorithm is adaptive, scalable and favors load balancing. The improvements suggested in this paper are handling of loss ants and resource reservation. Correia & Vazao [15] presents a MANET routing protocol, inspired in insect societies' biological models, the Simple Ant Routing Algorithm (SARA), which provides a simple and efficient routing solution. SARA uses a controlled neighbor broadcast route discovery procedure, aimed at reducing the routing overhead of existing solutions. In this controlled neighbor broadcast strategy, every node collects routing information received from its neighbors and updates its own routing information accordingly, but only one of them is responsible for forwarding this information. The selection of the node which will be responsible for this task is made. Simulation results have shown that, besides reducing the overhead incurred by the routing protocol, SARA also provides a solution to detect early congestion link situations and tries to re-route the traffic through alternative routes (if available). Correia & Vazão [16] introduced the Simple Ant Routing Algorithm (SARA) that offers a low overhead solution, by optimizing the routing process. Three complementary strategies were used in this approach: during the route discovery used a new broadcast mechanism in which each node broadcasts a control message (FANT) to its neighbors, but only one of them broadcast this message again. During the route maintenance phase, the overhead reduced, by only using data packets to refresh the paths of active sessions. Finally, the route repair phase is also enhanced, by using a deep search procedure as a way of restricting the number of nodes used to recover a route. Thus, instead of discovering a new path from the source to the destination, tried to start the discovery of a new path between the two end-nodes of the broken link. The results achieved show that SARA offers the smallest overhead of all the protocols under evaluation and presents an overhead reduction of almost 25% of the value achieved by the other proposals. Roy, et al [17] introduced one of the major issues in MANET is routing due to the mobility of the nodes. QoS routing plays an important role for providing QoS in wireless ad hoc networks. The biggest challenge in this kind of networks is to find a path between the communication end points satisfying user's QoS requirement. Here, a new QoS algorithm for mobile ad hoc network has been proposed. The proposed algorithm combines the idea of Ant Colony Optimization (ACO) with Optimized Link State Routing (OLSR) protocol to identify multiple stable paths between source and destination nodes. Yoshikawa, et al [8] proposed a new hybrid routing algorithm which combines Tabu search with Ant Colony Optimization. The proposed hybrid technique enables to find the shortest route including the blind alley. Experiments prove the effectiveness in comparison with conventional routing algorithm such as Dijkstra algorithm.

The Proposal CC-SA-ARA

Before introducing the proposal of Congestion Control-Simulating Annealing-ARA (CC-SA-ARA) routing algorithm which try to advance ARA performance by support ARA by Congestion Control and feeding first phase of ARA with SA to stimulate the best paths not considered in discovery phase due to the stagnation problems known in ACO which is presented by freezing although there is other solutions are exists.

Congestion Control

Congestion Control (CC), refers to the set of actions taken by the network to minimize the intensity, spread, and duration of congestion. It can be said that it is that aspect of a networking protocol that defines how the network deals with congestion. The proposed congestion control aims to examine the congestion state for all neighbors' nodes to select the best one which present a minimum congestion. This will be done by a proposed modest mathematical equation to measure the congestion, and it is as follow:

$$CSRi = TPPA - TPPS \qquad (1)$$

where

CSRi: Congestion State Node (no. i)

TPPAi: Time Ping Packet Acknowledgement (no. i)

TPPSi: Time Ping Packet Send (no. i)

This means the measure of congestion will considered by counting time between sending ping packets from current source to all neighbors represent the current destinations and receiving ping packets acknowledgements from all current destinations to the source. The minimum time will be depended and the related current destination will be taken as the best congestion state node to represent the new current source.

Simulated Annealing

Simulated Annealing (SA), is a global optimization algorithm that belongs to the field of Stochastic Optimization and Metaheuristics. Here we will use SA as a solution to solve the stagnation problems which freeze the space of search to some paths but in real there is a best path not discovered yet. This procedure of SA will applied after completing the first phase of ARA which ended by finding multipath from source to destination, here we will save them and apply SA to stimulate the search space to discover if there is a best path better than these are saved.

The Full Proposed CC-SA-ARA Algorithm

The network under consideration is represented as G = (V,E), a connected graph with N nodes. The metric of optimization is number of hops between the nodes and congestion for each node. The goal is to find the shortest and less congestion path between source node Vs and destination Vd, where Vs and Vd belong to V. The path length is given by the number of nodes along the path.

Input: Network as connected graph with N nodes G = (V, E).

Output: Shortest and less congestion path.

Graph Parameters

- Each link/edge e(i,j) ε E of the graph connecting node Vi and Vj has a variable φi,j indicating the artificial pheromone value.
- An ant located in node Vi uses pheromone φi,j of node Vj belong to Ni to compute the probability of node Vj as next hop.
- Ni is the set of one hop neighbors of node Vi.
- Ci is the set of congestion states of one hop neighbors of node Vi.

Node Parameters

Routing Table

Routing table at each node stores the list of reachable nodes and their pheromone value. It is represented as structure consisting of following fields:

Destination_id: This represents the address of the destination node.

