Cooperative Reinforcement Learning Approach for Routing in Ad Hoc Networks

Rahul Desai
Research Scholar, Sinhgad College of Engg.
Asst Professor, Army Institute of Technology
Pune, Maharashtra, India
E-mail: desaimrahul@yahoo.com

B P Patil
Professor, E and TC Department
Army Institute of Technology
Pune, Maharashtra, India
E-mail: bp_patil@rediffmail.com

***Abstract*—Most of the routing algorithms over ad hoc networks are based on the status of the link (up or down). They are not capable of adapting the run time changes such as traffic load, delay and delivery time to reach to the destination etc, thus though provides shortest path, these shortest path may not be optimum path to deliver the packets. Optimum path can only be achieved when quality of links within the network is detected on continuous basis instead of discrete time. Thus for achieving optimum routes we model ad hoc routing as a cooperative reinforcement learning problem. In this paper, agents are used to optimize the performance of a network on trial and error basis. This learning strategy is based work in swarm intelligence: those systems whose design is inspired by models of social insect behaviour. This paper describes the algorithm used in cooperative reinforcement learning approach and performs the analysis by comparing with existing routing protocols.**

Keywords—Ad hoc Networks, AODV, AOMDV, DSR, DSDV, MANET, SWARM, Q Routing

I. INTRODUCTION

An ad hoc network is a collection of wireless mobile nodes and having features of zero configurations. These are peer to peer networks and having dynamic topology [1,2]. These networks are mobile and need multiple network hops for packet transmission for one node to another across the network. In such a network, each mobile node is also acting as a router and thus forwards packets for other mobile nodes in the network that may not be within the direct reach.

The routing protocols for ad hoc networks are broadly classified into two classes: Proactive routing protocols (table driven) maintain unicast routes between all pairs of nodes without taking care of usage of routes. Second class includes On Demand routing protocols where the routing tables are created when the need arises. In this paper, various routing protocols of ad hoc networks are simulated and their performance comparison is done using delay, routing overheads, loss performance parameters and also using different mobility models.

Destination-Sequenced Distance-Vector Routing (DSDV) [3,4] is a table-driven algorithm where every node maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded. Each entry is with a sequence number assigned by the destination node. Routing updates are transmitted throughout the network in order to maintain table records consistent. Optimized List State Routing (OLSR) [5] is another link state algorithm which is said to be more optimized for ad hoc nature. It uses Multipoint relays (MPR) to efficiently propagate updates across the network. OLSR also uses only periodic updates for link state dissemination. Thus it reduces the overhead when the network is dense.

The Dynamic Source Routing protocol (DSR) [6] is based on source routing. The routes are stored in a route cache and if route is not available it initiates Route Discovery process by broadcasting request message. Destination node or any nearby node having a desired route reply with route reply message. Ad Hoc on Demand Distance Vector Routing (AODV) [6,7] is pure on-demand routing protocol. It uses traditional routing tables, one entry per destination. AODV uses destination sequence numbers as in DSDV to prevent routing loops and to determine freshness of routing information. In AODV, each node maintains at most one route per destination and as a result, destination replies only once to the first incoming request during a route discovery. When the path present in memory fails, it has to repeat route discovery process. To overcome this limitation, another Multipath extension to AODV called Ad Hoc On-Demand Multipath Distance Vector (AOMDV) [8-10] is used. AOMDV discovers multiple paths between source and destination in a single route discovery.

II. REINFORCEMENT LEARNING

Reinforcement learning is an area where agents perform some action in an environment to maximize the performance of a network. Supervised learning having samples input-output pairs and attempts to learn the function from input to output. It needs an amount of learning data in order to train the system. In Reinforcement Learning (RL), system tries to optimize the performance of a system by interacting with a dynamic environment through trial and error.

Figure 1 shows agent's interaction with the system. An agent checks the current state of system, chooses one action from those available in that state, observe the outcome and receives some reinforcement signal.

