Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

ELSEVIER

Ad Hoc Networks 2 (2004) 319–334

Ad Hoc Networks

www.elsevier.com/locate/adhoc

A reinforcement learning ticket-based probing path discovery scheme for MANETs

W. Usaha [a], J. Barria [b,*]

[a] School of Telecommunication Engineering, Suranaree University of Technology, Nakhon Ratchasima 30000, Thailand

[b] Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, South Kensington, London SW7 2BT, UK

Available online 20 April 2004

Abstract

In this paper, a path discovery scheme which supports QoS routing in mobile ad hoc networks (MANETs) in the presence of imprecise information is investigated. The aim is to increase the probability of success in finding feasible paths and reduce average path cost of a previously proposed ticket based probing (TBP) path discovery scheme. The proposed scheme integrates the original TBP scheme with a reinforcement learning method called the on-policy first-visit Monte Carlo (ONMC) method. We investigate the performance of the ONMC method in the presence of imprecise information. Our numerical study shows that, in respect to a flooding based algorithm, message overhead reduction can be achieved with marginal difference in the path search ability and additional computational and storage requirements. When the average message overhead of the ONMC method is reduced to the same order of magnitude of the original TBP, the ONMC method gains an improvement of 28% in success ratio and 7% reduction in the average path cost over the original TBP.

Keywords: Ticket-based probing; Reinforcement learning; Mobile ad hoc networks; Path discovery; Quality-of-service routing

1. Introduction

A mobile ad hoc network (MANET) consists of a set of mobile nodes (hosts) which are equipped with transmitters and receivers that allow them to communicate without the need of wire-based infrastructures.

Most of the existing routing protocols in MANETs have been focused on only best-effort data traffic. Routing schemes which can support connections with QoS requirements have only recently begun to receive attention. There are two keys to support QoS routing, namely, feasible route search and resource reservation [2–4,12,15]. Feasible route search can be done by distributed routing or source routing. In distributed routing, other nodes apart from the source node are involved in the feasible path(s) search and computation by identifying their neighboring nodes as the next hop router. On the other hand, in source routing, a feasible path(s) is computed solely at the source node. An alternative method is to perform flooding but have the source node calculate some measure of control over the amount of flooding. This is a mixed feature of distributed and source

[*] Corresponding author.
E-mail address: j.barria@imperial.ac.uk (J. Barria).

1570-8705/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2004.03.008

routing which is the underlying concept of the Ticket-Based Probing (TBP) scheme [2,3] where the amount of flooding here can be controlled by issuing a limited number of logical tickets at the source node.

The work reported in this paper builds on the earlier work of [2,3] in that it integrates the reinforcement learning (RL) framework into the TBP scheme. Despite several attractive features—as will be presented later on, the TBP scheme has some outstanding challenges. One of such issues 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. We enhance the original TBP scheme by using a RL technique. The RL-based TBP scheme learns a good rule for issuing tickets by interacting directly with the environment or by simulation. Note that there are other works which apply the RL framework to MANET routing: [16] proposed a multicast routing approach based on Q-learning concept, [13] suggested a possible application of their multiagent routing scheme in MANETs. However, these works do not deal explicitly with QoS requirements or message overhead. More recently in [8] a power-aware routing algorithm is presented. In [8] the authors extend their work on cognitive packet networks (CPN) routing protocol which provides intelligent QoS driven routing for peer-to-peer connections (see also [7,9]). The CPN protocol uses smart packets to discover and maintain lists of neighbor nodes, and to dynamically set up source-to-destination routes. Smart packets select routes using a RL algorithm rather than flooding. In this paper, we gather further experimental evidence on the advantages and limitations of RL techniques when employed to solve the underlying problem of QoS routing in MANETs using flooding based strategies [17]. The underlying aim of the scheme presented in this paper is to maximize the probability of success in finding feasible routes in dynamic topology networks in the presence of inaccurate information.

The paper is organized as follows. In the next section, we present an introduction to the partially observable Markov decision process (POMDP) model in which we state the QoS routing problem in MANETs. In Section 3, we introduce the RL technique used in this paper, namely the on-policy first-visit Monte Carlo method for POMDPs. Section 4 describes the TBP scheme to support QoS routing in MANETs. In this section, the original TBP scheme and the enhanced TBP scheme are presented. Section 5 presents results from the numerical study and the final section provides the conclusion.

2. A POMDP model for MANETs

Partial observability can occur when the topology of the MANET is highly dynamic. In such network, each mobile node acts as a router since there is no fixed infrastructure for routing support. Every mobile node is free to move and can enter or leave the network at any instant. In order to maintain up-to-date routing information at other mobile nodes, message exchanges between mobile nodes are 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.

Furthermore, within MANETs which support quality-of-service (QoS) routing, residual resource information in the network is critical. In particular, each mobile node maintains a state of the network, i.e., the delay and bandwidth information to all other destinations in the network. Note that such information depends on the route and consequently, on the current topology of the network. The information is propagated through the MANET according to some updating protocol. Since an accurate view of such information is difficult to obtain, each mobile node is faced with only an "observation" of its environment which is most likely incomplete and inaccurate. Based on its current network observation, each mobile node acts as an agent which 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. Assuming that each node moves independently from one another and its future movement (position, direction, and velocity) depends only on its current movement, the future topology of the network can depend only on the current topology of the network and is independent of its past. [1] If it is also assumed that the future residual resource information (e.g., delay and bandwidth at each link between two connecting mobile nodes) depends only on the current residual resource information and not its past, then it is possible to (approximately) model the state transitions as a Markov process. In addition, the actual state of the network is concealed from each agent due to mobility. We can therefore (approximately) model the decision-making problem in MANETs as a partially observable Markov decision process (POMDP), whose goal is to optimize 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.

Since our motivation to study POMDP RL algorithms arises from the partial observable information of network resources due to mobility in MANETs, the algorithms we seek should be distributed. Furthermore, such POMDP RL algorithms should also exhibit low computational complexity and demand low storage to reserve onboard processing power on the mobile node. Naturally, memoryless policies (i.e., policies based on direct observations) come to light as they tend to be neither computation nor hardware intensive. These policies, however, have limited success when applied to POMDPs. Fortunately, under certain conditions, optimal memoryless policies can still be guaranteed for some POMDP RL algorithms as discussed in the following section.

3. On-policy first visit Monte Carlo method for POMDPs

Results in [11] show that for a general class of non-Markovian decision processes (NMDP), an optimal memoryless policy can still be guaranteed if: (i) the first-visit Monte Carlo policy evaluation and (ii) the undiscounted reward criterion [2] are used. Based on the findings in [11], we use an on-policy first-visit Monte Carlo (ONMC) method which is originally employed for *completely observable* Markov decision processes (MDPs) [14] to find an observation-based policy in *partially observable* MDPs.

Consider a POMDP with finite sets of states, actions and observations denoted by X, A, and $\mathcal{O}$, respectively. We consider episodic tasks where there is a start state distribution and one or more terminating states. Assume that the terminating state(s) can be reached eventually regardless of whatever actions are taken. Let o_{N-1} correspond to the terminating state and the reward $g(o_{N-1}, a)$ is zero for all actions.

