Indonesian Journal of Electrical Engineering and Computer Science
Vol. 2, No. 3, June 2016, pp. 684 ~ 694
DOI: 10.11591/ijeecs.v2.i3.pp648-694

Analysis of Reinforcement Based Adaptive Routing in MANET

Rahul Desai*[1], B P Patil[2]
[1]Sinhgad College of Engineering, Army Institute of Technology, Pune, Maharashtra, India
[2]E & TC Department, Army Institute of Technology, Pune, Maharashtra, India
*Corresponding author, e-mail: desaimrahul@yahoo.com[1], bp_patil@rediffmail.com[2]

Abstract

This paper describes and evaluates the performance of various reinforcement learning algorithms with shortest path algorithms that are widely used for routing packets through the network. Shortest path routing is the simplest policy used for routing the packets along the path having minimum number of hops. In high traffic or high mobility conditions, the shortest path get flooded with huge number of packets and congestions occurs, So such shortest path does not provides the shortest path and increases delay for reaching the packets to the destination. Reinforcement learning algorithms are adaptive algorithms where the path is selected based on the traffic present on the network at real time. Thus they guarantee the least delivery time to reach the packets to the destination. Analysis done on a 6 by 6 irregular grid and sample ad hoc network shows that performance parameters used for judging the network - packet delivery ratio and delay provides optimum results using reinforcement learning algorithms.

Keywords*: Ad Hoc Network, AODV, AOMDV, DSDV, DSR, CQ Routing, DRQ Routing, Q Routing*

1. Introduction

Information is transmitted in the network in form of packets. Routing is the process of transmitting these packets from one network to another. While transmitting the packets from source to the destination, a number of intermediate hops came in picture. Various performance parameters are used to judge the quality of routing such as delay, packet delivery ratio, control overhead, throughput, jitter etc. Some of the most important parameter used for judging the quality is the delay and packet delivery ratio. Different routing algorithms such as shortest path routing, bellman ford algorithms are used. The most simplest and effective policy used is the shortest path routing. In shortest path routing the path with minimum number of hops is selected to deliver the packet from source to the destination. In shortest path routing, cost table and neighbor tables are present to store the appropriate information and tables are exchanged frequently for adaptation purpose.

The shortest path routing policy is good and found effective for less number of nodes and less traffic present on the network. But this policy is not always good as there are some intermediate nodes present in the network that are always get flooded with huge number of packets. Such routes are referred as popular routes. In such cases, it is always better to select the alternate path for transmitting the packets. This path may not be shortest in terms of number of hops, but this path definitely results in minimum delivery time to reach the packets to the destination because of less traffic on those routes. Such routes are dynamically selected in real time based on the actual traffic present on the network. Hence when the more traffic is present on some popular routes, some unpopular routes must be selected for delivering the packets. This is the main motivating factor for designing and implementing various adaptive routing algorithms on a network.

Learning such effective policy for deciding routes online is major challenge, as the decision of selecting routes must be taken in real time and packets are diverted on some unpopular routes. The main goal is to optimize the delivery time for the packets to reach to the destination and preventing the network to go into the congestion. There is no training signal available for deciding optimum policy at run time, instead decision must be taken when the packets are routed and packets reaches to the destination on popular routes.

Received March 7, 2016; Revised May 3, 2016; Accepted May 19, 2016

Ad Hoc networks are infrastructure less networks. These are consisting of mobiles nodes which are moving randomly. Figure 1 show an ad hoc network where multiple hops are used to deliver the packets to the destination. Routing protocols for an ad hoc network are generally classified into two types - Proactive and On Demand. Proactive protocols which are are table driven routing protocols which attempt to maintain consistent, up to date routing information from each node to every other node in the network. These protocols require each node to maintain one or more tables to store routing information and they respond to changes in network topology by exchanging updates throughout the network.

