ภาคผนวก

A Reinforcement Learning Approach for Path Discovery in MANETs with Path Caching Strategy

Wipawee Usaha
School of Telecommunication Engineering
Suranaree University of Technology, Nakhon Ratchasima, Thailand 30000
Email: wipawee@ccs.sut.ac.th

Abstract— **In this paper, we enhance an existing path discovery scheme called the Ticket-Based Probing (TBP) which supports QoS routing in mobile ad hoc networks (MANETs) to increase its accumulated reward. The scenario of QoS routing in MANETs with the presence of network information uncertainty is considered and modelled as a partially observable Markov decision process (POMDP). The proposed scheme integrates the original TBP scheme with a reinforcement learning method for POMDPs, called the on-policy first-visit Monte Carlo (ONMC) method, and a suitable path caching strategy. Simulation results shows that the inclusion of patch caching with the ONMC method can indeed achieve message overhead reduction with marginal difference in the path search ability and additional computational and storage requirements.**

I. Introduction

Routing in a mobile ad hoc network (MANET) is a challenging task due to node mobility. Difficulties arise even further in the development of routing schemes which support QoS connections. One key to support QoS routing is feasible route search [1], [2], [5]. Feasible route search can be done by distributed routing whereby other nodes apart from the source node are involved in the feasible path(s) search by identifying their neighboring nodes as the next hop router. It can also be performed by source routing where a feasible path(s) is computed solely at the source node.

Alternatively, certain methods like the Ticket-Based Probing (TBP) scheme [1] combine the features of distributed and source routing. More specifically, flooding is still invoked but the amount of flooding is controlled by issuing a limited number of logical tickets at the source node. Although the TBP scheme enjoys several advantages such as high tolerance to imprecise state information, some challenging issues still remain—one of which relates to the restricted flooding method: the computation of a suitable number of logical tickets issued at the source node. More specifically, the original TBP scheme relies on an heuristic rule of ticket computation. In [7], the original TBP scheme is enhanced by integrating it with a reinforcement learning (RL) technique. Results in [7] show that the RL-based TBP scheme is able to learn a "good" rule for issuing tickets by interacting directly with the environment or by simulation—at the expense of reasonable storage and computational requirements of on-line decision parameters.

In this paper, we study the effect of the inclusion of path caching to the RL-based TBP scheme. Our motivation is that by maintaining a path cache at each mobile node, we can avoid frequently invoking the path discovery scheme and therefore reduce the amount of routing overhead in the MANET. The contribution in this paper is the experimental evidence that, RL techniques equipped with suitable path caching strategies can be employed to reduce the amount of message overhead in QoS routing in MANETs [7].

The paper is organized as follows. In the next section, we present an introduction to the partially observable Markov decision process (POMDP) model which the QoS routing problem in MANETs is based on. Section III describes the TBP path discovery schemes and path caching to support QoS routing in MANETs. In this section, the original TBP scheme and the enhanced TBP scheme are presented. The following section shows the numerical study results and Section V provides the conclusion.

II. QoS routing in MANET as a partially observable Markov decision process

A vital component for QoS routing in MANETs is the residual resource information in the network. Such information depends on up-to-date information between mobile nodes. Message exchanges between mobile nodes are therefore required. These information exchanges are done periodically or when a topology change is detected. But even so, imprecise information can still arise due to delayed-arrival or lost update messages and restricted transmission of updating messages.

Because accurate network information is difficult to obtain, each mobile node is faced with only an "observation" of its environment which is most likely incomplete and inaccurate. With only the current network observation at hand, each mobile node must make certain decisions, e.g., how many control messages are needed to find a feasible path for some new connection arrival, when and how to perform path maintenance if an existing path is about to break, etc.

Under certain assumptions regarding the movement and resource information, it is possible to (approximately) model the state transitions as a Markov process [7]. Furthermore, since the accurate state of the network is hidden from each mobile node, we can (approximately) model the decision-making problem in MANETs as a partially observable Markov decision process (POMDP). The goal is to find a policy which optimizes some performance criterion in finite horizon. The finite horizon problem is considered here due to the *episodic* nature of message exchanges between the mobile nodes—an episode starts immediately after a message exchange and terminates at the subsequent message exchange.