978-1-4799-6272-3/15/$31.00(c)2015 IEEE

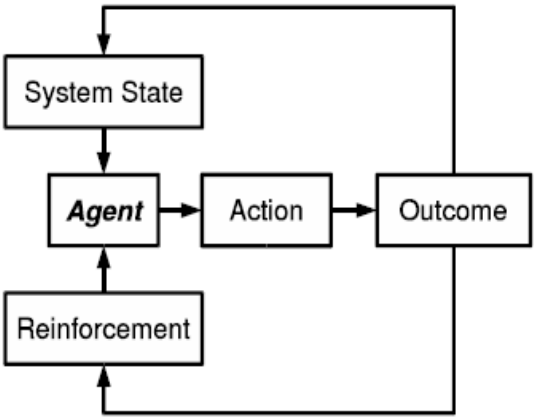

Figure 1: Reinforcement Learning Approach

Reinforcement learning algorithms may be broadly divided into two classes: those which attempt to learn a model of the system, and those which do not. Both model-free and model-based methods are capable of finding optimal policies. Model-free methods generally require less computation time per iteration of the algorithm, but more iterations to reach a (near) optimal policy.

In model based, where the agent leans a model of the environment and then uses it to control the routing policy of a network. In model free method, a controller is learned directly from the actual outcomes. The best example of model based method is the reinforcement learning which is mostly adapted for intelligent routing for a network.

In a model-free approach, no attempt is made to learn the state transition and reinforcement probabilities. Each individual reinforcement is used to feed information back to the actions that caused it. For a non-deterministic process, a model-free method gathers information about state transitions and reinforcements implicitly. [11,12]

III. Q LEARNING AND SWARM PROTOCOL

As an example of the difference between model-free and model-based learning methods, we consider Q-Learning. In this, every node contains Q values which represent the expected reinforcement of taking some action in a particular state. Highest Q value here represents shortest route when these Q values represents the actual accurate state of a network. Q values are presenting the actual state of a network at run time, so they are not constant, they are also updated. Initial Q values may not be accurate but these Q values get updated every time and thus converge to most stable Q values which thus represent most accurate state of a network. Thus complete state of a network represented in terms of Q values in a network [13-14].

Q learning is used to learn a status of the network in terms of Q values and then these Q values are used to decide the routing policy. Each node X in the network represents its own view of the state of the network through its Q-table Q_x. Given this representation of the state, the action 'A' at node X is to choose that neighbour Y such that it takes minimum time for a packet destined for node D to reach its destination if sent via neighbour Y.

Each node maintains a table of Q values Qx(Y, D), where D is the destination node and Y is the neighbor node for node X. Qx(Y, D) is node X's best estimated time that a packet would take to reach its destination D when sent via its neighboring node Y.

As shown in figure 2, when a node X receives a packet for a desired destination D, node X checks its Q table and thus selects that neighboring node ^Y for which the Q_x (^Y, D) value is minimum. It is important to note, however, that these Q values are not exact or accurate. So routing decision based on those Q values are not accurate and thus may not provide best solution. For making routing decisions accurate, these Q values should update after a short time frame. Instead, Q values are updated every time when a node transmits a packet to its neighbor.

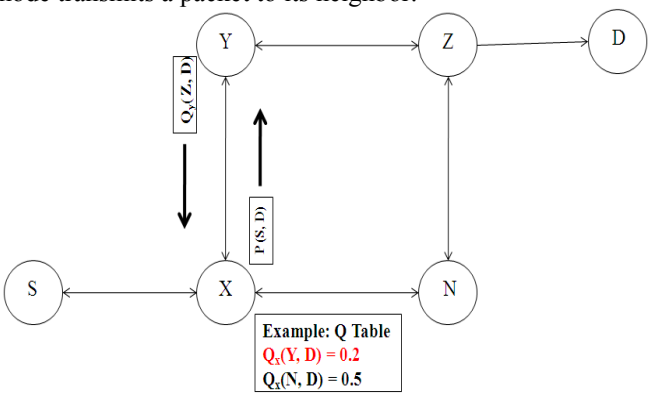

Figure 2: Example of Q Routing

Once the packet from source S is received by node X, node X checks its Q table (which is similar of routing table, but Q table consists of Q values which are directly proportional to actual delay that packet takes to reach to the destination), $Q_x(Y,D)$ and $Q_x(N,D)$. Node X decides the optimum route (which could be neighbour node Y or N) that packet takes to reach to the destination D. Once the packet reaches to the neighbour (Y or N) the neighbour node returns Q value ($Q_y(X,D)$ or $Q_y(N,D)$) back to node X. Node X updates its Q value ($Q_x(Y,D)$ and $Q_x(N,D)$).