Suppose that an episode, $\{o_0, a_0, g(o_0, a_0), \ldots, o_{N-1}, a_{N-1}, g(o_{N-1}, a_{N-1})\}$, where N is the duration of the episode, is generated under some stationary deterministic policy $\pi : \mathcal{O} \rightarrow A$. We consider only the episodes that the observation-action pair (o, a) occurs, where $o \in \mathcal{O}$ and $a \in A$.

Let t be the tth episode in which (o, a) occurs. Let N_t be the number of time steps in episode t, and $\tau_t(o, a)$ where $0 \leqslant \tau_t(o, a) \leqslant N_t - 1$ be the time step when the first-visit of (o, a) occurs. Let $Q^{\pi_t}(o, a)$ denote the expected reward when starting from an observation-action pair (o, a) and a fixed policy π_t is followed thereafter.

Let the initial policy be π_0 and initialize $Q^{\pi_0}(o, a)$ at the beginning of the episode. For each episode t, generate the actions according to π_t. At

[1] There are several mobility models in MANETs in which divert from this assumption. For example, mobile nodes may move depending on certain mobile nodes (e.g., in a battlefield or on the road), or their movements may depend on their past movement (e.g., in a search and rescue mission, their future and current movements depend on the path from their starting point), etc. In this paper, a conference room mobility model is adopted [2].

[2] For any given sequence of horizon $T, \{o_0, a_0, o_1, \ldots, o_{T-1}, a_{T-1}\}$, the total undiscounted return of the sequence is given by $\sum_{k=0}^{T-1} \alpha^k g(o_k, a_k)$ where the discount factor α is 1, $g(o_k, a_k)$ is the reward associated with taking action a_k when the current observation is o_k. The discounted reward is when $\alpha \in [0, 1)$. Thus, for the *undiscounted* return, all rewards received at the beginning of the sequence are equally significant as the ones received later in the sequence.

the end of episode t, the estimated observation-action value function of (o, a) is updated according to

$$Q^{\pi_t}(o,a) = Q^{\pi_{t-1}}(o,a) + \frac{1}{t}\left(\sum_{k=\tau_t(o,a)}^{N_t-1} g(o_k, a_k) - Q^{\pi_{t-1}}(o,a)\right).$$

Note that the summation term is the accumulative reward following only the *first* occurrence of (o, a) in episode t (thus, the terminology of *first-visit*). Then the greedy policy is found by

$$a^* = \arg\max\{Q^{\pi_t}(o,a)\} \tag{1}$$

and the ϵ-soft on-policy, $\epsilon \in [0, 1]$, is implemented as follows:

$$\pi_{t+1}(o) = \begin{cases} a^* & \text{with probability } 1-\epsilon+\epsilon/|A|, \\ a \in A-\{a^*\} & \text{with probability } \epsilon/|A|, \end{cases} \tag{2}$$

where $|A|$ is the size of the action space. The algorithm converges to a near optimal policy over ϵ-soft policies. The convergence results are derived in [17].

4. TBP scheme to support QoS routing in MANETs

In this section, the ONMC method for POMDPs is integrated into a path discovery scheme called the *Ticket-Based Probing* (TBP) scheme. The 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 [2,3]. The design objective of this algorithm is to maximise 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. The TBP scheme enjoys the following advantages: high tolerance to a high degree of imprecise state information; controlled amount of routing overhead (as opposed to the flooding-based path discovery algorithm); avoidance of any centralized path computation since it is a distributed routing process; path optimality consideration since it takes into account both the cost and the QoS of the path; and the support of multiple path discovery which helps reduce the level of QoS disruption.

In this paper, we study a *delay-constrained least-cost routing* problem. For this constrained routing problem, there are two tasks. Firstly, we need to determine the suitable number of tickets (M_0). In [2,3], $M_0 = Y_0 + G_0$ where Y_0 and G_0 are the number of yellow and green tickets, respectively. These two types of tickets have different purposes. The yellow tickets are for maximizing the chances of finding feasible paths while the green tickets are for maximizing the chances low cost paths.

Secondly, we need to distribute the tickets among the probes in such a way that it maximizes the probability of finding a feasible low-cost path.

In the TBP scheme, the yellow tickets are distributed along low-delay paths thus resulting in a high success probability of finding a feasible path. The strategy for distributing the green tickets is to favor low-cost paths, therefore, obtaining paths with smaller costs which may or may not satisfy the end-to-end requirement. Details of the ticket distribution can be found in [2,3].

QoS state metrics. A node i is assumed to keep the up-to-date local state about all outgoing links. The state information of link (i,j) could include: (1) delay (i,j) the channel delay of the link, including the radio propagation delay, the queueing delay, and background processing time [3]; (2) bandwidth (i,j), the residual (unused) bandwidth of the link [3], (3) the cost (i,j) which can be for example the link availability as a function of residual battery lifetime [1,6,8], forward packet loss ratio [10] and minimum required energy for successful reception [5].

4.1. Initial ticket calculation: original TBP scheme

Consider a connection request whose source, destination nodes and mean end-to-end delay requirement are s, d and $Dreq$, 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

$$D_n(d) = \sum_{(i,j)\in r^*} D_{ij}. \tag{3}$$

The parameter $\Delta D_n(d)$ is the variation of the mean end-to-end delay which is computed from

$$\Delta D_n^{\text{new}}(d) = \rho \Delta D_n^{\text{old}}(d) + (1-\rho)\beta |D_n^{\text{new}}(d) - D_n^{\text{old}}(d)|. \tag{4}$$

The parameter ρ is the forgetting factor which determines how fast $\Delta D_n^{\text{old}}(d)$ is forgotten, $(1-\rho)$ determines how fast $\Delta D_n^{\text{new}}(d)$ converges to $|D_n^{\text{new}}(d) - D_n^{\text{old}}(d)|$, and β is a parameter chosen to ensure a large value of $\Delta D_n^{\text{new}}(d)$. Note that by increasing β, we increase $\Delta D_n^{\text{new}}(d)$ and consequently, the certainty that the actual delay falls in the imprecise range. The parameter Y_0 is determined according to these heuristic rules [2,3]:

$$Y_0 = \begin{cases} 1 \ \text{if}\ Dreq > D_s(d) + \Delta D_s(d), \\ \left\lceil \dfrac{D_s(d) + \Delta D_s(d) - Dreq}{2 \times \Delta D_s(d)} \times \theta_Y \right\rceil \\ \quad \text{if}\ D_s(d) - \Delta D_s(d) \leqslant Dreq \leqslant D_s(d) + \Delta D_s(d), \\ 0 \ \text{if}\ Dreq < D_s(d) - \Delta D_s(d), \end{cases} \tag{5}$$

where 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:

$$G_0 = \begin{cases} 1 \ \ \text{if}\ Dreq > \Theta \times (D_s(d) + \Delta D_s(d)), \\ \left\lceil \dfrac{\Theta \times (D_s(d) + \Delta D_s(d)) - Dreq}{\Theta \times (D_s(d) + \Delta D_s(t)) - D_s(d)} \times \theta_G \right\rceil \\ \quad \text{if}\ D_s(d) \leqslant Dreq < \Theta \times (D_s(d) + \Delta D_s(d)), \\ \left\lceil \dfrac{Dreq - D_s(d) + \Delta D_s(d)}{\Delta D_s(d)} \right\rceil \times \theta_G \\ \quad \text{if}\ D_s(d) - \Delta D_s(d) \leqslant Dreq < D_s(d), \\ 0 \ \ \text{if}\ Dreq < D_s(d) - \Delta D_s(d), \end{cases} \tag{6}$$

