INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 10(4):492-499, August, 2015.

Restructured Ant Colony Optimization Routing Protocol for Next Generation Network

B. Chandramohan

Dr. B. Chandramohan
Kanchi Pallavan Engg College,
Anna University, Chennai, India
abc.phd@hotmail.com

Abstract: Wireless network is a major research domain in the past few decades. Wireless network evolves in many forms like cellular communication, ad hoc network, vehicular network, mesh network and sensor network. Next generation network is a recent cellular communication which provides heterogeneous connectivity on cellular communication. The routing in next generation wireless networks is an important research issue which requires many constraints than wired networks. Hence, Ant Colony Optimization (ACO) is applied in this paper for routing in heterogeneous next generation wireless network. The ACO is a swarm intelligence technique which applied for many engineering applications. ACO is an optimal technique for routing and travelling salesman problem. This paper proposed Restructured ACO which contains additional data structures for reducing packet loss and latency. Therefore, the proposed RACO provides higher throughput.
Keywords: Wireless Network, Next Generation Network, Routing, Swarm Intelligence, Ant Colony Optimization.

1 Introduction

Swarm intelligence is a new discipline of study that contains a relatively optimal approach for problem solving, which is the imitation inspired from the social behaviours of insects and of other animals. ACO, artificial bee colony algorithms and fire fly algorithm are few techniques in swarm intelligence. The ACO is an optimization technique which is widely applied for a variety of optimization problems in various engineering field of studies.

The few application of ACO in the recent year are, Job Scheduling (Li-Ning et al 2010), Project Scheduling (Wang Chen et al 2010), Production management and maintenance scheduling (Osama et al 2005), Cash Flow Management (Wei-Neng et al 2010), Manpower Scheduling and management (Hsin-Yun et al 2010), Travelling Salesman Problem (Xiao-ming et al 2010), Clustering and set partitioning (Ali and Babak 2010), Pattern Recognition (Zhiding et al 2010). Dorigo et al (1996, 1997, 2004) proposed Ant System, Ant Colony System and Ant Net, which are the significant implementation of ACO. Dorigo et al (1996) applied the simple probability rule and Dorigo and Luca (1997) applied the state transition rule for the decision model. Dorigo and Stutzle (2004) redefined the pheromone update policy of ACO, and the term argentine ant is replaced with forward ant.

In the network routing, Ant-Net Routing using ACO provides a optimality due to its real time computation and less control overhead. Kwang and Weng (2003) comparing all routing algorithms with ACO, concludes that ants are relatively small, so it can be piggybacked in data packets and more frequent transmission of ants may be possible in order to provide updates of routing information for solving link failures. Hence, using ACO for routing in dynamic network seems to be appropriate. Routing in ACO is achieved by transmitting ants rather than routing tables or by flooding LSPs. Even though it is noted that the size of an ant may vary in different systems or implementations, depending on their functions and applications, in general, the size of ants is relatively small, in the order of 6 bytes.

Comparison of network performance on OSPF and ACO is available in Chandramohan et al (2007). The ACO Based Redundant Link Avoidance Algorithm is proposed in our earlier paper (Chandramohan et al, 2010). Detailed study on ACO in various engineering domain is available in Chandramohan et al (2011, 2012). This paper further explains the data structure of proposed Restructured ACO (RACO) in section 2, the route discovery and route maintenance of proposed RACO in section 3, and routing decision model of RACO in section 4. and the results and performance analysis in section 5.

2 Data Structure of Proposed RACO

The existing ACO has two data structures, namely, pheromone matrix and traffic model. The proposed RACO restructures the ACO by introducing two more data structures called traffic model II and availability model. These two data structures are used for identifying congestion free and stable routes. In addition to the extended data structures, the routing decision of proposed RACO is based on compound probability rule. The compound probability rule is the mean value of the probability of existing random probability rule and the two additional probability rules proposed in this work. The proposed two additional probability rules are namely probability based on response complexity and probability based on packet loss complexity. In addition to this compound probability rule, the priority model based on availability model is defined for single path routing in the wireless routing.

