American Journal of Applied Sciences

Original Research Paper

A Q-Routing Protocol Using Self-Aware Approach for Mobile Ad hoc Networks

Amal Alharbi, Abdullah Al-Dhalaan and Miznah Al-Rodhaan

Department of Computer Science, College of Information Sciences, King Saud University, Riyadh, Saudi Arabia

Article history
Received: 02-03-2015
Revised: 21-11-2015
Accepted: 23-11-2015

Corresponding Author:
Amal Alharbi
Department of Computer Science, College of Information Sciences, King Saud University, Riyadh, Saudi Arabia
Email: amalhalharbi@gmail.com

Abstract: Mobile Ad hoc Networks (MANET) are self-organized networks that are characterized by dynamic topologies in time and space. This creates an instable environment, where classical routing approaches cannot achieve high performance. Thus, adaptive routing is necessary to handle the challenges in MANETs. Furthermore, it is necessary for nodes to be self-aware i.e., able to discover neighbors, links and paths when needed. This paper proposes a new adaptive Mobile Ad hoc Networks (MANET) routing algorithm to find and maintain paths that provide the needed Quality of Service (QoS) for network traffic using a low-complexity bio-inspired learning paradigm. It combines the self-aware approach in Cognitive Packets Network (CPN) with a Q-routing inspired path selection mechanism. CPN is a distributed adaptive routing protocol that uses three types of packets: Smart Packets for route discovery, Data Packets for carrying data payload and Acknowledgments to bring back feedback information for the Reinforcement Learning reward function. The research defines a Q-routing reward function as a combination of high stability and low delay path criteria to discover long-lived routes without disrupting the overall delay. The algorithm uses Acknowledgment-based feedback for Q-routing to make routing decisions that adapt on line to network changes allowing nodes to learn efficient routing policies. Simulation Results show how the reward function handles the network changing topology to select paths that improve QoS delivered.

Keywords: Cognitive Packet Network (CPN), Mobile Ad hoc Network (MANET), Q-Routing, Reinforcement Learning (RL), Self-Aware Networks (SAN)

Introduction

Mobile Ad hoc networks is a promising research field with rising number of real-world applications. However, MANET environment is randomly dynamic due to node mobility, limited power resources and variable bandwidth as well as other factors as shown in (Lent and Zanoozi, 2001). Therefore, to successfully communicate, nodes need an adaptive distributed routing protocol that adjusts when the network changes. Furthermore, it necessary for nodes to be self-aware such that a node is able to discover neighbors, links and paths when needed (on-demand). Researchers in Artificial Intelligence (AI) have contributed to the network communication field through adaptive routing protocols that use AI algorithms to find efficient routes. Reinforcement Learning (RL) is an AI technique that evaluates the performance of a learning agent regarding a set of predetermined goals (Sutton and Barton, 1998). For each step of the learning process, a reward is provided to the agent by its environment as feedback. At the beginning of the learning process, the agent (learner) chooses actions randomly and then appraise the rewards. After some time, the agent starts gathering knowledge about its environment and is able to take decisions that maximize the reward on the long run. Although RL has been known for a long time, it has been recently applied in several new areas. Some of these disciplines include game theory, simulation-based optimization, control theory and genetic algorithms. Some of these advances in RL are found in RL-Glue (Tanner and White, 2009), PyBrain (Schaul *et al*., 2010), Teaching Box (Ertel *et al*., 2012) and others. RL has also shown to be an appropriate framework to design adaptive network routing policies.

One of the breakthroughs in RL is an off-policy control algorithm called Q-learning (Boyan and Litmann,

1999). It approximates the action value function independent of the policy being used. The first application of RL in communication network packet routing was Q-routing which is based on Q-learning. Section 3.2 Q-routing concept and applications in MANETs.

MANETs are self-organized networks with no fixed infrastructure. Designing MANET protocols faces major challenges due to special characteristics of this type of network. In this study, we present an adaptive routing protocol for MANETs based on Q-learning to improve Quality of Service (QoS) delivered to applications. Nodes in our routing algorithm retrieve information from their environment then adapt to take actions. These actions are chosen according to the feedback in Acknowledgements. We introduce a RL mapping onto our routing model to learn a routing strategy that maintains best QoS.

The paper is organized as follows. Section 2 shows the research problem definition and the objectives of the research. Section 3 describes the background while section 4% the problem solution. Finally, section 5 shows the results analysis and section 6 is the conclusion.

