Q-Routing in Cognitive Packet Network Routing Protocol for MANETs

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

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

Keywords: Cognitive Packet Network (CPN), Mobile Ad Hoc Network (MANET), Q-Routing, Reinforcement Learning, Stability-based Routing.

Abstract: Mobile Ad hoc Networks (MANET) are self-organized networks which 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 random changing network topology. This research uses Reinforcement Learning approach with Q-Routing to introduce our MANET routing algorithm: Stability-Aware Cognitive Packet Network (CPN). This new algorithm extends the work on CPN to adapt it to the MANET environment with focus on path stability metric. 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 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 Q-routing to make routing decisions which adapt on line to network changes allowing nodes to learn efficient routing policies.

1 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 (Perkins, 2001). Therefore, to successfully communicate, nodes need an adaptive distributed routing protocol that adjusts when the network changes. 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 which 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 (decision maker) 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.

MANETs are self-organized networks with no fixed infrastructure. There has been many proposed routing algorithms for MANETs as shown in (Perkins, 2001). Designing MANET protocols faces major challenges due to special characteristics of this type of network. In this paper, we present an adaptive smart routing protocol for MANETs based on path stability evaluation. This routing algorithm extends the work on CPN with adjustments to suit the characteristics of a MANET. Our Stability-Aware CPN routing algorithm introduces an explicit neighbour discovery scheme and adjust the route maintenance scheme to result in long-lived routes with acceptable delay. Nodes in our routing algorithm first learn the network state, then make routing decision using Reinforcement Learning with Q-Routing.

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 presents the problem solution. Finally, section 5 shows the results analysis and section 6 is the conclusion.

Alharbi A., Al-Dhalaan A. and Al-Rodhaan M..
Q-Routing in Cognitive Packet Network Routing Protocol for MANETs.
DOI: 10.5220/0005082902340243
In *Proceedings of the International Conference on Neural Computation Theory and Applications* (NCTA-2014), pages 234-243
ISBN: 978-989-758-054-3

2 RESEARCH PROBLEM

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. Once the first ACK reaches the source node, the discovered path is stored in a Route Cache. Then, the algorithm reduces the rate at which the SP's are sent. The SPs that are sent after a route is discovered are to maintain and improve QoS delivered.

When a SP arrives at the destination node, an ACK packet is created and sent to the source node. The ACK uses the reverse route of the SP. The ACK passes every node on the discovered route and updates the weights in Random Neural Network (RNN) (Gelenbe,1993) according to the route performance.

When a SP arrives at an intermediate node, the node decides which neighbour to forward the SP. This decision is made using the RL/RNN algorithm.

As soon as the first ACK reaches 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. Each discovered route is evaluated according to the reward function defined in the RL decision algorithm. According to the route performance, the weights in RNN are updated. The subsequent SP's visiting the same node for the same destination and QoS will learn the efficient routing path depending on these weight updates.

In order to adapt CPN routing algorithm to the MANET environment, the major network characteristics should be handled. A MANET has a highly unstable topology. Node mobility is one of the significant factors effecting a routing protocol performance. Furthermore, nodes in ad hoc networks suffer from resource limitations. Our proposed algorithm adjusts the CPN routing algorithm to be suitable for this environment. It focuses on route stability to handle node mobility. It also combines Q-routing with CPN.

The Stability-Aware CPN routing protocol for MANETs is an adaptive protocol with self-improvement capabilities. It offers the network the ability to determine the QoS criteria according to the data being transferred in a distributed way. Each node in the network runs the protocol using RL with Q-routing . Network information is collected only for paths being used; there is no global network information exchange.

The research routing algorithm's main objective is to implement Q-routing in CPN focusing on path stability. The path stability is based on the *Associativity* property of the mobile nodes in time and space (Toh, 2004). Mobile nodes that show high association stability are chosen for routing as much as possible. The algorithm defines the reward function as a weighted combination of high stability and minimum delay.

Furthermore, the study aims to improve the protocol performance robustness to handle the network's dynamic topology problem. This is achieved by adjusting the routing protocol maintenance process as shown in section 4.2.

3 BACKGROUND

The Stability-Aware CPN Routing protocol for MANET is a unique research protocol, which introduces node stability over space and time into the CPN routing protocol. The first subsection reviews the CPN, while the second subsection reviews the stability-aware protocols. The last subsection shows the Q-routing based protocols.

