Learning methods in ad-hoc networks: a review

Michal Kudelski, Andrzej Pacut

Institute of Control and Computation Engineering
Warsaw University of Technology
00-665 Warsaw, Poland

Abstract. The paper presents a comprehensive review of learning methods used to solve various problems in ad-hoc networks. The learning methods are classified according to learning mechanisms and problems solved. Nine representative approaches are discussed in more detail.

1 Introduction

Since their emergence in the 1970s, wireless networks became increasingly popular in the area of telecommunications. As a fixed access to the global computer network — the Internet — becomes common, the researchers and industry efforts aim to make computer networking pervasive and ubiquitous.

One way to achieve this is to build ad-hoc networks. A basic mobile ad-hoc network (called MANET [9]) is a set of wireless mobile nodes that form dynamically a temporary network without using any existing infrastructure. Each node of such network can be autonomous and self-configurable hence no centralized administration is needed. It is also common that communication between the nodes can be indirect, namely, every node can also act as a wireless router and can be used as an intermediate node on a path from a source to a destination. Many variations of this structure are also possible. The ad-hoc network can be a part of an existing fixed network structure (hybrid ad-hoc networks [24]) or other highly capable infrastructure such as mesh networks [2] or cellular networks. Sensor networks [1] can be also identified as a sort of ad-hoc networks.

Solving problems in ad-hoc networks is a challenging task due to their highly dynamic environment and many constraints (capacity, energy, etc). On the other hand, there is a strong need for development because of many potential application areas like commercial networks, military, emergency services and many others. These two facts make ad-hoc networks a very desired, demanding and interesting field of research.

The main aim of this paper is to review the learning methods that can be successfully applied to solve typical ad-hoc networks problems. The additional objectives of this review are:

- to show the motivation of using learning methods in ad-hoc networks,
- to classify the learning approaches according to learning mechanisms and problems being solved,
- to show the efforts of Artificial Intelligence community in the field of ad-hoc networks.

The paper is organized as follows. Section 2 provides a general overview of learning methods in ad-hoc networks. This overview presents the major motivations and a basic classification of methods. Different learning approaches are presented in the next three sections. In Sec. 3, the reinforcement learning mechanism is briefly described and five RL based methods for solving different problems are discussed. Two ant-based routing algorithms, for pure ad-hoc networks and for hybrid networks, are presented in Sec. 4. In Sec. 5 other learning approaches are described: a learning automata based approach (Sec. 5.1) and a supervised learning approach (Sec. 5.2). We will conclude our review in Sec. 6.

2 Learning methods in ad-hoc networks

It is commonly known that the ad-hoc network makes a very demanding environment. There are many motivations for using learning approaches in such networks. The need of high adaptation abilities is probably the strongest one. Ad-hoc networks are dynamic, fast changing systems of unknown and changing structure and parameters. Adaptive learning structures seem to be a very adequate solution for such systems. Moreover, a non-deterministic and complex behavior of ad-hoc networks makes it difficult or even impossible to use classical methods. On the other hand, the utility of learning approaches in communication networks has already been proved for fixed networks [22].

Several categories of learning mechanisms can be distinguished in ad-hoc networks, namely the reinforcement learning (Sec. 3), SWARM intelligence (Sec. 4), learning automata (Sec. 5.1), supervised learning (Sec. 5.2) and other, including immune systems and genetic algorithms.

We can also group the learning approaches by the problems they solve, and they include routing (Secs. 3.1, 3.5, 4.1, 4.2), QoS provisioning (Sec. 3.2), energy management and topology control, node's movement prediction (Sec. 5.1), reliability optimization and preventing malicious node (Secs. 3.3, 5.2).

3 Reinforcement learning based methods

Reinforcement learning (RL) techniques are based on interactions between an agent(s) and an environment. In every state, the agent chooses an action according to his present policy and receives a *reinforcement* from the environment evaluating his actions. On the basis of the reinforcement and the state reached after taking the action according to agent's policy, the agent modifies its policy to increase its chance of attaining its goal, which is typically the maximization of the discounted sum of all reinforcements. There exists many learning algorithms which are modifications of the basic RL technique. The first applied to routing in telecommunication networks was Q-Routing proposed by Littman [4].