Research Problem

In the last decade, QoS has played a major role in Ad hoc networks with the increase use of multimedia traffic. Heavy use of multimedia needs stable and reliable networks that offer acceptable QoS (Punde and Pissinou, 2003). QoS routing mechanisms decide suitable paths for different application needs (Santhi *et al*., 2011). The main goal is to satisfy the application's traffic requirement while implying efficient utilization of network resources. For this purpose the algorithm must define multi-constrained paths considering both the network state and traffic needs. Thus the path computation algorithm is the basis of the QoS routing strategy. The path selection algorithm should select several alternative paths that satisfy the needed constraints. It should be as much as possible low complexity.

QoS-routing needs good network information gathering. However, information collection in MANETs is considered challenging due to node mobility and the dynamic environment. Thus, a MANET node should be able to proactively discover neighbors, links, paths when needed. This led to the initiation of Self-Aware Networks (SAN), where nodes can join and leave autonomously (Gelenbe, 2014). Paths are discovered dynamically on-demand. Nodes can learn the status of other nodes, links and paths quickly. In general, SANs are considered self-organized networks that use QoS driven approaches (Gelenbe *et al*., 2004). SANs depend on adaptive packet routing protocols such as the Cognitive Packet Network (CPN), which uses RL.

CPN performs routing using three types of packets: Smart Packets (SPs), Data Packets (DPs) and acknowledgments (ACK). SPs are used for route discovery and route maintenance. DPs carry the actual data. An ACK carry feedback information about the route performance. All packets have the same structure: A header, a cognitive map and the payload data. A cognitive map holds information about the nodes visited by the packet and the visiting time. To discover routes, SP's are source-initiated to move through the network gathering specific network information according to the specific Quality of Service (QoS) goals determined in each SP (Gelenbe, 2014).

When a SP arrives at the destination node, an ACK packet is created and sent to the source node. The source node copies the discovered route into all DP's ready to be sent to carry the payload from source to destination. DP's use source routing with the discovered route until a new ACK brings a new better route to the source node. The feedback information from ACK is essential for the Reinforcement Learning (RL) algorithm (Watkins, 1989). Each discovered route is evaluated according to the reward function defined in the RL decision algorithm. The subsequent SP's visiting the same node for the same destination and QoS will learn the efficient routing path depending on these updates (Halici, 2000).

Our research routing algorithm aims to adapt CPN to the MANET environment, focusing on path stability to handle node mobility. We apply the RL technique called Q-learning for its simplicity and reasonable memory requirements. We chose a reward function that is a combination of high stability and low delay to find long-lived routes without disrupting the total end-to-end delay.

The main objective of our Q-CPN routing algorithm is to improve QoS delivered through an adaptive distributed MANET routing mechanism. The major advantage of this routing scheme is improving robustness in the face of a dynamic network topology.

Background

The Q-CPN Routing protocol for MANET is a research protocol, which emphasizes on node stability over space and time into the CPN routing protocol. The first subsection reviews the related work in CPN, while the second subsection reviews the Q-routing based protocols in ad hoc networks.

Cognitive Packet Network

CPN was first introduced to create robust routing for the wired networks in (Gelenbe *et al*., 2001a). It has been tested and evaluated in later studies (Gelenbe *et al*., 2001b) to be adaptive to network changes and congestions. It uses Reinforcement Learning based on Random Neural Networks (Gellman and Liu, 2006). Genetic Algorithms have been used in CPN to modify

and enhance paths (Gelenbe *et al*., 2008). However, studies show that it improved performance under light traffic only and increased the packet delivery delay.

A study in (Gelenbe *et al*., 2004) investigated the number of SP's needed to give best performance. It resulted that SP's in about 10 to 20% of total data packet rate is sufficient to achieve best performance and that a higher percentage did not enhance the performance.

One extension of CPN is Ad hoc CPN (AHCPN) (Gelenbe and Lent, 2004), which uses combination of broadcast and unicast of SP's to search for routes. The authors introduced a routing metric "path availability" which modelled the probability to find available nodes and links on a path. Node availability was measured by the energy stored in the node (remaining battery lifetime).

Enhancements to AHCPN continued as research developed. A new routing metric "Path Reliability" was presented in (Lent, 2006), characterized by reliability of nodes and links. Node reliability was considered to be the probability that a node will not fail over a specific time interval which was estimated to be the average network lifetime. The QoS combined goal function includes maximum reliability and minimum path delay. Reliability was continuously monitored and if it dropped below a certain threshold, the source node was informed to start a new route discovery before link breakage.