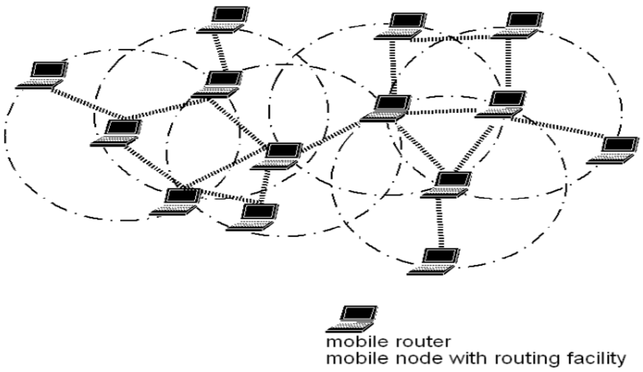

Figure 1. Example of Mobile Ad Hoc Network

DSDV is one of the proactive routing protocols developed for an ad hoc network. This is based on distance vector routing protocol and uses destination sequence numbers to avoid count to infinity problem. Second type of routing protocol on an ad hoc network is on demand routing protocols which are also known as reactive routing protocols. These routing protocols maintain routes whenever required.

The Dynamic Source Routing Protocol (DSR) is one of the on demands routing protocol which is characterized by the use of source routing. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a route cache. Ad Hoc on Demand Distance Vector Routing (AODV) [1, 2] is also on-demand routing protocol. AODV uses traditional routing tables, one entry per destination. This is in contrast to DSR, where DSR maintains multiple route cache entries for each destination. Being a single path protocol, it had to invoke a new route discovery process whenever this single path fails. To overcome this limitation, another Multipath extension to AODV called Ad Hoc On-Demand Multipath Distance Vector (AOMDV) is used.

The results obtained from analysis of various proactive and on-demand routing protocols [3, 4] shows, for low loads and low mobility, proactive protocols such as DSDV and OLSR gives better results. End-to-end delay is minimum in DSDV and OLSR protocols. On demand Routing protocols such as AODV, DSR and AOMDV are more effective in high traffic diversity as well as high mobility. Packet delivery ratio is highest in case of AODV and DSR Protocols. AOMDV also produces similar results as that of AODV but it produces larger over heads. In AODV protocol congestion occurs in selected shortest path and eventually it starts dropping the packets. Thus AODV and AOMDV protocol gives good performance at low loads but at high mobility and heavy load situations, both of them fail to work.

2. Literature Survey of Various Reinforcement Learning Algorithms – Problems and Proposed Solution

Reinforcement learning is learning where the mapping between situations to actions is carried out so as to maximize a numerical reward signal [5-6]. Reinforcement Learning is used for dynamically selecting path in crowded environment in the network. The learner is not told which actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In the most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics--trial-and-error search and delayed reward--are the two most important distinguishing features of reinforcement learning.

Reinforcement learning is defined not by characterizing learning methods, but by characterizing a learning problem. Basic idea is simply to capture the most important aspects of the real problem facing a learning agent interacting with its environment to achieve a goal. Such an agent must be able to sense the state of the environment to some extent and must be able to take actions that affect the state. The agent also must have a goal or goals relating to the state of the environment. The formulation is intended to include just these three aspects sensation, action, and goal in their simplest possible forms without trivializing any of them.

One of the challenges that arise in reinforcement learning is the trade-off between exploration and exploitation. To obtain a lot of reward, a reinforcement learning agent must prefer actions that it has tried in the past and found to be effective in producing reward. But to discover such actions, it has to try actions that it has not selected before. The agent has to exploit what it already knows in order to obtain reward, but it also has to explore in order to make better action selections in the future. The agent must try a variety of actions and progressively favor those that appear to be best. On a stochastic task, each action must be tried many times to gain a reliable estimate its expected reward [7].

There are two approaches to learning a controller for a given task. In model-based approach, the learning agent must first learn a model of the environment and use this knowledge to learn an effective control policy for the task, while in the model-free approach a controller is learned directly from the actual outcomes [7-8]. Reinforcement learning is an example of the model-based approach.

Q Routing [9-11] is reinforcement based learning algorithm. It is based on the Q learning principle [12-13] in order to learn the optimal path to the destination. Each node in the network has a reinforcement learning module in order to dynamically determine the optimum path to the destination. In order to implement regular adaptive routing, there is a need for a training signal to evaluate or improve the routing policy, which cannot be generated until the packet reaches the final destination. However, using reinforcement learning, the updates can be made more quickly and using only local information. Figure 2 illustrates the basic process of reinforcement learning. Let Qx(y, d) be the time that a node x estimates it takes to deliver a packet P bound for node d by way of x's neighbor node y including any time that P would have to spend in node x's queue. Upon sending P to y, x immediately gets back y's estimate for the time remaining in the trip [10-11].