Hence, there are four data structures used in the proposed RACO, namely 1) Pheromone Matrix Model, 2) Traffic Model I, which is used as Round Trip Time (RTT) traffic model, 3) Traffic Model II, which is used as Packet Loss traffic model and 4) Availability Model.

3 Route Discovery and Route Maintenance in RACO

Every source node in the network, in a regular interval (t), generates forward ants (FA) and the FA is circulated in the network for searching the destination. Destinations are locally selected according to the data traffic patterns generated by the local workload. Both intermediate and source nodes forward the FA in the same way. The FA carries the path source address, the destination address, the intermediate node Identification, and the path information. The FA generation rate can be a function of network dynamics, data rate, time, etc. The FA moves in the network searching for the destination using the probability routing table of intermediate nodes.

While moving, the FA collects the information about the time length or trip time, the congestion status, and the node identifier of the intermediate nodes. A sufficient number of the ants visit the neighbour corresponding to the highest probability in the routing table. However, a number of the FA have a probability to visit other nodes and other paths still have a probability to be visited. This will increase the number of the FA visiting nodes in the region around the best path. In addition, it allows a fair number of FA to visit other regions in the network. Unlike flooding, a FA will be forwarded to only one neighbour.

When a FA reaches its destination, the information carried by this FA path will be graded. Then, the FA will be killed and a Backward Ant (BA) will be generated in the destination. The BA carries its corresponding FAs path grade and paths intermediate nodes ID ant it will be send back to the source node by following the reverse path of its corresponding FA. As the BA moves in the reverse path, the intermediate nodes modify their four data structures based on the path grade carried by the BA and accordingly update their probability routing tables.

Finally, the source node receives the BA, updates its tables, and kills the BA. FA shares the same queues as data packets, so that they experience the same traffic load. The BA takes the same path as the concern FA travelled, but in opposite direction. BA do not share the same link queues as data packets (like FA), they use the higher-priority queues reserved for routing packets, since the only task of BA is to quickly propagate to the pheromone matrices (the information accumulated by the FA).

Ants adapt their exploration activity in this way to the varying data traffic distribution. While travelling towards their destination nodes, the FA keep memory of their paths and of the traffic conditions found. The identifier of every visited node and the time elapsed since the launching time to arrive at this node is stored in a memory stack S(i), where, i - is the node identifier. Further the ant builds a path by performing the following steps:

Step 1 (Distribution of Forward Ant): At each source node(s), each FA headed toward the destination node(d) by selecting the intermediate node(i), choosing among its neighbour(j) which is not visited already. The neighbours node(j) is selected with a probability P(i,j,d) computed as the normalised sum of the pheromone T(i,j,d) with a heuristic value N(i,j) taking into account the state (the length) of the jth link queue of the current node I is

$$P(i,j,d) = \frac{T(i,j,d) + \alpha * N(i,j)}{1 + \alpha(|N(i)| - 1)} \tag{1}$$

The heuristic value N(i,j) is a normalised value function (0, 1) of the length of the queue q(i,j) between the node i and its neighbour j is

$$N(i,j) = 1 - \frac{q(i,j)}{\sum_{i=1}^{N} q(i,j)} \tag{2}$$

The value of α determines the importance of the heuristic value with respect to the pheromone values stored in the pheromone matrix (T). The value N(i,j) reflects the instantaneous state of the nodes queues and assuming that the queues consuming process is almost stationary or slowly varying, N(i,j) gives a quantitative measure associated with the queue waiting time. The pheromone values, on the other hand, are the outcome of a continual learning process. This learning process will register both the current and the past status of the whole network. The two components of the ant decision system, T and N, used for the decision system will act both the combination of long-term learning process and an instantaneous heuristic prediction.

