

MANET with Q Routing Protocol

Rahul Desai¹, B P Patil²

Asst Professor in Information Technology¹ / Professor in Electronics and telecommunication²

Army Institute of technology
Pune - 411015 Maharashtra, India

Abstract: A Most of the routing algorithms are based on shortest path algorithms or distance vector or link state algorithms, they are not capable of adapting the run time changes such as traffic load, 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 state of the network is considered every time the packets are transmitted from the source. Thus the state of the network depends on a number of network properties like the queue lengths of all the nodes, the condition of all the links and nodes (whether they are up or down) and so on. Thus Q learning framework of Watkins and Dayan (1989) is used to develop and improve such adaptive routing algorithms. In Q Routing the cost tables are replaced by Q tables, and the interpretation, exploration and updating mechanism of these Q tables are modified to make use of the Q learning framework. This improves the Q Routing algorithm further by improving its quality and quantity of exploration. Q Learning is based on reinforcement Learning which is a popular machine learning technique, which allows an agent to automatically determine the optimal behavior to achieve a specific goal based on the positive or negative feedbacks it receives from the environment after taking an action. This paper describes Q routing protocols over Mobile ad hoc networks. Ad hoc network are wireless network with no infrastructure support. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within the direct reach.

Keywords: Ad Hoc Networks, MANET, DSDV, OLSR, WRP, Q Routing etc.

I. Introduction To MANET's

An ad Hoc network is a collection of wireless mobile nodes thus dynamically forming a temporary network without the use of existing network infrastructure or centralized administration. Because of limited transmission range of wireless network interfaces, multiple network hops may be needed for one node to exchange data with another across the network. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within the direct reach. Each node participates in an ad hoc routing protocol that allows it to discover multi hop paths through the network to any other node. The idea of an Ad hoc network is sometimes also called as an infrastructure less network, since the mobile hosts in the network dynamically establish routing among themselves to form their own network on the fly. These networks are typically set up for a limited period of time. The routing protocols are also tuned to the particular applications. Thus "mobile ad hoc network" (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links forming an arbitrary graph.

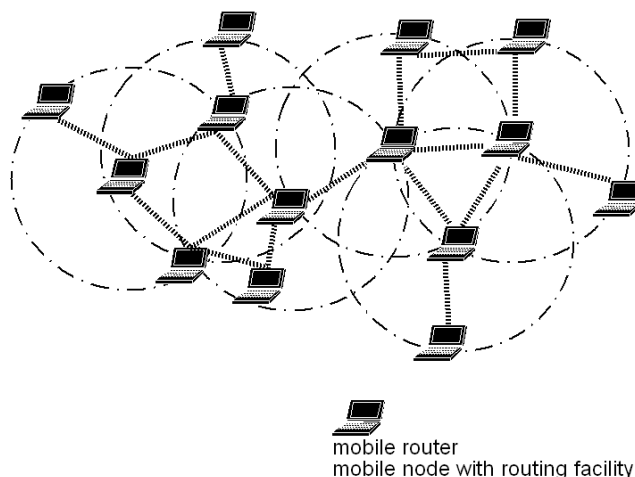


Figure 1. Example of Mobile Ad Hoc Network

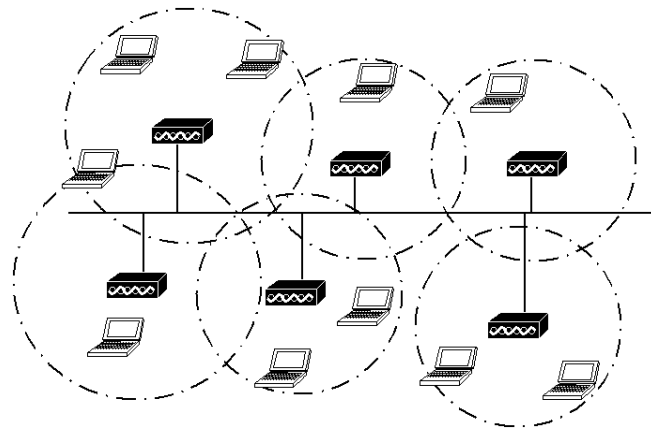


Figure 2. Example of Infrastructure Network

The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. Because of its mobility and non-infrastructure nature, the ad hoc network poses various new design requirements; the first is self-configuration (of addresses and routing).