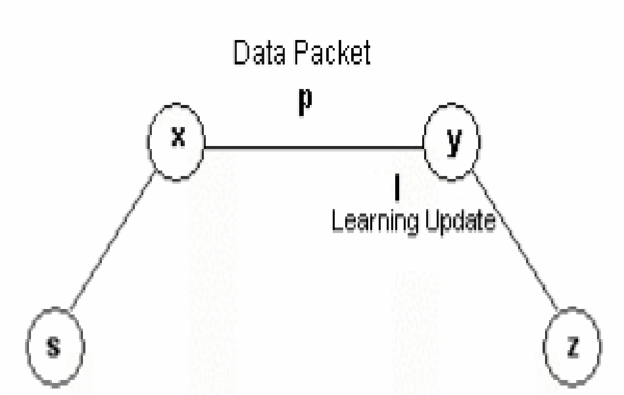

Figure 2. Reinforcement Learning

In Q Routing, each node maintains information about Q values for each of the possible next hops. These Q values represents the delivery time for the packets to reach to the destination. For every packet, the node makes a choice of the next hop that has the least estimate of the time it takes to reach the destination. Also, an update is also sent to the previous node regarding the present Q value. In order to keep the Q value as close to the actual values as possible and to reflect the changes in the state of the network, the Q value estimates need to be updated with minimum possible overhead [10-12]. Figure 3 shows Q routing forward exploration. As soon as the node X sends a packet P (S, D) destined for node D to one of the neighboring nodes Y, node Y send back to node X, its best estimate Q_y (Z, D) for the destination D. This value essentially estimates the remaining time in the journey of packet P (S, D). Upon receiving Q_y (Z, D), node X computes the new estimate. The exploration involved in updating the Q value of the sending node X using the information obtained from the receiving node Y, is referred to as forward exploration. With every hop of the packet P (S, D), one Q value is updated [11].

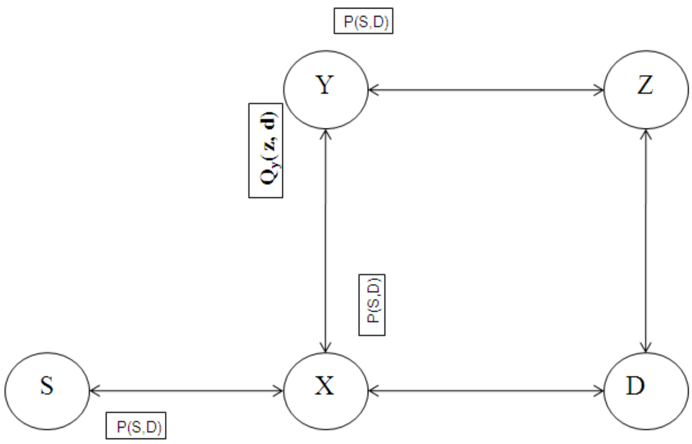

Figure 3. Q routing Forward Exploration

In another optimized form, Confidence Based Q Routing, each Q value Qx(Y, D) in the network is associated with a measure of confidence Cx(Y, D), which is a real number between 0 and 1. A value of 1 means that there is full confidence in the corresponding Q value and that this Q value reflects the current state of the network (completely reliable). In other words, this Q value has recently been updated. A value of 0, on the other hand, means that the corresponding Q value is random and does not necessarily reflect anything about the current state of the network. In other words, this Q value has never been updated. All Intermediate nodes along with Q value, also transmits Confidence value which will updated in confidence table. Figure 4 shows Confidence based Q routing forward exploration.