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

The intuitive reasoning behind the above rules is simple. If $Dreq$ is very large, then a single yellow ticket suffices. If $Dreq$ is within the estimated range, then more yellow tickets are assigned for more stringent $Dreq$. In the case where $Dreq$ 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. However, the system parameter, Θ, allows a certain degree of lenience to search for large-delay but least-cost paths. As $Dreq$ decreases, G_0 increases. But as $Dreq$ approaches $D_s(d) - \Delta D_s(d)$, it becomes more difficult to find a feasible path so finding a least-cost path is less significant than finding a feasible path. Therefore,

G_0 is decreased in order to reduce the routing overhead in the network while Y_0 is increased. When *Dreq* becomes too difficult to satisfy, no green tickets are issued. The selection of the system parameters (θ_Y, θ_G and Θ) is a practical design issue. These parameters can depend on level of overhead control imposed on the network, or the source–destination distance, etc. [3]. Note also that, theoretically, the TBP scheme becomes a flooding scheme when θ_Y or θ_G is infinity.

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

RL methods (such as the ONMC method) can be applied to the actual system or simulator to obtain a good ticket issuing policy which balances the trade-off in the number of issued tickets and the probability of discovering feasible paths. More specifically, the initial number of tickets, M_0, in the TBP path discovery scheme is determined by the ONMC method. So instead of calculating M_0 from an heuristic rule like in (5) and (6), 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. The role of the agent is played by each mobile node in the MANET. The network state uncertainty (partial observability) is due to the mobility of the nodes in the network.

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

$$\mathcal{O}_{sd} = \{[q_D(m), q_{\Delta D}(l)]\colon 1 \leqslant m \leqslant n, 1 \leqslant l \leqslant n_\Delta\},$$

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

Depending on $o_k \in \mathcal{O}_{sd}$ at time k, node s takes an action $a_k \in A = \{0,\dots,M_{\max}\}$ 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. We omit the green tickets ($G_0 = 0$) and consider only the yellow tickets so that $M_{\max} = \theta_Y$ in order to put more emphasis on finding feasible paths rather than low-cost paths. If the action taken is such that $a_k = M_0 > 0$, 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. More specifically, the reward scheme is defined as

$$g(o_k,a_k) = \begin{cases} \zeta_j - \log a_k & \text{if } a_k > 0 \text{ and at least} \\ & \text{one feasible path is found,} \\ -(\zeta_j - \log a_k) & \text{if } a_k > 0 \text{ and no feasible} \\ & \text{path is found,} \\ 0 & \text{if } a_k = 0, \end{cases} \tag{7}$$

where $\zeta_j \in \mathscr{R}^+$ is the immediate reward parameter for service type-j. The logic of the above reward scheme is straightforward: the more tickets issued at the source node, the more likely a feasible path(s) can be found but with a trade-off of introducing more message overhead into the network. Therefore, the obtained reward is less for large values of a_k. On the other hand, issuing tickets economically reduces the chances of finding a feasible path(s). Therefore, we penalize the events if no feasible paths are found when $a_k > 0$. If $a_k = 0$, the connection request is rejected so there is no message overhead nor reward generated from such action.

In the event that multiple feasible paths are discovered, the destination node d selects the least-cost path. It then returns an acknowledge message which includes the new end-to-end delay, $D_s^{\text{new}}(d)$, to node s by backtracking the selected path. Upon receiving the acknowledge message, node s updates its global network information with the new entries, i.e., $D_s^{\text{new}}(d)$ and $\Delta D_s^{\text{new}}(d)$, the latter having been computed from (4). 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. To maximize the probability of discovering a feasible path, note that high-cost paths (e.g. paths that require high transmission energy and/or paths that have low residual battery left) can be tolerated as long as a feasible path can be discovered.

The path discovery 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). For $\omega \in \Omega = \{0, 1, \ldots, \mathscr{K}\}$ where $\mathscr{K}$ is the number of delay-constrained service types in the network, $\omega = j$ for $j > 0$ represents a connection request placed at node s to node d with mean end-to-end delay requirement $Dreq(j)$, and $\omega = 0$ signifies the end of an episode. Therefore, using the on-policy first-visit Monte Carlo method in this scenario, we want to determine a near-optimal observation-based deterministic policy $\pi : \mathcal{O}_{sd} \times \Omega \rightarrow A$.

We consider episodes that (w, a) occur where $w \in \mathcal{O}_{sd} \times \Omega$. Let $\tau_t(w, a)$ be the first occurrence time of (w, a) in episode t where $0 \leqslant \tau_t(w, a) \leqslant N_t - 1$ and N_t is the duration of episode t. Note that at $k = N_t - 1$, $\omega_{N_t - 1} = 0$, and $g(w_{N_t - 1}, a)$ is zero for all actions. The implementation of the on-policy first-visit Monte Carlo method is presented in Appendix A.

5. Numerical results

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:

Accumulated reward
$$= \frac{\sum_t \text{Accumulated reward in episode } t}{\text{Total number of episodes}},$$

Success ratio
$$= \frac{\text{Total number of accepted connections}}{\text{Total number of connection requests}},$$

Average number of search messages
$$= \frac{\text{Total number of search messages sent}}{\text{Total number or connection requests}},$$

Average path cost
$$= \frac{\text{Total cost of all established connections}}{\text{Total number of established connections}}.$$

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. For the results presented next we assume the cost of each link to be uniformly distributed in [0, 1].

We consider a MANET of 36 nodes placed in a 15×15 square meter area as illustrated in Fig. 1. The topology of the MANET is randomly generated by a conference room mobility model [2]. In particular, each mobile node stays in a current location for a period of time which is called *pause time*. After this period is over, each mobile node moves to a new location randomly selected within the area and moves towards it at some constant velocity. The velocity is uniformly chosen between 0.3 and 0.7 m/s. The time it takes to move to the new location is called *moving time*. Each node has a circular transmission range with a radius of 3 m. A link is formed between any two mobile nodes located within this transmission range.

