Simulated Annealing Based Hierarchical Q-Routing: a Dynamic Routing Protocol

Antonio Mira Lopez
Power Costs Inc.
Norman, OK 73072
alopez@powercosts.com

Douglas R. Heisterkamp
Computer Science Department
Oklahoma State University
Stillwater, Oklahoma 75078
doug@cs.okstate.edu

***Abstract*—Network routing is the mechanism chosen to send packets from any source to a destination in the network. The goal of any routing algorithm is to find an efficient path to send a packet to any destination taking into account all the obstacles that may be taking place in the network at any time. The focus of this paper is to provide an efficient solution to the routing problem by making use of reinforcement learning and other heuristics. The paper proposes a routing algorithm called Simulated Annealing based Hierarchical Q-Routing which is based on an Q-Routing. Providing an hierarchy by defining network areas and roles to routers within these areas, allows us to have more updated global information and therefore much better decision making when routing packets across the network. The addition of simulated annealing as an exploration method also plays an important role in the improvement of the original Q-Routing approach. The temperature in simulated annealing adapts as a function of the waiting queue utilization of specific routers in the network. Experiments with different topologies and network loads show that the proposed method is able to improving convergence, loop avoidance, and scalability in comparison to Q-Routing.**

***Keywords*-Network Routing, Q-Learning, Reinforcement Learning, Simulated Annealing.**

I. Introduction

Internet is expanding at an incredible rate year after year. More efficient technologies need to evolve to keep up with the rapidly number of users that share the networks. With the increase in the number of users and the amount of traffic across the internet, the common problems in the communication among nodes will become more and more problematic. Some of these problems such as congestion create a huge impact in how fast packets can be delivered from one node to another node. Routers are the devices that manage how packets are delivered from a source to a destination. The internal algorithm that decides how the packet travels though routers is called a routing protocol. Routing protocols are critical when we think about how fast a packet makes it from source to destination. Their main purpose is to find an algorithm that provides the most optimal path from source to destination. There are some very well known internal routing protocols such as RIP, EIGRP, OSPF that are commonly used in networks today.

A router should be able to realize when there has been a change in the topology or when a link in the network has failed. In these situations the router must be able to adapt quickly to the new topology by changing the way it is distributing packets to places that were in the path of that particular link [4,10,13,14]. It must also do so when the network gets congested. This is called dynamic routing. In [1] the Q-learning reinforcement learning algorithm was used to create a dynamic routing algorithm called Q-Routing.

The objective of this work is to modify Q-Routing to overcomes its limitations and performs more efficiently under different network loads and different topologies. The second goal of the new algorithm will be improving scalability. Q-Routing's performance decreases drastically as the network topology becomes larger. The new proposed algorithm, due to its hierarchical model of the network, will be able to perform efficiently as the network grows larger. Q-Routing will be presented in Section II. The proposed algorithm, Simulated Annealing based Hierarchical Q-Routing, will be presented in Section III. The experiments and their results will be analyzed in Section IV.

II. Q-Routing

Q-Routing [1,2] was developed by Littman and it is the first routing algorithm to make use of reinforcement learning. Q-Routing is based on the concept of Q-learning [12]. Q-learning makes use of Q-values to perform updates. Similarly, Q-Routing uses Q-values which estimate how long it will take to send a packet to any particular destination through each one of the node's neighbors. The state s in the optimization problem of network routing is represented by these Q-values in the entire network [1].

In Q-Routing, each node contains a packet-routing module. The module performs actions such as routing messages arriving at a node to a neighbor with the smallest total delivery time. When a packet arrives, it sends an estimate of the remaining delivery time back to the node that sent it. When receiving a reply from a neighbor, it updates the estimate for that neighbor.

