DATA PRESERVATION WITHIN A SENSOR NETWORK

JENNIFER LY AND BIN TANG
COMPUTER SCIENCE - CSU DOMINGUEZ HILLS

ABSTRACT

This project proposes that reinforcement learning methods(eg: Q-Learning Algorithm) will be able to provide a general framework to help find the best storage node for data nodes to offload their data packets off to. The agent will be able to keep in mind the factors of offloading all data packets and maintaining the lowest communication cost possible(distance).

INTRODUCTION

In this proposal, we start with a simple sensor network that is composed of 4 nodes(Figure 1).From this example, we can see that the best solution is to have each data nodes offload to the storage node on their left. Both data nodes travel 1 hop in order to offload their data. This is where we consider reinforcement learning to train the data nodes to find the best storage node to offload their data to. Here, we design and implement (Q-Learning) to help find the best paths for all data nodes. We expect that this implementation will provide a generalized solution that can also address larger sensor networks(eg: a KxK grid), and achieve our two primary goals: data preservation and lowest possible communication costs.

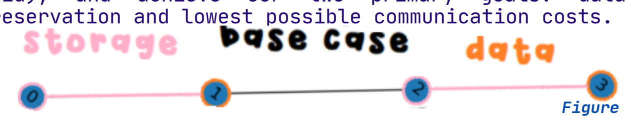


Figure 1

BACKGROUND

In a base station-less sensor network, there can be sensor nodes with an overflow of data packets. In this scenario, the main goal would be to preserve the data packets and not lose the overflow data. Therefore these sensor nodes that need to offload the extra data packets (now referred to data nodes) onto sensor nodes that can take on the overflow (storage nodes).But it takes energy to send the overflow to the storage nodes(communication costs); and the data nodes' main goal is to preserve the data. Therefore to address both constraints, we need to find the most optimal path assignment to find where a particular data node can offload its data to a storage node; while incurring the lowest possible cost of offloading the data from a data node to a storage node.

PROBLEM STATEMENT

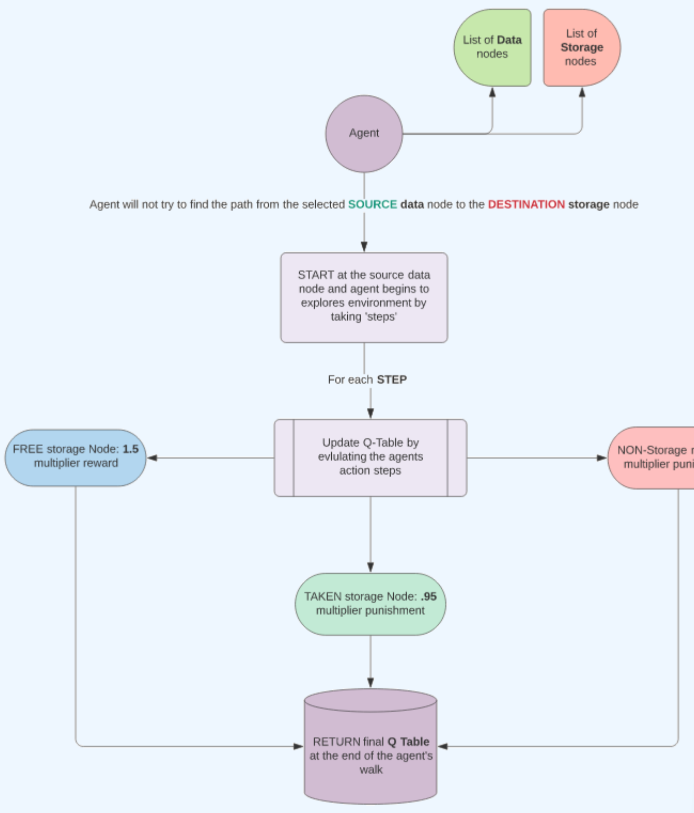
The use of Reinforcement learning techniques(eg Q-Learning) can provide an optimal assignment of data nodes to storage nodes that keeps the communication cost and data preservation intact.

HYPOTHESIS

The Q-Learning implementation will be able to provide the optimal assignment of data nodes to offload their data packet to the best selected storage node.