III. TBP PATH DISCOVERY SCHEME AND PATH CACHING

The on-policy first-visit Monte Carlo (ONMC) method for POMDPs [7] is employed here to find an observation-based policy in *partially observable* MDPs. The method is extended from *completely observable* Markov decision processes (MDPs) in [6].

The ONMC method for POMDPs is integrated into a path discovery scheme called the *Ticket-Based Probing* (TBP) scheme. The TBP scheme is a multipath distributed routing algorithm for supporting end-to-end delay or bandwidth requirements proposed to tolerate high degrees of imprecise state information [1]. The design objective of this algorithm is to maximize the probability of success in finding a feasible route in dynamic networks in the presence of inaccurate information. The basic idea of the algorithm is outlined as follows. When a source node s needs to find a route that satisfies a delay (or bandwidth) requirement to a destination node d, a number of probes (search messages) are sent from s towards d. The total number of probes used in the path discovery is controlled by the initial number of *logical* tickets, M_0. The parameter M_0 is computed at the source node s depending on the contention level of network resources and the inaccuracy of available information. When a neighboring node j receives a probe from node s, it makes copies of that probe and recomputes the number of tickets to be carried on the copied probes. The computation of the tickets at node j is based on the available end-to-end information (i.e., from node j to d) and cannot exceed the number of tickets in the probe that node j has received. The end-to-end information, which is obtained through probing on an on-demand basis, is used to guide the distribution of the tickets and the probes along the directions of *most probable* feasible paths towards the destination d. Each probe carries at least one ticket. Since no additional tickets are issued along the intermediate nodes and each probe searches one path, the number of paths found are also bounded by the number of tickets M_0 issued at the source node. Consequently, the amount of probes that enter the network is simply controlled by varying M_0.

A. Initial ticket calculation: Overview of the original TBP scheme

In this paper, we study a *delay-constrained least-cost routing* problem. Consider a connection request whose source, destination nodes and mean end-to-end delay requirement are s, d and D_{req}, respectively. Let D_{ij} be the mean link delay between node i and j. The mean end-to-end delay of the lowest delay route r^*, $D_n(d)$, is found by $D_n(d) = \sum_{(i,j)\in r^*} D_{ij}$. The parameter $\triangle D_n(d)$ is the variation of the mean end-to-end delay which is computed from

$$\triangle D_n^{new}(d) = \rho \triangle D_n^{old}(d) + (1-\rho)\,\beta \left| D_n^{new}(d) - D_n^{old}(d) \right| . \tag{1}$$

The parameter ρ is the forgetting factor which determines how fast $\triangle D_n^{old}(d)$ is forgotten, $(1-\rho)$ determines how fast $\triangle D_n^{new}(d)$ converges to $\left|D_n^{new}(d) - D_n^{old}(d)\right|$, and β is a parameter chosen to ensure a large value of $\triangle D_n^{new}(d)$. Note that by increasing β, we increase $\triangle D_n^{new}(d)$ and consequently, the certainty that the actual delay falls in the imprecise range. In [1], the number of tickets (M_0) is found from $M_0 = Y_0 + G_0$ where Y_0 and G_0 are the number of yellow and green tickets, respectively. The yellow tickets are for maximizing the chances of finding feasible paths while the green tickets are for maximizing the chances low cost paths.

The parameter Y_0 is determined according to these heuristic rules [1]:

$$Y_0 = \begin{cases} 1 & ,D_{req} > D_{hi} \\ \left\lceil \frac{D_s(d)+\triangle D_s(d)-D_{req}}{2\times\triangle D_s(d)} \times \theta_Y \right\rceil & ,D_{lo} \le D_{req} \le D_{hi} \\ 0 & ,D_{req} < D_{lo} \end{cases} \tag{2}$$

where $D_{hi} = D_s(d) + \triangle D_s(d)$, $D_{lo} = D_s(d) - \triangle D_s(d)$, θ_Y is a system parameter specifying the maximum allowable number of yellow tickets.