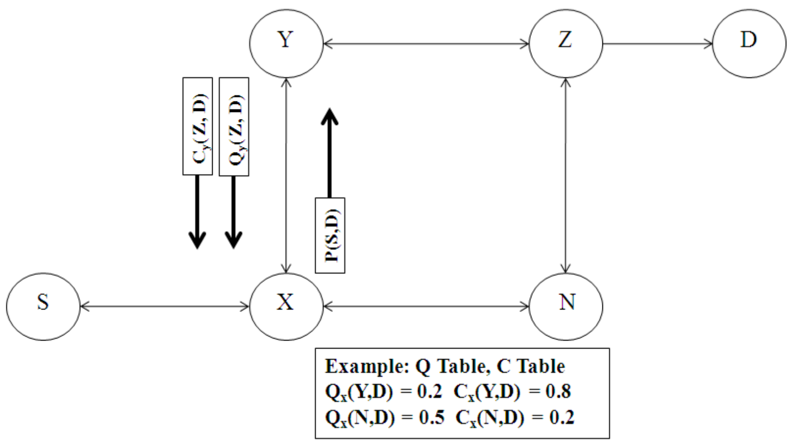

Figure 4. Confidence Q routing

Dual reinforcement Q Routing (DRQ) 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. Instead of trying to use the single reinforcement signal, an indirect reinforcement signal is extracted from the incoming information and is used to update the local decision maker. When a node X sends a packet to neighboring node Y, some additional routing information can be sent along with the packet. This information can be used to update node Y's decisions in the direction opposite to the direction of the packet. This update adds backward exploration to Q-Routing. Figure 5 shows forward and backward exploration involved in Q learning process [14].

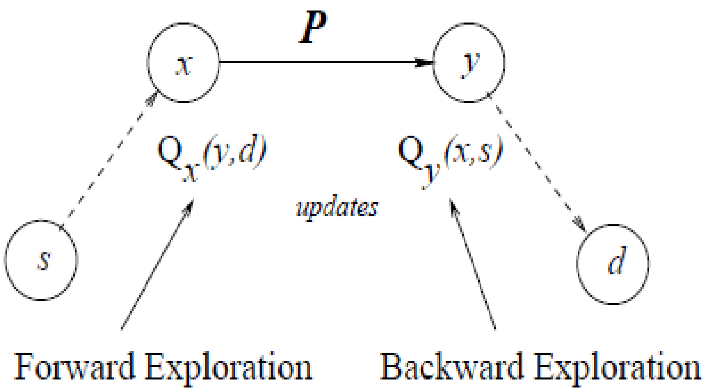

Figure 5. Forward and Backward Exploration

The Q-Routing algorithm makes use of forward exploration in updating the Q-values in the network. In forward exploration, the Q-value of the sending node is updated based on the information coming from the receiving node. DRQ routing [14] makes use of backward exploration as well. When a node X sends a packet P(S, D) to one of its neighbors Y, the packet can take along information about the Q-values of node X. When node Y receives this packet, it can use this information in updating its Q-values pertaining to the neighbor X. Later when node Y has to make a decision, it can use the updated Q values for X. Q value updates in backward exploration are more accurate than Q value updates in forward exploration. Figure 6 shows Dual reinforcement Q routing which involves backward exploration.

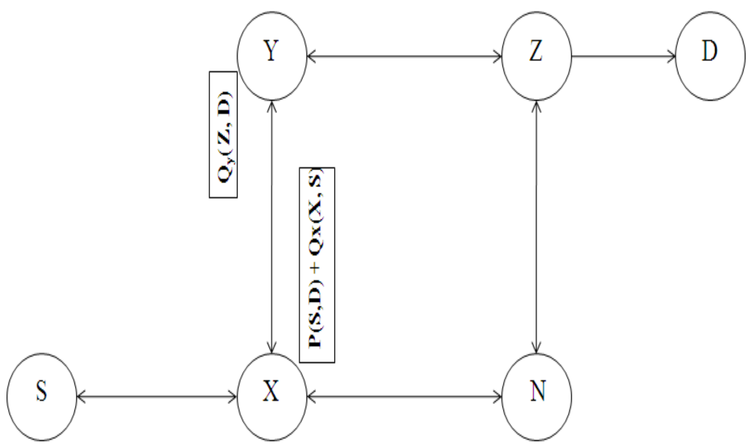

Figure 6. Backward Exploration

3. Results and Discussion

Network Simulator NS2 is used for experimentation. NS2 is the standard network simulator used for analysis of wired and wireless networks. Two different experiments are performed to judge the quality of reinforcement learning algorithms using different performance parameters, in first experiment 6×6 irregular grid is used to test the performance of reinforcement learning for random traffic. Second experiment is performed on an ad hoc network consisting of 10 to 100 nodes with random mobility of nodes and random traffic generated on the network.

In first experiment, the network topology used is the 6×6 irregular grid shown in the figure 7. Figure 7 shows the simulation of the same. In this network there are two possible ways of routing packets between the left cluster (nodes 1 through 18) and the right cluster (nodes 19 through 36): the route including nodes 12 and 25 (R1) and the route including nodes 18 and 19 (R2). For every pair of source and destination nodes in different clusters, either of the two routes, R1 or R2 can be chosen. Figure 8 shows a simulation of the 6×6 irregular grid in NS2.

