Reinforcement learning based reliable routing in Wireless Sensor Network

Wireless Sensor Network: Course Project Presentation

Presented by: Sudat Tuladhar
Department of Electrical Engineering
The University of Mississippi

May 1, 2019

Outline

Introduction

- 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**

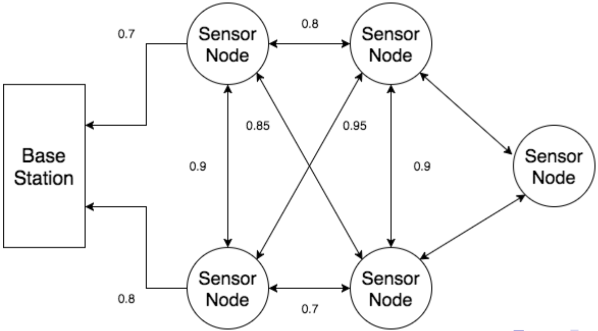
Literature Review: Routing Protocols

- **Energy Efficient**: Energy Usage Scheduling (H. Wang, et al. , 2008)
- **Delay-Less**: Analyze the wireless link conflicts by evaluating the end-to-end delay transmission (O. Basan, M. Jaseemuddin, 2011)
- **Secure**: Security and Energy-efficient Disjoint route to maintain network security (A. Liu, et al. , 2012)
- **Reliable**: Reliability of the topology, link between the nodes, protocol flow (L.Lin, et al. , 2007)

- Reliable routing based on Reinforcement Learning is a potential solution

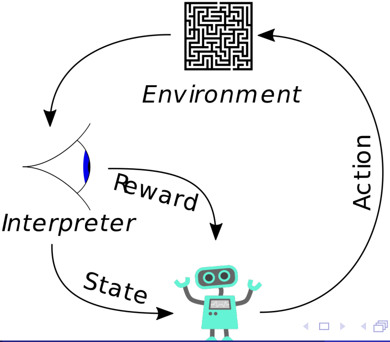
Scenario

- Nodes in WSN trying to send data to the Base Station (BS) reliably
- Each node transmits with same power, covers same distance
- Reliability of the link is unknown, destination location unknown
- Most reliable path for data to be routed from each sensor node to BS is to be found
- De-centralized solution



Reinforcement Learning

- Area of machine learning concerned with how **software agents** ought to take actions in an environment so as to maximize some notion of **cumulative reward**
- Differs from supervised learning in that **labeled input/output** pairs need not be presented
- Instead the focus is finding a **balance between exploration** (or uncharted territory) and **exploitation** (of current knowledge)



Reinforcement Learning Model

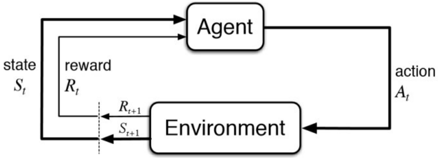


Figure: Reinforcement Learning Model

Q-Learning:

$$Q(\text{state}, \text{action}) \leftarrow (1-\alpha)Q(\text{state}, \text{action}) \\ + \alpha(R(\text{state}, \text{action}) + \gamma * Max[Q(\text{next state}, \text{all actions})])$$

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\overbrace{\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}}}^{\text{learned value}} - \underbrace{Q(s_t, a_t)}_{\text{old value}} \right)$$

Q Learning Algorithm: Action / Reward

Action:

- Uniform sampling: Each node transmits training packets to its neighbors

Reward (Immediate):

- +100 if destination is reached
- +0 if connection (good link quality)
- -1 if no connection (bad link quality)

Q Learning Algorithm: Exploration / Exploitation

: **Exploration**: Training

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

Exploitation: Transfer packet to the neighboring node that has highest link quality (Value in the Q table)

Comparison

Dijkstra's Algorithm:

- Time Complexity: O(VlogV)
- Doesn't work for Graphs with negative weight edges
- Suitable for static environment

Q learning:

- Time Complexity: O(VE) (more than Dijkstra's)
- Can handle positive and negative weight
- Suitable for stochastic environments
- More suitable for distributed system