The Q-Routing algorithm is presented in Figure 1 and can be describes as follows. A packet arrives at a router x and it is then processed. In the algorithm, this router would find itself in state s. The agent must select a neighbor y' such that y' has minimal $\mathbf{Q}_{\text{interarea}_x}(y,d)$ from all neighbors y of x and return the reward back to s. The Q-value for the chosen neighbor is then updated. In the update rule, t represents the

```
Let $x$ be the current node
while (True)
    Select a packet from the queue
    Let $d$ be the packet destination
    Let $y' = \underset{y \in \text{neighbor of } x}{\operatorname{argmin}} \mathbf{Q}_x(y, d)$
    Send packet to $y'$
    On reply from $y'$, apply update
    $\mathbf{Q}_x(y', d) = \mathbf{Q}_x(y', d)$
            $+ \alpha \left(\mathbf{Q}_{y'}(z', d) + t + q - \mathbf{Q}_x(y', d)\right)$
    end while
```

Figure 1. Q-Routing Algorithm

transmission delay, the time it takes for a packet to travel over the media to the next node. The variable q represents the time the packet spends waiting in the queue. In the update equation, $\mathbf{Q}_{\text{interarea}_y}(z', d) + t + q$ represent the new estimate and $\mathbf{Q}_{\text{interarea}_x}(y', d)$ represents the old estimate. The variable α represents the learning rate factor. In [1], $0.5 \leq \alpha \leq 0.7$ was used. The previously mentioned algorithm is purely greedy. It exploits its knowledge of the environment to search for the most optimal paths. For the type of scenario where the environment is constantly changing, an algorithm needs to perform some sort of exploration to avoid getting stuck in a local maxima.

Confidence-based Dual Reinforcement Q-Routing [6,11] and N Q-Routing Optimal Shortest Path [7,8] are attempts to improve the efficiency and robustness of Q-Routing. The improvement of initializing Q-values with the shortest path is used in the experiments of this paper for both Q-Routing and the proposed Simulated Annealing Based Hierarchical Q-Routing. This helps overcome the problem of inefficient learning phase in low network load situations.

III. Simulated Annealing Based Based Hierarchical Q-Routing

In the previous section the Q-Routing algorithm was described. The proposed method, Simulated Annealing Based Hierarchical Q-Routing, has been implemented to overcome the limitations present in the Q-Routing algorithm in order to provide a more robust and efficient routing algorithm. To reduce the number of loops and to provide faster convergence under changing network loads in the network a more hierarchical model of the network has been implemented. To provide exploration to the Q-Routing approach the idea of adding simulated annealing as an exploration method has been implemented. Exploration is very important when congestion changes the behavior of the network from the previously learned values. Simulated annealing provides a clear methodology to guide and control exploration based on a temperature parameter that we set base on current network congestion levels.

Q-Routing learns about the environment by updating its Q-values each time a reward packet is sent back from the potentially best neighbor. When the network suffers drastic changes time and time again these Q-values might take too much time to converge and therefore packets may be traveling though the network inefficiently. This could be detrimental since instead of decreasing congestion, this inefficiency is creating unnecessary congestion to the network. There is a need for a more global view of the network in order to make changes fast enough to adjust to these rapid changes in the network loads. A hierarchical model of the network is needed to accomplish this. The network space should be partitioned in such a way to maximizes the number of redundant paths between these areas and at the same time minimizes the number of areas needed to partition the topology.

For each area routers must be assigned a role. A router is assigned to be a *border router* if one of its links is directly connected to a router belonging to a different area. All other routers will be *local routers*.

When a packet is first created, the router must look to see if the destination belongs to a router in its same area or in a different area. If the area of the destination is external then the source must pick the border router that yields the minimum transmission time to the destination area. This Q value is of the form $\mathbf{Q}_{\text{interarea}_x}(b, d)$ where x represents the current node, b represents a border router from the same area as x and d represents the area of the destination.