3.1 Cognitive Packet Network

CPN was first introduced to create robust routing for the wired networks in (Gelenbe, 2001). It has been tested and evaluated in later studies (Gelenbe and Lent, 2001) to be adaptive to network changes and congestions. A number of learning algorithms have been researched before using Reinforcement Learning based on Random Neural Networks (Gellman, 2006). Genetic Algorithms have been used in CPN to modify and enhance paths (Gelenbe, 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.

There have been many studies evaluating the CPN performance. One research studied CPN in the presence of network worms (Skellari, 2008). It concluded in a better failure-aware CPN. It achieved that by introducing a detection mechanism which stores timestamps of the last SP and ACK that pass through the link. If no ACK was received after a SP has passed in some determined time, the link is considered under failure. However, there should be an appropriate estimate for the average delay under normal conditions for each link to be useful (Skellari, 2009).

One extension of CPN is Ad hoc CPN (AHCPN) (Gelebe 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 modeled 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). Thus SP's selected nodes that have the longest remaining battery with greater probability. The QoS Goal function was a combined function of two goals: maximum battery lifetime and minimum path delay. The result was good performance (short delay and energy-efficient), but there was a high number of lost packets which meant that the algorithm did not adapt to network changes quickly enough. Also, node availability in real systems is determined by many factors such as process load and work environment and not just battery lifetime.

The AHCPN was later adjusted as in (Lent and Zanoozi, 2005) which proposed a solution to control both energy consumption in nodes and mutual interference of neighbouring communications. The paper suggests an adjusted transmission power level when transmitting DP's and ACK's to save energy and reduce interference. Only SP's were transmitted using full power. The result was that nodes have more energy to participate in routing. Nodes with more energy were chosen in paths with higher probability.

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 wss 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.

3.2 Stability based Protocols

Stability-Aware routing algorithms aim to find the longest-lived routes. However, there are many approaches to study path stability.

In (Dube, 1997) the authors studied radio propagation effect on the signal effect strength. The study regard the link stability as the probability of the received signal strength higher than a predetermined threshold. They believed that the path bottleneck was the least stable link within it. In (Trivino, 2006), the Ad hoc On-demand Stability Vector routing protocol was proposed. It discovered routes and maintained them in relation to radio channels. In (Targen, 2007), the study introduced a new link stability classification. The study considered the links between low mobility nodes to be stationary links. On the other hand, transient links exist only for some short time and are more likely to cause link breakage. The authors introduced a routing protocol that used stationary links as much as possible to give more stable routes.

The study in (Toh, 2000) introduced the *Associativity* metric for determining node stability through time and space within the neighbourhood. Associative Based Routing (ABR) defined Associativity to determine a link's connection stability and thus path stability. The Associativity of a node with a certain neighbour is the degree of association over time and space. The Associativity property assumes that a mobile node goes through a stage of instability with high mobility followed by a stage of stability when it is dormant (connected to the same neighbours for some time) before the mobile node moves out of proximity (Manikandan, 2000). The dormant stage is the best time for a node to participate in routing, which is determined by a high Associativity level. Each node sends out periodic beacons with its identity and battery lifetime to signify its existence. If the number of beacons (i.e. *Associativity Ticks*) received from a certain neighbour is more than a specified threshold, the link to that neighbour is considered stable. The Associativity Threshold (Toh, 1997) is a function of the beaconing interval, the relative velocity between the two nodes, and the transmission range of a node. Associativity ticks are reset when either the mobile node itself or the neighbour move out of transmission range (Murad, 2007).

Signal Stability based Adaptive (SSA) routing protocol depends on signal strength and location stability (Sildhar, 2005). However, simulations show that it does not perform better than a simple shortest path algorithm.

There is also some significant research about link and path duration to propose that the residual lifetime of a link determines the expected path duration (Han, 2006). This kind of work studies the distribution of link lifetimes in a network. Each time a link breaks the average lifetime of the link is updated for future use in path duration estimation. However, the results of the study are closely related to the mobility model assumed. It also assumes that all nodes have the same movement pattern. It also depends on information gathered over a long time in order to reasonably estimate path duration and make routing decisions accordingly.