Inspired by the Q routing algorithm by Boyan and Littman (1994) [15-16], more optimized version called as Confidence based Q routing (CQ Routing), is is presented. The accuracy of routing decisions depends on how Q values in the network. If these Q values are accurate, shortest path is obtained. These Q values are always updated. In order to bring amount of reliability in the Q values, confidence

values are added in Q Routing. For every Q value in the network, confidence value (C value) is associated with them which lies in between 0 and 1. Low confidence values indicates less reliability of Q values while High confidence values represents more reliable value thus routing decisions are more optimum. Figure 3 represents the example of confidence based Q routing.

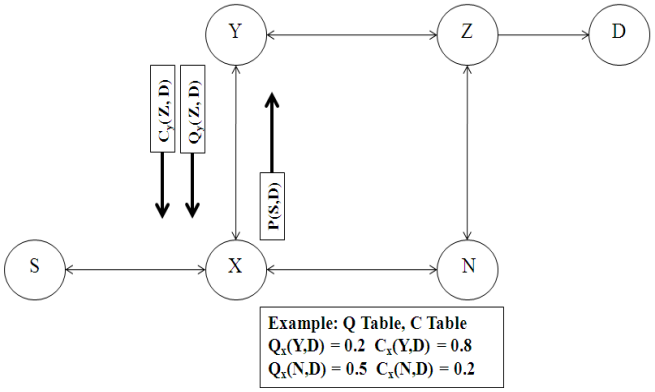

Figure 3: Example of CQ Routing

Dual reinforcement Q Routing (DRQ) [11] is a modified version of the Q-Routing algorithm, where learning occurs in both ways. Since, the learning process occurs in both ways the learning performance of the Q-Routing algorithm doubles. However, it adds more overheads to the network. Figure 4 shows an example of Dual reinforcement Q routing.

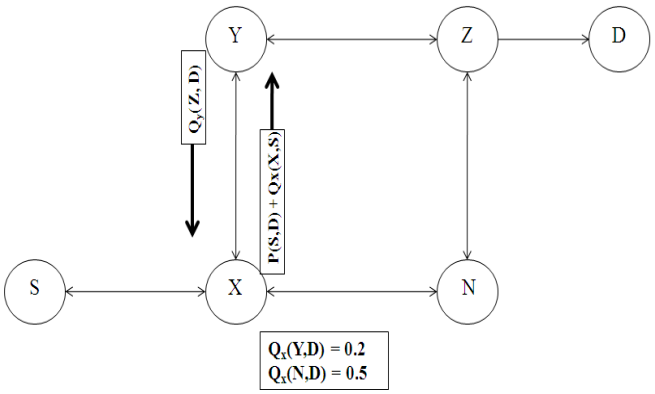

Figure 4: Example of DRQ Routing

Doing complete propagation of information in model-based Q-learning in response to each experience is very expensive. It also devotes a lot of resources making small updates to values, and updating states that are unlikely to be useful.

Prioritized Sweeping [17,18] attempts to concentrate computation effort on 'interesting' information: A state is considered interesting when it's value estimation changes, with larger changes considered more interesting. Prioritized sweeping's update strategy is to: maintain a list of 'predecessor' states for each state. These are states which have some non-zero transition probability to it under some action, maintain a priority for each state. This is the expected change to its value the next time it is calculated. Each time a state's value is updated, the change in its value is added to the priority of its predecessors. At each iteration, update the k states with the highest priority, setting their priority back to 0 afterwards.

SWARM uses mechanisms of swarm intelligence for selecting best routes to destinations. SWARM is scalable, adaptable, and autonomous and also having good Fault tolerance capability. [19-20]. Swarm systems are a source of inspiration for the design of various adaptive routing algorithms. [21]. Reinforcement function is reflecting the cost to the network of a given state transition. The state transition is dependent on the action performed and choice of action in this model is choice of which node to forward the packet to. Whenever any node receives a packet, it could be either dropped (When TTL reaches 0) or forwarded. Forwarding of a packet by a node is modeled using reinforcement learning model. The complete renouncement model used and algorithm is available at [22-24].