The other parameter, G_0, follows a slightly different set of rules:

$$G_0 = \begin{cases} 1 & , D_{req} > \Theta D_{hi} \\ \left\lceil \frac{\Theta(D_s(d)+\triangle D_s(d))-D_{req}}{\Theta(D_s(d)+\triangle D_s(t))-D_s(d)} \times \theta_G \right\rceil & ,D_s(d) \le D_{req} < \Theta D_{hi} \\ \left\lceil \frac{D_{req}-D_s(d)+\triangle D_s(d)}{\triangle D_s(d)} \times \theta_G \right\rceil & ,D_{lo} \le D_{req} < D_s(d) \\ 0 & ,D_{req} < D_{lo} \end{cases} \tag{3}$$

where θ_G specifies the maximum allowable number of green tickets, $\Theta > 1$ specifies the threshold beyond D_{req} which we allow to search for large-delay paths.

The intuitive reasoning behind the above rules is simple. If D_{req} is very large, then a single yellow ticket suffices. If D_{req} is within the estimated range, then more yellow tickets are assigned for more stringent D_{req}. In the case where D_{req} is less than the best estimated end-to-end delay, no tickets are issued since such a tight requirement is unlikely to be satisfied. The connection request is rejected or some negotiation for a less stringent requirement is made. The green tickets undergo a similar strategy. The selection of the system parameters (θ_Y, θ_G and Θ) is a practical design issue which can depend on level of overhead control imposed on the network [1].

B. Initial ticket calculation: TBP scheme based on the ONMC method

The ONMC method for POMDPs can be applied to the actual system or simulator to obtain a good ticket issuing policy—one that balances the trade-off in the number of issued tickets and the probability of discovering feasible paths. More specifically, instead of calculating M_0 from an heuristic rule like in (2) and (3), M_0 is selected from some finite set in a sequential decision-making process in the presence of state uncertainty with the objective of maximizing some performance criterion.

Consider a $\mathcal{N}$-node MANET. Each mobile node maintains end-to-end delay information to all the destination nodes in the network. For each source node s, a policy is determined separately for each destination node d in the network. Hence, for each source-destination node pair (s, d), the observation set is defined as

$$\mathcal{O}_{sd} = \{[q_D(m), q_{\triangle D}(l)] : 1 \leq m \leq n, 1 \leq l \leq n_\triangle\}$$

where n $(n_\triangle)$ is the number of discrete end-to-end delay (end-to-end delay variation) intervals and $q_D(m)$ $(q_{\triangle D}(l))$ is the m^{th} (l^{th}) interval on $[0, \infty)$. The variable $q_{\triangle D}(l)$ is included to reduce the uncertainty of the actual end-to-end delay.

Based on $o_k \in \mathcal{O}_{sd}$ at time k, node s takes an action $a_k \in A = \{0, \ldots, M_{\max}\}$ by selecting some $M_0 \in A$ tickets, where $M_{\max}$ is the maximum allowable number of tickets. To maximize the probability of discovering a feasible path, note that high-cost (e.g. longer hops) paths can be tolerated as long as a feasible path can be discovered. The green tickets are omitted $(G_0 = 0)$ and only the yellow tickets are considered so that $M_{\max} = \theta_Y$ in order to put more emphasis on finding feasible paths rather than low-cost paths. If the selected action is $a_k = M_0 > 0$, the tickets are distributed in the manner as the original TBP scheme. If at least one feasible path is found once the path discovery is completed, a reward $g(o_k, a_k)$ is generated. Otherwise, the action is penalized. Such reward scheme is defined as

$$g(\cdot, a_k) = \begin{cases} \zeta_j - \log a_k & , a_k > 0, X = \varkappa \\ -(\zeta_j - \log a_k) & , a_k > 0, X = 0 \\ -\log a_k & , a_k > 0, X > \varkappa \\ 0 & , a_k = 0 \end{cases} \quad (4)$$

where $\zeta_j \in \mathcal{R}^+$ is the immediate reward parameter for service type-j, X is the number of discovered feasible paths, $\varkappa$ is the desired maximum number of discovered feasible paths. Note that this scheme favors issuing tickets which can find up to $\varkappa$ paths only—issuing too few or too many tickets than necessary is penalized.