In a different approach to deal with the uncertainty in MANETs, many researchers considered probabilistic methods to estimate node stability and path lifetimes. In (Tseng, 2003), the authors attempted to predict the route lifetimes in different mobility patterns. Sophisticated probability methods were used to determine the expected value of each link lifetime in a route. Their analysis revealed an expected outcome that predicting route lifetimes highly depend on the node mobility pattern assumed. The author of (Camp, 2002) proposed a formal model to predict the lifetime of a route based on Random Walk model. The research defined a probability Existence function to compute a route lifetime, which allowed to estimate residual lifetime of a route to use in routing decisions for stable routes. A complete probabilistic analysis in MANETs using Random Way point mobility model was conducted in (Chung, 2004). The result showed that the exponential distribution was a good function to predict route behaviour for stability evaluation.

3.3 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 was based on the RL scheme called Q-Routing. The results reveal that adaptive Q-Routing performed better than shortest path algorithm in static networks under changing network load and connectivity. In another study, the authors used RL to perform a policy search to optimize routing decisions that resulted in multiple source-destination paths to deal with high network load (Brown, 1999). Authors in (Kumar and Miikulainen, 1999) developed a confidence–based Q-routing algorithm with dual RL. Their objective was mainly to increase quantity and quality of exploration in Q-routing. In (Chang et al, 2004), 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. In (Tao et al, 2005) the authors combine Q-routing with Destination Sequence Distance Vector routing protocol for mobile networks. To deal with mobility, the concept of path lifetime was introduced to reflect path stability. In (Forster, 2007), the author used RL to propose a solution to multiple destination communication in WSN using Q-Routing.

Other authors propose a routing model that is fit for ad hoc networks using different RL algorithms. One paper used the Prioritized Sweeping RL model technique to propose a Collaborative Reinforcement Learning (CRL) routing algorithm for MANETs called SAMPLE (Kulkani and Rao, 2010). This routing protocol converged fast to routing solutions providing QoS. It was based on a reward function that approximates the number of transmissions needed to transmit a packet. There was some extended research on SAMPLE to optimize it in relation to packet delivery, energy-consumption, QoS, and scalability.

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.

The author in (Wang, 2012) proposed a self-learning routing protocol based on Q-learning that uses Signal to Interference plus Noise Ratio (SINR), delay, and throughput to deliver the desired QoS. The algorithm uses a Bayesian Network to estimate congestion levels at neighbouring nodes. Their results showed that their algorithm improves performance in dense, high load networks.

In (Sachi and Parkash, 2013), introduce a MANET routing algorithm by combining Q-learning with Ad hoc Distance Vector (AODV) to achieve higher reliability. It considers QoS parameters such as traffic, channel capacity, energy, bandwidth, and packet loss ratio.

4 STABILITY-AWARE CPN

Stability-Aware CPN routing algorithm for MANETS defines the routing process as a Reinforcement Learning problem. This allows learning the network topology in short time without periodic advertisement of network information or global routing information exchange as in the classic algorithms.

4.1 Path Stability

A MANET is a type of network that is self-configured with no infrastructure. Nodes are free to move and join arbitrarily. However, there are some patterns that the nodes follow which allows routing protocols to select the best nodes for the routing process. One such node property is the Node Associativity (Toh, 2004) with its neighbours over time and space. It reveals the connection stability of nodes in MANETs. Each mobile host periodically sends a beacon to each of its neighbours every beacon interval time (p). Each time a mobile host receives a beacon from a certain neighbour, the number of Associativity Ticks (count) in relation with this neighbour is increased by one unit. A mobile node in an Ad hoc environment usually goes through an initial stage of migrating with a certain velocity (v). Then the node spends some pause time dormant within its neighbours as shown in Figure. 1.

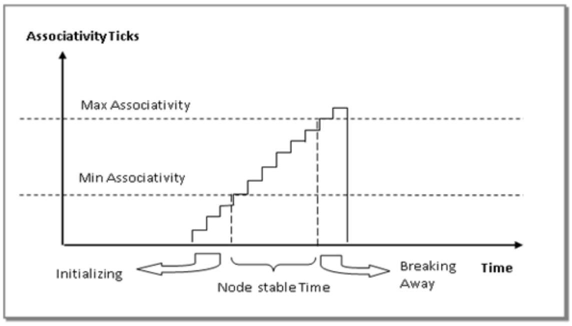

Figure 1: Determining Node Stable time using Associativity.