When a packet is sent from one router to another local router in the same area, it is done so using basic Q-Routing. The reason is that since areas are subsets of the topology, the distance from one node to another node in the same area is relatively small and Q-Routing is sufficient to route this packets efficiently. Because of the close proximity between these routers, the Q-values will be able to adjust quicker to changes in the environment. Most of these packets will not be sent to a destination but rather to border routers since these routers will make a routing decision to send them to another area.

When a packet arrives at a border router two things can happen. Border routers maintain information about all areas directly connected to them. If a packet arrives at a border router and its destination happens to be from an area directly connected to it, the border router will be able to send the packet using basic Q-Routing. If the destination area is not adjacent, then the border router will pick a border router from a directly connected adjacent area that yields the minimum transmission time to the destination area. This will be done by using $\mathbf{Q}_{\text{interarea}}$ values of the form $\mathbf{Q}_{\text{interarea}_x}(b, d)$ where x is the current border router, b is a border router belonging to an adjacent connected area to x, and d is the area of the destination. Simulated annealing will be implemented in the decision process of border routers routing packets to external areas.

```
Arbitrary initialization all $\mathbf{Q}(s, a)$ values;
repeat (for each episode):
    Choose a random (initial) state $s$;
    repeat (for each step in the episode):
        Select action $a_r$ in $\mathbf{A}(s)$ arbitrarily;
        Select action $a_p$ in $\mathbf{A}(s)$ according to policy;
        Let $a \leftarrow a_p$
        Generate a random number $v$ with $0 \leq v \leq 1$
        If $v < e^{\frac{\mathbf{Q}(s,a_r)-\mathbf{Q}(s,a_p)}{\text{temperature}}}$ then let $a \leftarrow a_r$
        Execute the action $a$
        Receive reward $r$ and new state $s'$
        $\mathbf{Q}(s, a) \leftarrow \mathbf{Q}(s, a)$
                $+\alpha(r + \gamma \max_a \mathbf{Q}(s', a) - \mathbf{Q}(s, a))$
```

Figure 2. The SA-Q-Learning Algorithm

```
Let $B$ be the set of closest border routers in adjacent
    areas to the congested area and all routers in the
    congested area.
Let $q_{w_x}$ = waiting queue utilization at current border
    router $x$.
If (Congestion Level() != $q_{w_x}$)
Send a congestion packet to $B$
```

Figure 3. Congestion Advertisement

The idea of implementing simulated annealing into the Q-Routing problem has been extracted from [3] in which simulated annealing integrated into Q-learning. The resulting algorithm is called the SA-Q-Learning algorithm and is presented in Figure 2.

For the routing problem, simulated annealing is implemented into the Q-Routing algorithm the same way it was implemented in the Q-learning algorithm. There are some important parameters that we need to address before the implementation is explained. In the routing problem, the temperature variable is represented by the waiting queue utilization in the border routers. This variable will be adjusted dynamically as the network load changes in the network. Simulated annealing is used as an exploration method in Q-Routing. Border routers compute their waiting queue utilization every time a packet arrives to their waiting queue. The waiting queue is partition in levels according to the percentage of queue utilization. If this value changes to a new level, a trigger is executed and the border router sends a packet to specific routers in the topology. This process is called Congestion Advertisement and is presented in Figure 3.