IV. PERFORMANCE EVALUATION

We are comparing SWARM protocol with existing routing protocol. NS2 is Network Simulator (Version 2) is an event driven simulation tool useful for designing and simulating various types of networks including wireless and ad hoc networks. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours.

We select constant bit rate (CBR) traffic sources. In NS-2 directory the following file exists: (/ns-2.34/indep-utils/cmu-scen-gen/cbrgen.tcl). This file is the traffic-scenario generator tool to generate CBR traffic connections between the nodes. Data packets are of size of 512 bytes. Data packets are sent by sources at the rate of 2 to 4 packets/second.

We also create motion file having 10 nodes movement within an area of 800 by 800. Speed is varying from 0 to 10 m/s, pause time changing from 0 to 500 in steps of 50s. Thus simulation parameters used are as shown in table 1.

TABLE 1 : SIMULATION PARAMETERS

Parameter	**Value**
Number of nodes	20 to 100 nodes
Mobility model	Default Case, Random Waypoint Mobility Model
Simulation time	500 s
Topology Size	800 m × 800 m
Routing protocols analysed	DSDV, DSR, AODV, AOMDV and SWARM Protocol.
Packet size	512 bytes
Mobility rate	25m/s to 125 m/s

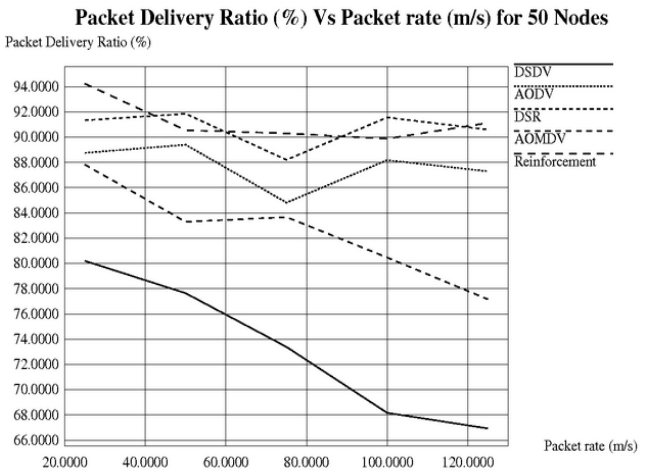

Figure 5: Packet Deliver Ratio vs. Pause Time for 50 Nodes

Figure 5 and 6 shows Packet deliver ratio (PDR) vs. Pause Time. Nodes are varying from 10 nodes to 100 nodes and also mobility changes from 25m/s to 125 m/s. It shows that DSDV as it is proactive routing protocol returns worst result and PDR lies in between 65% to 80%. Other three protocols AODV, DSR and AOMDV which are on demand routing protocols; gives packet delivery ratio between 80% to 95%. SWARM protocol described earlier gives consistent result and PDR consistently lies in range of 90% to 95%.

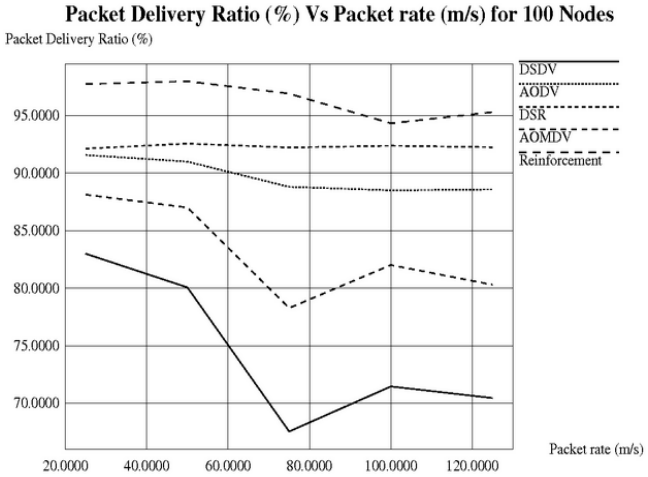

Figure 6: Packet Deliver Ratio vs. Pause Time for 100 Nodes

Mobility changes from 25m/s to 125 m/s. By changing nodes to 100 Nodes, it is observed that PDR ratio for SWARM protocol returns better results as compared with 50 nodes. PDR goes beyond 95% and returns best result comparatively other routing protocols.