The topology of the network at any given time depends on the stability of each link. A straightforward rule is used here: any link which has been formed longer than 5 s is declared a stable link, otherwise it is a transient link where it will not be included in the path search. Connection requests are generated at a source node at rate 0.2 connections per second. 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 $(\tilde{D}_{ij})$. 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] ms. Each announced mean link delay is subjected to imprecision so that it is uniformly distributed in the range $\tilde{D}_{ij} \in [D_{ij} - \Delta_{ij}, D_{ij} + \Delta_{ij}]$, $\Delta_{ij} = \xi_{\text{imp}} D_{ij}$ and ξ_{imp} is

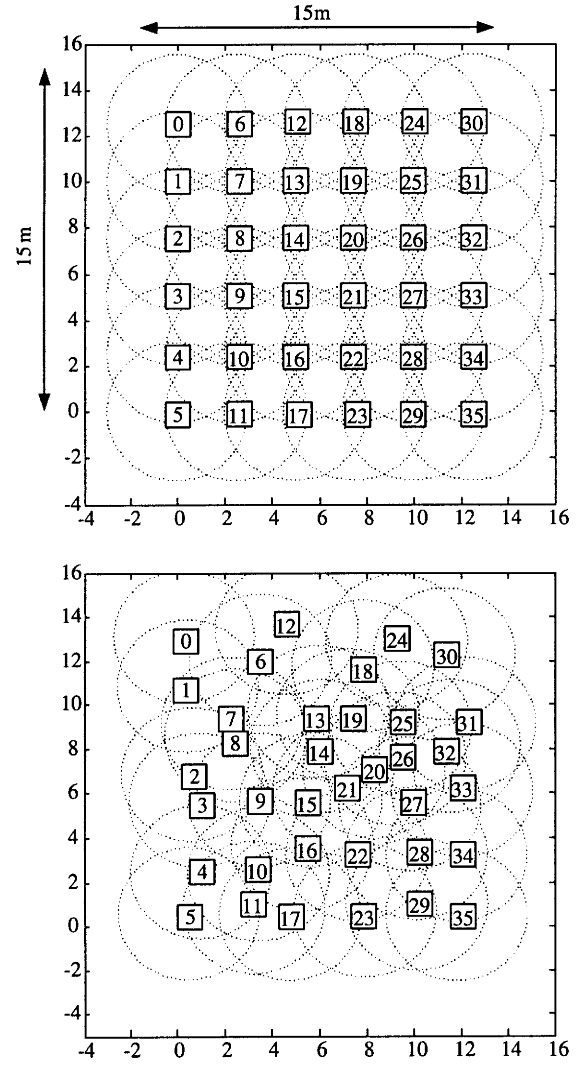


Fig. 1. Test network model of 36 nodes in a 15×15 square meter conference room. Each node has a circular transmission range with a radius of 3 m indicated by the dotted circle around it. The above figure shows the initial coordinates of each node; the lower figure depicts the coordinates after starting simulation.

the imprecision rate. [3] The parameters ρ and β in (4) are 0.95 and 1, respectively so that $\Delta D_n^{\mathrm{new}}(d) = 0.95\Delta D_n^{\mathrm{old}}(d) + 0.05|D_n^{\mathrm{new}}(d) - D_n^{\mathrm{old}}(d)|$. The maximum hop count allowed in a path is 10.

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), and a flooding-based TBP scheme (FLO). The FLO scheme issues $M_{\max}$ tickets for all types of delay-constrained services. 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. Note also that all algorithms exchange distance vectors periodically with the same interval, i.e., every 30 s. Therefore, the same amount of message overhead is generated from the distance vectors exchange in each algorithm, so there is no need to take measurement of this overhead. For this reason, the only measurement of overhead is the number of search messages sent occurred from the feasible path search.

The FLO scheme implemented here constantly issues $M_{\max}$ tickets for all types of delay-constrained services. For the remaining algorithms, the action set (when the connection request is not rejected) is given by

$$M_0 \in A = \{1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}.$$

Note that this action set has a coarse granularity. Better results would be expected if a finer granularity is obtained considering the structural and the dynamic characteristics of the underlying network.

The mean end-to-end delay and delay variation (in ms) are quantized into these intervals,

$$q_D(m) \in \{[0,10), [10,20), \ldots, [240,250), [250,\infty)\},$$
$$m = 1, \ldots, 26,$$

$$q_{\Delta D}(l) \in \{[0,10), [10,\infty)\}, \quad l = 1,2,$$

where $q_D(m)$ is the mth quantized interval of the mean end-to-end delay between nodes s and d, and $q_{\Delta D}(l)$ is the lth quantized interval of the

[3] The imprecision rate specifies the largest percentage of deviation allowed between the actual and advertised link delay, $\xi_{\mathrm{imp}} = \max|D_{ij} - \widetilde{D}_{ij}|/D_{ij}$.

mean end-to-end delay variation between the two nodes.

The ONMC schemes is trained for 4×10^6 connection requests under a 30-s distance vector update interval. Once completed, its performance is evaluated and compared with the TBP and FLO schemes—all schemes are evaluated using a simulation run of 1×10^6 connection requests.

5.1. Accumulated reward per episode

Fig. 2 compares the accumulated reward per episode of all algorithms as a function of the mean end-to-end delay requirement at 0.5 imprecision rate. The pause time is 60 s and the update interval is 30 s. Note that as the mean end-to-end delay requirement increases, the easier it is to satisfy and thus increased accumulated reward per episode is observed for all algorithms. The TBP scheme generates the least reward. At mean end-to-end delay requirements 120–180 ms, the FLO and ONMC schemes are comparable—indicating that for such stringent mean end-to-end delay requirements, a large number of tickets is preferred. In [17], it is observed that under 0.5 imprecision rate, the accumulated rewards per episode generally lower than those under 0.1 imprecision rate for all algorithms because it is more difficult to discover feasible paths when the imprecision rate increases.

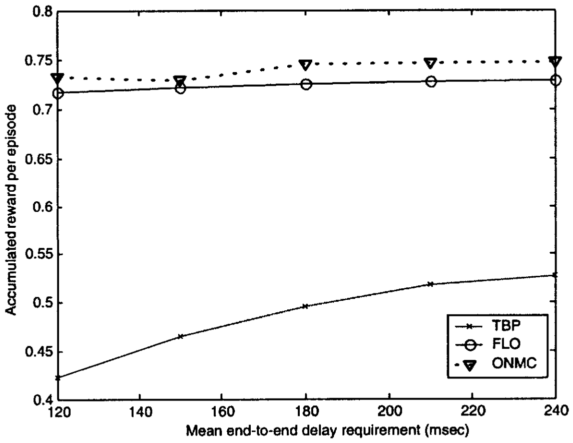


Fig. 2. Accumulated reward per episode with 0.5 imprecision rate.

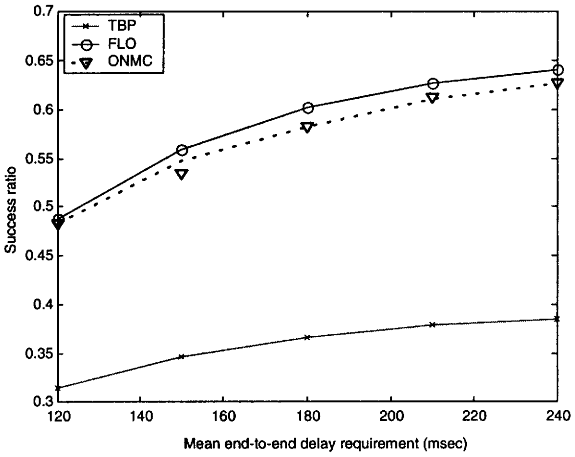


Fig. 3. Success ratio with 0.5 imprecision rate.