Q-Routing Based Protocols

The algorithm in (Boyan and Littman, 1999) first introduced RL in networking to solve the problem of routing in static networks. Their adaptive algorithm Q-routing was based on the RL scheme called Q-Learning. The results reveal that adaptive Q-Routing performed better than shortest path algorithm in static networks under changing network load and connectivity.

Chang and Ho (2005), the authors proposed a straight forward adaptation of the basic Q-routing algorithm to Ad hoc mobilized networks. The main objective was to introduce traffic-adaptive Q-routing in Ad hoc networks. Tao *et al*. (2005) the authors combine Q-routing with Destination Sequence Distance Vector routing protocol for mobile networks. Forster and Murphy (2007), the author used RL to propose a solution to multiple destination communication in WSN using Q-Routing.

Authors in (Santhi *et al*., 2011) propose a MANET Q-routing protocol considering bandwidth efficiency, link stability and power metrics. They applied Q-routing to Multicast Ad hoc Distance Vector (MAODV) and their results showed enhancements in QoS delivered compared to the original MAODV.

Sachi and Parkash (2013) a MANET routing algorithm by combining Q-learning with Ad hoc Distance Vector (AODV) to achieve higher reliability.

Q-Routing in CPN

Q-CPN routing algorithm for MANETS defines the routing process as a Reinforcement Learning (RL) problem. This allows learning the network topology in short time without periodic advertisement of network information or global routing information exchange as in the traditional non-adaptive routing algorithms.

Reinforcement Learning Mapping

Network Routing can be modelled as a RL problem to learn an optimal control policy for network routing. A node is the agent which makes routing decisions. The network is the environment. The RL reward is the network performance measures.

We assume N = {1, 2,...., n} is a set of nodes in mobile Ad hoc network and the network is connected. We also assume that each node has discrete time *t*, where each time step is a new decision problem to the same destination. At time *t* node *x* wants to send a Smart Packet (SP) to some destination Node *d*. Node x must take a decision to whom it sends the SP with minimum delay and maximum path stability possible. The node is the agent and the observation (state) is the destination node. The set of actions the agent (node x) can perform is the set of neighbors to whom it can forwards the packet to.

One simple but powerful RL technique is called Q-learning (Peshkin and Savova, 2002). It is a learning method that does not need a model and a type of value-iteration. It requires no prior knowledge of the system and the agent's perspective is very local. Only Q-tables are needed for implementation. For these advantages, Q-learning is appropriate for routing in dynamic networks. Q-routing is based on Q-learning to learn a network state presented as routing information stored in Q-tables.

In order for the agent to learn a representation of the system (network), Q-values (Peshkin and Savova, 2002) are used. A Q-value is defined as Q(state, action) and has a value which represents the expected rewards of taking action from state. Each state is a destination node *d* in the network and the action is the next hop neighbor node *n*. Each node *x* has its own view of the different states of network and its own Q-values for each pair (state, action) in its Q-table written as $Qx(s,a)$. The structure of the Q-Table is shown in Fig. 1.

The more accurate these Q-values are of the actual network topology, the more optimal the routing decisions are. Thus, these Q-values should be updated correctly to reflect the current state of the network as close as possible. This update, which is also called the learning algorithm, has to be with minimum processing overhead.

The main algorithm for Q-CPN routing mechanism is shown in Fig. 2.

	Neighbor1	Neighbor2	Neighbor3
Destination1	$Q^x(d1,n1)$	$Q^x(d1,n2)$	.....
Destination2	$Q^x(d2,n1)$	$Q^x(d2,n2)$	.....
Destination3	$Q^x(d3,n1)$	$Q^x(d3,n2)$	.....

Fig. 1. The Q-table for node x

1- Set initial Q-values for each node.
2- Get first packet from packet queue of node x.
3- If packet is ACK , go to Learning-Algorithm.
4- If packet is SP, go to Selection-Algorithm.
5- If packet is DP , forward packet to next hop according to Cognitive Map.
6- Go to step 2.

Fig. 2. Main Q-CPN ALGORITHM

Q-CPN Learning Algorithm

Our algorithm relies also on exploitation/exploration framework of Q-routing. It relies on a forward probabilistic exploration method. The ACK packet carries the maximum future reward (Q-value) as well as the timestamp needed to calculate the delay.

