Faculty of Science and Technology
Department of Computer Science

Peer Observations of Observation Units

Camilla Stormoen

INF-3981 Master's Thesis in Computer Science ... May 2018

Abstract

What is wrong with the world? Motivation 1-3 sentences, Arch, Des, Imp, Exp 1,2-3 sentences, results and main conclusion.

Acknowledgements

Contents

List of Figures

1 Introduction

FRA CAPSTONE: *The Arctic tundra in the far northern hemisphere is challenged by climate changes in the world today and is one of the ecosystems that are most affected by these changes[10]. The Climate-ecological Observatory for Arctic Tundra (COAT) is a long-term research project developed by five Fram Center[1] institutions. Their goal is to create robust observation systems which enable documentation and understanding of climate change impacts on the Arctic tundra ecosystems. COAT was in autumn 2015 granted substantial funding to establish research infrastructure which allowed them to start up a research infrastructure during 2016-2020[10].*

Wireless Sensor Network (WSN) is a system that consists of hundreds or thousands of low-cost micro-sensor nodes. These nodes monitor and collect physical and environmental conditions. The various activities in the sensor nodes consume lots of energy and the battery of the sensor node is difficult to recharge in wireless scenarios and also because the sensor nodes are located at remote areas in the Arctic tundra.

This thesis presents the architecture, design and implementation of a peer observation that can observe and accumulate data from in-situ observation units.

1. http://www.framsenteret.no/english

1.1 Motivation

The motivation behind this project is...

The purpose is to fetch and accumulate data observed by observation units for further use.

The observation units to be used for the prototype comprises Observation Unit Processes executing on PCs and/or Raspberry Pi.

1.2 Contributions

The dissertation makes the following contributions:

- A
- B

1.3 Assumptions

Avgrense viktig!

1.4 Limitations

Avgrense viktig!

1.5 Outline

This thesis is structured into X chapters including the introduction.

Chapter 2 describes ..

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter X

2 Routing Techniques in WSNs?

Som eget kapittel eller ha det under Related Work? Si noe om routing protocols som hierarchical (evt flat-based og location-based) og si noe om routing protocol operations som multipath (evt. query-based, QoS-based, coherent-based etc..)?

Har også fra WSN-bok ("Protocols and Architectures for Wireless Sensor Networks, Holger Karl, Andreas Willig) kap. 11 som heter "Routing Protocols" som sier noe om gossiping, energ-efficient unicast, broadcast/multicast, geographic routing og mobile nodes..

3

Related Work

WSNs main task is to periodically collect information of the interested area and broadcast the information to a Base Station (BS). An easy approach to achieve this task is to make each sensor node transmit their data directly to the BS. But the problem is that the BS can be far away from the sensor node so a direct data transmission would not be possible, or if the routing path from the sensor node to the BS is long, the sensor node may/will die due to energy consumption. There are multiple hierarchical protocols that has been proposed as a solution to this problem.

To improve the overall energy dissipaton of WSNs, Low-Energy Adaptive Clustering Hierarchy (LEACH) [1] introduce a hierarchical clustering algorithm for sensor networks. It is self-organized and use randomization to distribute the energy load evenly among the sensors in the network. The sensor nodes organize themselves into local clusters where one node is the local BS or CH. The CH are not fixed to avoid nodes to drain their battery and to spread the energy usage over multiple nodes. The nodes self-elect a new CH depending on the amount of energy left at the nodes at different time-intervals. LEACH is divided into different rounds where each round include a setup phase and a steady-state phase [2]. In the setup phase will each node decide whether to become a CH or not. When a CH is chosen, each node will select its own CH based on the distance between the node and the CH and join the cluster. In the steady-state phase will the CH fuse the received data from the node members in the cluster and send it to BS.

In diversity, will nodes in our approach first connect to a cluster and then start a CH election rather than elect a CH first and then nodes joining the cluster. A resemblance between the two approaches is that neither of them consider a nodes energy level when calculating the CH.

LEACH do not consider a nodes energy level when calculating the CH and has been a benchmark for improving algorithms such as the centralized clustering algorithm LEACH-C [7] and distributed clustering algorithm such as LEACH-E [9] and LEACH-B [10]. They concentrate on energy consumption reducing a nodes residual energy and more relevant criterions [6].