In this section we introduce five RL based methods applied to solving four different problems in ad-hoc networks.

3.1 Routing

The problem of finding the best routes from a source node to a destination node is a basic issue in a communication network. In ad-hoc networks the problem can be reduced

to finding the neighborig node that would be the best intermediate node on the path to the destination (in terms of the given metrics). We chose *LQ-Routing* algorithm [30] as a prominent example of a reinforcement learning approach to routing in ad-hoc networks. LQ-Routing can be viewed as a combination of Q-Routing and Destination-Sequenced Distance-Vector algorithm (DSDV) [23], which is known as a typical proactive routing protocol for mobile ad-hoc networks.

In DSDV, routing updates are propagated periodically by each host as *advertisements*. Each advertisement is marked with a sequence number to avoid an infinite circulation of advertisements. In the original Q-Routing algorithm, Q-value $Q_x(y,d)$ is defined as the expected routing time from the current node x to the destination node d through the selected neighbor node y. This value is stored in the *routing table* and used to select the next hop as a part of the routing policy. The message delivery process is as follows. When a host x receives a packet P destined for d, it selects a neighbor y such that $Q_x(y,d)$ is minimal over all neighbors, and sends it out to y. Upon receiving the packet, y sends back the value of $min_{z \in N_y}\{Q_y(z,d)\}$, where N_y denotes the set of neighbors of y. Then x updates its new estimate of $Q_x(y,d)$, namely:

$$Q_x(y,d) \leftarrow (1-\alpha)Q_x(y,d) + \alpha\{Q_y(z,d) + q_y + \gamma\}, \tag{1}$$

where q_y denotes the expected waiting time in the FIFO queue, γ denotes the expected transmission delay between any two neighboring hosts, and α is the *learning rate*. The update mechanism is in fact more complex, to include certain randomness that enables for *exploration* of the environment.

LQ-Routing associates Q-values acquired by the learning process in Q-Routing with the estimated message delivery times. The *lifetime* is introduced to avoid a degradation of the convergence speed of the underlying learning process, which is caused by the dynamic changes in ad-hoc networks. A path lifetime, $PL_x(y,d)$, measures the stability of a route from x to the destination d through the neighbor y and is updated due to the number of successful advertisement transmissions from y to x. The Q-value is slightly modified to use the path lifetime:

$$LQ_x(y,d) = \frac{Q_x(y,d)}{\max\{\epsilon, PL_x(y,d)\}}, \tag{2}$$

where ϵ is a small constant value. The neighbor with the smallest LQ-value is selected. Path lifetime values are propagated by advertisements with consecutive numbers assigned, similarly to the mechanism used in DSDV. By combining DSDV with Q-Routing, the routing policy is adaptively acquired from dynamically changing environment.

Performance of the proposed scheme was simulated in NS2 environment. The results indicate that LQ-Routing may outperform DSDV under heavy load levels, regardless of the mobility of the nodes. There are still some open issues, like a more realistic path lifetime evaluation.

There are also other RL approaches to solving the routing problem in ad-hoc networks, including Q-Routing inspired reactive routing scheme [8], a path discovery schemes [31, 32], power-aware routing scheme [15], SAMPLE routing scheme [12], and QMAP multicast routing scheme [29].

3.2 QoS provisioning

Quality of Service provisioning is another problem of ad-hoc networks that can be solved using the reinforcement learning approach. We present as an example the *Wire-Fitted Reinforcement Learning Provisioning* (WFRLP) scheme proposed in [36].

WFRLP is a joint bandwidth allocation and buffer management scheme for QoS provisioning in Differentiated Services framework in wireless ad-hoc networks. The system is modeled as a Semi-Markov Decision Process (SMDP) and the RL algorithm is used to maximize the average long term reward and to minimize QoS violations at the same time.