There are 2 types of congestion packets that are sent in the network. The type 2 congestion packet is the one created by the congested border router. The type 3 congestion packet is the one that the closest border routers at adjacent areas send to other border routers in their area. The process of receiving a congested packet is called Congestion Update and is presented in Figure 4. The optimal interarea Q value, $\mathbf{Q}^{\star}_{\text{interarea}_x}$, is determined by shortest path algorithm.

```
Let $q$ be the congested border router that sent the
    congested packet.
Let $val_q$ be the congested value to be updated.
Let $x$ be the current node.
Let $\mathbf{A}$ be the set of all areas in the topology.
If (congestion packet type == 2)
    Congestion$_x(q)$ = $val_q$
    penalty = $\mathbf{Q}^{\star}_{\text{interarea}_x}(q) \cdot val_q$
    For every $a \in \mathbf{A}$
        $\mathbf{Q}_{\text{interarea}_x}(q, a)$ = $\mathbf{Q}^{\star}_{\text{interarea}_x}(q, a)$ + penalty
    If (area($x$) != area($q$))
        Send a congestion packet type 3 to other
        border routers in the area.
If (congestion packet type == 3)
    Congestion$_x(q)$ = $val_q$
    penalty = $\mathbf{Q}^{\star}_{\text{interarea}_x}(q) \cdot val_q$
    For every $a \in \mathbf{A}$
        $\mathbf{Q}_{\text{interarea}_x}(q, a)$ = $\mathbf{Q}^{\star}_{\text{interarea}_x}(q, a)$ + penalty
```

Figure 4. Congestion Update of Inter-Area Q-values.

```
Let $x$ be the current border node.
Let $\mathbf{A}$ be the set of border routers in adjacent areas
    directly connected to $x$.
Let $d$ be the destination area.
Select $r' \in \mathbf{A}$ such that $r'$ has minimal
$\mathbf{Q}_{\text{interarea}_x}(r', d)$
If (Congestion Level$_x(r'$, $d$) != 0)
    Select $r'' \in \mathbf{A}$, $r'' \neq r'$, such that $r''$
        has minimal $\mathbf{Q}_{\text{interarea}_x}(r'', d)$.
    Let $\mathbf{B}$ be the set of border routers in local area
        of $x$ NOT including $x$.
    temperature = Congestion Level$_x(r', d)$
    Generate a random number $v$ with $0 \leq v \leq 1$
    If $v < e^{\frac{\mathbf{Q}^{\star}_{\text{interarea}_x}(r'',d)-\mathbf{Q}^{\star}_{\text{interarea}_x}(r',d)}{temperature}}$ then
        $r' \leftarrow r''$
Return $r'$
```

Figure 5. Inter-Area Border Selection using Simulated Annealing.

The remaining step of SAHQ is the decision process of the border routers. The process is presented in Figure 5.

IV. Experiment Results

The experiments have been performed using a Java simulator. Each experiment has been performed using basic Q-Routing and the proposed algorithm Simulated Annealing based Hierarchical Q-Routing. The topology used for this experiment is the one previously used by Littman and Boyan, the 6X6 grid topology [1,5]. The grid with area partitioning