If multiple feasible paths are discovered, the destination node d selects the least-cost path. It then returns an acknowledge message which includes the new mean end-to-end delay, $D_s^{new}(d)$, to node s by backtracking the selected path. Upon receiving the acknowledge message, node s updates its network information with the new entries, i.e., $D_s^{new}(d)$ and $\triangle D_s^{new}(d)$, the latter having been computed from (1). Note that all other entries to other destination nodes remain the same. If no feasible route is found, no acknowledgment is returned and the global information at node s remains unchanged.

The process is repeated for every connection request at node s until an exchange of distance vectors occurs at node s. Such exchange occurs periodically or whenever a topology change is detected, causing an update to the entries of the global information at node s—independent of the previous actions taken (i.e., the number of M_0 selected). Therefore, using the on-policy first-visit Monte Carlo method in this scenario, we want to determine a near-optimal observation-based deterministic policy $\pi : \mathcal{O}_{sd} \rightarrow A$.

C. Path Caching

In [7], the TBP scheme based on the ONMC method is invoked to discover new paths for every connection request. To avoid frequently invoking the path discovery algorithm for every connection request, a path cache can be maintained at each mobile node [3], [4]. Path caching strategies are thus likely to help reduce overhead in MANETs.

In this paper, we use a simple path caching strategy which is readily supported by the TBP scheme. The entries of the path cache are the set of redundant paths discovered from the TBP scheme. The size of the path cache depends on the desired degree of path redundancy. Since paths can be broken at any time, the entries in the path cache can become out-of-date. To deal with such dynamic nature, each path entry in the path cache is validated by a timeout procedure. That is, each path entry requires a *refreshing* message periodically in order to remain in the path cache. The refreshing message is periodically initiated from the destination node, and propagates to intermediate nodes along the path. Once the refreshing message is received at a node, the timer for that path entry is reset and the refreshing message is propagated upstream towards the source node. If no refreshing message is received within a time period, the path entry is deleted.

IV. Numerical Study

The performance of the modified TBP schemes based on the ONMC method are evaluated on MANETs through simulations. To assess their performance, the following four metrics are considered: i) *Accumulated reward* which is equal to the accumulated reward over all episodes divided by the total number of episodes, ii) *Success ratio* which is equal to the total number of accepted connections divided by the total number of connection requests, iii) *Average path cost* which is equal to the total cost of all established connections divided by the total number of established connections and iv) *Average number of search messages* which is equal to the total number of search messages sent divided by the total number of connection requests,. Note that one search message is counted each time a probe is sent over a link. Therefore, a probe which has traversed l hops in the network has created l search messages.

We consider a MANET of 36 nodes placed in a 15×15 square meter area. The topology of the MANET is randomly generated by a random way point mobility model. The velocity is uniformly chosen between 0.3 to 0.7 meters per second. Each node has a circular transmission range with a radius of 3 meters. A link is formed between any two mobile nodes located within this transmission range.

Connection requests are generated at a source node at rate 0.2 connections per second. The cost of each link is uniformly distributed in $[0, 1]$. Each link connecting nodes i and j has two types of link delays associated to it, namely, the actual (D_{ij}) and announced mean link delay $\left(\tilde{D}_{ij}\right)$. The latter type is advertised though the network and used to calculate the mean end-to-end delay $D_j(d)$, for all nodes j and d in the MANET. Each actual mean link delay is uniformly distributed in $[0, 50]$ msecs. Each announced mean link delay is subjected to imprecision so that it is uniformly distributed in the range $\tilde{D}_{ij} \in [D_{ij} - \Delta_{ij}, D_{ij} + \Delta_{ij}]$, $\Delta_{ij} = \xi_{imp} D_{ij}$ and ξ_{imp} is the imprecision rate.[1] The parameters ρ and β in (1) are 0.95 and 1. The maximum hop count allowed in a path is 10. All algorithms exchange distance vectors every 30 seconds interval.