Step 2 (Avoid Cyclic Routing): If an ant returns to an already visited node, the cyclic nodes are removed and all the memory about them are deleted. This process helps the system to avoid the count-to-infinity problems.

Step 3 (Generate Backward Ant):
* Condition1 [Generate BA and delete FA] - When the destination node(d) is reached, the FA generates another ant called the backward ant and transferred the information stored in the memory of FA to BA. Then the FA is to be deleted.
* Condition2 [Delete FA on lifetime Basis] - The FA is deleted if its life time becomes greater than a value of MaxLife before it reaches the destination node, which is similar to TTL (Time to Live) in the TCP.

Step 4 (Flow of BA): The flow of FA and BA have some distinctions. The BA takes the same path as the concern FA travelled, but in the opposite direction. The FA share the same link queues as data packets, but the BA do not share the same link queues as data packets, instead BA uses the higher-priority queues reserved for routing packets. Because the task of

BA is to quickly propagate to the pheromone matrices. The pheromone matrix is updated by incrementing the pheromone T(i,f,d) and decrementing the other pheromones T(i,j,d), where, i-represents the node ID, d-is the destination node, f-is the current neighbour and j-is the existing neighbour, where j is not equal to f. The trip time is a good decision parameter because it indicates the appropriateness of the physical and logical condition of the path, which includes the number of hops, transmission capacity of the concerned link, processing time and queuing delay. A path with a high trip time also considered as a good path, if its trip time is significantly lower than the other trip time.

The traffic model is updated with the values stored in the BAs memory. The time elapsed to arrive to the destination by the FA starting from the current node is used to update, according to the following equations:

$$\mu_i = \mu_i + \delta(RTT(i,d) - \mu_i) \tag{3}$$

$$\beta_i^2 = \beta_i^2 + \delta(RTT(i,d) - \mu_i)^2 \tag{4}$$

where RTT(i,d) is the new observed trip time from node i to the destination d. μ is the sample mean of the traffic model and β^2 is the variance of the traffic model. δ is the representation of the weighs of the number of most recent samples that will really affect the average.

The value of δ is calculated as $\delta = 0.1$, assuming that the latest fifty observations really influence the estimate (approximately), i.e. (5/50). Suppose 100 latest observations really influence the estimate, then $\delta = 5 \ / \ 100 = 0.05$. Node i, waits for a specific predefined time unit (maximum allowable trip time) for the BA of each sending FA. If the BA is not reached within the time period, then the node i considered the ant is destroyed in the intermediate node due to heavy congestion or by wrong propagation in the path.

For every interval (t), the number of packet loss is calculated and this information is stored in the traffic model II.

$$PL_{new} = PL_{old} + PL_{observed} - \mu * PL_{old} \tag{5}$$

where PL_{new} is the new value for packet loss, PL_{old} is the existing value of packet loss, $PL_{observed}$ is the currently calculated value of packet loss, and μ is the standard deviation of existing values. This is an optional model that used only for wireless environment. This availability model also may be opted for wired environment in order to test the frequent link failures. This model stores the information of the availability of the next hop based on its battery power and / or the mobility. When a BA reaches node i, then the information stored regarding the mobility or link failure in the memory is collected and this model is updated. The value of the availability model is between (0,1). The value 0 represents the next hop is unavailable and the value 1- represents the next hop is available. If the existing value is 1, the new value is 0, and then the new value is updated. If the existing value is 0 (or 0 to 1), the new value is 1, and then the current value is defined as, current value = observed value $-(\frac{1}{1+existingvalue})$.