Fuzzy-LEACH (F-LEACH) [4] [5] have three different fuzzy descriptors such as energy, concentration and centrality used to complement the cluster head selection process. The BS performs the CH election in each round by computing the chances of a node becoming a CH by calculating the three fuzzy descriptors. F-LEACH also assumes that the BS elects the appropriate CH because it has a complete information about the whole network.

In contrast to F-LEACH will our approach elect a CH by the nodes in the cluster and not in a BS. The CH election will not consider "variables" such as battery level or number of nodes in the cluster.

PEGASIS is a chain-based protocol with the idea to form a chain among the sensor nodes so each node will receive and transmit data to a close neighbor. The sensor nodes will also take turns on being the leader for transmitting data to the BS and therefor distribute the energy load evenly among the sensor nodes. The chain can be organized by the nodes themselves using a greedy algorithm starting from some node or the BS can compute the chain and broadcast it to all the nodes in the network [3].

As mentioned earlier,

To increase the robustness of devices and lower power consumptions, ZebraNet [11] provides a low-power wireless system for position tracking of wildlife by using peer-to-peer network techniques. This reduces the researchers effort to manage the sensors and collecting logged data for their research. The radios on their devices also operates on different frequencies and have different bandwidth, range and other characteristics.

A diversity to our approach is that ZebraNet stores multiple copies of the same data across multiple nodes while our approach forwards the data to a node

Gossip-based protocols, or epidemic protocols, are popular protocols due to their ability to reliably pass information among a large set on interconnected nodes. Jelasity et al. [12] provide a Gossip-based communication protocol (GBCP) where each node have peers to gossip with in a large-scale distributed system [8]. These nodes can quickly join and leave the network at any given point of time. The general principle of their framework is that every node (1) maintains a relatively small local membership table that provides a partial view of all nodes and (2) periodically refreshes the table using a gossiping procedure.

The difference between GBCP and our approach is that our approach does not use a gossip protocol to update its table of nodes, but instead relies on the communication with other nodes to know about the election of a new CH, who CH is and when a node should accumulate data and send it to the CH.

4

Architecture

This chapter describes the architecture of the system. The main functionality can be divided into 3 sub-sections: a nodes lookup service, discovery of other nodes and incoming network requests. The architecture of the system is presented in 4.1.

4.1 Node Lookup Service

The lookup service is responsible for storing a list of all previous discovered nodes and their address.

4.2 Discovery Of Other Nodes

Each node can only discover nodes withing a possible range. Figure 5.2 shows how a possible range of a node may be discovered. When a new node in the network has been discovered will meta-data about the node such as address be stored on locally on the device in the cluster.

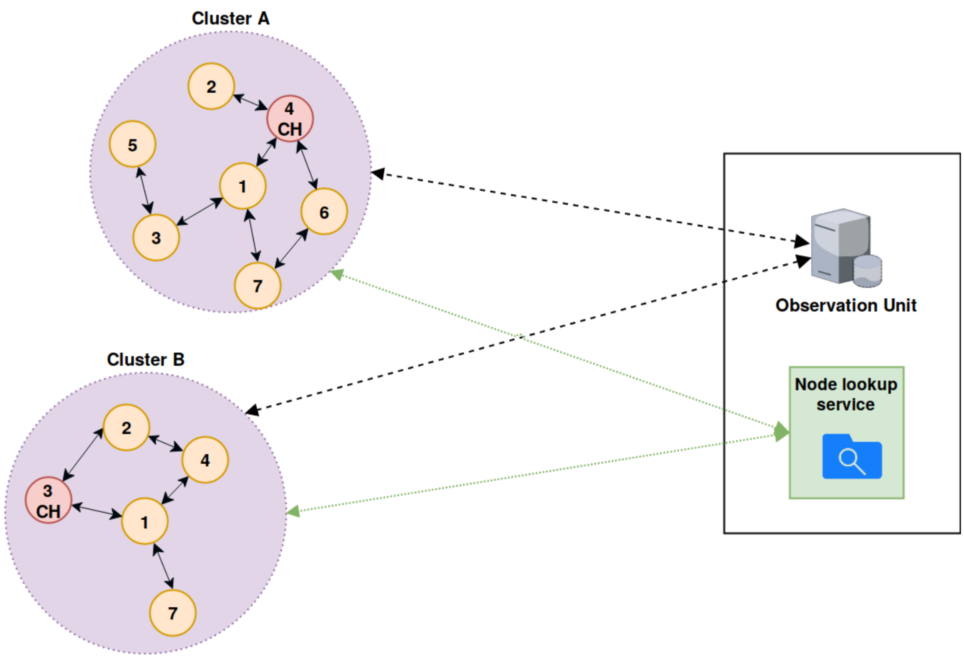

Figure 4.1: Figure shows the architecture of the system.