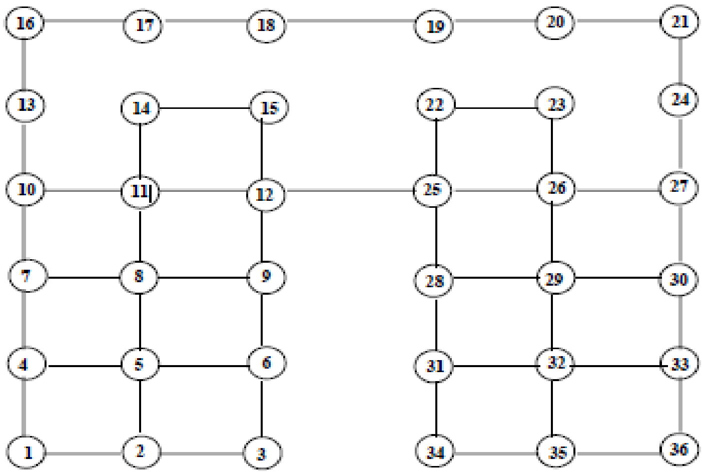

Figure 7. The 6×6 Irregular Grid

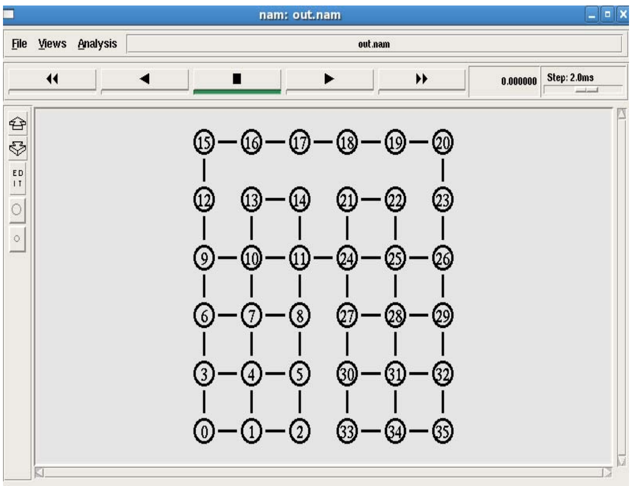

Figure 8. NS2 simulation for 6×6 Irregular Grid

Route R1 is always selected by shortest path routing. For low loads (1 packet/simulation time), shortest path routing is giving best results. Average packet delivery time is very less as compared with reinforcement learning methods (figure 9)

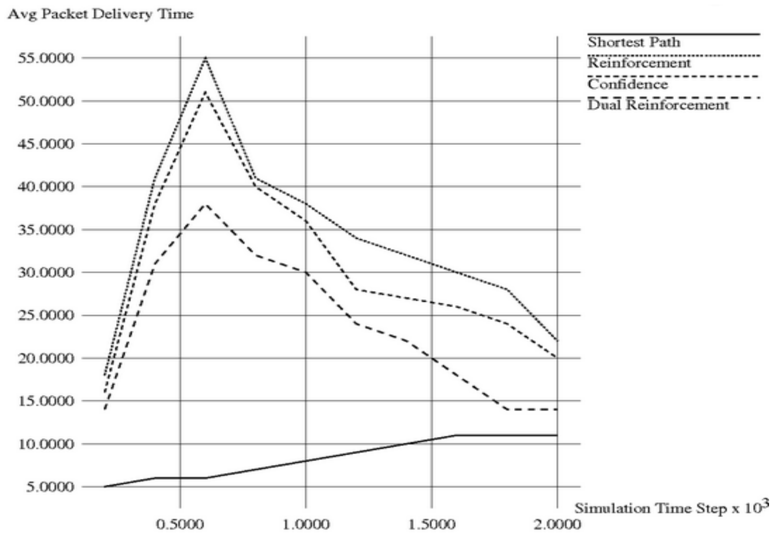

Figure 9. Average Packet Delivery Time vs. Simulation Time step for low loads

At medium (2 packets/simulation time) (figure 10), or high load conditions (3 packets/simulation time) (figure 11), imposed over the network, it is found that the shortest path routing breaks down and the average packet delivery time grows linearly as the simulation time progresses. This is because the packet queues at particular nodes 12 and 25 increases without bound. A lot of queue waiting time is incurred by packets going through these nodes. In reinforcement routing, simulation time steps 1500 to 2000 are required to find out the optimum paths, and there after they settle down to most stable routing policy.