Simulations are run for three algorithms, namely, the original Ticket-Based Probing scheme (TBP), the TBP scheme based on on-policy first-visit Monte Carlo method (ONMC), the TBP scheme based on on-policy first-visit Monte Carlo method (ONMCP), with path caching. All these algorithms omit the green tickets ($G_0 = 0$) and consider only the yellow tickets so that $M_{\max} = \theta_Y = 100$ tickets, to put more emphasis on finding feasible paths rather than low-cost paths. It should be noted that for all the algorithms, the connection request is immediately rejected if the mean end-to-end delay requirement exceeds the best possible end-to-end delay available.

For the all algorithms, the action set (when the connection request is not rejected) is given by $M_0 \in A = \{1, 10, 20, \ldots, 100\}$. The mean end-to-end delay and delay variation (in milliseconds) are quantized into these intervals, $q_D(m) \in \{[0, 10), [10, 20), \ldots, [250, \infty)\}$ and $q_{\Delta D}(l) \in \{[0, 10), [10, \infty)\}$, where $m = 1, \ldots, 26$, $l = 1, 2$, $q_D(m)$ is the m^{th} quantized interval of the mean end-to-end delay between nodes s and d, and $q_{\Delta D}(l)$ is the l^{th} quantized interval of the mean end-to-end delay variation between the two nodes. The ONMC and ONMCP schemes are trained for 4×10^6 connection requests, after which, their performance is evaluated and compared with the TBP scheme. All schemes are evaluated using a simulation run of 1×10^6 connection requests.

Figure 1 shows that the accumulated reward per episode increases as the mean end-to-end delay requirement increases. The reason is because as the mean end-to-end delay requirement increases, it becomes easier to satisfy and thus increased accumulated reward per episode is observed for all algorithms. As the mean end-to-end delay requirement increases, the ONMCP outperforms the ONMC scheme since fewer tickets are issued when path caching is used (path caching reduces the frequency of invoking the feasible path search). Figure 2 shows that the TBP scheme produces the least average number of search messages although at the expense of low accumulated reward per episode. The ONMCP generates less average number of search messages than the ONMC method due to the presence of path caching.

Due to space limitation, results of the success ratio and average path cost of the algorithms are reported without figures. It is found that the success ratio and average path cost of the ONMC outperforms the ONMCP scheme only by upto 3% despite the fact that the latter issues upto 35% fewer tickets. The reason is that the ONMC issues tickets more frequently and therefore has better chances of obtaining least cost paths and feasible paths (i.e. higher success ratio) than the ONMCP method. The TBP scheme attains the least success ratio and highest average path cost.

The final experiment tests the affect of mobility on all algorithms (by increasing the time which the node is stationary, so called pause time) under a fixed mean end-to-end delay requirement. Figure 3 shows that the accumulated reward per episode differs only slightly between the ONMCP and ONMC schemes. The TBP scheme has the least accumulated

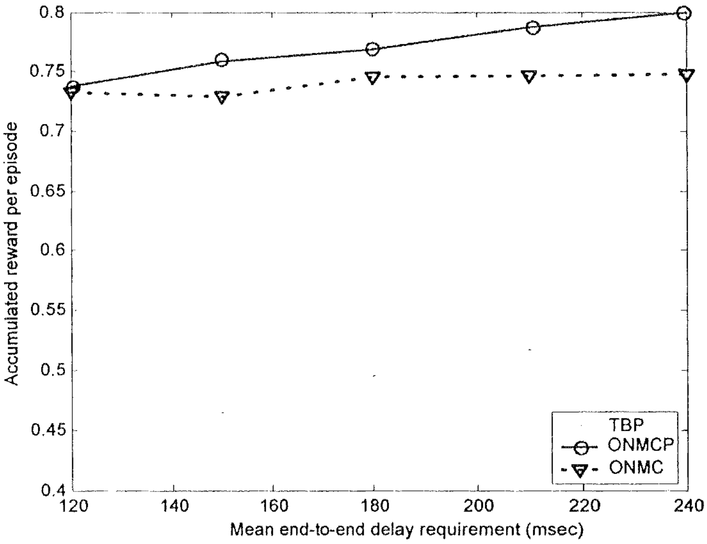

Fig. 1. Accumulated reward per episode with 0.5 imprecision rate.