For a path to be stable, the conducting nodes should be stable as well as their links as much as possible. Clearly, the node mobility model effects the routing path stability. In this section we focus on stability of the multi-hop routing path, which is determined here by the level of connection-stability (Associativity) of the conducting nodes. We define the Path-Stability-Ratio under the Random Way Point mobility model as the percentage of stable nodes along the routing path. It is calculated at the Destination Node using the equation 1.

$$Path-Stability-Ratio = \frac{Number\ of\ stable\ nodes\ in\ path}{Total\ Number\ of\ nodes\ in\ path} \quad (1)$$

4.2 Q-Routing in CPN

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 neighbours to whom it can forwards the packet to.

In order for the agent to learn a model of the system (network), Q-values (Peshkin and Savova, 2002) are used. A Q-value is defined as Q(state1,action1) and has a value which represents the expected rewards of taking action1 from state1. Each state is a destination node d in the network. Thus after some time steps (stages of RL), the Q-values should represent the network accurately. This means that at each state, the highest Q-value is for actions (neighbours) that represent the best choices (Brown, 1999). 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 $Q^x(s,a)$. The structure of the Q-Table is shown in Figure 2.

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

Figure 2: The Q-Table for node x.

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 also has to be with minimum processing overhead. In our algorithm, update of Q-values occurs whenever a node has to make a routing decision to forward a SP depending on the current reward calculated at each neighbor and the old Q-value estimate. Rewards are calculated depending on information gathered at each node from Acknowledgments carrying performance measures (delay and path-stability-ratio).

$$Q_t^x(d1,n1) = \alpha Q_{t-1}^x(d1,n1) + (1-\alpha)R_t \qquad (2)$$

The update is performed according to equation 2. Where $Q_t^x(d1,n1)$ is the new estimate and $Q_{t-1}^x(d1,n1)$ is the old estimate , and R_t is the current reward. Also α is a learning constant typically close to 1, 0< α < 1 . Here in this algorithm α= 0.8 , which means that delayed reward in the future are more important than immediate rewards.

4.3 Reward Function

The Goal Function of the routing process is a common goal for all agents (nodes) in the network. The goal is to minimize a weighted combination of the delay and the inverse of the path-stability-ratio. This goal function is expressed mathematically in equation 3.

The goal is calculated at every node where path-stability-ratio is the total stability ratio of the path from this node to the destination node. Also the delay is the total delay from this node to the destination node.

$$G = \frac{1}{Path - stability - Ratio} + Delay \qquad (3)$$

The reward depends on this goal function. It is defined as the inverse of the routing goal. As shown in Equation 4. The Reinforcement Learning algorithm aims to select optimal actions (neighbours) 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 path starting from the node x.

$$R = 1/G \qquad (4)$$

4.4 Routing Protocol Processes

The research routing protocol is described in this section through explaining the major protocol processes and its effect on the network performance.

4.4.1 Neighbor Discovery

MANETs are considered self-configured networks with no communication infrastructure. Thus, neighbour discovery is an essential part of initialization of a MANET (Perkins, 2001). A node has to be able to know at least the one-hop neighbours to communicate with any other node in the network. In Stability-Aware 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.

The effect of the beaconing period on the node's power consumption is critical. A very small beacon period (i.e. 10ms) dissipates the node's power rapidly. It also condenses the network with too many control packets without any significant advantage (Toh, 2000). Each node keeps a neighbour table to hold neighbour information needed to update Q-values. A neighbouring node increments it’s Associativity Ticks for a neighbour each time it receives a beacon from that specific neighbour. Associativity Ticks are initialized when the neighbour moves away of radio range. A neighbour is considered moved away, when a node x does not receive any beacons from this neighbour for three times the beacon period. Inactive nodes and nodes that are low in battery become passive and refrain from sending beacons.