Simulation Results: Sensor Layout-1

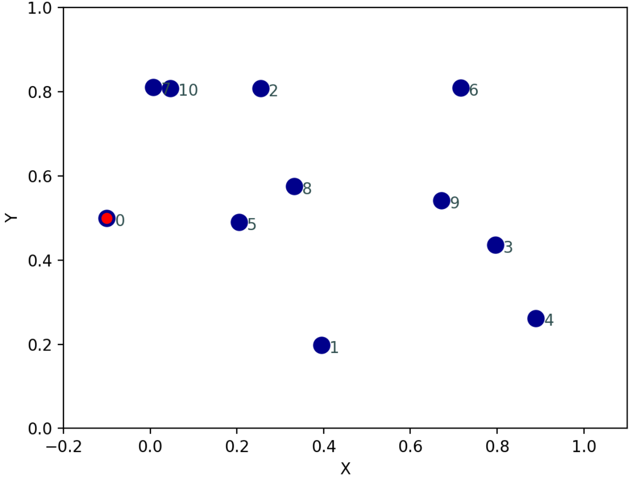


Figure: Sensor Layout-1

Simulation Results: Sensor Layout-1

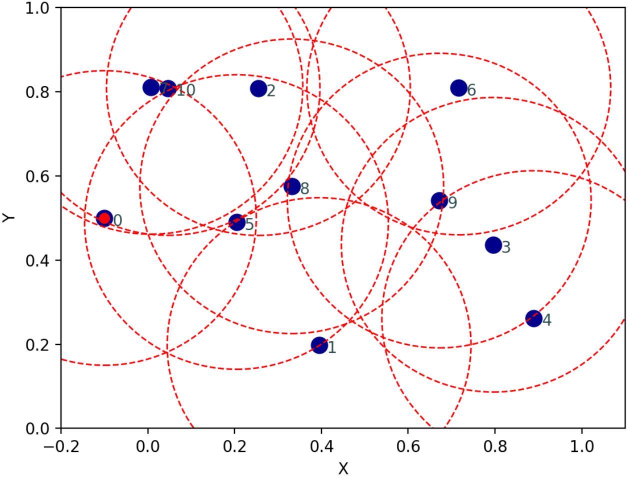


Figure: Sensor Layout-1 Range

Simulation Results: Sensor Layout-1

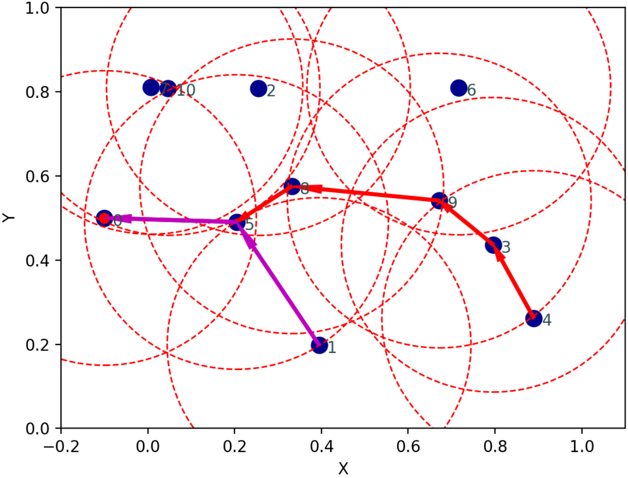


Figure: Sensor Layout-1 Route

Simulation Results: Sensor Layout-2

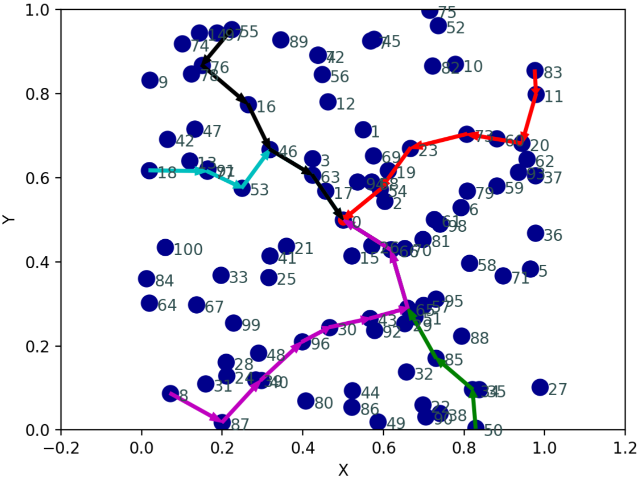


Figure: Sensor Layout-2 Radius 0.15 Route

Simulation Results: Sensor Layout-2

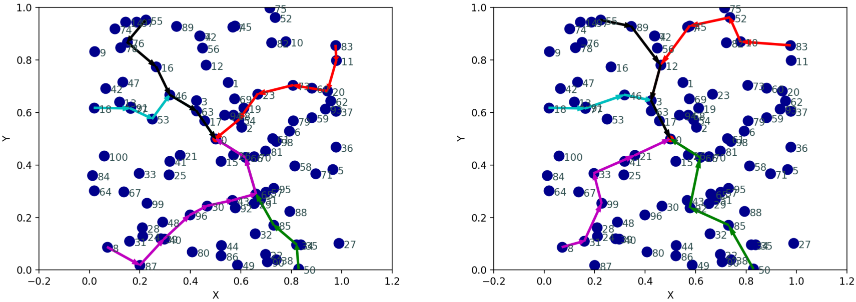


Figure: Sensor Layout-2 Radius 0.15 Vs 0.2 Route

Remaining Work and Challenges

Remaining :

- Integrate Link Quality into reward

Future Work :

- Mote Simulation Perspective
- Energy Efficient

Challenges :

- Fine-tuning hyper-parameters of model

Conclusion

- Need of a reliable routing protocol
- Reinforcement learning a potential solution to learn the wireless sensor network environment and find reliable routing paths from nodes to destination in stochastic environment