Suppose there are K classes of network services in the system. Each class i defines a minimum amount of bandwidth $b_{i,min}$, and the absolute packet delay d_i and a normalized loss l_i (i.e. packet buffer loss or dropping) constraints: $d_i < d_{i,max}$ and $l_i < l_{i,max}$. The *state descriptor* is defined as:

$$S = [b_1, n_1, b_2, n_2, ..., b_K, n_K], \tag{3}$$

where b_i is the current bandwidth used by class i, and n_i is the current number of packets in the i-th class queue. The following system events for the state transitions were identified: a) changes in the routing path, b) MAC layer notifications, such as transmission failures and c) packet arrivals. When any of these events occurs, the bandwidth is allocated and the buffer management is performed. The action is defined as a vector:

$$a = [a_{b1}, a_{d1}, a_{b2}, a_{d2}, ..., a_{bK}, a_{dK}], \tag{4}$$

where a_{bi} is the bandwidth level allocated to class i, and a_{di} is the number of packet buffer drops for class i. The reward function combines the objective of maximizing the average long term reward with minimizing the average QoS violations. Additional specific objectives are introduced to avoid changes of the current service rate allocation and to avoid dropping traffic.

To solve the problem of bandwidth allocation and buffer management, a model-free RL algorithm known as the Semi-Markov Average Reward Technique (SMART) is applied. Action values $Q(s_t, a_t)$ are estimated by using the temporal difference method. The action with the highest value is performed in a given state, with a small probability of exploration. If action a_t is chosen at t-th decision period at state s_t, the corresponding $Q(s_t, a_t)$ is updated as follows:

$$\begin{aligned} Q_{new}(s_t, a_t) = (1 - \alpha_t) Q_{old}(s_t, a_t) + \alpha_t \{ r(s_{t+1}, s_t, a_t) - \sigma_t \tau_t \\ + \max_{a_{t+1} \in A} Q_{old}(s_{t+1}, a_{t+1}) \}, \end{aligned} \tag{5}$$

where α_t is the learning rate parameter, $r(s_{t+1}, s_t, a_t)$ is the actual cumulative reward earned between two successive decision epochs starting in state s_t, with action a_t and ending in state s_{t+1}, and τ_t is the time difference between state s_t and s_{t+1}. The reward rate σ_t is updated according to:

$$\sigma_t = (1 - \beta_{t-1}) \sigma_{t-1} + \beta_{t-1} \frac{T(t-1)\sigma_{t-1} + r(s_{t+1}, s_t, a_t)}{T(t)}, \tag{6}$$

where β_t is the learning rate parameter, and $T(t)$ is the total time spent in all visited states until the t-th decision period. To facilitate the convergence of the RL algorithm a novel function approximation technique (wire-fitted CMAC) is used.

Simulation performed in NS2 environment indicates that WFRLP is suitable for fast and real-time learning and is able to attain robust convergence. In comparison with the heuristic mechanism built in NS2 (JoBS), a better average reward is achieved.

QoS provisioning may be also improved by using specialized routing schemes. Two QoS routing mechanisms based on RL are introduced in [14] and [28].

3.3 Preventing malicious nodes in ad hoc networks

Maneenil and Usaha [19] propose a RL based method of building a reputation scheme for selecting neighboring nodes in a path search. The method uses an on-policy Monte Carlo (ONMC) methodology, where sample episodes are used to estimate the value function.

In each episode, a path to the destination is searched for. Each node maintains the reputation information to all of its neighboring nodes in a form of reputation values. Reputation values are quantized and form a discrete state space. An action is defined as choosing the neighbor for further transmission. In each following node, the neighbor is selected according to reputation values, and the process is repeated until the destination is found or the maximum count of hops is reached. If the route search is successful, a constant positive reward is assigned to every node on all successful paths. Otherwise, a null reward is assigned to all involved nodes.

Experimental results show that this approach can achieve up to 89% increase in throughput for the static topologies in comparison with the reputation scheme using a fixed threshold. Up to 29% increase is achieved for dynamic topologies.

3.4 Mobilized ad-hoc networks

Yu-Han Chang [7] deals with a problem of *mobilized ad-hoc networks*, where the nodes movements are controlled. The problem can be devided into two subproblems: the routing problem in ad-hoc networks and the problem of finding a nodes movement policy that maximizes network connectivity.

Chang proposes a RL approach to solve both routing and node control problems. A slightly modified Q-Routing [4] is used for routing, and Q-Learning [35] is applied for mobility control. The learning process consists of two phases: the node movement problem is handled off-line, and then the acquired movement policy is used during the execution phase.