This work is ordered as follows. We described survey of existing routing protocols/algorithms in section II, introduction and detailed description of Q routing protocol in section III, performance comparison of Q routing protocol over existing routing protocol in section IV, optimization techniques used over Q routing protocol in section V and finally section VI specifies the conclusion.

II. Survey of existing routing protocols over manet

In MANET, nodes are not aware of each other hence they need to discover each other by broadcasting to neighboring nodes their presence. They also need to listen for other broadcasts in case a new node is added. The nodes might not only broadcast their own information, but are able to broadcast how to reach other nodes as well. Since there are many possibilities to design an ad hoc network, many types of related protocols specified for such network were discovered. Some of them are enhancements over others, and others are a combination of protocols.

A. Pro-active Routing

Proactive protocols maintain unicast routes between all pairs of nodes regardless of whether all routes are actually used. Therefore, when the need arises, the traffic source has a route readily available and does not have to incur any delay for route discovery. These protocols also can also find optimal routes (shortest path) given a model of link costs. However, these protocols are not directly suitable for resource poor and mobile ad hoc networks because of their high overheads and/or somewhat poor convergence behavior. Therefore, several optimized variations of these protocols have been proposed for use in ad hoc networks. These protocols are broadly classified into the two traditional categories: distance vector and link state. Destination sequenced Distance Vector (DSDV) [1] was one of the earliest protocols developed for ad hoc networks. Primarily design goal of DSDV was to develop a protocol that preserves the simplicity of RIP, while guaranteeing loop freedom. The main idea in DSDV is the use of destination sequence numbers to achieve loop freedom without any inter-nodal coordination. The distance/metric information for every destination, typically exchanged via routing updates among neighbors in distance vector protocols, is tagged with the corresponding destinations sequence number. DSDV also uses triggered incremental routing updates between periodic full updates to quickly propagate information about route changes.

Wireless Routing Protocol (WRP) [1] is another distance vector protocol optimized for ad hoc networks. The algorithms of this class use the next hop and second-to-last hop information to overcome the counting-to-infinity problem; this information is sufficient to locally determine the shortest path spanning tree at each node. Optimized List State Routing (OLSR) [2] is an optimized version of traditional link state protocol such as OSPF. It uses the concept of Multipoint relays (MPRs) to efficiently disseminate link state updates across the network. Only the nodes selected as MPRs by some node are allowed to generate link state updates. Moreover, link updates contain only the links between MPR nodes and their MPR-selectors in order to keep the update size small. Thus, only partial topology information is made available at each node. However, this information is

sufficient for each node to locally compute shortest hop path to every other node because at least one such path consists of only MPR nodes. OLSR also uses only periodic updates for link state dissemination.

B. On Demand Routing

A different approach from table driven routing is source initiated on demand routing. Main idea in on demand routing is to find and maintain only needed routes. The obvious advantage with discovering routes on demand is to avoid incurring the cost of maintaining routes that are not used. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined. Once a route has been established, it is maintained by a route maintenance procedure until either the destination becomes inaccessible along every path from the source or until the route is no longer designed. Reactive means discover route only when you need it. This saves energy and bandwidth during inactivity but congestion occurs during high activity. Significant delay might occur as a result of route discovery. It is good for light loads but collapse in large loads.

The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks (DSR) [3,4] 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) [1,5, 6] is pure on-demand routing protocol. It shares DSR's on demand characteristic in that it also discovers routes on an "as needed" basis via a similar route discovery process. However, AODV adopts a very different mechanism to maintain routing information. It uses traditional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route cache entries for each destination.

III. Q routing protocol

In section II, various conventional routing algorithms, both proactive and on demand were discussed in detail. In this section, new adaptive routing algorithms which are based on reinforcement learning approach, called as Q-routing (Littman and Boyan 1993a; Littman and Boyan 1993b; Boyan and Littman 1994) [9,10], is discussed in detail.

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. Reinforcement learning is an example of the model-based approach which is used for the task of adaptive network routing. Here the model of the system is learned in terms of Q-values. Each Q-value is of the form $Q(s, a)$ representing the expected reinforcement of taking action a in state s . Thus in state s , if the Q-values are learned to model the system accurately, the best action is the one with the highest Q-value in the vector $Q(s, *)$. The Q-values are learned using an update rule that makes use of the current reinforcement $R(s; a)$ computed by the environment and some function of Q-values of the state reached by taking action a in state s . In the Q routing algorithm [11], Q learning is used to learn the task of finding an optimal routing policy given the current state of the network. The state of the network is learned in terms of Q-values distributed over all nodes in the network.