4.3 Incoming Network Requests

A node may receive incoming requests from other nodes in the network. The request handler will handle the request based on the type of request. The types of requests a node may receive are listed below.

4.3.1 Connect To Neighbours

When a node receive a list of neighbours in range, it will try to connect to the neighbours that are within the nodes range. It will only connect to the neighbour node if it receives a OK-message.

Receive OK From Neighbours

When a node receives a OK-message it will connect itself to the neighbour. The neighbour will also then have connected to the new node.

4.3.2 Cluster Head Election Request

A node may receive a CH election request when a node has joined the cluster.

Cluster Head Election Request

When a node receives a CH election request it will perform a leader election and forward the result to it's neigbours which will do a leader election as well.

Cluster Head Election Calculation Request

If there is a leader in the cluster already and a new election should be proposed, a CH election calculation request is sent from the leader. Nodes receiving this request will calculate their CH election number. This is explained further in Section xx.

4.3.3 Data Transmission

Notify Neighbours About Sending Data To Leader

A node may receive a request that it should send its data to the leader. This request is forwarded to the nodes neighbour and so on.

Send Data To Leader

This request forwards a nodes data to the next node in the path to the leader. If the nodes data receiving this requests isn't sent to the leader of the cluster, will the data be accumulated with the received data and then forwarded to the next node.

5

Design

In this chapter we will look at the design of the system and present the design of each component of the architecture. Figure 5.1 shows how the cluster network may appear (in the system). Nodes are connected to other nearby nodes represented by arrows and together they form a cluster network.

5.1 Discovery Of Other Nodes

5.1.1 Broadcasting

When a new node start it will contact the node lookup service to discover other nodes in the network. The node will then initiate a broadcast. Broadcast is limited due to a radio range limitations where only nodes that are within this range will receive the broadcast, shown in Figure 5.2. *Node 1* will only reach *node 5* and *node 7*.

5.2 Cluster Head Election

A CH election may occur in two scenarios listed below.

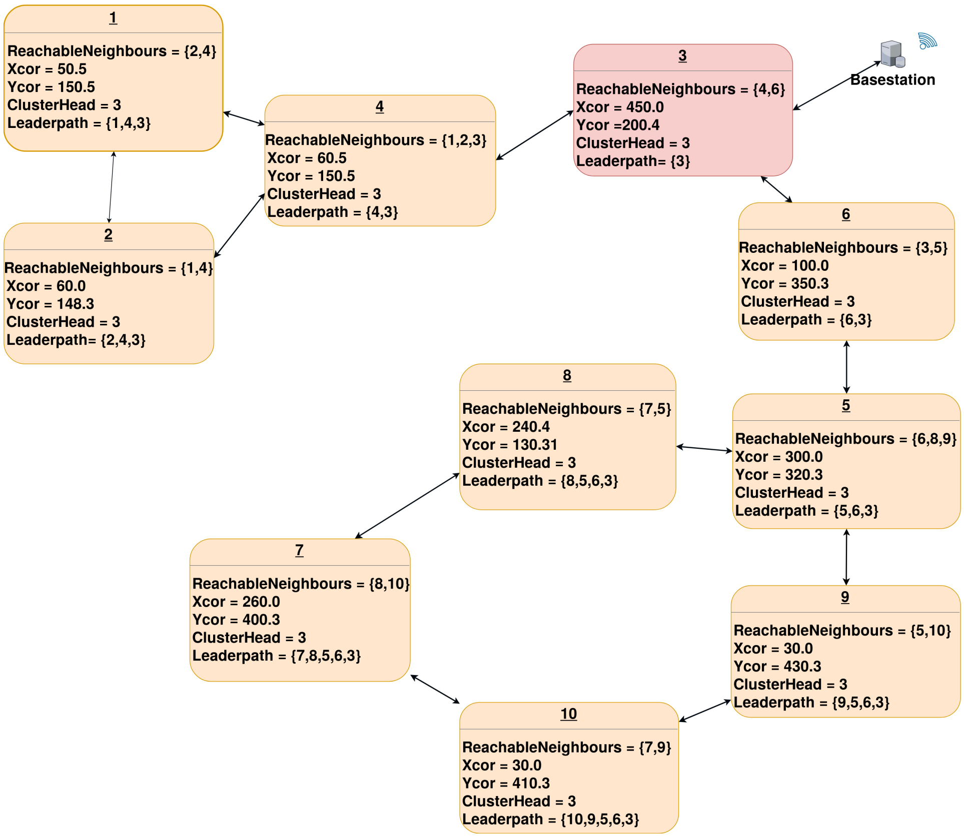

Figure 5.1: Figure show design of the system.