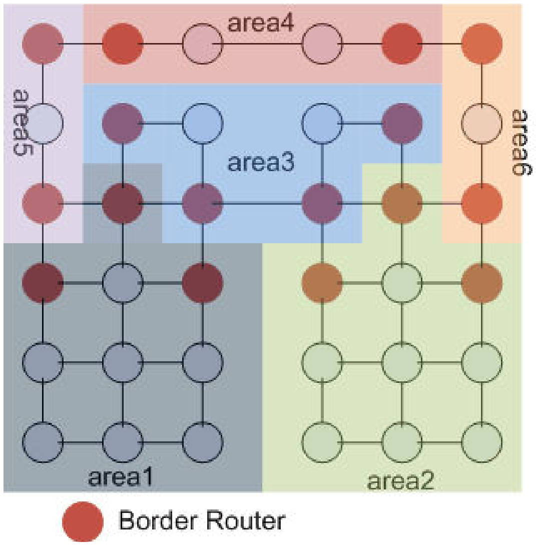

Figure 6. 6x6 grid network with Areas

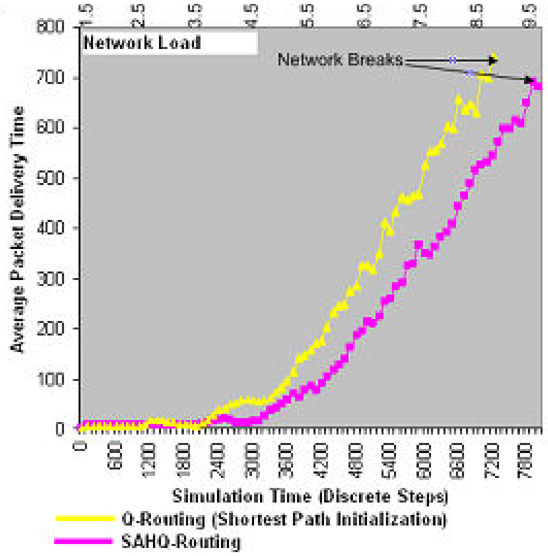

Figure 7. High Constant Network Load

is presented in Figure 6. Experimental results for additional network topologies and load conditions may be found in [9].

Figure 7 shows the result of experiments with high constant network in the 6x6 grid. It can be seen that under a rapid increase of network load SAHQ-Routing is able to balance the network faster than the Q-Routing algorithm. Q-Routing network breaks at time 7000. SAHQ-Routing is able to run until time 8000. Through the use of a more hierarchical network design and the use of simulated annealing the traffic is balanced quicker across the network. As the network load increases the difference in performance becomes more apparent.

Figure 8 shows the results of experiments with variable network load. They show that not only SAHQ-Routing

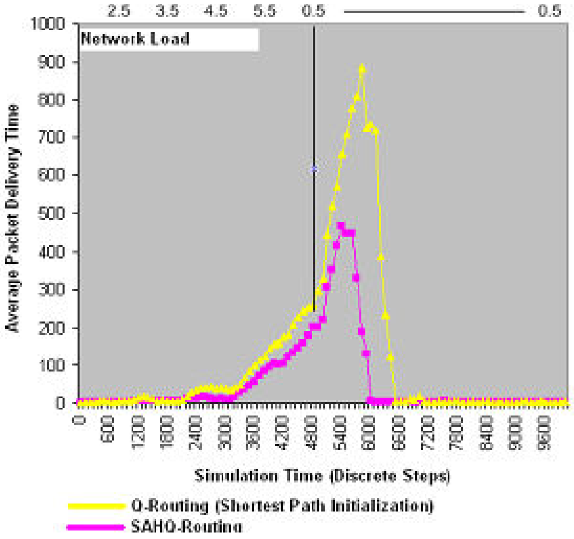

Figure 8. Variable Network Load

is able to route packets more efficiently under increasing network loads, but it is also able to recover from congestion faster. The reason why this happens is that Q-Routing relies on its Q-values to route packets efficiently. It learns a complete new set of values as Congestion increases. The problem is that by the time congestion ends, the Q-values are not optimized to work under low network loads and therefore it must start learning in order to adjust these values. During this time period the network already had many packets still being routed and since the Q-values are no longer adjusted to the new environment, packet will be routed in a non-efficient manner. Loops will be very frequent during these time period. On the other hand, SAHQ-Routing works in a more hierarchical manner. It uses real time information to make decisions based on current network load. The simulated annealing congestion variable representing the temperature allows us to adjust inter-area traffic values rapidly to switch back from high to low congestion. Congestion ends at simulation step 5000. SAHQ-Routing is able to realize this and make changes quickly to continue routing packets efficiently. Q-Routing must learn and updates its Q-values for some time until is able to route packets in an efficient manner. SAHQ-Routing is able to recover 600 simulation steps faster than Q-Routing. In a real case scenario this becomes very important since a routing protocol must be able to route packets efficiently under any circumstances.