5.2. Success ratio

Fig. 3 shows the success ratio of the algorithms obtained from the same experiment in Section 5.1. As expected, the FLO scheme produces the best success ratio since it requires the maximum number of tickets for the path discovery process. Therefore, the probability of finding at least one feasible path is maximized in the FLO scheme. The next best success ratio corresponds to that of the ONMC scheme, and finally the TBP scheme. In [17], it is observed that the success ratio at 0.1 imprecision rate is higher than those observed in the 0.5 imprecision rate for all algorithms. Note also that the success ratio increases as the mean end-to-end delay requirement becomes less stringent since feasible paths become easier to find.

5.3. Average path cost

Fig. 4 compares the average path cost of the algorithms from the same experiment in Section 5.1. The figure shows that the FLO scheme has the lowest average path cost whereas the TBP scheme has the highest path cost of all. The path cost from the ONMC scheme is close to that of the FLO scheme. Such results arise from the different number of tickets issued at the source nodes: the FLO utilizes the most tickets thus has better chances in finding least cost paths.

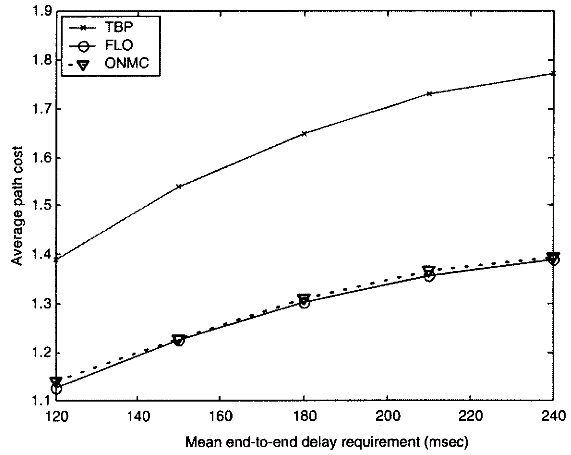


Fig. 4. Average path cost with 0.5 imprecision rate.

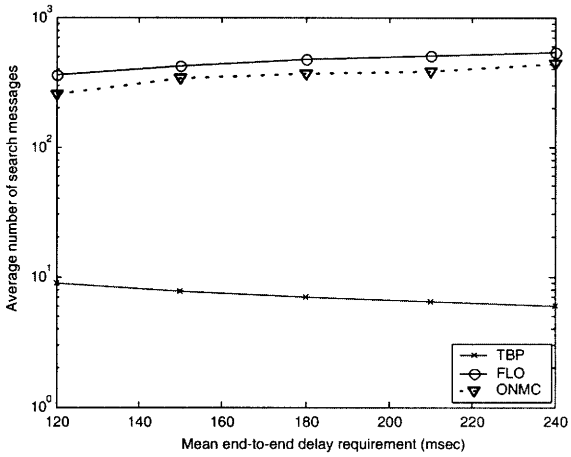


Fig. 5. Average number of search messages with 0.5 imprecision rate.

It is also observed that as the mean end-to-end delay requirement increases, so does the average path cost. This is because at higher mean end-to-end delay requirements, the paths discovered tend to be longer thus increasing the path cost and therefore bringing up the average path cost.

5.4. Average number of search messages

Fig. 5 compares the average number of search messages (on a logarithm scale) of the algorithms from the same experiment in Section 5.1. The figure shows that the FLO scheme generates the highest average number of search messages since the maximum allowed number of tickets is always issued. On the other hand, the TBP scheme generates the least average number of search messages following the heuristic rule in (5). However, this is at the expense of low success ratio and accumulated reward per episode, and higher average path cost with respect to other algorithms. The ONMC scheme produces an average number of search messages between that of the FLO and TBP schemes.

It was observed in [17] that both 0.1 and 0.5 imprecision rate cases, the average number of search messages under the FLO scheme increases with the mean end-to-end delay requirement. The reason is because there are more nodes which can receive the probes since the criteria [4] to receive a probe becomes easier to satisfy as the mean end-to-end delay requirement increases. Since each node forwards the probes to its own neighbors, the ticket distribution propagates over a larger number of nodes in the network even though the number of tickets issued at the source node is fixed (i.e. $M_{\max}$)—giving rise to an increased average number of search messages. For the case of the TBP scheme, a reduction in the average number of search messages is observed. This is due to the heuristic rule in (5) which decreases the number of tickets issued at the source node as the mean end-to-end delay requirement increases. The intuitive reasoning behind this rule is that more tickets should be issued at the source node for connection requests with stringent mean end-to-end delay requirements since feasible paths are difficult to find; and as the mean end-to-end delay requirement increases, the number of tickets required in the search should be reduced subsequently since

[4] A node j can receive a probe from node i only if the best case delay scenario is satisfied, i.e., $Delay(p) + D_{ij} + D_j(d) - \Delta D_j(d) < Dreq$, where $Delay(p)$ is the collected path delay on the path probe p has traversed so far, D_{ij} is the link delay, $D_j(d)$ is the delay from node j to destination node d, $\Delta D_j(d)$ is the mean end-to-end delay variation and $Dreq$ is the mean end-to-end delay requirement.

the feasible path becomes easier to discover. The gradual reduction of the average number of search messages under the TBP scheme is observed in both imprecision cases.

For the ONMC scheme, the average number of search messages increases with the mean end-to-end delay requirement as observed with the FLO scheme—however, at a lower value. This can be explained by the lower number of issued tickets at the source node resulting from the trained ticket-issuing policies.

Although the reduction of the average number of search messages under the ONMC scheme is not observed as the TBP scheme, the average number of search messages is still controlled. This is observed from the fact that, as the mean end-to-end delay requirement increases, the average number of search messages increases only slightly—indicating that the propagation of ticket distribution in the network is very marginal despite the more easily-satisfied mean end-to-end delay requirement. This implies that the number of tickets issued by the source node is reduced under the ONMC scheme. The reason is that feasible paths are easier to discover at higher mean end-to-end delay requirements, hence less penalty and more reward is received when fewer tickets are issued. The ONMC scheme learns to reduce the tickets through these reward signals.

5.5. Robustness to mobility

Figs. 6–9 shows how mobility affects the accumulated reward per episode, success ratio, average path cost and average number of search messages. The algorithms are tested under 0.1 imprecision rate, the maximum number of allowable tickets ($M_{\max}$) is 100, for mean end-to-end delay requirement 180 ms and update interval of 30 s. The mobility of the test network is varied by increasing the pause time from 0 to 30, 60, 90 and 120 s—the network becoming more stationary as the pause time increases. Fig. 6 shows that for pause times 0 and 30 s, the accumulated reward per episode differ only slightly between the FLO and ONMC schemes, while the TBP scheme has the least accumulated reward per episode. As the nodes become more stationary, the accumulated reward per episode is increased for all algorithms since more links are stable and it becomes easier to discover paths.