Q-Routing is adapted to dynamically changing topology by two simple modifications. The action's value is set to infinity when communication with a neighbor is lost, and is set to 0 when a new neighbor appears. The movement policy problem is seen as a partially-observable Markov decision process solved with Q-Learning. Each node accesses only the local network state observations (neighboring nodes and their connections) and can choose one of complex actions (e.g., randomly explore, circle around a node in search for more connections, etc.). The reward corresponds to the percentage of successful transmissions.

The simulations show that the acquired policy is better than other known movement policies and better than a hand-coded policy (provided that the same information is given).

3.5 Routing in Cognitive Packet Networks

Cognitive Packet Networks (CPN, [16]) are networks in which intelligent capabilities are concentrated in the packets, rather than in the nodes and protocols. Packets within a cognitive packet networks make decisions (e.g. routing decisions) themselves. Executable code and data needed for decision making are embedded into the packets. Nodes only serve as buffers, mailboxes and processors that can be used by the cognitive packets.

In [16] it is shown how learning can support intelligent behavior of cognitive packets. Authors deal with the routing problem in ad-hoc networks. Random neural networks are used as a decision model and a RL technique is applied to learn a decision policy. Simulation results show that applying learning can improve the overall performance of the CPN.

4 SWARM Intelligence methods

The Ant Colony Optimization (ACO) metaheuristic was inspired by the behavior of ants [11]. This multi-agent approach solves many discrete optimization problems, like the shortest path problem or the traveling salesman problem. In 1996 and 1998 the first ant routing algorithms were proposed for telecommunication networks [22].

4.1 Routing

AntHocNet [10] is a hybrid multipath ant routing algorithm for mobile ad-hoc networks. The multipath routing problem is the problem of finding multiple paths from a source node to a destination node and distributing the traffic between these paths according to their quality. AntHocNet does not maintain the paths to all possible destinations at all times, but only sets up paths when they are needed (reactive phase) and then monitors these path as long as the transmission continues (proactive phase).

When the source node s starts communication with the destination node d while not having an appropriate routing information, it broadcasts a *reactive forward ant* F_d^s. The reactive ant is further unicast or broadcast at each node, depending on whether the routing information is available, and its goal is to find a destination d. The routing information is represented by a pheromone value $T_{nd}^i \in R$ which estimates a quality of the path from the source i to the destination d through the neighbor n. The next hop is chosen according to the probability

$$p_{nd} = \frac{(T_{nd}^i)^{\beta_1}}{\sum_{j \in N_d^i} (T_{jd}^i)^{\beta_1}} \quad , \quad \beta_1 \geq 1, \tag{7}$$

where β_1 is the exploration parameter and N_d^i denotes the set of neighboring nodes over which a path to d is known. Each forward ant stores its path details. When the destination is reached, the forward ant is transformed into the *backward ant* and travels back using exactly the same path. In each intermediate node i, the pheromone values T_{nd}^i are updated according to

$$T_{nd}^i = \gamma \; T_{nd}^i + (1 \; - \; \gamma) \; \tau_d^i \quad , \quad \gamma \in [0, 1], \tag{8}$$

where

$$t_d^i = \left(\frac{T_d^i + h \; T_{hop}}{2} \right)^{-1} \quad , \tag{9}$$

T_d^i is the traveling time estimated by the ant, h is the number of hops, T_{hop} is a fixed value of the time to take one hop in unloaded conditions, and γ is the learning parameter.

The nodes in AntHocNet forward data stochastically, according to the probabilities similar to (7), but with a higher exponent $\beta_2 \geq \beta_1$. During data transmission, a *proactive forward ant* is generated every n-th data packet. Proactive ants act similar to the reactive ants, serving two purposes: they monitor the existing paths and explore new paths. Additional mechanisms are built in for link failures management (failure notification messages and repair ants) and neighborhood monitoring (periodic hello messages).

Simulations performed in [10] compare AntHocNet with AODV, which is known as a typical reactive routing protocol for ad-hoc networks. The results show that AntHocNet outperforms AODV in simulation scenarios with highly dynamic topology. AntHocNet achieves lower delay and higher delivery ratio. However, there are some open issues like limiting the routing overhead and improving proactive information exchange.