When a node receives an ACK, it performs these steps:

- Compute end-to-end delay using timestamps
- Calculate the reward using Equation 1,3 and 4
- Update the Q-value using Equation 2
- Get its maximum Q-value and attach it to ACK
- Forward the ACK to next node

This algorithm is performed *K* times, which is the number of nodes along the path. The "Get Max Q-value" procedure is executed *m* times, which is the number of neighbors to the node. Thus the total time complexity of the learning (update) algorithm for Q-CPN is *O*(*K x m*).

The Goal Function of the routing process is a common goal for all agents (nodes) in the network. The goal is to minimize a combination of the delay and the node's Associativity Ratio. This goal function is expressed mathematically in Equation 1. Associativity Ratio is discussed in section 4.4 Equation 6.

The goal is calculated at every node where the delay is the end-to-end delay from this node to the destination node:

$$G = Delay - A / A_{Threshold} \quad (1)$$

The RL algorithm aims to select optimal actions (neighbors) at each time step in order to maximize the network routing performance on the long run. This implies calculating the reward for all the nodes along the chosen routing path:

$$Q_t^x(d,n) = (1-\alpha)Q_{t-1}^x(d,n) + \alpha R_t \quad (2)$$

The Q-value update is performed according to Equation 2. Where $Q^t{}_x(d,n)$ is the new estimate and $Q^x{}_{t-1}(d,n)$ is the old estimate and Rt is the additive expression of immediate reward plus the future reward as shown in Equation 3. Also α is a learning constant typically close to 1, 0<α<1. Here in this algorithm α = 0.8. It means that the rewards are more important than past experience Q-values, which results in faster learning:

$$R_t = r_t + \gamma \max_z Q_{t-1}^y(d,z) \quad (3)$$

where, r_t is the immediate reward calculated by Equation 4:

$$r = 1/G \quad (4)$$

and $\gamma \max_z Q_{t-1}^y(d,z)$ is the discounted future reward, which is the maximum Q-value from the Q-table of the next hop neighbor. The discount factor γ = 0.9.

Q-CPN Selection Algorithm

There is a challenge in RL problems to find the suitable balance of exploitation and exploration. An agent must prefer to choose actions that it knows their good quality (exploit). However, the agent must also explore in order to discover better routes in the future. Soft-Max approach uses Gibbs or Boltzmann distribution, where probability of selecting an action is based on the Q-value of the action as shown in Equation 5:

$$P = \left(eQ^{(s,a)/t}\right) / \left(\sum_a {}^{(Q(s,a)/t)}\right) \; t > 1 \quad (5)$$

Since a MANET environment is dynamic, Soft-Max exploration best suits our problem. The Q-CPN selection algorithm is shown in Fig. 3.

```
Repeat
   Receive SP at node i
   Get Q-values of all neighbours of node i
   Calculate all link probabilities
   Select chosen nieghbor
   Send SP to next hop neighbour
   Decrease Temp t
Until SP reach Destination
```

Fig. 3. Q-CPN selection algorithm

This selection algorithm is performed for each SP received at each node on the path. If we assume the max number of hops allowed for a SP is N_{max} and the number of neighbors (links) is m, then the total time complexity for the selection algorithm is $O(N_{max} \; x \; m)$. That is due to the procedure "Calculate all link probabilities" which is performed m times at each node on the path.

Q-CPN Neighbor Discovery and Link Quality

MANETs are considered self-configured networks with no communication infrastructure. Thus, neighbor discovery is an essential part of initialization of a MANET (Lent and Zanoozi, 2001). A node has to be able to know at least the one-hop neighbors to communicate with any other node in the network. At any time links may disappear while others might become more stable. Link instability is a major problem in dynamic environments. A good routing protocol needs to update its information about the stability of connections to a node's neighbor.

In Q-CPN, periodic beacons are used to signify node existence. The beacon contains information such as source-identification which is the transmitting node and the timestamp when it got sent. Each node keeps a neighbor table to hold neighbor information needed to update Q-values. A neighboring node increments it's Associativity Ticks for a neighbor each time it receives a beacon from that specific neighbor. Associativity Ticks are initialized to zero value when the neighbor moves away of radio range (Toh, 1997). A neighbor is considered moved away, when a node x does not receive any beacons from this neighbor for three times the beacon period. When a node receives a beacon timeout for a neighbor, a node immediately adjusts the Q-values for this neighbor to all destinations to be a very small number (-∞) to make sure it is not chosen again. On the other hand, when a new node comes into radio range its Q-value is initialized optimistically to the value zero.