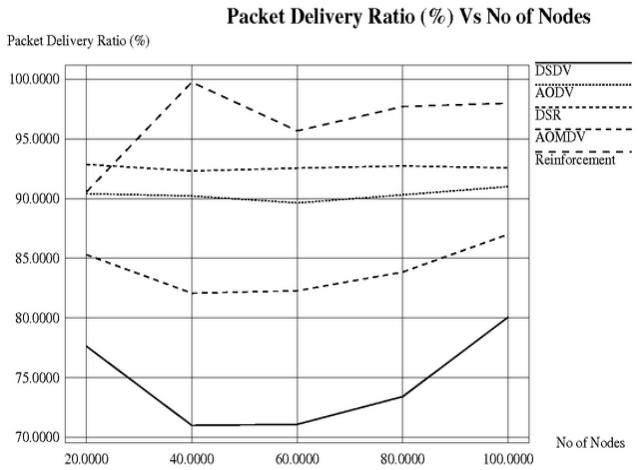

Figure 7: Packet Deliver Ratio vs. No of Nodes

Figure 7 shows PDR vs. No of nodes. Nodes are changing from 20 to 100 nodes. Here also SWARM protocol returns better results where PDR goes beyond 95% and returns best result comparatively other routing protocols. Other parameters such as delay and routing overhead are also tested for comparing SWARM protocol with other routing protocols; it is observed that SWARM protocols generate more delay and more routing overhead as compared with AODV and DSR protocols.

V. CONCLUSION

On demand routing protocols such as AODV and DSR returns good performance in terms of packet delivery ratio but at high mobility and heavy load situations, both of them fail to work. SWARM protocol based on reinforcement learning algorithm returns consistent result as compared with AODV and DSR protocols. SWARM protocol generates high Normalized routing overhead due to its broadcasting nature and also delay is comparatively more as compared with AODV and DSR protocol. It is also necessary to optimize the performance of SWARM protocol and test it further with existing routing protocols.

REFERENCES

[1] S Radha, S Shanmugavel, “Mobility Models in Mobile Ad Hoc Network”, 2007, IETE Journal of Research, Vol.53, No. 01

[2] A K Sharma And Amit Goyal, “A Power Efficient Self Adjusting Routing Protocols for Mobile Ad Hoc Networks”, 2007, IETE Journal of Research, Vol 53,No.4, July-Auguest 2007.

[3] Asma Toteja, Raynees Gujral, Sunil Thalia, “Comparative performance Analysis of DSDV, AODV and DSR Routing Protocols in MANETs, using NS2”, 2010 International Conference on Advances in computing Engineering, IEEE Computer Society.

[4] Khan, K.; Zaman, R.U.; Reddy, K.A.; Reddy, K.A.; Harsha, T.S., "An Efficient DSDV Routing Protocol for Wireless Mobile Ad Hoc Networks and its Performance Comparison," *Second UKSIM*

European Symposium on Computer Modeling and Simulation, 2008. EMS '08. , pp.506,511, 8-10 Sept. 2008

[5] Ouacha, A.; Lakki, N.; El Abbadi, J.; Habbani, A.; El Koutbi, M., "OLSR protocol enhancement through mobility integration," *10th IEEE International Conference on Networking, Sensing and Control (ICNSC), 2013*, pp.17,22, 10-12 April 2013

[6] Bai, R.; Singhal, M., "DOA: DSR over AODV Routing for Mobile Ad Hoc Networks," *Mobile Computing, IEEE Transactions on* , vol.5, no.10, pp.1403,1416, Oct. 2006

[7] D B Johnson, D A Maltz, Y. Hu and J G Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR) http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt, Feb 2002, IETF Internet Draft.