4 Routing Decision Model

The routing decision model computes and identifies the optimal path from the information stored in the data structures. In the RACO, the routing model applies the compound and priority based routing model for routing decision.
Response Complexity (C_r) of particular path is calculated using

$$C_r = \frac{rtt_s - rtt_a}{rtt_s} \tag{6}$$

where, rtt_s is standard RTT (Round Trip Time), and rtt_a is actual RTT. The standard RTT is defined based on the applications, for data communication the standard RTT is defined as 150ms and for the multimedia communication the same is defined as 100ms in order to avoid problems like jitter.
Probability based on Response Complexity (P_c) is

$$P_c = \frac{C_r}{\sum C_r} \tag{7}$$

Similarly, the Packet Loss Complexity (C_{pl}) of particular path is computed using

$$C_{pl} = \frac{C_{pl}}{\sum C_{pl}} \tag{8}$$

and the Probability based on Packet Loss Complexity (P_{pl}) is

$$P_{pl} = \frac{P_{pl}}{\sum P_{pl}} \tag{9}$$

The Random proportional rule is applied for Pheromone density ($P_{T,j}$) is

$$P_{(T,j)} = \frac{(T_j + k)^h}{\sum_{i=1}^{n} (T_i + k)^h} \tag{10}$$

where k and h are constant, k = 0,1 and h = 0,1. From the above, the compound probability is obtained as

$$P = \frac{P_c + P_{pl} + P_{(T,j)}}{3} \tag{11}$$

For Multi Path Routing, data is propagated via all available paths based on compound probability. For Single Path Routing, the first priority path is selected and used for communication.

4.1 Pseudo Code for Route Discovery phase of Proposed RACO

A. Procedure RACO (t, t_{end}, t1)
Inputs:
t – current time
t_{end} – Time length
t1 – Time interval between ants generation
//Initialization Phase
for-each i in C do
PM – initialize pheromone matrix
TM1 – initialize traffic model I
TM2 – initialize traffic model II
AM – initialize Availability model
R – Initialize Routing Table
//Generate and Propagate Forward Ant
while $t \leq t_{end}$ do
in-parallel
if (t mod t1) = 0 then
$destination = select(distribution_equation)$
LaunchForwardAnt(Source, Destination)

end-if
for-each (ActiveForwardAnt (Source, Current, Destination)) do
$while(current \neq destination)$ do
$next_{hop} = SelectLink(current, destination, link_{queues}, T)$
$PutAntOnLinkQueue(current, next_{hop})$
$WaitOnDataLinkQueue(current, next_{hop})$
$CrossLink(current, next_{hop})$
$Memorize(current, next_{hop})$
$current = next_{hop}$
end-while
//Generate and Propagate Backward Ant
LaunchBackwardAnt(Source, Destination)
end-for each
for-each (ActiveBackwardAnt(Source, Current, Destination)) do
$while(current \neq destination) \quad do$
$next_{hop} = PopMemory$
$WaitOnHighPriorityLinkQueue(current, next_{hop})$
$CrossLink(current, next_{hop})$
from=current
$current = next_hop$
UpdateLocalTrafficModel(PM,TM,AM,current,from,source,memory)
GetNewPheromone (PM,current,from,source,memory)
UpdateLocalRoutingTable (PM,TM,AM,R,source)
end-while
end-for each
$end - in_{parallel}$
end-while
end-for each
end-procedure

5 Conclusion

The proposed RACO is implemented in Network Simulator 2 (NS2). The simulation performed for 10 seconds and the performances are recorded. The test are carried out in a variety of design and topology of network which include wireless network, and HNGN. Also the tests are carried out in various number of nodes; and on various load condition. The Routing Packet Size, Route Discovery Time, Packet Delivery Ratio and Network efficiency are shown in tables 1, 2, 3 and 4, respectively.