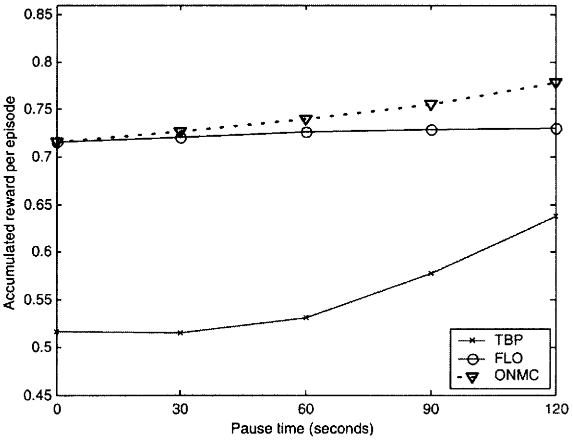


Fig. 6. Accumulated reward per episode under varying pause times.

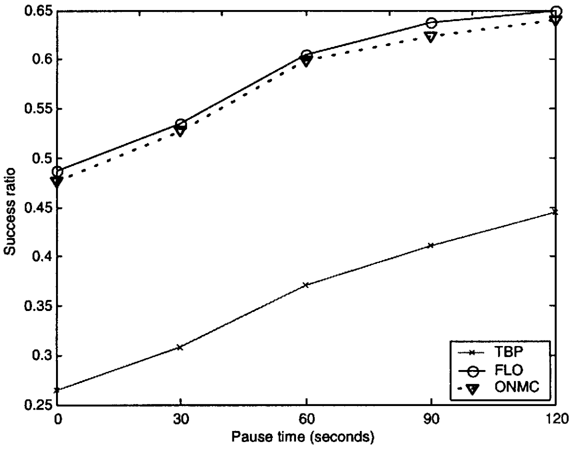


Fig. 7. Success ratio under varying pause times.

Fig. 7 shows the success ratio plotted against the pause time. All algorithms exhibit a consistent increase in success ratio as the nodes become more stationary. Once again, the FLO scheme has the highest success ratio of all while that of ONMC scheme is almost as good as the FLO scheme, and the TBP scheme shows the least success ratio. This can similarly be explained by the reduced mobility which promotes feasible path searches.

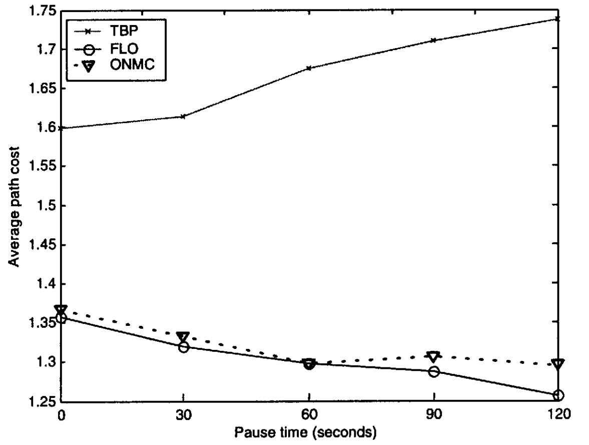


Fig. 8. Average path cost under varying pause times.

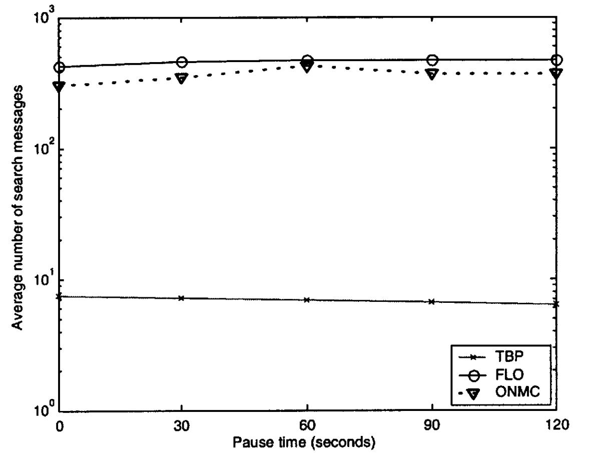


Fig. 9. Average number of search messages under varying pause times.

The average path cost is illustrated in Fig. 8. The TBP scheme shows an increase in the average path cost as the pause time increases. This is due to the fewer number of tickets issued at the source node. Note that more stable links are available when the pause time increases and this, in turn, produces more choices of paths between the source and destination node. With a few tickets issued by the original heuristic rule in (5), the TBP can only explore a small number of paths and is likely to discover feasible paths with higher costs. On the other hand, the FLO and ONMC scheme gives a gradual decline in the average path cost as the pause time is increased. The decline in the FLO is monotonic. However, that of ONMC scheme is non-monotonic due to the reduced number of tickets issued at the source node when network becomes more stationary—with fewer tickets, fewer low cost and feasible paths are discovered.

Fig. 9 confirms this scenario: at pause times 90–120 s, the average number of search messages from the ONMC scheme is reduced (due to the reduced number of tickets). The reason is that feasible paths can be discovered more easily, hence the ONMC scheme learns to issue fewer tickets so as to minimize the number of search messages.

5.6. *Effect of the maximum allowable number of tickets*

So far, the maximum number of allowable tickets ($M_{\max}$) is kept constant at 100. The results presented next show the effect of decreasing $M_{\max}$. The purpose of this experiment is to examine the performance of the algorithms as each algorithm is forced to use fewer tickets (thereby reducing the discrepancy in the amount of message overhead in each algorithm). The algorithms are tested under 0.1 imprecision rate, for mean end-to-end delay requirement 180 ms, pause time 60 s and update interval of 30 s. The parameter $M_{\max}$ is varied from 5 to 100 tickets with action sets as follows:

$$\begin{aligned}
\mathcal{A}_5 &= \{1,2,3,4,5\},\\
\mathcal{A}_{10} &= \{1,2,3,4,5,6,7,8,9,10\},\\
\mathcal{A}_{20} &= \{1,2,4,6,8,10,12,14,16,18,20\},\\
\mathcal{A}_{40} &= \{1,10,20,30,40\},\\
\mathcal{A}_{60} &= \{1,10,20,30,40,50,60\},\\
\mathcal{A}_{80} &= \{1,10,20,30,40,50,60,70,80\},\\
\mathcal{A}_{100} &= \{1,10,20,30,40,50,60,70,80,90,100\}.
\end{aligned}$$

Fig. 10 compares the success ratio from this experiment. As expected, the FLO scheme shows the highest success ratio with the FLO scheme giving 1–6% higher success ratio than the ONMC scheme. In addition, the success ratio of the ONMC scheme still outperforms the TBP scheme by 28–62%. Note that the granulity of the action

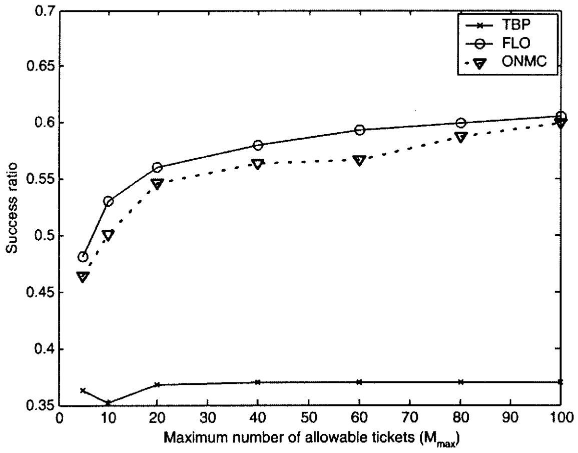


Fig. 10. Effect of the maximum number of allowable tickets on the success ratio.