The topology used in the next experiment is shown in Figure 9. Each area of the topology can get to any other area in the topology though various paths. This way if a link goes down or is congested the traffic can use another path to send the packet to its destination. Each 3×3 square of nodes represents an area in the topology. This experiment is performed under variable network loads. In this experiment, due to the redundant design of the topology the traffic across

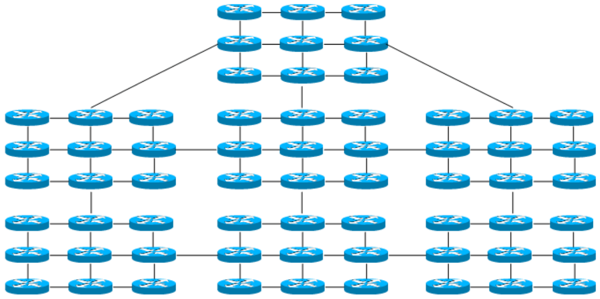

Figure 9. Redundant Network Topology

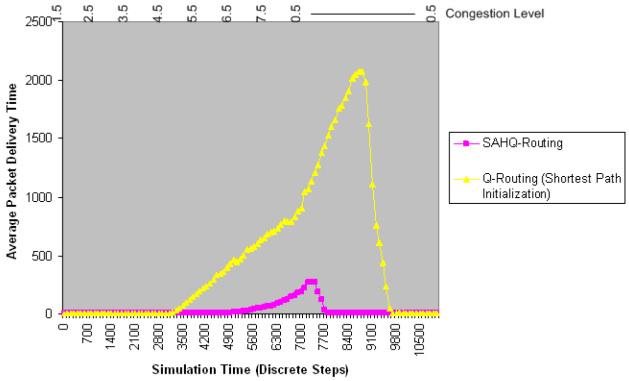

Figure 10. Average Packet Delivery Time

the network should stay balanced at higher network loads. Q-Routing takes longer to learn about larger topologies. Q-Routing decreases in routing performance as the network topology becomes larger. Q-Routing uses Q-values that are maintained in the routers to make routing decisions. These estimates take longer to be updated and therefore routing becomes inefficient. The type of topology used in this experiment also makes SAHQ-Routing perform much more efficiently than Q-Routing does. This is once again because of the hierarchical management of delivering packets. There is a lot of redundancy in the network and therefore there are multiple paths to avoid congestion. The sooner routers realize, the sooner congestion will be able to be avoided. SAHQ-Routing due its hierarchical functionality is able to adjust much quicker in larger networks and under changing network environments. Figure 10 shows the results of the experiment under various network loads. The difference in performance is overwhelming. Q-Routing has a difficult time adjusting to the changing network load and reaches a high peak of over 2000 average packet delivery time. SAHQ-Routing due to its preventive nature to avoid congestion is able to maintain a stable network flow and reaches a high peak of less than 500 average packet delivery time.

V. Conclusion

The internet is increasing at a very rapidly rate. There is a need for routing optimization in order to speed up the way packets are delivered from hosts to destinations. There exist routing algorithms such as OSPF that use Shortest Path routing algorithm to perform the delivery of packets. These routing algorithms are efficient when the network load is not very high in the network but can become very inefficient when the network load increases. If the network load increases, the network becomes congested and bottlenecks begin to take place. The network then becomes inefficient. Littman and Boyan created the Q-Routing algorithm in 1994 so that the routers would be able to adjust to new changing environments dynamically. The Q-Routing algorithm uses reinforcement learning to achieve this but still has some problems when adjusting to these new environments such as recovering from Congestion. A new method, Simulated Annealing based Hierarchical Q-Routing, is proposed to overcome and improve the already existing Q-Routing algorithm.

The experiments conducted show that SAHQ-Routing is able to route packets more efficiently than Q-Routing in low, medium and high constant network loads. The reason is that Q-Routing must learn about the network before it is able to make efficient routing. SAHQ-Routing is initialized to shortest path so that its values are optimal at initialization.