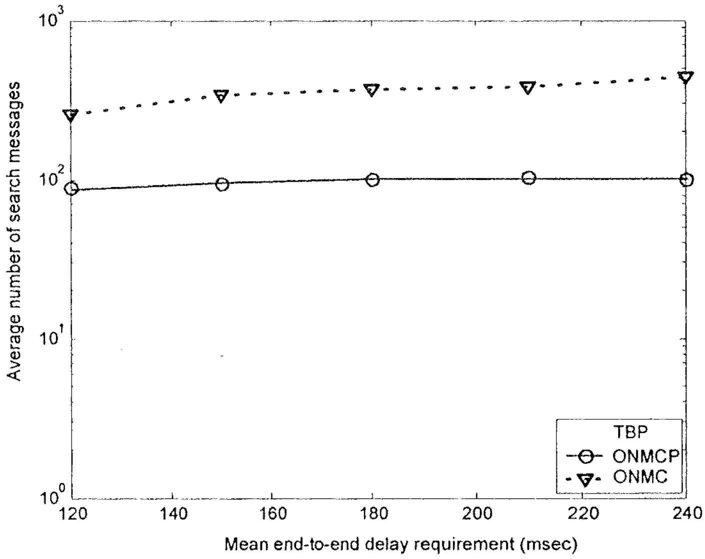

Fig. 2. Average number of search messages with 0.5 imprecision rate.

[1] The imprecision rate specifies the largest percentage of deviation allowed between the actual and advertised link delay, $\xi_{imp} = \max\left(\left|D_{ij} - \tilde{D}_{ij}\right| / D_{ij}\right)$.

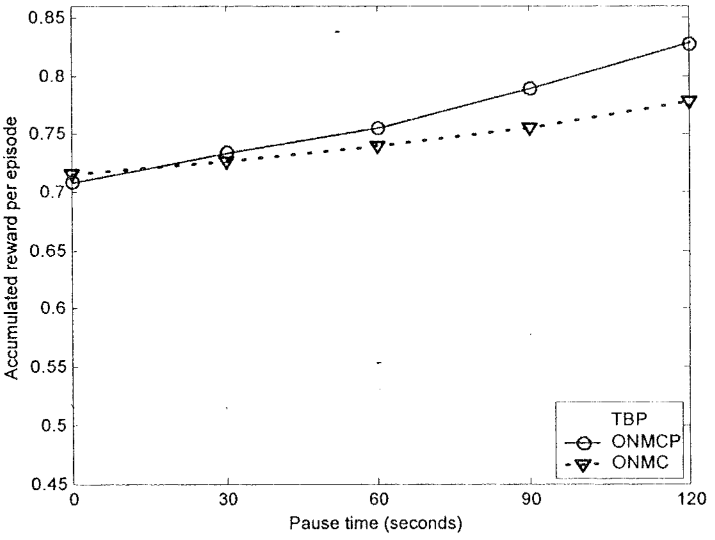

Fig. 3. Accumulated reward per episode with varying pause times.