Each mobile host periodically sends a beacon to each of its neighbors every beacon interval time (p). We choose this interval in our algorithm to be three seconds. A link with a certain neighbor is considered stable if the Associativity level is above a certain threshold (Toh and Vassiliou, 2000). This threshold is determined by Equation 6, where v is the velocity of the mobile node and r is the diameter of the radio range of the wireless transmission:

$$A^{Threshold} = \frac{r}{pv} \tag{6}$$

Result Analysis

The numerical experiments for studying the use of Q-routing in CPN using Acknowledgement feedback were simulated using MatLab R2013a. An experimental small ad hoc networks of 12 nodes was used to evaluate algorithm behavior regarding network changes such as link failure and congestion.

Sample Network Experiments

The network with twelve nodes is shown in Fig. 4. The source node is node S and destination node is node D. The first experiment sends SP's from source to destination and studies the relationship between the rewards and the corresponding Q-values of the links. The first path discovered consists of S-N1-N2-N4-N10-D and an ACK is sent on the reverse route with the rewards computed for each link. The corresponding Q-values for the path is shown in Fig. 4. Due to the high recorded rewards, all the link Q-values are also high.

However, for the path S-N1-N3-N6-N8-D as shown in Fig. 4, the rewards recorded were not as high as the first path and thus accordingly, the Q-values are lower than the links of the first path. Link N6-N7 has Q-value zero. The path does not reach the destination so there is no ACK sent from the destination to the source and thus the Q-value is never updated as shown in Fig. 4.

The Third path S-N1-N2-N5-N9-D records close Q-values to path 1, which indicate that the rewards and performance of third path were close to that of the first path.

To show how the Q-values reflect the network performance, we assume a congestion happens at the link connecting N5-N9 and the delay is recorded to be higher than the same path at an earlier time. The Q-CPN update algorithm reflects that congestion and we see the Q-values of the links connecting the source node to this link all have lower values than before. While link N9-D is not affected as shown in Fig. 5.

For detecting link failure. The neighbor beacon time-out detects the broken links and updates the Q-values accordingly for this neighbor to all destinations to be a very small number (-∞) to make sure it is not chosen again. The algorithm recovers quickly and find new good paths.