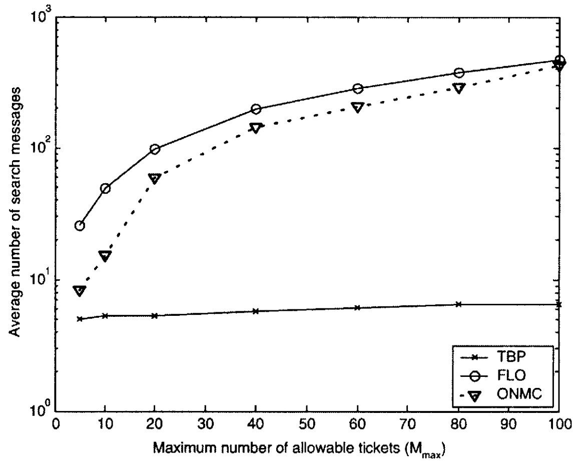


Fig. 11. Effect of the maximum number of allowable tickets on the average number of search messages.

space plays a crucial role on the performance of the ONMC scheme. A finer granulity of action space gives better choices of actions to select from. A coarse granulity action space may force the ONMC scheme to select a higher number of tickets than necessary therefore increasing the message overhead. We are currently performing further investigation on the selection of granulity of the action space and quantization of $q_D(m)$ and $q_{\Delta D}(l)$ under various structures and dynamics of the MANET.

The corresponding results for the average number of search messages are shown in Fig. 11. All methods exhibit reduction in the average number of search messages. Note that the discrepancy in the average number of search messages between the FLO and the ONMC scheme is evident as $M_{\max}$ is reduced. Furthermore, we observe that when $M_{\max}$ is 5 tickets, the ONMC and TBP schemes have the same order of magnitude of the average number of search messages—however, the ONMC scheme gains 28% higher success ratio than the TBP scheme.

Fig. 12 shows the corresponding average path cost. A discrepancy in the average path cost of up to 3% is observed between the ONMC and FLO schemes. The ONMC scheme shows an improvement in the average path cost between 7% and 29% over the TBP scheme. With the same magnitude of average number of search messages (i.e. when $M_{\max}$ is 5 tickets), the ONMC scheme gives 7% lower average path cost than the TBP scheme.

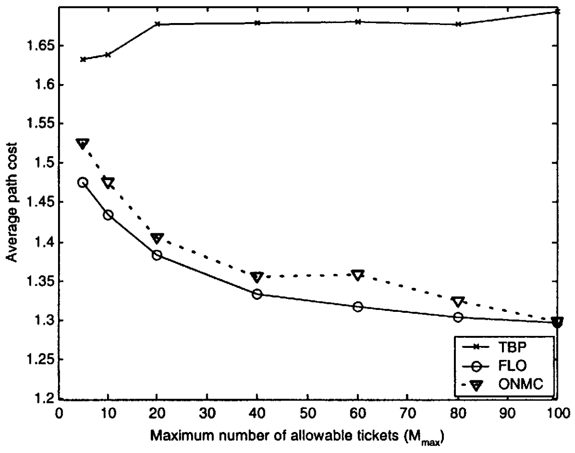


Fig. 12. Effect of the maximum number of allowable tickets on the average path cost.

5.7. Implementation issues

The TBP framework discussed so far is suitable for medium size networks (say, for instance, less than 100 nodes). The reason for this is because each mobile node i must maintain the values of the mean end-to-end delay, $D_j(d)$, and mean end-to-end delay variations, $\Delta D_j(d)$, for all neighbors

$j \in \mathbb{V}_i^S$ to all destinations $d \in \{1,\ldots,\mathcal{N}\}$. Such values are used to forward the received tickets (see [2,3]) and are exchanged via a distance-vector protocol periodically. If a MANET has at most 100 nodes and the maximum node connectivity degree of 10, and if each $D_j(d)$ and $\Delta D_j(d)$ require 4 bytes, the maximum storage requirement for these values at each node i is no more than 8 kB ($100\times10\times2\times4$ bytes). For the ONMC scheme, each node i requires no more than 227 kB ($99\times26\times2\times11\times4$ bytes) to store entries for all destinations. That is, there are 99 destination nodes—for each destination node, one entry is needed to store an observation-action value at $(q_D(m), q_{\Delta D}(l), M_0)$, where $q_D(m)$, $m = 1,\ldots,26$ and $q_{\Delta D}(l), l = 1,2$ and $M_0 \in A = \{1,10,20,30,40,50,60,70,80,90,100\}$. In terms of the number of iterations required to compute one on-line decision, the ONMC method requires $O(|\mathcal{O}||A|)$ iterations where $|\mathcal{O}|$ and $|A|$ are the sizes of the observation and action spaces.

6. Conclusion

In this paper, the TBP scheme based on the ONMC method studied in [17], is applied to support QoS routing in a MANET environment. The reinforcement learning (RL)-based ONMC method, relies on a look-up table representation which stores a value function for every observation and action pair.

The simulation study shows that the TBP schemes based on the ONMC method can achieve good ticket-issuing policies, in terms of the accumulated reward per episode, higher success ratio and lower average path cost, when compared to the original heuristic TBP scheme and a flooding-based TBP scheme. The RL-based TBP path discovery scheme (based on the ONMC method) here proposed, is flexible enough to foster various other objectives and costs functions proposed in the recent literature. In the present version of the RL-based TBP algorithm, decisions are made only once at the source node as new call requests are offered to the network.

Preliminary numerical results reported here suggest that the ONMC scheme can control the amount of flooding in the network. More specifically, it achieves 22.1–58.4% reduction in the average number of search messages compared to the flooding-based TBP scheme with marginal reduction 0.5–1.7% in success ratio. In addition, the ONMC scheme can attain 13–24.3% higher success ratio than the original heuristic TBP scheme at the expense of higher average message overhead. However, as the maximum number of allowable tickets is reduced to a level in which the average message overhead of the ONMC and the original TBP schemes are of the same magnitude, the ONMC scheme still gains 28% higher success ratio and 7% lower average path cost over the original heuristic TBP scheme.