There are also other known SWARM intelligence routing algorithms for ad-hoc networks. Two ant-based routing algorithms are proposed in [3] and [37]. Topology control algorithm is introduced in [27]. A GPS routing algorithm is proposed in [6].

4.2 Routing in hybrid networks

In [24], an ant routing algorithm ANSI is proposed for *hybrid ad-hoc networks*. Such networks consist of a *highly capable infrastructure* such as a wired network, a mesh network or a cellular network, and a set of mobile nodes that build an ad-hoc network. Mobile nodes can communicate with the fixed nodes via gateways.

Only the reactive routing is performed in the ad-hoc sections of the network. The main difference in comparison to AntHocNet is that only a single backward ant is generated (only by the first forward ant). This means that only one route is established, what should result in lower contention in the mobile part of the network. In addition to the reactive route discovery, hello messages are periodically generated, that contain information about the network load level.

In highly capable sections of the network, additional proactive routing is performed apart from the reactive path discovery. Non-mobile nodes exchange information about their connections and about mobile nodes in their range, so they can assist the reactive routing process within the mobile nodes when possible. Non-mobile nodes can also perform a stochastic multipath routing.

Simulation results are promising, especially for hybrid networks. ANSI is able to effectively utilize high capable links and therefore to outperform AODV.

5 Other learning approaches

5.1 Learning automata

A learning automata to solve the problem of cache allocation for web services for an individual user is introduced in [17]. Each user movements are predicted, based upon a learning automata scheme, and the cache is allocated according to the predicted user's location.

Learning Automata (LA) are finite state adaptive systems that interact continuously with an environment. Through a probabilistic trial-and-error response process, they learn to choose or adapt to the behavior that generates the best response. To use LA for the

path prediction problem, the space is divided into hexagonal cells and a state transition matrix is defined. Each entry of this matrix contains the following information: a previous cell id, a current cell id, a future cell id, a time slot, a probability value, and a time stamp. The probability value measures the likelihood that the user, previously located at the previous cell, migrates from the current cell to the future cell within a specific time slot. Hence, the state of the process is defined as a triplet of the previous cell, the current cell, and the time slot.

In the first step of the learning process, an input is provided to the LA from the environment. This input triggers one of the possible responses from the LA. The environment receives the response and then provides a feedback to the LA. Such feedback is used by the LA to update its state transition matrix and improve its behavior. When the LA selects the right response (the prediction is accurate), the positive feedback received by the environment causes the respective state transition to be rewarded, and otherwise it is penalized, according to:

$$\begin{aligned} &transition\ (i \to j)\ received\ positive\ feedback: \\ &\qquad \begin{cases} P_{ij} = P_{ij} + \gamma(1 - P_{ij}), \\ P_{ik} = P_{ik}(1 - \gamma), \quad k \neq j; \end{cases} \\ &transition\ (i \to j)\ received\ negative\ feedback: \\ &\qquad \begin{cases} P_{ij} = P_{ij} - \gamma(1 - P_{ij}), \\ P_{ik} = P_{ik}(1 + \gamma), \quad k \neq j; \end{cases} \end{aligned} \tag{10}$$

where γ is a design parameter, $0 < \gamma < 1$.

Simulations show that this method can achieve up to 70% accuracy of the first hit after 8-9 weekes of training.

LA can be also used to solve other problems of ad-hoc networks. In [33] authors introduce a learning automata based approach for adaptive selection of the congestion window size for the TCP protocol. In [20] a learning automata based adaptive MAC protocol is proposed. Other LA-based approaches may be found in [13, 21].

5.2 Supervised learning

The goal of supervised learning is to predict the output value based on an input vector [34]. Learning is performed on a set of training samples. In [34] it is shown that the supervised learning can be used to solve the problem of link quality estimation in sensor networks.

Some features characterizing a node (e.g. buffer size, signal level, etc.) were chosen and a set of samples is gathered for different load levels. These samples are then used in the off-line training process. Estimated link quality classifiers are built that can be further used for the on-line link quality estimation. Simulations suggest that this supervised learning framework can help to make routing and reliability decisions in sensor networks.