4.4.2 Route Discovery

In the Stability-Aware CPN, the route discovery process is triggered when a source node needs to send data packets to an unknown destination node. The source node first checks its route cache for a known route to that destination. If there is no route in the cache, the source node creates and send SP's with a smart packet ratio 2% of the total data packets to be sent. SP's search the network gathering information from each node visited to find good routes to the target destination. At the beginning nodes do not have a complete picture of the whole ad hoc network. However, with time and learning nodes start to have a good picture about the state of network. Thus later SP's learn from previous SP's of the same QoS and same destination. Nodes start to take good routing decisions using RL with Q-Routing to select next hops for SP's wisely. The decision algorithm uses network information stored in the Neighbour Tables, Mailboxes, and Q-Tables. When SP finally reaches its destination, an ACK is sent to the source node. The source node uses information in the ACK to store a new route entry in the Route Cache.

4.4.3 Gathering Network Information

The SP's collect specific network information as they move around and store it in a distributed fashion as follows. Route Caches located only at source nodes, stores complete path for all active destinations. Cognitive Maps (CM) exist in all type packets to store addresses and network metrics (Battery, Arrival time, and Associativity) of visited nodes. ACK's distribute this information to update mailboxes along the path. Packets store complete route in their CM. Mailboxes are located at every node, they keep statistics about performance of active paths such as average delay, degree of associativity. This information is used by RL algorithm to decide on next hop. Q-Tables are located at nodes for the RL. Q-Tables are updated by the RL algorithm. Neighbour Tables are created and maintained at every node to keep information about neighbours and their associativity degree and forwarding delay.

4.4.4 Route Maintenance

The Route Maintenance process is of great importance in the operations of a routing protocol. This process if performed efficiently, decreases the packet loss ratio and the total packet delay. However, it should introduce minimum control overhead. In Stability-Aware CPN route maintenance of active routes is achieved by sending a small fraction of SP's (1%) to search for alternate routes. Only active routes are maintained. Once a better route is discovered, the old one is considered invalid. An intermediate node can detect link breakage while forwarding a Data Packet, and thus sends a Route-Error packet to the source node. When a source node receives a Route-Error packet, it stops using this route and sets the Rout-invalid flag in the routes entry in the Route Cache. If the source node still has some packets to send to that destination, it checks its Route Cache for an alternate route and use it. If the Route Cache has no alternate routes, then the source node issues a new Route Discovery process.

4.4.5 Routing Policy

The basic Q-routing algorithm shown in figure 3 is a greedy algorithm. It acquires an estimate only from the best neighbour. Each time a node sends a packet, it updates only the Q-value corresponding to the best neighbour. Thus, it is possible to return sub-optimal policy by not exploring other non-maximum neighbours. Other versions have tried to overcome this drawback in different ways as in (Chetret, 2009).

1- Set initial Q-values for each node.
2- Get first packet from packet queue of node x.
3- Choose the best neighbor node y and forward the packet to y.
4- Get estimated value from y, which is y's best time estimate for packet delivery.
5- Node x updates its Q-value for the best neighbor using equation

$$Q_x(d,y)_{new} = Q_x(d,y)_{old} + \eta[y\ new\ estimate - Q_x(d,y)_{old}]$$

6- Go to step 2.

Figure 3: Basic Q-Routing.

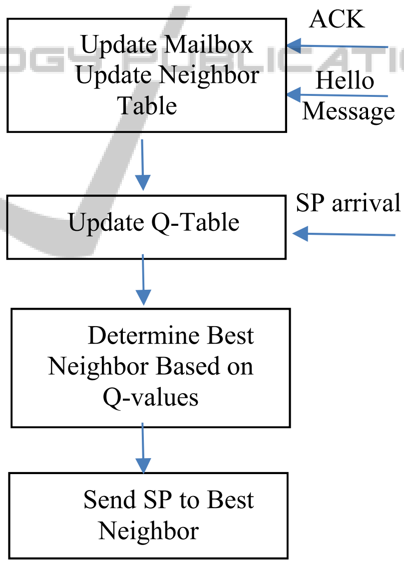

Figure 4: Routing Policy at each node.

Our algorithm steps are shown in Figure 4. Feedback from ACK updates the tables in each node it visits, including the delay and Associativity degree of the nodes on the path. Hello messages update neighbour tables with immediate neighbour information. Nodes do not communicate their estimates. Instead, each node has a mailbox where information is updated continuously with each ACK arriving for SP or DP. When a node has to take a decision to send a packet: 1- the node checks it tables. 2- The node calculates the expected reward from each neighbour and updates corresponding Q-values. 3- The node selects the highest Q-value neighbour and forwards the packet to it.