Table 1: Routing Packet Size (in Bytes)

DSR	DSDV	AODV	ACO	RACO
60	44	44	16	32

Table 2: Route Discovery Time (in ms)

DSR	DSDV	AODV	ACO	RACO
420	4300	3200	216	182

Table 3: Packet Delivery Ratio (in percentage)

No of Nodes	DSR	DSDV	AODV	ACO	RACO
100	87	89	92	96	100
200	85	88	91	95	100
500	79	84	87	91	93
1000	79	82	87	89	92
1500	76	81	85	87	91
2000	76	81	85	86	91

Table 4: Network Efficiency (in percentage)

No of Nodes	DSR	DSDV	AODV	ACO	RACO
100	71	73	88	90	94
200	69	71	85	88	90
500	67	69	83	86	88
1000	66	68	81	83	86
1500	64	66	74	74	83
2000	62	64	74	74	79

Bibliography

[1] Ali M. and Babak (2010); A. A new clustering algorithm based on hybrid global optimization based on a dynamical systems approach algorithm, *Expert Systems with Applications*, 37: 5645-5652.

[2] Chandramohan, B., Prasanna Kumar P, Anantha Venkata Ramana, Sridharan D (2007); Real time routing protocol (Antnet) using ACO and performance comparison with OSPF, *IEEE Int. Conf. on Emerging Trends in High Performance Architecture Algorithms and Computing*, 47-53.

[3] Chandramohan, B. and Baskaran, R.(2010); Improving Network Performance using ACO Based Redundant Link Avoidance Algorithm, *International Journal of Computer Science Issues*, 7(3): 27-35.

[4] Chandramohan, B. and Baskaran, R. (2011); Survey on Recent Research and Implementation of Ant Colony Optimization in Various Engineering Applications, *International Journal in Computational Intelligent Systems*, 4(4): 566-582.

[5] Dorigo, M., Maniezzo, V. and Colorni, A. (1996); Ant System: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, 26(1): 29-41.

[6] Dorigo, M. and Luca, M.G.(1997); Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation*, 1(1): 53-66.

[7] Dorigo, M. and Stutzle, T. (2004); *Ant Colony Optimization*, MIT Press, Cambrige MA.

[8] Hsin-Yun, L., Hao-Hsi, T., Meng-Cong, Z. and Pei-Ying, L. (2010); Decision support for the maintenance management of green areas, *Expert Systems with Applications*, 37: 4479-4487.

[9] Kwang, M. S. and Weng, H.S. (2003); Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions, *IEEE Transactions on Systems, Man, and Cybernetics*, 33(5): 60-572.

[10] Li-Ning, X., Ying-Wu, C., Peng, W., Qing-Song, Z. and Jian, X.(2010); A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems, *Applied Soft Computing*, 10: 888-896.

[11] NS2, available online at: www.isi.edu/nsnam/ns/

[12] Osama, H.H., Tarek, N.S. and Myung, J.L. (2005); Probability Routing Algorithm for Mobile Ad Hoc Networks Resources Management, *IEEE Journal on Selected Areas in Communications*, 23(12): 2248-2259.

[13] Wang Chen, Yan-jun, S., Hong-fei, T., Xiao-ping, L. and Li-chen, H. (2010); An efficient hybrid algorithm for resource-constrained project scheduling, *Information Sciences*, 180: 1031-1039.

[14] Wei-Neng, C., Jun Zhang, Henry Shu-Hung, C., Rui-Zhang, H. and Ou Liu (2010); Optimizing Discounted Cash Flows in Project Scheduling An Ant Colony Optimization Approach, *IEEE Transactions on Systems, Man, and Cybernetics*, Part C: Applications and Reviews, DOI:10.1109/TSMCC.2009.2027335, 40(1): 64-77.

[15] Xiao-ming, Y., Sheng, L. and Yu-ming, W. (2010); Quantum Dynamic Mechanism-based Parallel Ant Colony Optimization Algorithm, *International Journal of Computational Intelligence Systems*, Suppl. 1, 101-113.

[16] Zhiding, Y., Oscar, C.A., Ruobing, Z., Weiyu, Y. and Jing, T. (2010); An adaptive unsupervised approach toward pixel clustering and color image segmentation, *Pattern Recognition*, 43: 1889-1906.