Reinforcement learning based reliable routing in Wireless Sensor Network

Sudat Tuladhar
Department of Electrical Engineering

***Abstract*—This report provides a discussion on using reinforcement learning (Q-learning) on a Wireless Sensor Network (WSN) routing, based on link reliability. The paper presents routing results on static, sparse random and dense random scenario.**

***Index Terms*—Wireless Sensor Network, routing, reinforcement learning**

I. Introduction

A. Wireless Sensor Network routing

The characteristics of a wireless sensor network (WSN) generally necessitates deployment of a number of sensor nodes to monitor an object, environment or event. Identification of efficient routing paths can extend the life of individual nodes in WSN by putting the nodes in sleep mode when not in use and activating only those in routing paths. Apart from minimizing energy consumption, design of sensor network protocols and algorithms also demand requirements such as fault tolerance and reliability.

B. Literature Review

The routing problems of WSN have been addressed in a large number of works. On reviewing the work, various features such as Energy, security, delay and error that pose challenges are identified [1]. Energy efficient protocols [2] have developed a multi-rate routing scheme to optimize routing in distributed source coding. Delay-less protocols [3] have considered both the operations of the underlying directional MAC protocols and the physical interference to develop a color conflict graph abstraction. Secure protocols [4] have introduced a novel three phase disjoint routing scheme, called the Security and Energy-efficient Disjoint route, to maintain network security. Reliable protocols [5] have used packet delivery rate as the metric measure and proposed a model to detect the efficiency of multi-hop radio networks. The work in this report focuses on solving the reliable routing in WSN using a reinforcement learning technique: Q-learning.

C. Q-Learning

Q-learning is a model-free reinforcement learning technique to learn a policy by an agent to take suitable action under different circumstance. It does not require a model of the environment, and can handle problems with stochastic transitions and rewards, without requiring adaptations. It differs from supervised learning in that labelled input/output pairs need not be presented. Instead the focus is finding a balance between exploration (or uncharted territory) and exploitation (of current knowledge). For Markov decision process, Q-learning finds a policy that is optimal in the sense that it maximizes the expected value of the total reward over any and all successive steps, starting from the current state.

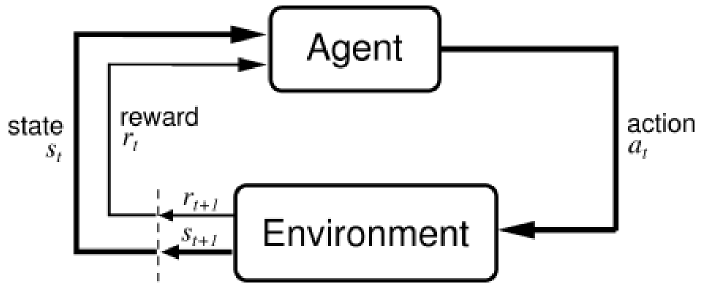

Fig. 1. Q-learning: Agent and Environment

As shown in Fig. 1, reinforcement learning involves an agent, a set of states *S*, and a set of actions *A* in every state. By performing an action a $\in$ *A*, the agent transitions from current state to next state. Executing an action in a specific state provides the agent with a reward *r* (a numerical score). The goal of the agent is to maximize its total (future) reward. It does this by adding the maximum reward attainable from future states to the reward for achieving its current state, effectively influencing the current action by the potential future reward. This potential reward is a weighted sum of the expected values of the rewards of all future steps starting from the current state as in (1).

$$Q^{\pi}(s,a) = \mathbf{E}[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots] \tag{1}$$

where γ is a discount factor that discounts the future reward.

Before learning begins, Q is initialized to a possibly arbitrary fixed value. During training (exploration), at each time step *t* the agent selects an action a_t, observes a reward r_t, enters a new state s_{t+1}, and Q is updated as in (2). The core of the algorithm is a single value iteration update (2), using the weighted average of the old value and the new information. $\alpha \in (0,1)$ is the learning rate. $\gamma \in (0,1)$ is a discount factor and has the effect of valuing rewards received earlier higher than those received later.

$$Q^{new}(s_t,a_t) \leftarrow (1-\alpha)Q(s_t,a_t) + \alpha(r_t + \gamma \underset{a\in A}{max} Q(s_{t+1},a)) \tag{2}$$

The Q-Learning algorithm for training (exploration) phase is summarized in Fig. 2.

```
Initialize $Q(s,a)$ arbitrarily
Repeat (for each episode):
    Initialize $s$
    Repeat (for each step of episode):
        Choose $a$ from $s$ using policy derived from $Q$
        Take action $a$, observe $r$, $s'$
        Update
            $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
        $s \leftarrow s'$;
    Until s is terminal
```