In terms of implementation, the savings in the amount of generated search messages obtained by the RL-based TBP schemes is at the expense of reasonable storage and computational requirements of on-line decision parameters. The storage requirements grow linearly with the number of nodes in the network. The ONMC method requires $O(|\mathcal{O}||A|)$ iterations where $|\mathcal{O}|$ and $|A|$ are the sizes of the observation and action spaces. Note that $|\mathcal{O}|$ depends on the granulity of the quantized delay intervals whereas $|A|$ depends on the number of tickets.

From the results of our experimental work gathered so far, it can be said that RL techniques can play an important role in controlling search messages overhead in environments in which the outcome of a decision is only partially observable. It is important to note that parameters of the algorithm and the granularity of, for example, the action set A is important and further investigation is being carried out at present on these issues. We are currently investigating methods to further reduce the average number of search messages and the integration of TBP schemes with other POMDP RL approaches which will be reported in a forthcoming paper.

Acknowledgements

The paper is part of the first author's Ph.D. research work at Imperial College London which had been supported by the Royal Thai Government.

Appendix A. Algorithm ONMC

On-policy first-visit Monte Carlo method for M_0 selection. Let π_0 be the initial policy. Initialise $Q^{\pi_0}(w,a)$.

```
1. for episode $t = 1$ to $T$ do
2.    for $k = 0$ to $N_t - 1$ do
      (a) At time step $k$, node $s$ has an observation $o_k \in \mathcal{O}_{sd}$ of the network, $\omega_k = j$.
      (b) If $Dreq(j) < D_s(d) - \Delta D_S(d)$,
      (c)       $M_0 = 0$. Reject the connection request.
      (d) Else,
      (e)       $M_0 = a$ is selected according to policy $\pi_t(w_k)$.
      (f) Get immediate reward $g(w_k, a_k)$ and next observation $o_{k+1} = \widetilde{D}_s^{\text{new}}(d)$.
3.    end
4.    Perform updates:
5.    $Q^{\pi_t}(w,a) = Q^{\pi_{t-1}}(w,a) + \frac{1}{t}\left(\sum_{k=\tau_t(w,a)}^{N_t-1} g(w_k,a_k) - Q^{\pi_{t-1}}(w,a)\right)$.
6.    $\pi_{t+1}(w) = \begin{cases} a^* & \text{with Probability } 1 - \epsilon + \epsilon/|A| \\ a \in A - \{a^*\} & \text{with probability } \epsilon/|A| \end{cases}$
      where $a^* = \arg\max\{Q^{\pi_t}(w,a)\}$.
7. end
```

References

[1] S. Banerjee, A. Misra, Minimum energy paths for reliable communication in multi-hop wireless networks, in: Proceedings of the MOBIHOC'02, 2002.

[2] S. Chen, Routing support for providing guaranteed end-to-end quality-of-service, Ph.D. Thesis, University of Illinois at Urbana-Champaign, IL, 1999.

[3] S. Chen, K. Nahrstedt, Distributed quality-of-service routing in ad-hoc networks, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1488–1505.

[4] K. Chen, S.H. Shah, K. Nahrstedt, Cross-layer design for data accessibility in mobile ad hoc networks, Journal of Wireless Personal Communications 21 (2002) 49–76.

[5] S. Doshi, S. Bhandare, T.X. Brown, An on-demand minimum energy routing protocol for a wireless ad hoc network, Mobile Computing and Communication Review 6 (6) (2002) 50–66.

[6] L. Feeney, M. Nilsson, Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, in: Proceedings of the IEEE INFOCOM, 2001.

[7] E. Gelenbe, R. Lent, Z. Xu, Measurement and performance of a cognitive packet network, Computer Networks 37 (2001) 691–701.

[8] E. Gelenbe, Self-aware network and QoS, in: Proc. ISCIS XVIII, Lecture Notes in Computer Science, vol. 2869, Springer, Berlin, 2003, pp. 1–14.

[9] E. Gelenbe, R. Lent, A power aware routing algorithm, in: Proceedings of SPECTS'03, Summer Simulation Multiconference, Society for Computer Simulation, 2003.

[10] E. Gelenbe, M.Gellman, P. Su, Using loss and delay as QoS goals in Cognitive Packet Networks, in: Proceedings of SPECTS'03, Summer Simulation Multiconference, Society for Computer Simulation, 2003.

[11] M.D. Pendrith, M.J. McGarity, An analysis of direct reinforcement learning in non-Markovian domains, in: Proceedings of the 15th International Machine Learning Conference, 1998.

[12] 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.

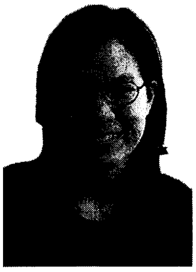
[13] E. Peshkin, V. Savova, Reinforcement learning for adaptive routing, in: Proceedings of the International Joint Conference on Neural Networks'02, 2002.

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

[15] R. Sivakumar, P. Sinha, V. Bharghavan, CEDAR: a core-extraction distributed ad hoc routing algorithm, IEEE Journal on Selected Areas in Communications 17 (8) (1999) 1454–1465.

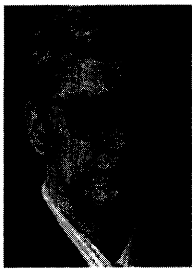
[16] R. Sun, S. Tatsumi, G. Zhao, Application of multiagent reinforcement learning to multicast routing in wireless ad hoc networks ensuring resource reservation, in: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2002.

[17] W. Usaha, Resource allocation in networks with dynamic topology, Ph.D. Thesis, University of London, London, UK, 2004.



Wipawee Usaha received a B.Eng. (Hons) degree from Sirindhorn International Institute of Technology, Thammasat University, Thailand, and M.Sc. and Ph.D. degrees in Electrical and Electronic Engineering, all from Imperial College London, University of London, UK, in 1996, 1998, and 2004, respectively. She is currently with the School of Telecommunication Engineering, Suranaree University of Technology, Thailand.

Dr. Usaha has received His Majesty King Bhumibol Scholarship in 1996 awarded for the best graduate (in Electrical Engineering) in Sirindhorn International Institute of Technology, Thammasat University. Her research interests include resource allocation in communication networks and reinforcement learning.



J.A. Barria (http://jab.ee.ic.ac.uk/jbarria.html) is a Senior Lecturer in the Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, at Imperial College London. He has a B.Sc. degree in Electronic Engineering (U. of Chile, 1980), a Master of Electronic Engineering (PII, The Netherlands, 1982), a Ph.D. degree (Imperial, 1992) and an MBA degree (Imperial, 1998). He is a member of IEE, member of IEEE and a Chartered Engineer. He has published in major international journals and conferences on areas concerned with telecommunication networks. British Telecom Research Fellow (2001–2002). He has been actively researching areas concerned with robust communication system design, load control and routing strategies, fault-tolerant networks and resource management. Other areas of interest include the application of intelligent agent systems and distributed algorithms for telecommunication networks. He is the joint holder of several FP5 and FP6 European Union project contracts all concerned with aspects of communication network design and management.