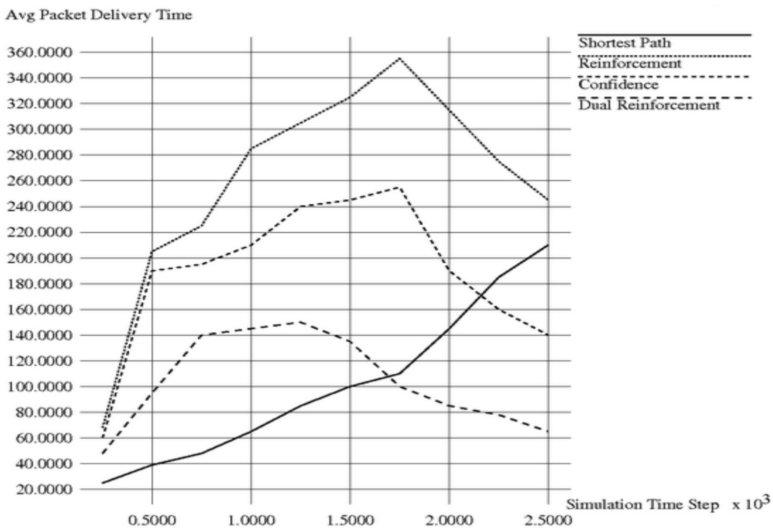

Figure 10. Average Packet Delivery Time vs. Simulation Time step for Medium loads

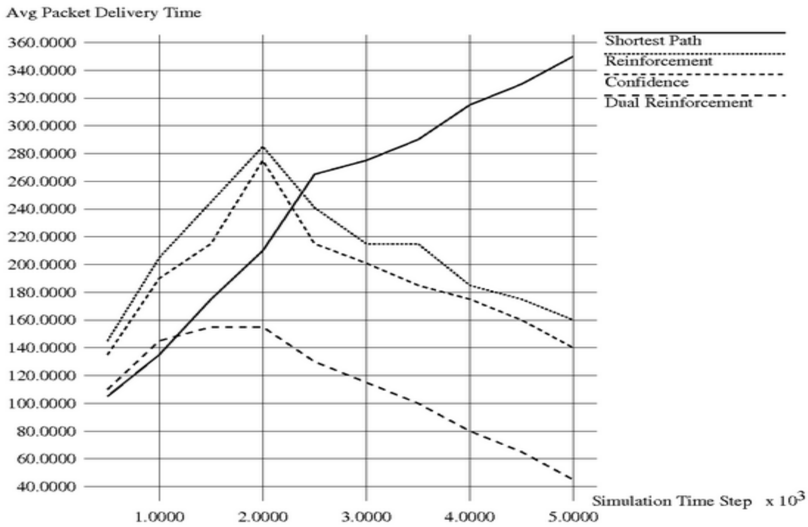

Figure 11. Average Packet Delivery Time vs. Simulation Time step for High loads

Second experiment is carried out on an ad hoc network and various reinforcement routing algorithms are compared with existing routing protocols such as AODV, DSR and AOMDV.

Table 1. Simulation Parameters Used for Analysis

Parameter	Value
No of nodes	10 to 100 nodes
Mobility model	Random Way Point Mobility Model
Simulation time	200 s
Topology Size	800 m × 800 m
Routing protocols analyzed	DSR, AODV, AOMDV, Q Routing, Confidence Q Routing and Dual Reinforcement Routing
Mobility rate	25m/s to 125 m/s
Pause time	0, 50, 100, 150 and 200 s

The various parameters used in second experiment for analysis of reinforcement learning are shown in Table 1. Packet delivery ratio is analyzed for medium size network (50 nodes) and packet rate changing from 25 m/s to 125 m/s. It is observed as the mobility rate increases, routing protocols such as AOMDV and DSR protocols starts dropping the packets as it becomes difficult for them to adapt the changes in the network in short amount of time. For obtaining the consistent packet ratio for medium size network, reinforcement routing works better than existing protocols. Figure 13 shows the performance of an ad hoc network while changing the number of nodes at low load network.