Fig. 2. Q-learning Algorithm

After training (exploration) phase is complete, during execution (exploitation) phase, the agent in a particular state, always takes an optimal action at that particular state determined using (3)

$$a_t = arg \underset{a \in A}{max} \, Q_t(s_t, a) \tag{3}$$

Comparing Q-learning to Dijkstra's Algorithm, Q-learning can handle positive and negative weights instead of just positive weights in Dijktra's. Even though the time complexity of Q-learning is $\mathcal{O}(VE)$ which is more than Dijktra's $\mathcal{O}(V \log V)$, it is more suitable for stochastic environments and distributed systems.

II. System Model

We consider a scenario where W Sensor Nodes (SN) in a WSN are trying to send data to the Base Station (BS) in an efficient way through a network whose reliability is unknown. One example of a scenario is shown in Fig. 3. The links in the network, represented by the adjacency matrix P_{ij} $\forall i \in W, \forall j \in W$ denote the channel link reliability which is unknown to SNs. We assume that each node transmits with same power and hence covers same distance.

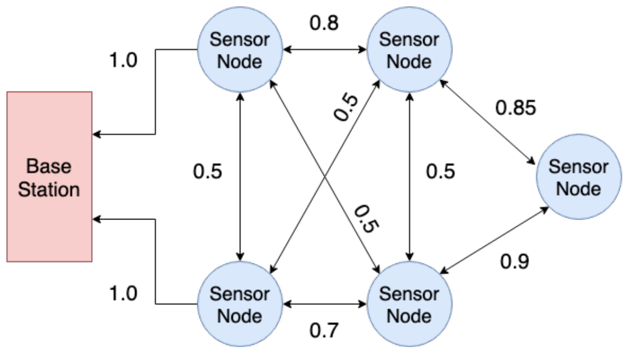

Fig. 3. Wireless Sensor Network

A. Problem Statement

The objective is to find the most reliable path for data to be routed from each sensor node to BS. The Q-learning technique provides a decentralized method to achieve the objective.

III. Solution: Q-Learning

At the core, Q-learning algorithm is a dynamic programming where each node N_i $\forall i \in W$stores the quality/goodness $\mathbf{Q}$ of other nodes as shown in Table I. N_0 is the destination (BS) of packets from each sensor nodes N_i. Each sensor nodes takes action a from the action space A and gets reward r. This provides decentralized way of computing the reliable path. The exploration and exploitation phases are described in Fig. 2.

TABLE I
Q-Table

Nodes	**Nodes**					
	N_0	N_1	N_2	N_3	$\dots$	N_W
N_1						
N_2						
N_2						
$\vdots$						
N_W						

A. State

Each sensor node N_i being in state N_j is equivalent to transferring a packet from node N_i to node N_j.

B. Action Space

Each node takes a random action uniformly, by trying to transmit the training packets to its neighbors.

C. Reward

The immediate reward function is defined as in (4).

$$r = \begin{cases} 100 & \text{if destination is reached} \\ P_{ij} & \text{if within transmitting distance} \\ 0 & \text{if i == j : same node} \\ 0 & \text{if i and j are not at transmitting distance} \end{cases} \tag{4}$$

where $P_{ij} \sim Uniform(0, 1)$ represents the link reliability between the nodes N_i and N_j.

IV. Performance Analysis

We consider three general scenarios: static, sparse random and dense random for the analysis. The red node N_0 is the marked destination.

A. Scenario: Static

We consider a fixed sensor layout with two different link reliability scenario 'Static-1' and 'Static-2' as shown in Fig. 4 and Fig. 6. The routing path from the farthest node to the BS is found using Q-learning for both scenario and is shown in Fig. 5 and Fig. 7. The route prefers the upper path [5, 4, 2, 0] in Fig. 5 and zig-zag path [5, 3, 2, 0] in Fig. 7 because those links provide the highest reliability for the packets to be delivered.