Other application of supervised learning can be found in [5], where this methodology is used for intrusion detection in mobile ad-hoc networks.

6 Conclusions

In this paper we made a comprehensive review of learning methods used within the wide area of ad-hoc networks. We categorized learning approaches according to learning mechanisms and problems solved. Nine representative approaches were presented in more detail.

The review does not cover all the learning methodologies used in ad-hoc networks. For instance, other methodologies include back propagation learning [26], evolving fuzzy neural networks [18] used to solve the routing problem, and immune system applied to nodes misbehavior detection [25].

Acknowledgments

This work has been supported by Warsaw University of Technology Research Program grant.

Bibliography

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "A survey on sensor networks", *IEEE Commun. Mag. 40(8)*, pp. 102-114, 2002.

[2] Ian F. Akyildiz, Xudong Wang, Weilin Wang, "Wireless mesh networks: a survey", *Computer Networks 47*, pp. 445-487, 2005.

[3] B. Awerbuch, D. Holmer, H. Rubens, "Swarm Intelligence Routing Resilient to Byzantine Adversaries", *International Zurich Seminar on Communications*, pp. 160–163, 2004.

[4] J. A. Boyan and M. L. Littman, "Packet Routing in Dynamically Changing Networks: A Reinforcement Learning Approach", *Advances in Neural Information Processing Systems*, vol. 6, pp. 671-678, Morgan Kaufmann Publishers Inc., 1994.

[5] J.B.D. Cabrera, C. Gutierrez, R.K. Mehra, "Infrastructures and algorithms for distributed anomaly-based intrusion detection in mobile ad-hoc networks", *IEEE MILCOM*, vol. 3, pp. 1831–1837, 2005.

[6] D. Camara, A.A.F. Loureiro, "GPS/Ant–Like Routing in Ad Hoc Networks", *Telecommunication Systems 18*, pp. 1-3, 85-100, 2001.

[7] Yu-Han Chang, "Mobilized ad-hoc networks: A reinforcement learning approach", *Proceedings of International Conference on Autonomic Computing*, pp. 240–247, 2004.

[8] D. Chetret, C. Tham, L. Wong "Reinforcement Learning and CMAC-based Adaptive Routing for MANETs", *IEEE ICON*, vol. 2, pp. 540–544, 2004.

[9] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing protocol Performance Issues and Evaluation Considerations", *Internet draft from http://www.ietf.org/rfc/rfc2501.txt*, 1999.

[10] G. Di Caro, F. Ducatelle and L. M. Gambardella, "AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks", *European Transactions on Telecommunications*, vol. 16, pp. 443–455, 2005.

[11] M. Dorigo, V. Maniezzo, and A. Colorni, "Positive feedback as a search strategy", *Politecnico di Milano Technical Report 91016*, 1991.

[12] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, "Using Feedback in Collaborative Reinforcement Learning to Adaptively Optimize MANET Routing", *IEEE Transactions On Systems, Man, and Cybernetics*, vol. 35, pp. 360–372, 2005.

[13] A.I. El-Osery, D. Baird, W. Abd–Almageed, "A Learning Automata Based Power Management for Ad–Hoc Networks", *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3569–3573, 2005.

[14] P. Fu, J. Li, and D. Zhang, "Heuristic and Distributed QoS Route Discovery for Mobile Ad hoc Networks", *IEEE CIT*, pp. 512–516, 2005.

[15] E. Gelenbe, R. Lent, "Power–aware ad hoc cognitive packet networks", *Ad Hoc Networks 2*, pp. 205-216, 2004.

[16] E. Gelenbe, Z. Xu and E. Seref, "Cognitive Packet Networks", *Proceedings of IEEE Tools with Artificial Intelligence*, pp. 47–54, 1999.

[17] S. Hadjiefthymiades and L. Merakos, "Proxies + Path Prediction: Improving Web Service Provision in Wireless-Mobile Communications", *Mobile Networks and Applications 8*, pp. 389-399, 2003,

[18] C. Huang, L. Chen,Y. Lin, Y. Chuang, W. Kuang Lai, S. Hsiao, "A Zone Routing Protocol for Bluetooth MANET with Online Adaptive Zone Radius", *IEEE ICICS*, pp. 579–583, 2005.