Next_id: This represents the address of the adjacent node used to reach destination node.

Pheremone: This represents the value used by the node to calculate the probability of each adjacent node to be the next hop in order to reach the destination.

Neighbor List

Neighbor list is used to store the information of all the neighboring nodes.

Congestion List

Congestion list is used to store the congestion state information of all the neighboring nodes.

SA Parameters

- *Problem Size*, the paths discovered from the first phase of ARA.
- *Iteration*$_{\max}$, the maximum no. of iterations in SA to stimulate the search space (stimulate new un discovered paths).
- *temp$_{max}$*, the maximum heat to shake the search space.
- *S$_{best}$*, the best solution.

Process

Step 1: Discovery Phase

- At source node create FA and broadcast to nodes in neighbor list.
- At source node, wait for BA. If BA not received within timeout period, generate FA with new sequence number and broadcast to nodes in neighbor list. If BA is received within timeout period send data packets along the path generated.
- At any node, when it receives FA, it does the following

```
    If current node = destination node
    {
    Set type of control packet to BA with same sequence number as FA.
    - Reserve resource at current node.
    - Send BA to node from which it has received FA.
    }
    Else
    {
    - hop count = hop count + 1
    - φi,j = φi,j+ Δ   //update pheromone value
    - determine the congestion state using [Eq-1]
    Send FA to nodes in neighbor list and congestion list
    }
```

- At any node, when it receives BA, it does the following

```
    If (current node is not source node)
    {
    Reserve resource at current node
        Send BA to node from which it has received FA
    }
    Else //BA reaches source node
    {
    -Get pheromone values of all links using neighbor list.
    -Compute probability for all nodes in neighbor list using
    pi,j = φi,j / (Σφi,j ) for j ε Ni = 0
    for j does not belong to Ni                              (2)
    The probability pi,j of a node Vi has the constraint that
    Σ pi, j =1, j ε Ni
  - Get congestion state values of all links using Congestion list.
  - Compute congestion state for all nodes in congestion list using [Eq-1]
Send packets to that link which has highest probability value.
    }
```

Step 2: Stimulating Best Paths

```
    S_current ← Initial Paths (Problem Size);
    S_best ← S_current;
for i = 1 to iterations_max do
    S_i ← Create Neighbor path (S_current);
    temp_curr ← Calculate Temperature (i, temp_max)
if Cost (S_i) ≤ Cost (S_current) then
```

```
          $S_{current} \leftarrow S_i$;
if Cost ($S_i$) ≤ Cost ($S_{best}$) then
          $S_{best} \leftarrow S_i$;
          end
Else if Exp> $\left(\frac{Cost(S_{current}) - Cost(S_i)}{temp_{curr}}\right)$ rand () then
          $S_{current} \leftarrow S_i$;
          end
    end
          Return $S_{best}$;
```

Step 3: Route Maintenance

At each node, when it receives data packet, it does the following

```
If current node = destination node
          {
          Extract data
          Send packet (acknowledge packet) to previous node
          }
Else
          {
   - Get pheromone values of all links using neighbor list.
   - Compute probability for all nodes in neighbor list using [Eq-2].
   - Get congestion state values of all links using Congestion list.
   - Compute probability for all nodes in congestion list using [Eq-1].
   - Send packets to that link which has highest probability value and less congestion.
   - Increment pheromone value for the highest probability link.
          }
```

- At regular intervals decrement pheromone value by α. If the pheromone value = 0 for any destination then call route discovery phase.
- If acknowledgement is not received at current node before timeout send route error to previous node (handling by error handling phase).
- Refresh route after time out.

Step 4: Route Failure Handling

At any node if a route error message is received, it performs the following function

```
If alternate path exist to reach destination
          - Send packets through other route
Else
          {
          - Set pheromone = 0 in routing table //deactivate link
          - Send route error message to previous node.
          }
```

If route error message reaches source, it calls route discovery phase.

End of Process

Experimental Works and Results

In our Experimental works we concentrate on two dimensions in problems of routing algorithms these are congestion and stagnation; the both could be measured by the following parameters:

Message Delivery Ratio: Message Delivery Ratio is defined as the ratio between the number of messages sent by sources and number of messages received by destination.

Average End-to-End Delay: Time taken for the packets to reach the destination.

To get a better idea of the performance of CC-SA-ARA we compared it with traditional ARA, We present results obtained by simulations done under VB.NET environment by constructing connected graph represent a network with 20 nodes, and we control the mobility of nodes in this network. Our experimental works was done by applying the traditional ARA and proposed CC-SA-ARA on the constructed networks, and from the implementation we get the following results. [Fig-1] presents the message delivery ratio of MANET when using traditional ARA and proposed CC-SA-ARA and [Fig-2] presents the end-to-end delay of MANET when using traditional ARA and proposed CC-SA-ARA, we see the superiority of the proposed CC-SA-ARA over the traditional one. There is also most critical and encouraged point in our proposal the results were obtained with high mobility of MANET.