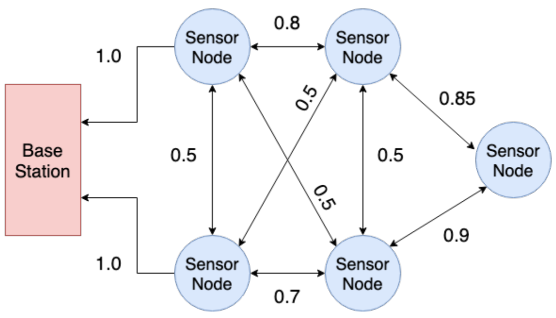

Fig. 4. WSN Setup: Static-1

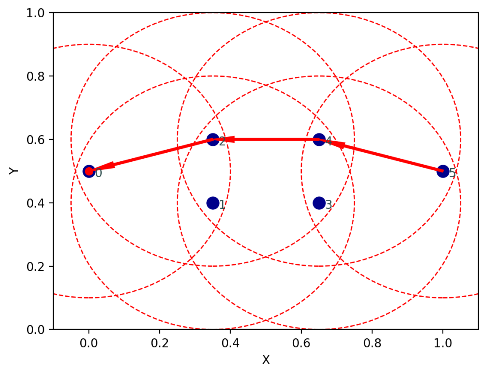

Fig. 5. WSN Route: Static-1

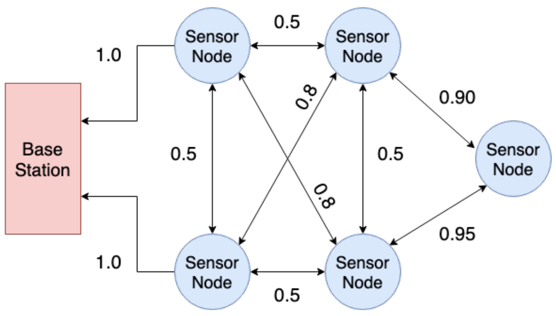

Fig. 6. WSN Setup: Static-2

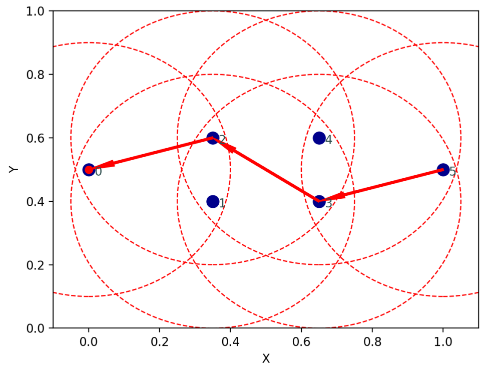

Fig. 7. WSN Route: Static-2

B. Scenario: Sparse Random

We consider a sparse random distribution of sensor nodes ($W = 20$). The destination node N_0 (marked red) has been centered. The most reliable routing path is found from every node to the destination. For visual purpose only paths from nodes 1, 2 and 3 are shown in Fig. 8 and 9.

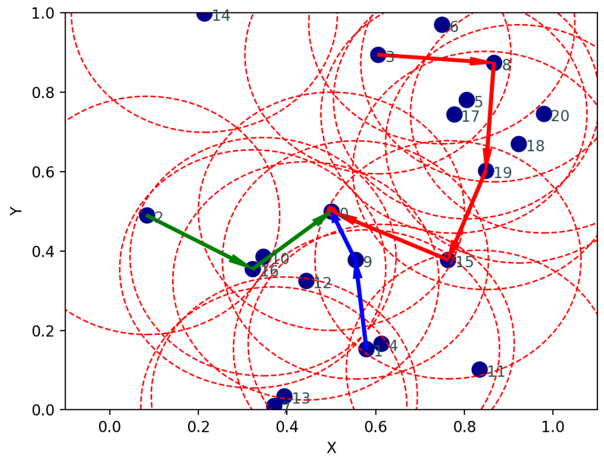

Fig. 8. WSN Route: Sparse Random (Run-1)

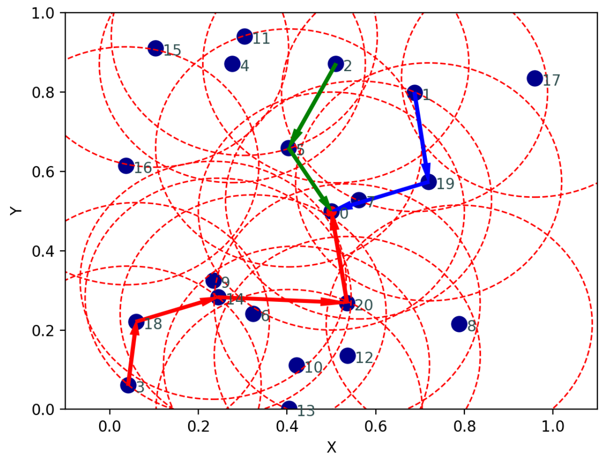

Fig. 9. WSN Route: Sparse Random (Run-2)

C. Scenario: Dense Random

We consider same dense random distribution of sensor nodes ($W = 100$) with different sensor transmit power. The most reliable routing path is found from every node to the destination. For visual purpose only paths from nodes 8, 18, 50, 55 and 83 are shown in Fig. 10 and 11. It can be noticed that when the transmit power of sensor node increases, the better routing path involving less number of nodes is found.