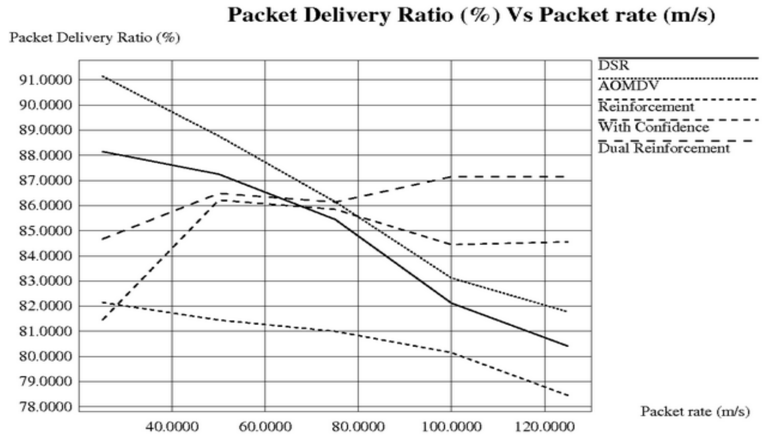

Figure 12. Packet Delivery Ratio vs. Packet Rate

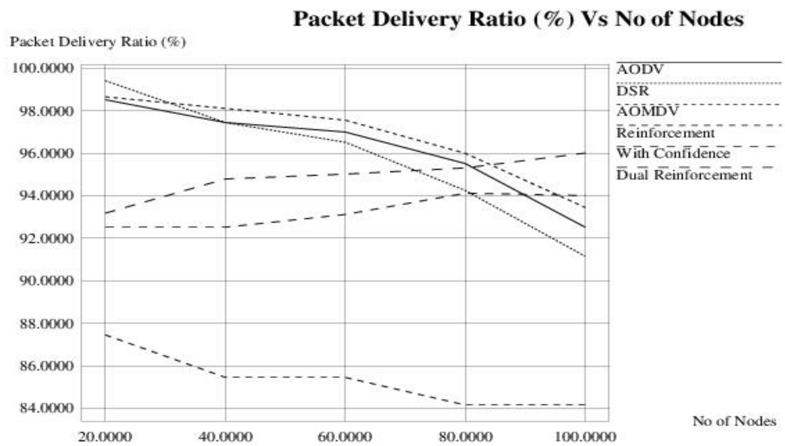

Figure 13. Packet Delivery Ratio vs. No of Nodes

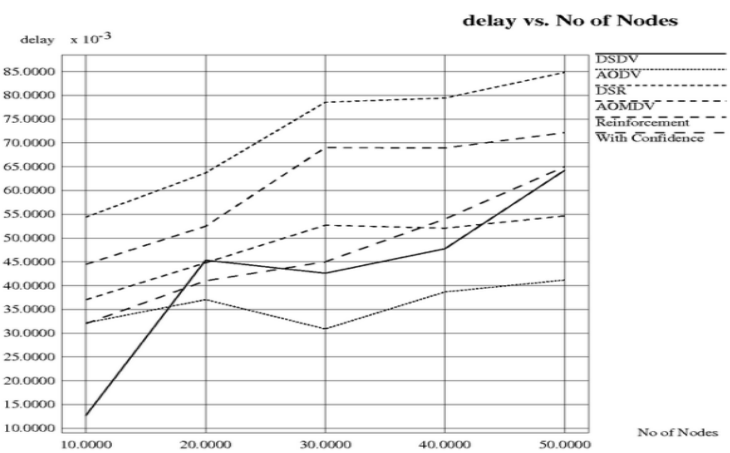

Figure 14. Delay vs. No of Nodes

It is observed that as we start increasing the number of nodes, (figure 14) delay will starts increasing. DSDV as a proactive routing protocol provides minimum delay, but dual reinforcement Q routing provides prominent results as compared with AODV, DSR and AOMDV.