In Q routing, Q learning is used to first learn a representation of the state of the network in terms of Q values and then these values are used to make control decisions. The task of Q learning is to learn an optimal routing policy for the network. The state s in the optimization problem of network routing is represented by the Q values in the entire network. 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 neighbor y such that it takes minimum time for a packet destined for node d to reach its destination if sent via neighbor y . In Q-Routing each node x maintains a table of Q values $Q_x(y, d)$, where d is the destination node and y is the neighbor node. According to Boyan and Littman (1994), the value $Q_x(y, d)$ can be interpreted as "Q_x(y, d) is node x's best estimated time that a packet would take to reach its destination node d from node x when sent via its neighboring node y excluding total waiting time and transmission delays over the entire path that it would take starting from node y ."

The base case values for this table are: $Q_x(y, x) = 1$ for all $y \in N(x)$, that is, if a packet is already at its destination node, it should not be sent out to any neighboring node. And also $Q_x(y, y) = \partial$, in other words, a packet can reach its neighboring node in one hop. ∂ is the transmission delay over the link from node x to y . In the steady state, when the Q values in all the nodes represent the true state of network, the Q values of neighboring nodes x and y should satisfy the following relationships:

A) The general inequality: $Q_x(y, d) = q_y + \hat{\partial} + Q_y(z, d)$

This equation essentially states that if a packet destined for node d , currently at node x , is sent via x 's neighbor y , then the maximum amount of time it will take to reach its destination is bound by the sum of three quantities: (1) the waiting time q_y in the packet queue of node y , (2) the transmission delay $\hat{\partial}$ over the link from node x to y , and (3) the time $Q_y(z, d)$ it would take for node y to send this packet to its destination via any of node y 's neighbors (z).

B) The optimal triangular equality: $Q_x(y, d) = q_y + \hat{\partial} + Q_y(\wedge z, d)$

This equation is a special case of the above general inequality and it states that the minimum time taken to deliver a packet currently at node x and destined for node d and via any of neighbor $y \in N(x)$, is the sum of three components: (1) the time this packet spends in node y 's queue, (2) the transmission delay $\hat{\partial}$ between node x and y , and (3) the best time $Q_y(\wedge z, d)$ of node y for the destination d . The general inequality is used later to formalize the notion of an admissible (or valid) Q value update. The triangular equality is used to compute the estimated Q value for the update rules.

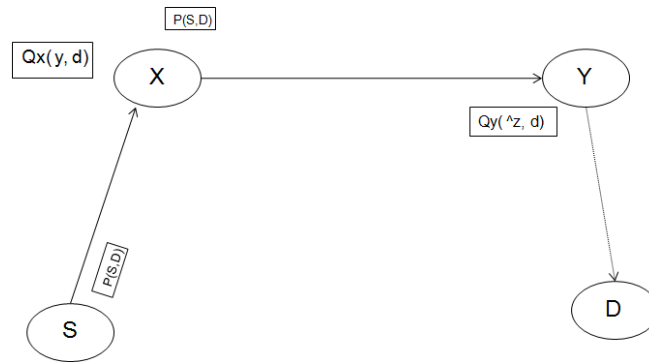


Figure 3. Q Routing Example

When a node x receives a packet $P(s, d)$ destined for node d , node x looks at the vector $Q_x(*, d)$ of Q values and selects that neighboring node $\wedge y$ for which the $Q_x(\wedge y, d)$ value is minimum. This is called the minimum selector rule. This way, node x makes a locally greedy decision by sending the packet to that neighbor from which this packet would reach its destination as quickly as possible. It is important to note, however, that these Q values are not exact. They are just estimates and the routing decision based on these estimates does not necessarily give the best solution. The routing decisions are locally optimal only with respect to the estimates of the Q values at these nodes and so is the overall routing policy that emerges from these local decisions. In other words the control action (the routing decision) is only as good as the model of the network represented by the Q values in the network. The closer these estimates are to the actual values, the closer the routing decision is to the optimal routing decisions.