The most important part of the experiment was to see the difference in performance between SAHQ-Routing and Q-Routing under changing network loads. Two different topologies were used to see how both algorithms are able to adjust to changes in the environment. The first topology is the 6x6 grid used in the original Q-Routing simulations done by Boyan and Littman. The results of the simulation showed that SAHQ-Routing was able to maintain a more balanced traffic flow under changing environments and therefore the average queue routing time was lower than that of Q-Routing. Q-Routing is able to adjust to increasing network load by updating its Q-values. The problem with Q-Routing is that if the network rapidly suffers a change in the network load, Q-Routing takes some time to update its Q-values to the new environment and packets are not routed efficiently for a period of time. SAHQ-Routing uses a hierarchical model of the network to be able to advertise congestion to other areas so that congestion can be avoided. With a hierarchical model there more of a global view and we can exchange information between the highest layer of the network in order to avoid congestion. This way information can be propagated across the network much quicker and decision can be made using values very closed to the real ones.

New features can be added to the algorithm to enhance its performance. The addition of a discovery algorithm at each router's initialization would make this process dynamic without having to change information manually each time. The addition of priority queuing would definitely speed up propagation of management information across the network. In the current algorithm management packets are treated the same way as regular traffic. This means that in times of congestion these packets may take some time to arrive to their destination and by that time something else may have changed in the network.

References

[1] J. Boyan and M. L. Littman, “Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach”, *Advances in Neural Information Processing Systems* 6, 1994.

[2] J. Boyan and M. Littman, *A Distributed Reinforcement Learning Scheme for Network Routing*, Technical report, Department of Computer Science, Carnegie Mellon University, 1993.

[3] M. Guo, Y. Liu, and J. Malec, “A new Q-Learning Algorithm Based on the Metropolis Criterion”, *IEEE transactions on Systems, man, and Cybernetics-Part B: Cybernetics*, Vol. 34, No. 5, Pages 2140-2143, October 2004.

[4] X. Jing, C. Liu, and X. Sun, “Artificial Cognitive BP-CT Ant Routing Algorithm”, *Proceedings of International Joint Conference on Neural Networks*, Montreal, Canada, July 31 - August 4, 2005.

[5] S. Khodayari, and M.J. Yazdanpanah, “Network Routing Based on Reinforcement Learning In Dynamically Changing Networks”, *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI ’05)*, 2005.

[6] S. Kumar and R. Miikkualainen, “Confidence-based Q-Routing: an on-queue adaptive routing algorithm” *Proceedings of Neural Networks in Engineering*, 1998.

[7] A. Mellouk, S. Hoceïni, S. Larynouna. “Flow based Routing for Irregular Traffic using Reinforcement Learning Approach in Dynamic Networks”, *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC’06)*, 2006.

[8] A. Mellouk, S. Hoceïni, S. Larynouna, “Adaptive Probabilistic Routing Schemes for Real Time Traffic in High Speed Dynamic Networks”, *IJCSNS International Journal of Computer Science and Network Security*, Vol. 6 No 5B, pages 36-42, May 2006.

[9] A. Mira Lopez, *Simulated Annealing Based Hierarchical Q-Routing: A Dynamic Routing Protocol*, Master Thesis, Oklahoma State University, December 2007.

[10] A. Nowe, K. Steenhaut, M. Fakir, and k. Verbeeck, “Q-learning for adaptive load based routing”, *IEEE transactions on Systems, Man, and Cybernetics*, Vol. 4, Pages 3965-3970, October 1998.

[11] Shailesh, Kumar, *Confidence based Dual Reinforcement Q-Routing: an On-line Adaptive Network Algorithm*, Master Thesis, The University of Texas at Austin, 1998.

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

[13] P.R.J Tillotson, Q. H. Wu, and P.M Hughes, “Multi-Agent Learning for Control of Internet Traffic Routing”, *Learning Systems for Control*, IEE Seminar, 2000.

[14] S. Whiteson, and P. Stone, “Towards Autonomic Computing: Adaptive Network Routing and Scheduling”, *Proceedings of the International Conference on Autonomic Computing (ICAC’04)*, 2004.