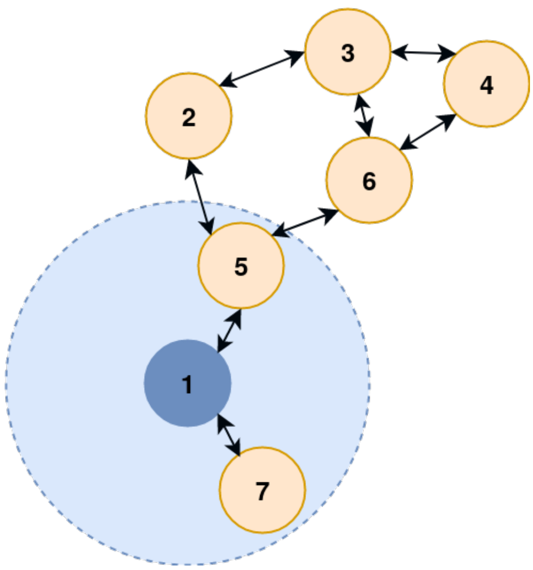

Figure 5.2: Figure show broadcast range of a OU.

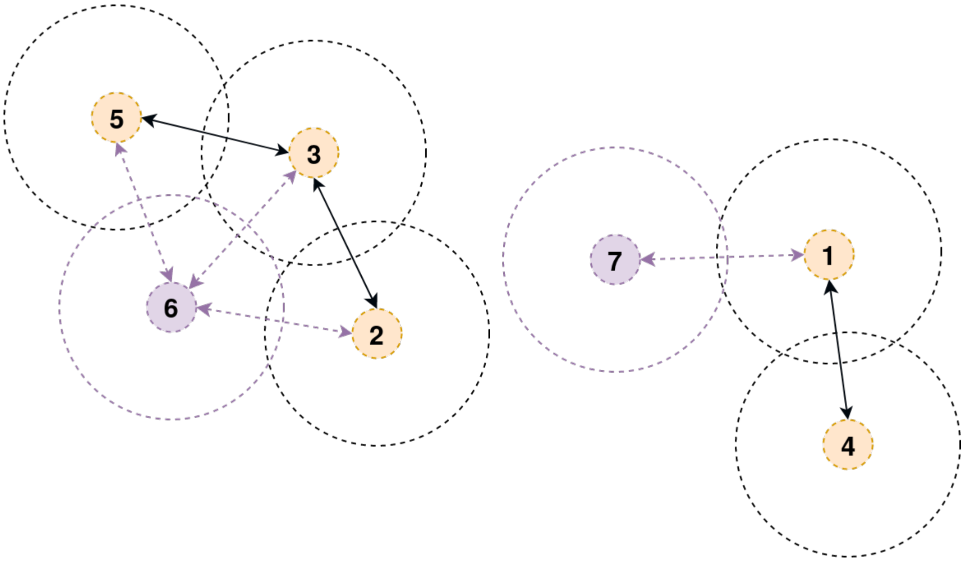

Figure 5.3: Figure show how new nodes (purple) contacts their neighbours.

5.2.1 New Node In Cluster Wants Election

See Figure 5.5.

5.2.2 Cluster Head Wants Election

See Figure 5.6.

5.3 Accumulate Data

A node may at times forward data to another node in the cluster.