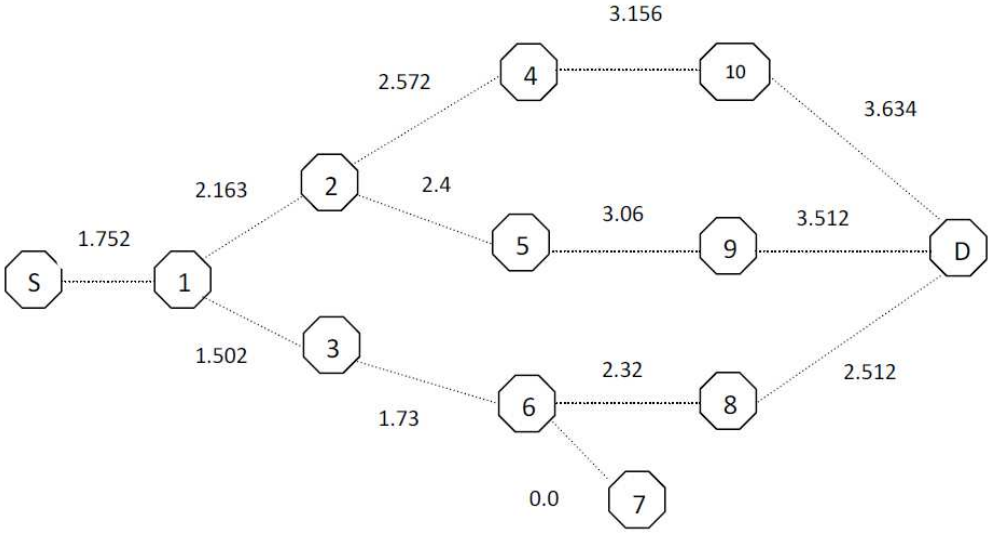

Fig. 4. Sample network with Q-values on links

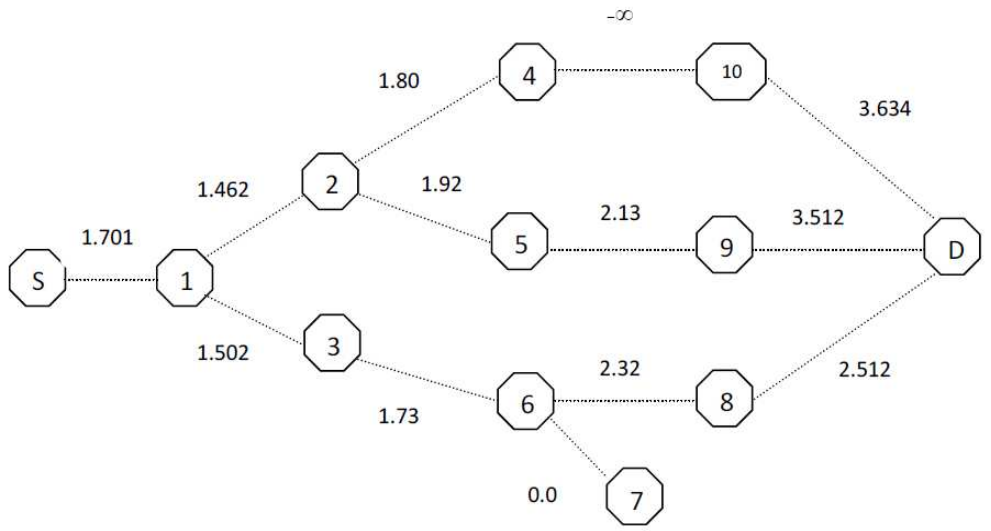

Fig. 5. Network after link congestion and broken link q-values updates

Route Discovery Time

Route Discovery Time (RDT) is the time between the departure of the first SP searching for some route and the arrival of the first ACK to the source node with a route for that destination.

Only a few SP were needed to discover routes at lower SP rates. As SP rate is increased, the results show that higher RDT is recorded since most of SP's collide causing congestion. Then the average RDT is plotted against the number of SP released for route discovery as shown in Fig. 6.

Protocol Simulation and Discussion

To simulate the Q-CPN for MANETs, it was compared to non-adaptive Ad hoc Distance Vector (AODV) (Lent and Zanoozi, 2001) Simulated network is conducted by randomly distributing 35 nodes over 1300×1300 m square area.

The simulation time for each run is 700 sec with the goal function based on delay only. The result data is averaged for each point. The node mobility model used is the Random Way Point (Camp *et al*., 2002) with the node speed 5 meters per second (m/sec) and node pause time varying from 10 to 300 sec. Traffic is set to Constant Bit Rate (CBR) with 1024 byte data packet. The sending packet rate is set to 100 packets/sec. The performance metrics studied are Packet Delivery ratio and Average End-to-End Packet Delay time as shown in Fig. 7 and 8.

