Asian Journal of Information Technology 12 (1): 20-26, 2013
ISSN: 1682-3915

MANET with Q Routing Protocol

[1]Rahul Desai and [2]B.P. Patil
[1]Department of IT, [2]Department of E and TC, Army Institute of Technology,
Pune, Maharashtra, India

Abstract: 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 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 behaviour to achieve a specific goal based on the positive or negative feedbacks it receives from the environment after taking an action. This study 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.

Key words: Ad hoc networks, MANET, DSDV, OLSR, WRP, Q Routing

INTRODUCTION

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

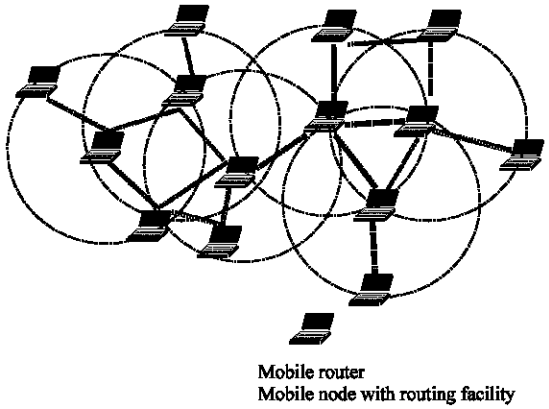

Fig. 1: Example of mobile ad hoc network

(MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links forming an arbitrary graph (Fig. 1 and 2).

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

Corresponding Author: Rahul Desai, Department of IT, Army Institute of Technology, Pune, Maharashtra, India

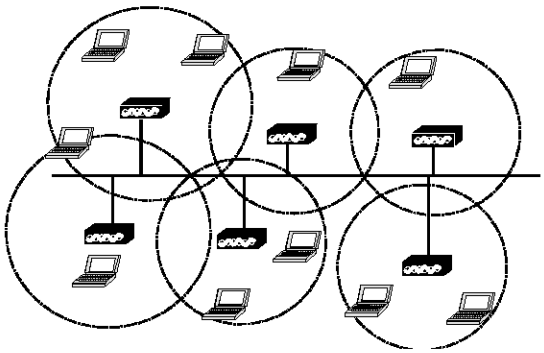

Fig. 2: Example of infrastructre network

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).

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 neighbouring 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.

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) (Toteja *et al*., 2010) 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) (Toteja *et al*., 2010) 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) (Clausen and Jacquet, 2003) 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.

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) (Zafar *et al*., 2011;

Johnson *et al*., 2002) 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) (Toteja *et al*., 2010; Shrivastava and Saluja, 2012; Ali and Ali, 2011) 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.

Q ROUTING PROTOCOL

In this study, various conventional Routing algorithms, both proactive and on demand were discussed. In this study, new adaptive Routing algorithms which are based on reinforcement learning approach called as Q-Routing (Boyan and Littman, 1994; Kelley, 2005) is given.

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 states. 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 (Haraty and Traboulsi, 2012), 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) = ∂ in other words, a packet can reach its neighbouring node in one hop. The ∂ 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 neighbouring nodes x and y should satisfy the following relationships.

The general inequality (Q_x (y, d) = q_y+∂+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 neighbour y then the maximum amount of time it will take to reach its destination is bound by the sum of three quantities: the waiting time q_y in the packet queue of node y, the transmission delay ∂ over the link from node x to y and 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).

The optimal triangular equality (Q_x (y, d) = q_y+∂+Q_y (^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 neighbour $y \in N$ (x) is the sum of three components: the time this packet spends in node y's queue, the transmission delay ∂ between node x and y and the best time Q_y (^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 (Fig. 3).

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 neighbouring node ^y for which the Q_x (^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 neighbour from

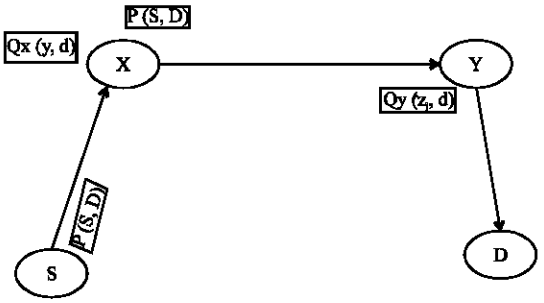

Fig. 3: Q Routing example

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 (^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 for Q_x(y, d) as follows:

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

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

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

Where; ηf is the learning rate:

$$Q_x(y, d) = Q_x(y, d)+\eta f ((Q_y(^z, d)+q_y +\partial)-Q_x(y, d))$$

When the learning rate ηf is set to 1, the update rule reduces to the optimal triangular equality. Since, the value Q_y (^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 <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 PacketReceive$_y$ (x) step describes what node y does when it receives a packet from its neighboring node x and the PacketSend$_x$ (P (s, d)) step describes what node x does when it has to send a packet P (s, d) for destination d.

PacketSend$_x$ p(s, d) at node x:

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

PacketReceive$_x$ (x) at node y:

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

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: decision overhead is defined as the time that the sending node x takes to decide what its best neighbour is. 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. Estimate transmission overhead is defined as the time taken by this estimate to travel from the receiver node x to the sender node y. 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.