TESTING PROCEDURE

1. Create a 4x4 sensor grid network ***(Figure 2)***
2. Assign 3 random data nodes by node ID
3. Assign 3 random storage nodes by node ID
4. **FOR** 100 episodes the agent does the following:



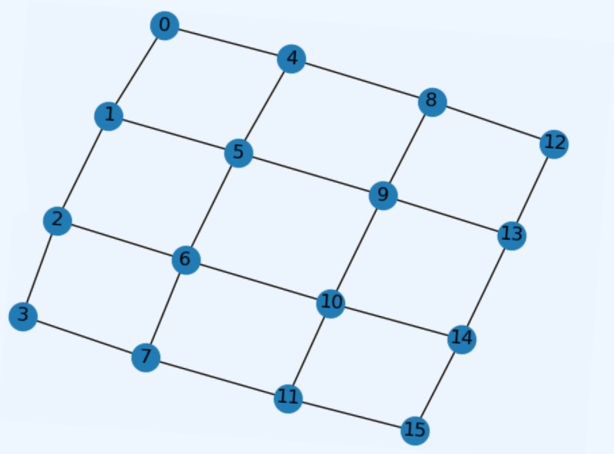


Figure 2

5. Add the Q Tables together
6. Check data nodes MAX value from final Q Table
   i. CHECK if assignments are unique
      1. IF so:
      2. Display assignments and DONE
   ii. IF not:
      1. Rank data nodes by impact (MAX - MIN)
      2. Assign storage nodes to data nodes by rank
7. Run Minimum Cost Flow to compare costs

RESULTS

- Base case scenario was validated
- Out of 25 trials
  - 22 trials the Q Learning implementation found the optimal solution ***(Figure 3)***
  - 3 trials were disqualified due to agent never finding the path from a given data node and storage node
- When solutions was found, the 'cost' was the same as MCF

```
DATA NODES  [4, 11, 5]
STORAGE NODES  [9, 12, 3]
```

```
OFFLOAD DATA FROM NODE: 5 TO NODE: 9 with MAX Value being: 1415455.0
OFFLOAD DATA FROM NODE: 4 TO NODE: 12 with MAX Value being: 534835.375
OFFLOAD DATA FROM NODE: 11 TO NODE: 3 with MAX Value being: 350270.625
THEY WERE BOTH THE SAME!
```

DATA	SOURCE	Q Learning Cost	MCF Cost	Time MCF(sec)
[1,10,13]	[2,7,12]	5496558.13	5496558.13	0.038
[4, 6, 9]	[0, 11 ,14]	4081517.25	4081517.25	0.034
[5,6,10]	[2,7,11]	324324312.44	324324312.44	0.034
[3,5,7]	[0,6,11]	6521596.44	6521596.44	0.043
[1,5,10]	[3,6,11]	4782138.08	4782138.083	0.038
[1,2,6]	[5,10,12]	2798220.5	2798220.5	0.039
[3,10,13]	[0,5,11]	3584550.7	3584550.7	0.033
[10,14,15]	[0,4,8]	4103840.3	4103840.3	0.034
[1,6,9]	[11,13,15]	5496558.13	5496558.13	0.075
[6,8,12]	[0,11,14]	1724662.32	1724662.32	0.023
[1,2,3]	[10,11,12]	2528124.5	2528124.5	0.036
[2,7,10]	[11,8,9]	7137189	7137189	0.036
[2,6,10]	[1,9,11]	3929242.82	3929242.82	0.029
[10,11,14]	[1,4,7]	2615657.39	2615657.39	0.032
[1,4,10]	[2,6,11]	93829381.07	93829381.07	0.034
[2,4,10]	[9,11,13]	3863239.13	3863239.13	0.03
[6,9,14]	[11,13,15]	5216558.13	5216558.13	0.04
[4,7,9]	[0,11,13]	5163366	5163366	0.033
[0,4,11]	[2,7,15]	3548422.59	3548422.59	0.029
[0,7,11]	[6,9,15]	5194127.77	5194127.77	0.032
[2,9,13]	[0,5,12]	5136552.75	5136552.75	0.035

CONCLUSIONS

- Promising start and supports looking into more robust implementations
- Develop method to automatically adjust parameters as needed
- Take a variable amount of data packets and storage capacities
- Consider use of existing Reinforcement Learning libraries

REFERENCES

Chen-Khong Tham and J. -. Renaud, "Multi-Agent Systems on Sensor Networks:A Distributed Reinforcement Learning Approach," 2005 International Conference on Intelligent Sensors,Sensor Networks and Information_Processing,2005,pp.423-429,doi: 10.1109/ISSNIP.2005.1595616.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grants numbers 1923956, 2034030, and 2036143. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.