5 RESULT ANALYSIS

The numerical experiments for studying the use of Q-routing in CPN using Acknowledgement feedback were analysed using MatLab. Two experimental small ad hoc networks of 4 and 12 nodes were used to evaluate algorithm convergence behaviour.

5.1 Four Node Network

The network with four nodes is shown in figure 5. The source node is node 0 and destination node is node 3. The optimal path is through nodes 0, 1, and 3 that give maximum rewards. The algorithm shows fast convergence and give corresponding Q-tables as shown in Tables 1, 2, 3 for each node 0, 1, and 2. Q-values are continuously updated through feedback about the reward the path delivered carried by ACK. After some finite number of episodes, Q-values no longer change and learning converges. Thus the Q-values show more realistic values about the neighbours connecting each node.

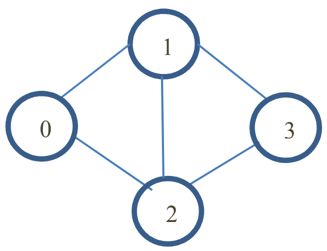

Figure 5: Four Node Network.

Table 1: Q-Table for node 0 (source).

	Node 1	Node 2	Neighbor3
Destination3	1.2	0.3	-----

Table 2: Q-Table for node 1.

	Node 0	Node 2	Neighbor3
Destination3	------	0.2	1.3

Table 3: Q-Table for node 2.

	Node 0	Node 1	Node 3
Destination3	-------	0,3	0.5

5.2 Protocol Simulation

To simulate the stability-Aware CPN for MANETs, it was compared to non-adaptive Ad hoc Distance Vector (AODV) (Perkins,2001) using OPNET 14.5 modeler. Simulated network is conducted by randomly distributing 50 nodes over 1300m x 1300m square area. The simulation time for each run is 300 seconds. The result data is averaged for each point. The node mobility model used is the Random Way Point with the node speed 25 meter per second (m/sec) and node pause time varying from 10 to 300 seconds. Traffic is set to Constant Bit Rate (CBR) with 1024 byte data packet. The sending packet rate is set to 5 packets per second. The performance metric studied are Packet Delivery ratio and Average End-to-End Packet Delay time as shown in Figure 6 and 7.

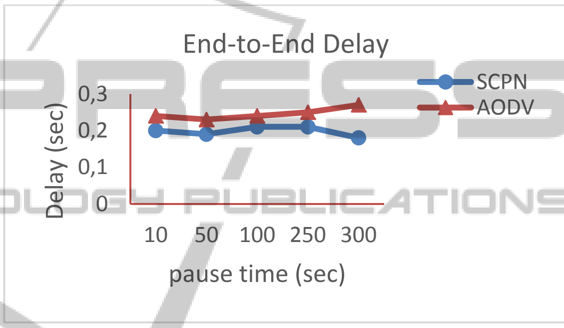

Figure 6: End-to-end delay in mobility simulation.

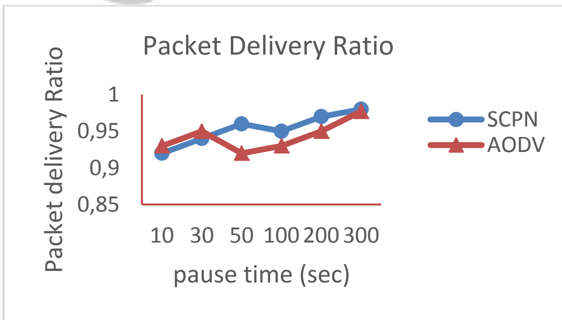

Figure 7: Packet Delivery Ratio in mobility simulation.

5.3 Computational Time

Software computational time of the decision algorithm is considered an important characteristic to evaluate. The study in (Brown, 1999) reveals that the original decision algorithm in CPN with RNN/RL is an $O(n^2)$ every time a decision is taken, plus additional $(2n)$ operations for normalizing matrices weights for every weight update process. This is undesirable for a mobile ad hoc node because of the nodes limitations in processing power and memory. The original algorithm also defines and maintains two weight matrices for each QoS-

Destination pair at each node. It also stores Excitation probabilities for every neighbour of each node in the network.