In order to keep the Q value estimates 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. Boyan and Littman (1994) proposed the following update mechanism, constituting the Q Routing algorithm. As soon as the node x sends a packet $P(s, d)$, destined for node d , to one of its neighboring nodes y , node y sends back to node x its best estimate $Q_y(\wedge z, d)$ for the destination d . This value essentially estimates the remaining time in the journey of packet $P(s, d)$. Upon receiving $Q_y(\wedge z, d)$, node x computes the new estimate for $Q_x(y, d)$ as follows:

$$Q_x(y, d)^{\text{est}} = Q_y(\wedge z, d) + q_y + \hat{\partial}$$

Using the estimate $Q_x(y, d)^{\text{est}}$, node x updates its $Q_x(y, d)$ value as follows:

$$Q_x(y, d)^{\text{new}} = Q_x(y, d)^{\text{old}} + \eta f(Q_x(y, d)^{\text{est}} - Q_x(y, d)^{\text{old}})$$

where ηf is the learning rate.

$$Q_x(y, d) = Q_x(y, d) + \eta f((Q_y(\wedge z, d) + q_y + \hat{\partial}) - Q_x(y, d))$$

When the learning rate ηf is set to 1, the update rule reduces to the optimal triangular equality. Since the value $Q_y(\wedge z, d)$ and others from which it was derived (the $Q_y(*, d)$ vector) were not accurate, the learning rate is set to some value less than 1 (e.g. 0.8-0.9). 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.

The complete Q Routing algorithm can be summarized in two steps. The $\text{PacketReceive}_y(x)$ step describes what node y does when it receives a packet from its neighboring node x , and the $\text{PacketSend}_x(P(s, d))$ step describes what node x does when it has to send a packet $P(s, d)$ for destination d .

1	If (not EMPTY(PacketQueue(x))) go to step 2
2	$P(s, d) = \text{Dequeue the first packet in the PacketQueue}(x)$.
3	Compute best neighbor $\hat{y} = \min(Q_x(y, d))$
4	ForwardPacket $P(s, d)$ to neighbor $\hat{y}$.
5	Wait for $\hat{y}$'s estimate.
6	ReceiveEstimate $(Q_{\hat{y}}(\hat{z}, d) + q_{\hat{y}})$ from node $\hat{y}$.
7	UpdateQvalue $(Q_x(y, d))$.
8	Get ready to send next packet (goto 1).

Figure 4. $\text{PacketSend}_x(P(s, d))$ at NODE x .

1	Receive a packet $P(s, d)$, for node d from neighbor x
2	Calculate best estimate for node d ; $Q_y(\hat{z}, d)$
3	Send $(Q_y(\hat{z}, d) + q_y)$ back to node x .
4	If $(d = y)$ then ConsumePacket $_y(P(s, d))$ else goto 5.
5	If (PacketQueue(y) is FULL) then DropPacket $(P(s, d))$ else goto 6.
6	AppendPacketToPacketQueue $_y(P(s, d))$
7	Get ready for receiving next packet (goto 1).

Figure 5. $\text{PacketReceive}_x(x)$ at NODE y .

The term overhead refers to the time taken to execute steps in the routing algorithm that would be either completely missing or executed in constant time in the non adaptive shortest path algorithm. The overhead incurred by an adaptive routing algorithm for exploitation and exploration should be carefully analyzed in order to evaluate how feasible it is. There are four distinct overheads associated with each hop of every packet routed in the network: (1) Decision overhead is defined as the time that the sending node x takes to decide what its best neighbor is. (2) Estimate computation overhead is defined as the time taken by the receiving node y in computing the estimate that it sends back when it receives a packet from the sender node. (3) Estimate transmission overhead is defined as the time taken by this estimate to travel from the receiver node x to the sender node y . (4) Q -value update overhead is defined as the time node x takes to update its Q value $Q_x(y, d)$ once the sender node receives the appropriate estimate from the receiving node.

IV. Performance comparison

In section III, the basic framework of Q -Routing was reviewed. In this section, we are going to evaluate for its ability to adapt under various load levels and initial conditions. Comparison of Q -Routing with non-adaptive shortest path routing and Bellman Fort routing are used to highlight the performance of Q -Routing. The various main network properties considered include the network load levels, traffic patterns, network topology and mobility speed. Other metrics used to analyze the performance of the network are: Routing Overhead (RO) or Routing Load, end-to-end Delay, delay jitter, percentage out-of-order delivery, round trip time (RTT), number of data packets dropped, throughput, efficiency and path optimality.