[19] K. Maneenil, W. Usaha, "Preventing malicious nodes in ad hoc networks using reinforcement learning", *2nd International Symposium on Wireless Communication Systems*, pp. 289–292, 2005.

[20] P.Nicopolitidis, G.I.Papadimitriou, A.S.Pomportsis, "An Adaptive MAC Protocol for Ad–Hoc Wireless LANs", *58th Vehicular Technology Conference*, vol. 2, pp. 1383–1386, 2003.

[21] B.J. Oommen, S. Misra, "A Fault–Tolerant Routing Algorithm for Mobile Ad Hoc Networks Using a Stochastic Learning–Based Weak Estimation Procedure", *IEEE WiMob*, pp. 31–37, 2006.

[22] A. Pacut, M. Gadomska, A. Igielski, "Ant-Routing vs. Q-Routing in Telecommunication Networks", *Proceedings of the 20-th ECMS Conference*, pp. 67–72, 2006.

[23] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", *Proceedings of ACM SIGCOMM*, pp. 212-225, 1994.

[24] Sundaram Rajagopalan, Chien-Chung Shen, "ANSI: A swarm intelligence-based unicast routing protocol for hybrid ad hoc networks", *Journal of Systems Architecture 52*, pp. 485-504, 2006.

[25] S. Sarafijanovic, J. Boudec, "An Artificial Immune System Approach With Secondary Response for Misbehavior Detection in Mobile ad hoc Networks", *IEEE Transactions on Neural Networks*, vol. 16, pp. 1076–1087, 2005.

[26] Y. Shang, M.P.J. Fromherz, Y. Zhang, L.S. Crawford, "Constraint-based Routing for Ad–hoc Networks", *IEEE ITRE*, pp. 306–310, 2003.

[27] C. Shen, Z. Huang, C. Jaikaeo, "Ant–Based Distributed Topology Control Algorithm for Mobile Ad hoc Networks", *Wireless Networks 11*, pp. 299–317, 2005.

[28] P. Su, M. Gellman, "Using adaptive routing to achieve Quality of Service", *Performance Evaluation 57 (Elsevier)*, pp. 105-119, 2004.

[29] R. Sun, S. Tatsumi, G. Zhao, "Q_MAP: a novel multicast routing method in wireless ad hoc networks with multiagent reinforcement learning", *Proceedings of IEEE TENCON*, vol. 1, pp. 667–670, 2002.

[30] T. Tao, S. Tagashira, S. Fujita, "LQ-Routing Protocol for Mobile Ad-Hoc Networks", *IEEE ICIS*, pp. 441–446, 2005.

[31] W. Usaha, J. Barria, "Reinforcement learning ticket–based probing path discovery scheme for MANETs", *Ad Hoc Networks 2*, pp. 319-334, 2004.

[32] S. Varadarajan, N. Ramakrishnan, M. Thirunavukkarasu, "Reinforcing reachable routes", *Computer Networks 43*, pp. 389-416, 2003.

[33] B. Venkata Ramana, B. S. Manoj, and C. Siva Ram Murthy, "Learning–TCP: A Novel Learning Automata Based Reliable Transport Protocol for Ad hoc Wireless Networks", *2nd International Conference on Broadband Networks*, vol. 1, pp. 484–493, 2005.

[34] Y. Wang, M. Martonosi and Li-Shiuan Peh, "Supervised Learning in Sensor Networks: New Approaches with Routing, Reliability Optimizations", *SECON*, vol. 1, pp. 256–265, 2006.

[35] C. J. Watkins, "Learning with delayed rewards", *Ph.D. Thesis, Univ. of Cambridge*, 1989.

[36] Daniel Yagan and Chen-Khong Tham, "Adaptive QoS Provisioning in Wireless Ad Hoc Networks: A Semi-MDP Approach", *Wireless Communications and Networking Conference*, vol 4, pp. 2238–2244, 2005.

[37] S. Ziane, A. Mellouk, "A Swarm Intelligent Scheme for Routing in Mobile Ad hoc Networks", *Proceedings of IEEE ICW*, pp. 2–6, 2005.