On the other hand, our decision algorithm for Stability-Aware CPN is based on RL/Q-Routing. It is considered an $O(n)$ where n is the number of neighbors a node on the average has. Our algorithm avoids using weight matrices and weight normalizing overhead. This reduces processing complexity and memory storage needs. The decision algorithm stores averaged QoS data in Q-Tables with one Q-value for each neighbour. Also our Decision algorithm defines for each node one Q-Table that represents the whole network state.

6 CONCLUSIONS

The Cognitive Packet Network is an experimental routing protocol that uses Computational Intelligence in network routing. This research studied the implementation of path stability in the goal function of the CPN routing algorithm to adapt it to the MANET environment. Most of the implementation of Q-routing in MANETs were enhancement of Ad hoc Distance Vector (AODV). Our algorithm combined Cognitive Packet Network routing protocol with Q-routing with some adjustments to accommodate the MANET environment. Stability-Aware CPN routing algorithm for MANETs gives comparable results to conventional MANET routing protocols without disrupting the overall end-to-end delay. Some further studies should be conducted to evaluate the routing protocol in terms of the amount of packet control overhead.

REFERENCES

Rahul D., Patil B. (2013) ‘MANET with Q-Routing Protocol’, *Medwell Journal of Information Technology,* vol. 12. Issue 1, pp.20-26.

Sachi, M., Prakash, C.(2013) ‘QoS Improvement for MANET Using AODV Algorithm by Implementing Q-Learning Approach’, *International Journal of Computer Science and Technology IJCST* vol.4 , issue 1, pp. 407-409.

Wang K. (2012) ‘A MANET Routing Protocol Using Q-Learning Method Integrated with Bayesian Network ’, *Proceedings of International Conference on Communication Systems (ICCS),* Singapore, 2012, IEEE, pp. 270-274.

Santhi, G., Nachiappan, A. Ibrahime, M.,Raghunadhane, R. (2011) ‘Q-Learning Based Adaptive QoS Routing Protocol for MANNETS’ , *Proceedings of the International Conference on Recent Trends in Information Technology*. Chennai: IEEE. Pp. 1233-1238.

Kulkani, S., Rao, R. (2010) ‘Performance Optimization of Reinfrocement Learning Based Routing Algorithm Applied to Ad hoc Networks’, *International Journal of Computer Networks and Communications*, vol. 2, pp. 46-60.

Chetret, D. L., (2009) ‘Adaptive Routing in MANETS using RL and CMAC’. Master Thesis submitted to Department of Electrical and Computer Engineering, National University of Singapore.

Skellari, G. (2009), ‘Maintaining Quality of Service and Reliability in Self-Aware Networks”, PhD Thesis for Imperial College of London.

Gelenbe, E., Liu P., and Lain, J. (2008) ‘Genetic Algorithms for Autonomic Route Discovery’, *in Proceedings of IEEE Conference on Distributed Intelligent Systems,* Washington, USA.

Skellari, G., Hey L., and Gelenbe, E. (2008), ‘Adaptability and Failure Resilience of the Cognitive Packet Network’, *27th IEEE Conference on Computer Communications,* Phoenix, Arizona, USA.

Mehandthan, N. (2008) “Exploring the Stability-Energy Consumption-Delay Network Lifetime Trade-off of MANET Routing Protocol”, *Journal of Networks* , vol.3, No.2.

Targen, J., Chuang, B. (2007) ‘A Novel Stability-Based Routing for Mobile Ad hoc Networks”, *IEICE Transactions on Communications*, B(4) : 876-884.

Murad, A., Al-Mahadeen, B. (2007) ‘Adding Quality of Service Extensions to Enhanced ABR Routing protocol’, *American Journal of Applied Science.*

Forster, A., Murphy, A. (2007) ‘Feedback Routing for Optimized Multiple Sinks in WSN with Reinforcement Learning”, Proc. 3rd International Conference of Intelligent Sensors, Sensor Networks, Information Process (ISSNIP).

Han, Y., (2006) ‘Maximizing Path Duration in MANETS’, Maryland, USA.

Trivino-Cabrera A., Nieves-Perez, I. (2006) ‘Ad hoc Routing based on Stability of Routes’, *Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access*, ACM, New York, USA. Pp. 100-103.

Gellman, M., Liu, P. (2006), ‘Random Neural for Adaptive Control of Packet Networks’, *Proceedings of 16th International on Artificial Neural Networks*, Greece, pp.313-320.