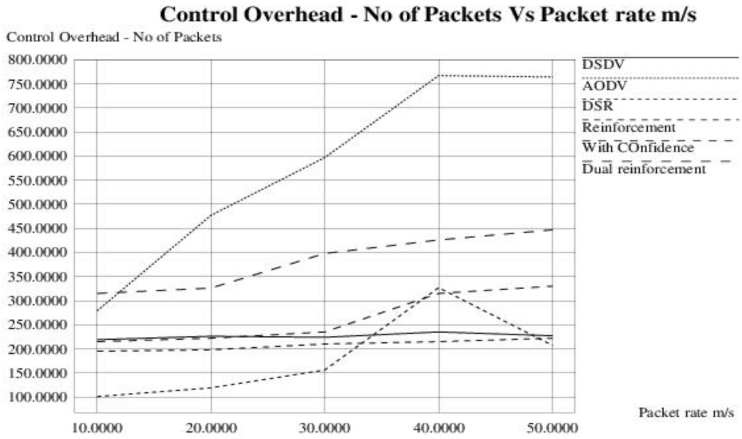

Figure 15. Control Overhead vs. Packet rate

As far as control overhead is considered (figure 15) it is observed that in proactive protocols such as DSDV the routing overhead is high. AOMDV protocol though it is on demand routing protocol, routing overhead is high as it the extension of AODV protocol to find the maximum number of path between source and destination. Reinforcement routing generates less and consistently equal number of packets throughout the network.

4. Conclusion

In this paper, analysis and comparison of reinforcement routing, Confidence based reinforcement routing and dual reinforcement routing were presented. Confidence based reinforcement and Dual reinforcement routing are showing prominent results as compared with shortest path routing for medium and high load conditions. At high loads, dual reinforcement Q routing performs more than twice a fast as Q-Routing. For an ad hoc network, In AODV protocol congestion occurs in selected shortest path and eventually it starts dropping the packets. Thus AODV and AOMDV protocol gives good performance at low loads but at high mobility and heavy load situations, both of them fail to work. In dual reinforcement routing, as backward exploration is involved including confidence measure, less time is required in order to settle down the Q values thus they more accurately predict the state of network at run time. It is found that, though mobility rate changes at high rate as well as high traffic, dual reinforcement routing obtains more accurate result as compared with Q routing.

References

[1] G Vetrichelvi and G Mohankumar. “Performance Analysis of Load Minimization in AODV and FSR”. *International Journal of Information and Network Security (IJINS)*. 2012; 1(3): 152-157.
[2] Malek Al-Gabri, Chunlin LI and Layuan Li. “Improving ZigBee AODV Mesh Routing Algorithm Topology and simulation analysis”. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(2): 1528-1535.
[3] R Desai and BP Patil. "Analysis of routing protocols for Ad Hoc Networks". *Circuits, Systems, Communication and Information Technology Applications (CSCITA), 2014 International Conference on*, Mumbai. doi: 10.1109/CSCITA.2014.6839244. 2014: 111-115.
[4] Rahul Desai, BP Patil. “*Performance Analysis of Ad Hoc Routing Protocols*”. 3rd International Conference on Recent Trends in Engineering & Technology (ICRTET’2014). ISBN No.: 978-93-5107-220-1. 2014: 11-15.
[5] Richard S Sutton and Andrew G Barto. “Reinforcement Learning: An Introduction”. A Bradford Book, the MIT Press Cambridge, Massachusetts London, England.

[6] Bing Xia, Wahab MH, Yang Yang, Zhong Fan, Sooriyabandara M. "Reinforcement learning based spectrum-aware routing in multi-hop cognitive radio networks". 4th International Conference on Cognitive Radio Oriented Wireless Networks and Communications. CROWNCOM '09, doi: 10.1109/CROWNCOM.2009.5189189. 2009; 1(5): 22-24.

[7] Ouzecki D, Jevtic D. "*Reinforcement learning as adaptive network routing of mobile agents*". MIPRO, 2010 Proceedings of the 33rd International Convention. 2010: 479,484.

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

[9] Rahul Desai, BP Patil. "MANET with Q Routing Protocol". *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)* – 12-382. 2013: 255, 261.

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

[11] Kumar S. "Confidence based Dual Reinforcement Q-Routing: An On-line Adaptive Network Routing Algorithm". Master's thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX-78712, USA Tech. Report AI98-267. 1998.

[12] Xin Li, Chuanzhi Zang, Wenwei Liu, Peng Zeng, Haibin Yu. "Metropolis Criterion Based Fuzzy QLearning Energy Management for Smart Grids". *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(8): 1956-1962.

[13] Yunfei Zhang, Clarence W de Silva. "RSMDPbased Robust Qlearning for Optimal Path Planning in a Dynamic Environment". *IAES International Journal of Robotics and Automation (IJRA)*. 2014; 3(1): 1-12.

[14] Kumar S, Miikkulainen R. "Dual Reinforcement Q-Routing: An On-line Adaptive Routing Algorithm," Proc. Proceedings of the Artificial Neural Networks in Engineering Conference. 1997.