PERFORMANCE COMPARISON

The basic framework of Q-Routing was reviewed. In this study, researchers 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.

OPTIMIZATION TECHNIQUES IN Q Routing

Inspired by the Q Routing algorithm by Boyan and Littman (1994), Branch (2011) and Bhatnagar and Babu (2008), a new adaptive Routing algorithm, Confidence based Dual Reinforcement Q Routing (CDRQ Routing) with higher quality and increased quantity of exploration is presented (Fig. 4). 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.

Kumar (1998a) and Mastronarde and van der Schaar (2011) 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

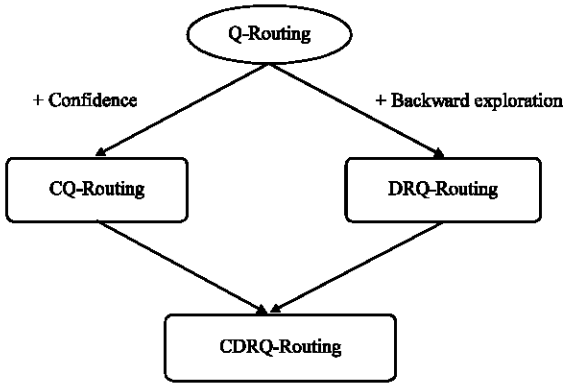

Fig. 4: Optmization of Q Routing

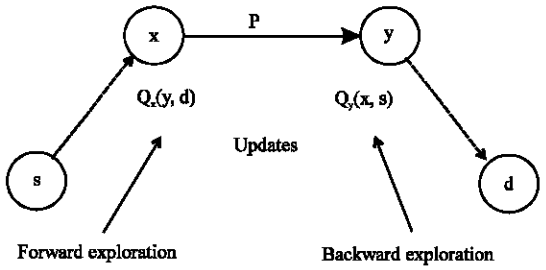

Fig. 5: Dual reinforcement Q Routing (DRQ Routing)

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 neighbours, 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 neighbouring node for the update has been updated more recently.

In CQ Routing (Kumar, 1998b; Yap and Othman, 2004), 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 Fig. 5. 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 $PacketReceive_y$ (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.

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.

REFERENCES

Ali, A. and S. Ali, 2011. Performance Analysis of AODV, DSR and OLSR in MANET. LAP Lambert Academic Publishing, USA., ISBN: 9783845412306, Pages: 84.

Bhatnagar, S. and K.M. Babu, 2008. New algorithms of the Q-learning type. Automatica, 44: 1111-1119.

Boyan, J.A. and M.L. Littman, 1994. Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach. In: Advances in Neural Information Processing Systems, Tesauro, G. and D. Touretzky (Eds.). Morgan Kaufmann, San Francisco, CA., USA., pp: 671-678.

Branch, K., 2011. Modified Q-learning Routing algorithm in fixed networks. Austr. J. Basic Applied Sci., 5: 2699-2703.

Clausen, T. and P. Jacquet, 2003. Optimized link state Routing protocol. Internet Draft, IETF Manet Working Group. http://tools.ietf.org/html/draft-ietf-manet-olsr-11.

Haraty, R. and B. Traboulsi, 2012. MANET with the Q-Routing protocol. Proceedings of the 11th International Conference on Networks, February 29-March 5, 2012, Saint Gilles, Reunion, pp: 187-192.

Johnson, D., D. Maltz and J. Jetcheva, 2002. The dynamic source Routing protocol for mobile ad hoc networks. http://www.monarch.cs.rice.edu/monarch-papers/dsr-chapter00.pdf.

Kelley, D., 2005. Reinforcement learning with application to adaptive network Routing. J. Theor. Applied Inf. Technol.,

Kumar, S., 1998a. Confidence based dual reinforcement Q-Routing: An on-line adaptive network Routing algorithm. M.S. Thesis, University of Texas, Austin.

Kumar, S., 1998b. Confidence based dual reinforcement Q-Routing: An on-line adaptive network Routing algorithm. Report No AI-98-267, University of Texas at Austin Austin, TX., USA., pp: 108.

Mastronarde, N. and M. van der Schaar, 2011. Fast reinforcement learning for energy-efficient wireless communication. IEEE Trans. Signal Process., 59: 6262-6266.

Shrivastava, R. and R.K. Saluja, 2012. Performance evaluation of extended AODV using different scenarios. Int. J. Smart Sensors Ad Hoc Networks, 1: 56-62.

Toteja, A., R. Gujral and S. Thalia, 2010. Comparative performance analysis of DSDV, AODV and DSR Routing protocols in MANETs, using NS2. Proceedings of the International Conference on Advances in Computer Engineering, June 20-21, 2010, Bangalore, Karnataka, India, pp: 330-333.

Yap, S.T. and M. Othman, 2004. An adaptive algorithm: Enhanced confidence based Q Routing algorithm in network traffic. Mala. J. Comput., 17: 21-29.

Zafar, H., N. Alhamahmy, D. Harle and I. Andonovic, 2011. Survey of reactive and hybrid Routing protocols for mobile ad hoc networks. Int. J. Communi. Networks Inf. Secur., 3: 1-79.