In the first set of experiments the ability of Q -Routing to learn an effective routing policy in terms of average packet delivery time was demonstrated over different load conditions. The conclusion is that Q -Routing is capable of adapting to low and medium load conditions but takes a long time to converge at high load conditions. In another set of experiments Q -Routing was compared with distance vector routing, the Distributed Bellman-Ford algorithm. At low loads the best version of Bellman-Ford (learns the shortest path policy) as will Q -Routing. At medium loads, the more general version of Bellman-Ford must be used, but even then it learns an inferior path compared to Q -Routing. Exploration overhead in Bellman-Ford is 7-10 times that of Q -Routing for same speed of learning.

Moreover, the quality of exploration in Q -Routing is not as good as it could be because the same learning rate (0.85) is used for updating Q -values without any regard as to how closely the estimated Q -value represents the state of the network. The main contribution of this work is to extend the quality and quantity of exploration of

Q-Routing to yield a superior routing algorithm with regard to the speed of adaptation, the quality of final routing policy learned, ability of the routing policy to handle higher load levels and finally their ability to adapt quickly to the changes in network topology and traffic patterns. The resulting algorithm called Confidence based dual reinforcement Q-Routing (CDRQ-Routing) is developed. In next section, the various modified versions of Q routing called as dual reinforcement Q routing, confidence based Q routing and combination of both (confidence and dual reinforcement) are described.

V. Optimization techniques in Q routing

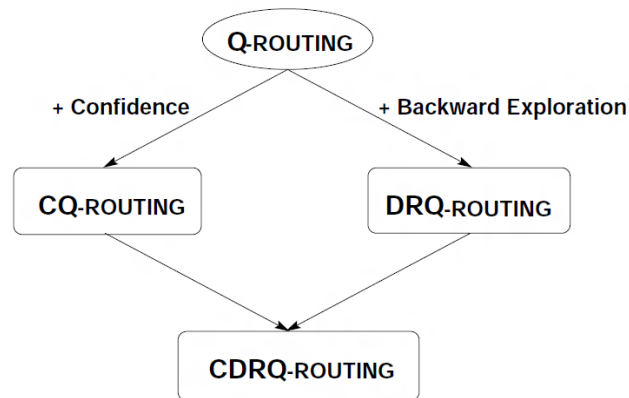


Figure 6. Optimzation of Q Routing.

Inspired by the Q routing algorithm by Boyan and Littman (1994) [12,13], a new adaptive routing algorithm, Confidence based Dual reinforcement Q routing (CDRQ Routing), with higher quality and increased quantity of exploration, is presented. The quality of the routing policy depends mainly on how closely the Q values in the network represent its current state. Therefore, they must be continuously updated. Depending on the network load and traffic patterns, however, some of the Q values may not get updated for a long time. Decisions based on such unreliable Q values are unreliable. In order to quantify the amount of reliability in the Q values, confidence measures are introduced in Q Routing. For every Q value in the network, there is a corresponding confidence value (C value) between 0 and 1. Essentially, a low C value implies that we have low confidence in the corresponding Q value because it has not been updated for a long time. When a Q value with low confidence is to be updated, it is advisable to update it more; in other words, the learning rate for this Q value should be high. Similarly, if the confidence in the new estimate of a Q value is high then also the learning rate should be high. These confidence values are themselves updated so that they decay exponentially with every time step if the corresponding Q value is not updated. On the other hand, if the Q value is updated in the last time step then the corresponding C value should also be updated. Thus every Q value update is associated with a corresponding C value update. Since the learning rate depends on the reliability (C value) of the Q value being updated and that of the estimated Q value, the quality of exploration is improved by updating these Q values more.

Goetz, Kumar, and Miikkulainen (1997) [7,14] developed the Dual Reinforcement Learning algorithm. Instead of trying to use the signal 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. 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 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 . The only overhead is a slight increase in the size of the packets. Q value updates in backward exploration are more accurate than Q value updates in forward exploration. A Q value update is more accurate if the estimate coming from the neighboring node for the update has been updated more recently.