[8] Yuhua Yuan; Huimin Chen; Min Jia, "An Optimized Ad-hoc On-demand Multipath Distance Vector(AOMDV) Routing Protocol," *Asia-Pacific Conference on Communications, 2005*, pp.569,573, 5-5 Oct. 2005

[9] Rajesh Shrivastava, Rashween Kaur Saluja "Performance Evaluation of Extended Aodv Using Different Scenarios" International Journal of Smart Sensors and Ad Hoc Networks (IJSSAN) ISSN No. 2248-9738 Volume-1, Issue-3, 2012

[10] May Zin Oo; Othman, M., "Performance Comparisons of AOMDV and OLSR Routing Protocols for Mobile Ad Hoc Network," *Second International Conference on Computer Engineering and Applications (ICCEA), 2010*, pp.129,133, 19-21 March 2010

[11] Shailesh Kumar "Confidence based Dual Reinforcement Q-Routing: an On-line Adaptive Network Routing Algorithm" Report AI98-267 May 1998

[12] S Kumar, Confidence based Dual Reinforcement Q Routing : An on line Adaptive Network Routing Algorithm. Technical Report, University of Texas, Austin 1998.

[13] Davi Kelly, "Reinforcement Learning with Application to Adaptive Network Routing" Journal of Theoretical and Applied Information Technology, 2005 JATIT.

[14] Ramzi A. Haraty and Badieh Traboulsi "MANET with the Q-Routing Protocol" ICN 2012 : The Eleventh International Conference on Networks

[15] Mahmoud Alilou, Mohammad Ali Jabraeil Jamali, Behrooz Talebzadeh and Maysam Alilou "Modified Q-learning Routing Algorithm in Fixed Networks" Australian Journal of Basic and Applied Sciences, 5(12): 2699-2703, 2011 ISSN 1991-8178

[16] Shalabh Bhatnagar, K. Mohan Babu "New Algorithms of the Q-learning type" Science Direct Automatica 44 (2008) 1111- 1119. Website: www.sciencedirect.com.

[17] Gang Zhao; Tatsumi, S.; Ruoying Sun, "RTP-Q: a reinforcement learning system with an active exploration planning structure for enhancing the convergence rate," *1999 IEEE International Conference on Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings.*, vol.5, pp.475,480, 1999

[18] Garcia-Hernandez, M. G.; Ruiz-Pinales, J.; Ledesma-Orozco, S.; Avina-Cervantes, G.; Onaindia, E.; Reyes-Ballesteros, A., "Combination of acceleration procedures for solving stochastic shortest-path Markov decision processes," *International Conference on Intelligent Systems and Knowledge Engineering (ISKE), 2010*, pp.89,94, 15-16 Nov. 2010

[19] Gautam, S., "Swarm Routing Protocol for Mobile Ad Hoc Networks," *Second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT), 2010*, pp.94,96, 2-3 Dec. 2010

[20] Bitam, S.; Mellouk, A., "QoS Swarm Bee Routing Protocol for Vehicular Ad Hoc Networks," *Communications (ICC), 2011 IEEE International Conference on* , vol., no., pp.1,5, 5-9 June 2011

[21] Di Caro, Gianni; Ducatelle, F.; Gambardella, L.M., "Swarm intelligence for routing in mobile ad hoc networks," *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, pp.76,83, 8-10 June 2005

[22] Dirani, M.; Altman, Z., "A cooperative Reinforcement Learning approach for Inter-Cell Interference Coordination in OFDMA cellular networks," *Proceedings of the 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010*,pp.170,176, May 31 2010-June 4 2010

[23] Ji-Hwan Son; Hyo-Sung Ahn, "Cooperative Reinforcement Learning: Brief Survey and Application to Bio-insect and Artificial Robot Interaction," *IEEE/ASME International Conference on Mechtronic and Embedded Systems and Applications, 2008. MESA 2008.* pp.71,76, 12-15 Oct. 2008

[24] Kao-Shing Hwang; Jeng-Yih Chiou; Tse-Yu Chen, "Cooperative reinforcement learning based on zero-sum games," *SICE Annual Conference, 2008* , pp.2973,2976, 20-22 Aug. 2008