Lent, R. (2006) ‘Smart Packet-Based Selection of Reliable Paths in Ad hoc Networks’, *IEEE.*

Toa, T. Tagashira, S. Fujita, S. (2005) ‘LQ-Routing Protocol for MANETs’, *In proc. Of Fourth Annual ACIS International Conference on Computer and Information Science.*

Liu, C., Kaiser J. (2005), ‘A Survey of Mobile Ad hoc Networks Routing Protocols’. University of Ulm, Germany.

Lent, R., Zanoozi, R. (2005) 'Power Control in Ad hoc Cognitive Packet Network', *Texas Wireless Symposium*.

Chang, Y. , Ho, T. (2004) 'Mobilized Ad hoc Networks: A Reinforcement Learning Approach.' Proc. Of First International Conference on Autonomic Computing IEEE Computer Society USA, pp. 240-247.

Chung, W. (2004) 'Probabilistic Analysis of Routing on Mobile Ad hoc Networks', *IEEE Communication Letters* 8(8): 506508.

Toh, C. K. , Lee , E. C. , and Ramos N. (2004), 'Next Generation Tactical Ad hoc Mobile Wireless Network' ,*Technology Review Journal.*

Gelenbe E., Lent R., and A. Nunez, (2004), 'Self-Aware Networks and Quality of Service', *Proceedings of IEEE* vol. 92(9): 1478-1489.

Gelenbe, E., Lent, R. (2004), 'Power-Aware Ad hoc Cognitive Packet Network', *Elsevier.*

Punde, J., Pissinou, N. (2003) 'On Quality of Service Routing in Ad hoc Networks", *Proceedings of the 28th IEEE International Conference on Local Computer Networks,* IEEE Computer Society, Washington, USA.pp.276-278.

Tseng, Y., Chang, C. (2003) 'On Route Lifetime in Multi-hop Mobile Ad hoc Networks', *IEEE Transactions on Mobile Computin*g 2(4): 366-376.

Camp, T., Davies, J. (2002) 'Survey of Mobility Models for Ad hoc Networks Research", *Wireless Communication & Mobile Computing (WCMC):* Special issue on MANET: Research Trend & Application 2 (5): 483-502.

Peshkin, L., Savova, V. (2002) 'Reinforcement Learning for Adaptive Routing', IEEE.

Gelenbe, E., Lent R., and Xu Z.,(2001), 'Design and Performance of Cognitive Packet Network', *Performance Evaluation*, 46,pp.155-176.

Halici, U. (2000), 'Reinforcement Learning with Internal Expectation for The Random Neural Network', *Eur.J.Opns.Res. 126(2)2,* pp.288-307.

Perkins, C. E. (2001), *Ad Hoc Networking*, Addison-Wesley.

Gelenbe, E., Lent, R. and Xu Z.,(2001) 'Measurement and Performance of Cognitive Packet Network', *Journal of Computer Networks*, 46,pp.155-176.

Toh, C. K., Vassiliou, V. (2000) 'The effects of Beaconing on the Battery life of Ad hoc Mobile Computers'.

Manikandan, S., Naveen, L. and Padmanban, R., (2000) 'Optmized Associativity-Based Threshold Routing for Mobile Ad hoc Netwroks', 2000.

Boyan, J., Littman, B. (1999) 'Packet Routing in Dynamic Changing Networks: A Reinforcement Learning Approach', *In Advances in Neural Information Processing Systems*, vol.12, pp. 893-899.

Brown, T. X, (1999) 'Low Power Wireless Communication via Reinforcement Learning', *In Advances in Neural Information Processing Systems*.

Sutton, R., Barto, S. (1998) *Reinforcement Learning: an Introduction,* MIT Press: Cambridge.

Toh, C.K. (1997) 'Associativity Based Routing for Mobile Ad hoc Networks', *IEEE Conference on Computer & Communications (IPCCC),* Phoenix, USA.

Dube, R., Rais, C. R. and Wang, K.Y. (1997) 'Signal Stability Based Adaptive Routing SSA for Ad hoc Mobile Networks', *Personal Communication, IEEE*, 4(1): 36-45.

Gelenbe, E. (1993), 'Learning in the Recurrent Random Neural Network'. *Neural Computing*, 5(1), pp. 154–164. 1993.