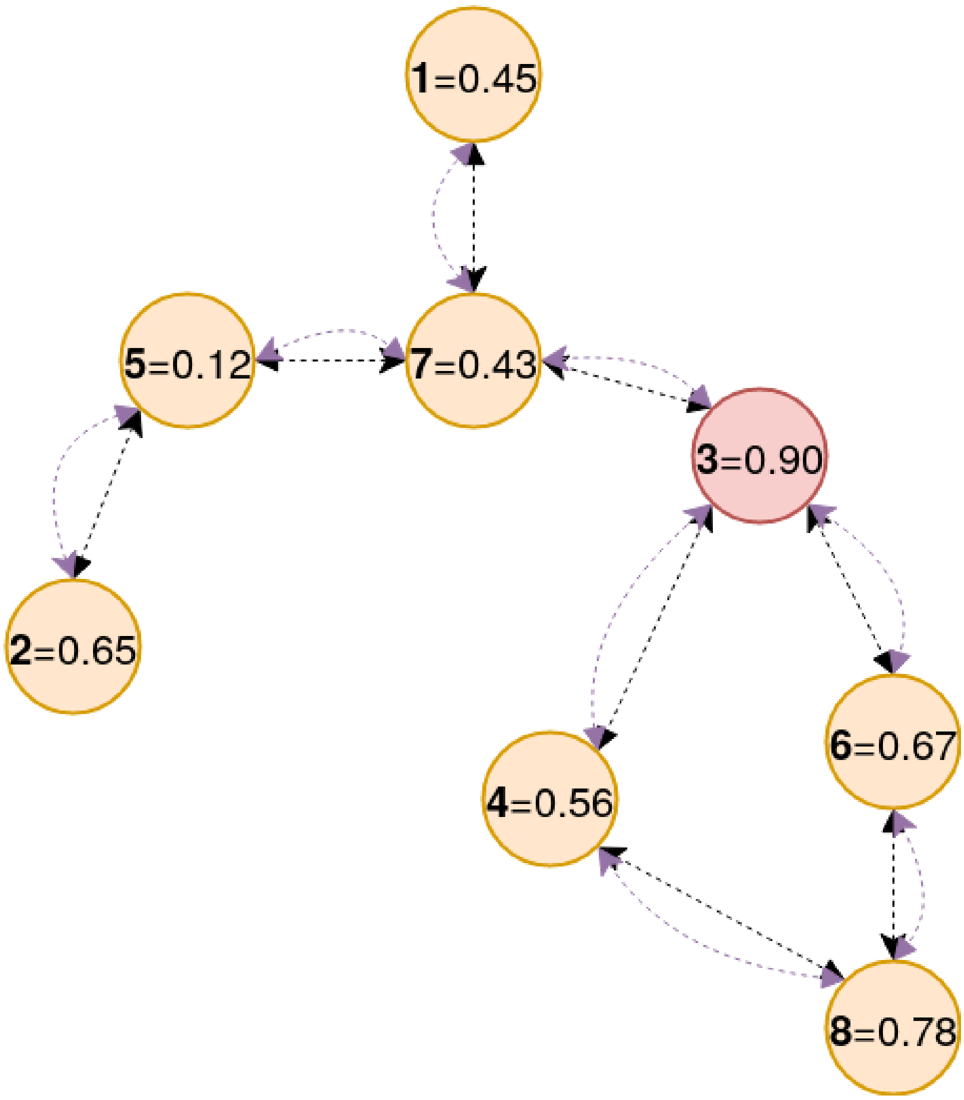

Figure 5.4: Figure show how a CH election is gossiped in the cluster.