DOI: 10.3844/ajassp.2015.1014.1022

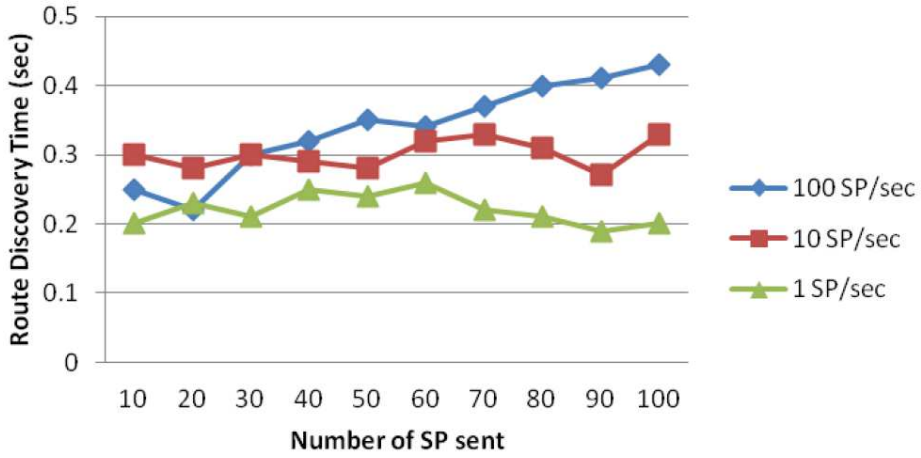

Fig. 6. Route discovery time for different sp rates

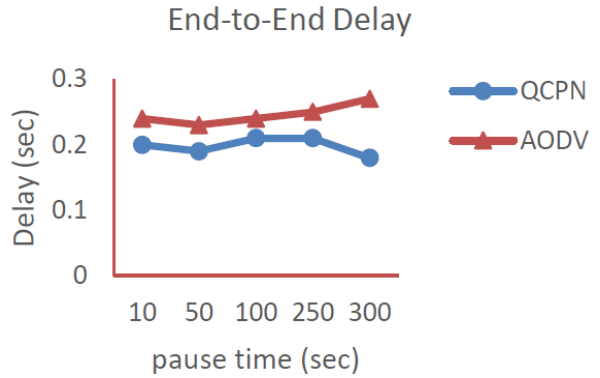

Fig. 7. End-to-end delay against the node pause time

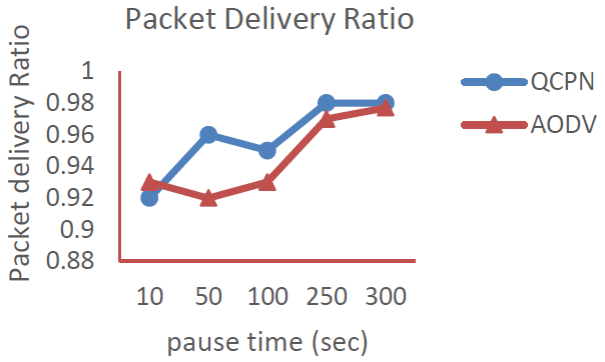

Fig. 8. Ratio of the delivered packets against the node pause time

The algorithm shows better performance than AODV in respect to end-to-end delay and packet delivery ratio due to the learning (update) function which captures path performance and detects any congestions or broken links quickly, changing paths when necessary. While AODV is a shortest path algorithm, focusing on the lowest delay even if that causes heavy use and congestions at some links.

Conclusion

The Cognitive Packet Network is an experimental routing protocol that uses Reinforcement Learning in network routing. This research studied the implementation of Q-routing in CPN routing algorithm to adapt it to the MANET environment. This enables the network nodes to be self-aware, organizing themselves with minimum information exchange and processing complexity. It defines a combination routing goal function relying on the end-to-end delay and the node stability. We show that the application of Q-routing in MANET routing results in low cost solutions without over occupying the node. The main advantage of Q-CPN is robustness to node failure and adaptive behavior to the network dynamics. Thus, self-aware adaptive routing mechanisms are preferred over shortest path algorithms in dynamic self-organized networks such as MANETs.

Acknowledgement

Special thanks to the Network Research Group in the Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia.

Funding Information

The King Saud University Research Unit at the Computer Science Department, College Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia, funded this work.

Author's Contributions

Amal Alharbi: Edited the paper, performed experiments and data analysis.

Abdullah Al-Dhalaan: Supervisor for the work organized the study and performed the final review.

Miznah Al-Rodhaan: Co-supervisor for the work designed the research plan and reviewed the paper.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and no ethical issues involved.

References

Boyan, J. and B. Littman, 1999. Packet routing in dynamically changing networks: A reinforcement learning approach. Adv. Neural Inform. Process. Syst., 12: 893-899.

Camp, T., J. Boleng and V. Davies, 2002. A survey of mobility models for ad hoc network research. Wireless Commun. Mobile Comput., 2: 483-502.

Chang, Y. and T. Ho, 2005. Mobilized ad-hoc networks: A reinforcement learning approach. Proceedings of the 1st International Conference on Autonomic Computing, May 17-18, IEEE Xplore Press, pp: 240-247. DOI: 10.1109/ICAC.2004.1301369

Ertel, W., M. Schneider, R. Cubek and M. Tokic, 2012. The teaching-box: A universal robot learning framework. Proceedings of the International Conference on Advanced Robotics, Jun. 22-26, IEEE Xplore Press, Munich, pp: 1-6.

Forster, A. and A. Murphy, 2007. FROMS: Feedback routing for optimizing multiple sinks in WSN with reinforcement learning. Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, Dec. 3-6, IEEE Xplore Press, Melbourne, Qld, pp: 371-376. DOI: 10.1109/ISSNIP.2007.4496872

Gelenbe, E. and R. Lent, 2004. Power-aware ad hoc cognitive packet networks. Ad Hoc Netw., 2: 205-216. DOI: 10.1016/j.adhoc.2004.03.009

Gelenbe, E., 2014. A software defined self-aware network: The cognitive packet network. Proceedings of the IEEE 3rd Symposium on Network Cloud Computing and Applications, (CCA' 14), Rome, Italy.