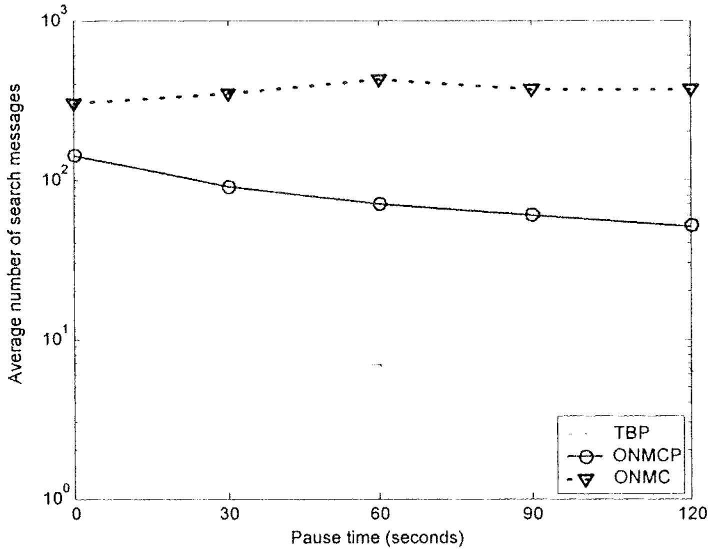

Fig. 4. Average number of search messages with varying pause times.

reward per episode. Since nodes become more stationary, the accumulated reward per episode is increased for all algorithms because it becomes easier to discover paths. In Figure 4, the average number of search messages of the ONMCP and ONMC schemes is reduced because feasible paths can be discovered more easily as the nodes become more stationary. Hence, both schemes learn to issue fewer tickets so as to minimize the number of search messages. However, the ONMCP generates less average number of search messages since the path cache avoids frequently invoking the path discovery algorithm.

The success ratio and average path cost are only reported without figures here due to space limitation. All algorithms exhibit a consistent increase in success ratio as the nodes become more stationary, therefore promoting feasible path searches. The success ratio of the ONMCP is comparable to the ONMC scheme and the TBP scheme shows the least success ratio. In respect to the average path cost, the TBP scheme has the highest average path cost. This is because it can only explore a small number of paths and is likely to discover feasible paths with higher costs. On the contrary, the ONMCP and ONMC scheme give a gradual decline in the average path cost as the pause time is increased.

V. Conclusion

In this paper, the TBP scheme based on a reinforcement learning (RL) technique and path caching, called the ONMCP scheme, is applied to support QoS routing at the network level in a MANET. The simulation study shows that the proposed scheme can achieve good ticket-issuing policies, in terms of the accumulated reward per episode when compared to the original heuristic TBP scheme and the ONMC scheme without path caching. Simulation results also show that message overhead reduction can be achieved with marginal difference in the success ratio and average path cost.

As a final note, the design of different path caching strategies (e.g., cache structure, time-out, capacity) will have a wide affect on the performance of feasible path discovery algorithms. We are currently investigating the integration of the RL-based TBP scheme with other choices of path caching strategies and employ them in different mobility scenarios.

Acknowledgment

The author would like to thank Dr. J. Barria of Imperial College London, for his valuable insights and constructive comments on the thesis which this work is based upon.

References

[1] S. Chen, K. Nahrstedt, "Distributed quality-of-service routing in ad-hoc networks," *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 8, August 1999, pp. 1488-1505.

[2] K. Chen, S. H. Shah, K. Nahrstedt, "Cross-layer design for data accessibility in mobile ad hoc networks," *Journal of Wireless Personal Communications*, Vol. 21, 2002, pp. 49-76.

[3] Y. Hu, D. B. Johnson, "Caching strategies in on-demand routing protocols for wireless ad hoc networks," *Proceedings of the 6th International Conference on Mobile Computing and Networking*, August 2000, pp. 231-242.

[4] P. Papadimitratos, Z. J. Haas, E. G. Sirer, "Path set selections in mobile ad hoc networks," *Proceedings of the MOBIHOC'02*, June 2002.

[5] C. E. Perkins, E. M. Royer, S. R. Das, "Quality-of-service for ad hoc on-demand distance vector routing," *Internet-Draft, draft-ietf-manet-aodvqos-00.txt*, Work In Progress, July 2000.

[6] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, Massachusetts: The MIT Press, 1998.

[7] W. Usaha, *Resource Allocation in Networks with Dynamic Topology*, PhD Thesis, University of London, London, U.K., 2004.