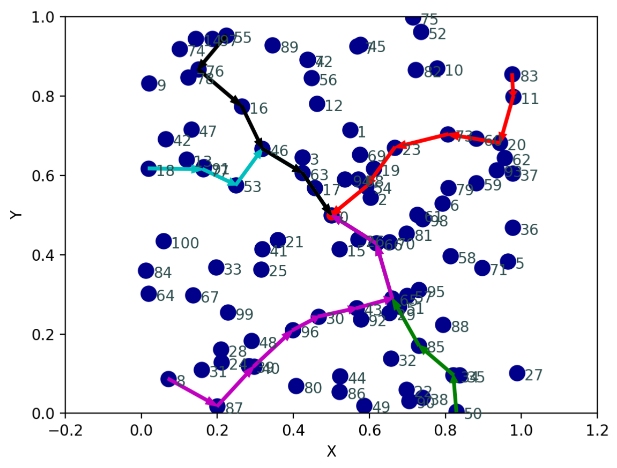
Fig. 10. WSN Route: Dense Random (Radius 0.15)

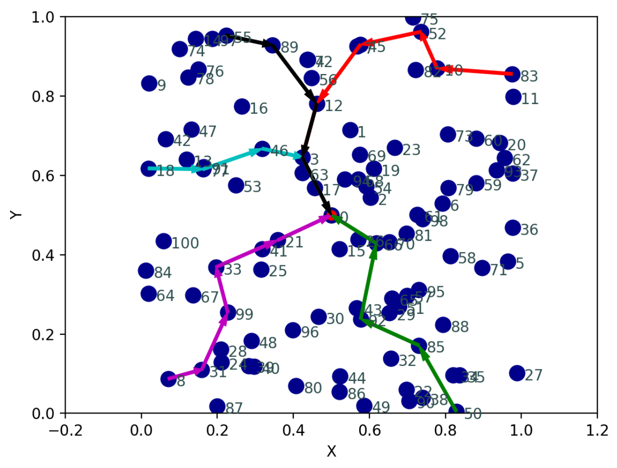
Fig. 11. WSN Route: Dense Random (Radius 0.2)

V. Future Works

Apart from the works done in this project, there are some improvements that can be done to further increase the reliability of routing. Currently the nodes only improve their routing information during the exploration phase and not in exploitation phase. In real scenario, the channel link quality is stochastic and do change in exploitation phase also. Hence in real scenario, the Q-learning tables should also be updated for normal packets as done for training packets.

Currently the link quality between nodes is considered to be uniformly distributed. A better model might be to consider link quality as a distance metric between nodes with respect to maximum distance that a node can transmit. This is a reasonable model and is directly in relation with the path loss of the channel. Also the reward function currently only considers the link to destination and stochastic link between the nodes. The reward function could be improved by incorporating remaining energy of the nodes which in turn finds a path that is also energy efficient. The trade-off between energy efficient and reliable routing path is yet to be studied. Finally the implementation of the model using hardware implementation or mote simulation is necessary to verify the results.

VI. Conclusion

In this report, we investigated the routing in wireless sensor nodes based on link reliability using one of reinforcement learning technique: Q-learning. The performance of Q-learning routing for different scenario: static, sparse random and dense random is discussed. Based on the results, it is concluded that Q-learning is an effective technique for reliable routing of packets from sensor nodes to the base station.

References

[1] A. Sarkar and T. S. Murugan, "Routing protocols for wireless sensor networks: What the literature says?" *Alexandria Engineering Journal*, vol. 55, no. 4, pp. 3173–3183, 2016.

[2] H. Wang, D. Peng, W. Wang, H. Sharif, and H.-H. Chen, "Cross-layer routing optimization in multirate wireless sensor networks for distributed source coding based applications," *IEEE Transactions on Wireless Communications*, vol. 7, no. 10, pp. 3999–4009, 2008.

[3] O. Bazan and M. Jaseemuddin, "A conflict analysis framework for QoS-aware routing in contention-based wireless mesh networks with beamforming antennas," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3267–3277, 2011.

[4] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and energy-efficient disjoint multipath routing for WSNs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 7, pp. 3255–3265, 2012.

[5] L. Lin, N. B. Shroff, and R. Srikant, "Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources," *IEEE/ACM Transactions on Networking (TON)*, vol. 15, no. 5, pp. 1021–1034, 2007.