Gelenbe, E., P. Liu and J. Lain, 2008. Genetic algorithms for autonomic route discovery. Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, Jun. 15-16, IEEE Xplore Press, Prague, pp: 371-376. DOI: 10.1109/DIS.2006.32

Gelenbe, E., R. Lent and Z. Xu, 2001. Design and performance of cognitive packet network. Performance Evaluat., 46: 155-176. DOI: 10.1016/S0166-5316(01)00042-6

Gelenbe, E., R. Lent and Z. Xu, 2001. Measurement and performance of a cognitive packet network. J. Comput. Netwo., 37: 691-701. DOI: 10.1016/S1389-1286(01)00253-5

Gelenbe, E., R. Lent and A. Nunez, 2004. Self-aware networks and QoS. Proc. IEEE, 92: 1478-1489. DOI: 10.1109/JPROC.2004.832952

Gellman, M. and P. Liu, 2006. Random neural networks for the adaptive control of packet networks. Proceedings of the 16th International on Artificial Neural Networks, Sept. 10-14, Greece, pp: 313-320. DOI: 10.1007/11840817_33

Halici, U., 2000. Reinforcement learning with internal expectation for the random neural network. Eur. J. Operational Res., 126: 288-307. DOI: 10.1016/S0377-2217(99)00479-8

Lent, R. and R. Zanoozi, 2001. Power control in ad hoc cognitive packet networks. Texas Wireless Symposium. Perkins.

Lent, R., 2006. Smart packet-based selection of reliable paths in ad hoc networks. Proceedings of the 5th International Workshop on Design of Reliable Communication Networks, Oct. 16-19, IEEE Xplore Press. DOI: 10.1109/DRCN.2005.1563915

Peshkin, L. and V. Savova, 2002. Reinforcement learning for adaptive routing. Proceedings of the International Joint Conference on Neural Networks, May 12-17, IEEE Xplore Press, Honolulu, HI, pp: 1825-1830. DOI: 10.1109/IJCNN.2002.1007796

Punde, J. and N. Pissinou, 2003. On quality of service routing in ad hoc networks. Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks, Oct. 20-24, IEEE Xplore Press, pp: 276-278. DOI: 10.1109/LCN.2003.1243138

Sachi, M. and A. Prakash, 2013. QoS improvement for MANET using AODV algorithm by implementing q-learning approach. Int. J. Comput. Sci. Technol., 4: 407-409.

Santhi, G., A. Nachiappan, M. Ibrahime and R. Raghunadhane, 2011. Q-learning based adaptive QoS routing protocol for MANETS. Proceedings of the International Conference on Recent Trends in Information Technology, Jun. 3-5, IEEE Xplore Press, Chennai, Tamil Nadu, pp: 1233-1238. DOI: 10.1109/ICRTIT.2011.5972411

Schaul, T., J. Bayer, D. Wierstra, T. Sun and M. Felder *et al*., 2010. PyBrain. J. Machine Learn. Res., 11: 743-746.

Sutton, R.S. and A.G. Barto, 1998. Reinforcement Learning: An Introduction. 1st Edn., MIT Press, Cambridge, ISBN-10: 0262193981, pp: 322.

Tanner, B. and A. White, 2009. RL-Glue: Language-independant software for reinforcement learning experiments. J. Machine Learn. Res., 10: 2133-2136.

Tao, T., S. Tagashira and S. Fujita, 2005. LQ-routing protocol for mobile ad-hoc networks. Proceedings of the 4th Annual ACIS International Conference on Computer and Information Science, Jul. 14-16, IEEE Xplore Press, pp: 441-446. DOI: 10.1109/ICIS.2005.80

Toh, C.K. and V. Vassiliou, 2000. The effects of beaconing on the battery life of ad hoc mobile computers.

Toh, C.K., 1997. Associativity-based routing for ad hoc mobile networks. Wireless Personal Commun. Int. J., 4: 103-139. DOI: 10.1023/A:1008812928561

Watkins, C., 1989. Learning from delayed rewards. PhD Thesis, Cambridge University, London, England.