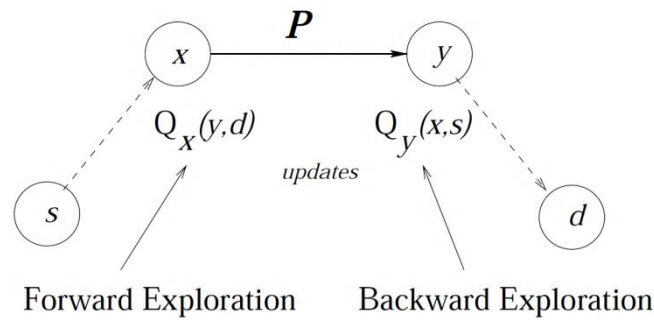


Figure 7. Dual reinforcement Q Routing(DRQ Routing).

In CQ Routing [8,15], the quality of exploration was improved by learning faster when the Q values represent the current state of the network more closely. In DRQ Routing, on the other hand, the quantity of exploration was improved by adding another direction of exploration to Q Routing. CDRQ Routing combines the features of both CQ routing and DRQ Routing. Thus, with each hop of a packet $P(s, d)$ from node x to node y , the Q and C-values of both nodes x and y are updated in the forward and backward exploration, respectively. Like Q Routing, the complete CDRQ Routing algorithm can be summarized in terms of two steps: The $\text{PacketReceive}_y(x)$ step describes what the node y does when it receives a packet from one of its neighboring nodes, x and the PacketSend_x step describes what node x does when it has to send a packet.

VI. Conclusion

Q-routing is always better as compared with existing protocols used over MANET as Q routing considers the network state at run time. Q-routing is easy to implement over fixed networks but it is very difficult to implement over mobile ad hoc networks because of special characteristic of mobile ad hoc networks. Though Q-Routing is better than Bellman Ford algorithms (shortest path routing), still various optimization techniques must be used over Q-routing. Confidence based dual reinforcement (CDRQ routing) performed better than Q routing. The other optimization techniques used could be probabilistic CDRQ routing (PRCQ routing) and Predictive Q routing could be used.

VII. References

- [1] 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.
- [2] T. Clausen et al. Optimized Link State Routing Protocol, <http://www.ietf.org/internet-drafts/draft-ietf-manet-olsr-11.txt>, July 2003, IETF Internet Draft.
- [3] Haseeb Zafar, Nancy Alhamahmy, David Harle and Ivan Andonovic "Survey of Reactive and Hybrid Routing Protocols for Mobile Ad Hoc Networks" International Journal of Communication Networks and Information Security (IJCNIS) Vol. 3, No. 3, December 2011
- [4] 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.
- [5] 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
- [6] S. Ali, A. Ali, "Performance Analysis of AODV, DSR and OLSR in MANET", Master's Thesis, M.10:04, COM/School of Computing, BTH, 2010.
- [7] Shailesh Kumar "Confidence based Dual Reinforcement Q-Routing: an On-line Adaptive Network Routing Algorithm" Report AI98-267 May 1998
- [8] S Kumar, Confidence based Dual Reinforcement Q Routing : An on line Adaptive Network Routing Algorithm. Technical Report, University of Texas, Austin 1998.
- [9] J A Boyan, M L Littman, Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach. In Advances in Neural Information Processing Systems 6 (NIPS6), Morgan Kaufmann, San Francisco, CA pp. 671-678, 1994.
- [10] Davi Kelly, "Reinforcement Learning with Application to Adaptive Network Routing" Journal of Theoretical and Applied Information Technology, 2005 JATIT.
- [11] Ramzi A. Haraty and Badieh Traboulsi "MANET with the Q-Routing Protocol" ICN 2012 : The Eleventh International Conference on Networks
- [12] 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
- [13] Shalabh Bhatnagar, K. Mohan Babu "New Algorithms of the Q-learning type" Science Direct Automatica 44 (2008) 1111-1119. Website: www.sciencedirect.com
- [14] Nicholas Mastrorade and Mihaela van der Schaar "Fast Reinforcement Learning for Energy-Efficient Wireless Communication" IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 59, NO. 12, DECEMBER 2011
- [15] Soon Teck Yap and Mohamed Othman "An Adaptive Algorithm: Enhanced Confidence based Q Routing Algorithm in network Traffic" Malaysian Journal of Computer, Vol. 17 No. 2, December 2004, pp.21-29.