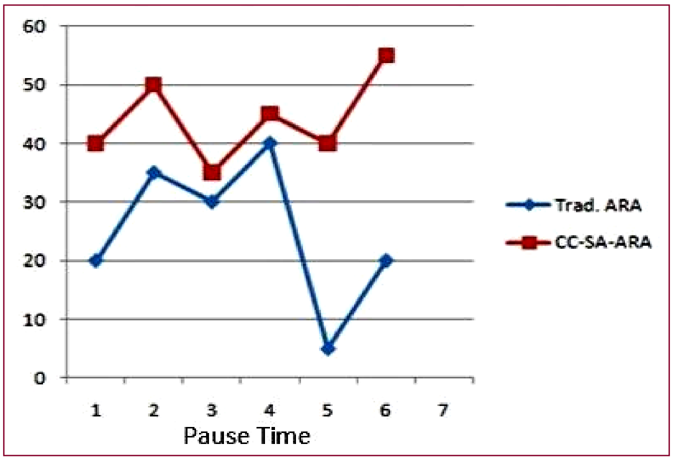

Fig. 1- Message Delivery Ratio

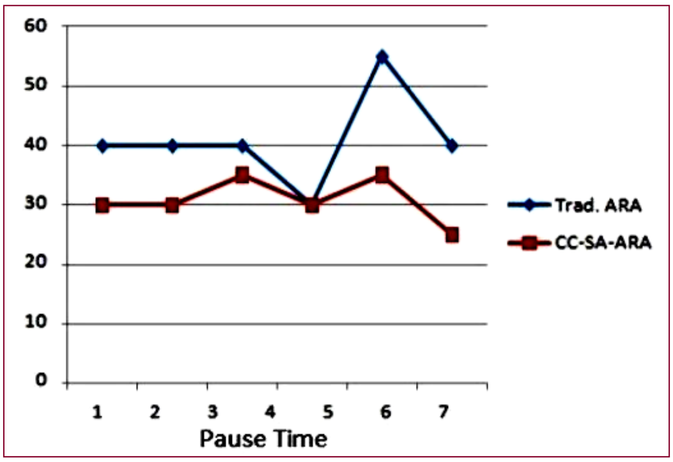

Fig. 2- Jittar

Conclusions

In this paper we presented a new (ARA) for MANET which is based on ant algorithms, also supported by proposed modest equation to apply a congestion control and supported by simulating annealing to stimulate the best path in discovery phase. The traditional ARA performed poorly in highly mobile scenarios (message delivery ratio and jitter). The proposed CC and injection of SA improve the performance of ARA in all scenarios (message delivery ratio and jitter) especially in those with high mobility.

Conflicts of Interest : None declared.

References

[1] Bheemalingaiah M., Naidu M.M., Rao D.S., Varaprasad G. (2009) *Journal of Theoretical and Applied Information Technology*, 5(4), 416-431.

[2] Li X. (2006) *Multipath Routing and Qos Provisioning in Mobile Ad Hoc Networks*, PhD Thesis, Queen Mary University, London.

[3] Garg D. and Gohil P. (2012) *International Journal of Smart Sensors and Ad Hoc Networks*, 2(3&4), 8-13.

[4] Caro G.D., Ducatelle F. and Gambardella L.M. (2005) *Euro. Trans. Telecomms.*, 16, 443-455.

[5] Perkins C.E., Royer E.M. (1999) *Second IEEE Workshop on Mobile Computing Systems and Applications*, 90-100.

[6] Royer E.M., Toh C.K. (1999) *IEEE Personal Communications*, 6 (2), 46-55.

[7] Deepalakshmi P. and Radhakrishnan S. (2009) *International Journal of Recent Trends in Engineering*, 1(1), 459-462.

[8] Yoshikawa M., Otani K. (2010) *International Multiconference of Engineers and Computer Scientists*, 3, 17-19.

[9] Dorigo M., Di Caro G., Gambardella L.M. (1999) *Artificial Life*, 5 (2), 137-172.

[10]Gunes M. and Spaniel O. (2003) *Conference on Network Control and Engineering for QoS*, 120-138.

[11]Kalaavathi B., Madhavi S., VijayaRagavan S., Duraiswamy K. (2008) *IEEE International Conference on Computing, Communication and Networking*, 1-9.

[12]Gunes M., Sorges U., Bouazizi I. (2002) *IEEE International Conference on Parallel Processing*, 79-85.

[13]Günes M., Kähmer M., Bouazizi I. (2003) 2*nd Mediterranean Workshop on Ad-Hoc Networks*.

[14]Cauvery N.K., Viswanatha K.V. (2008) *World Academy of Science, Engineering and Technology*, 46, 30-35.

[15]Correia F., Vazao T. (2008) *IEEE International Conference on Information Networking*, 1-8.

[16]Correia F., Vazão T. (2010) *Ad Hoc Networks*, 8(8), 810-823.

[17]Roy B., Banik S., Dey P., Sanyal S., Chaki N. (2012) *Journal of Emerging Trends in Computing and Information Sciences*, 3(1), 10-14.