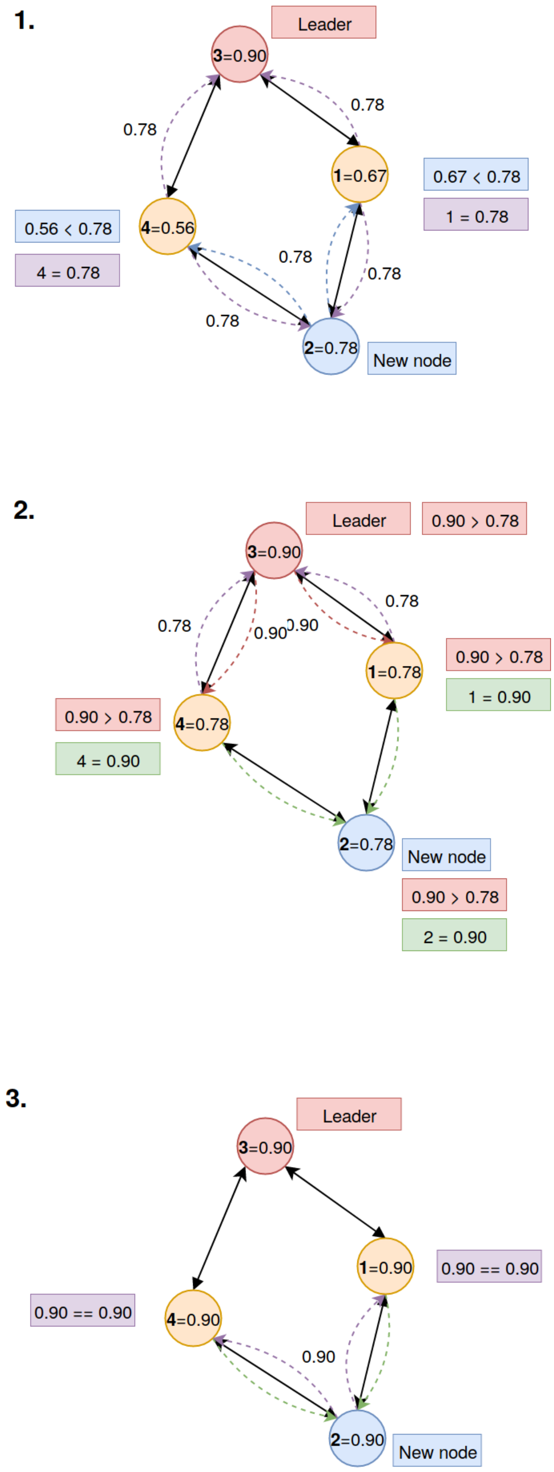

Figure 5.5: Figure show how a new node starts a leader election.

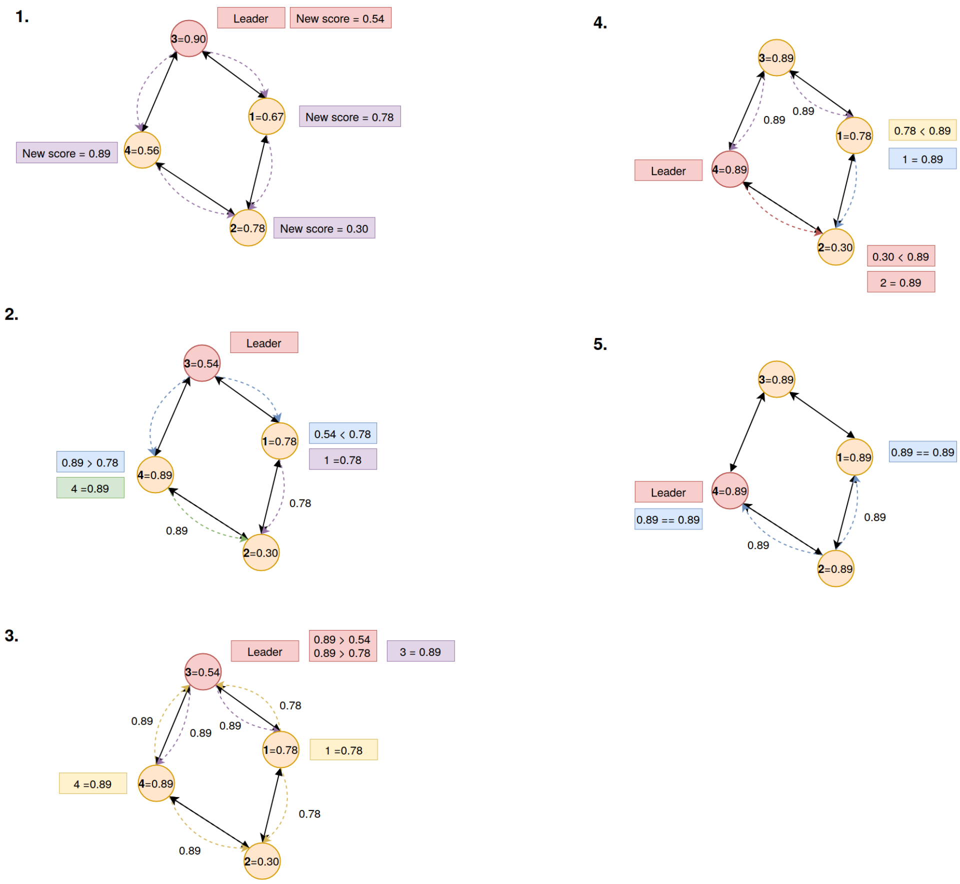

Figure 5.6: Figure show how a new node starts a leader election.

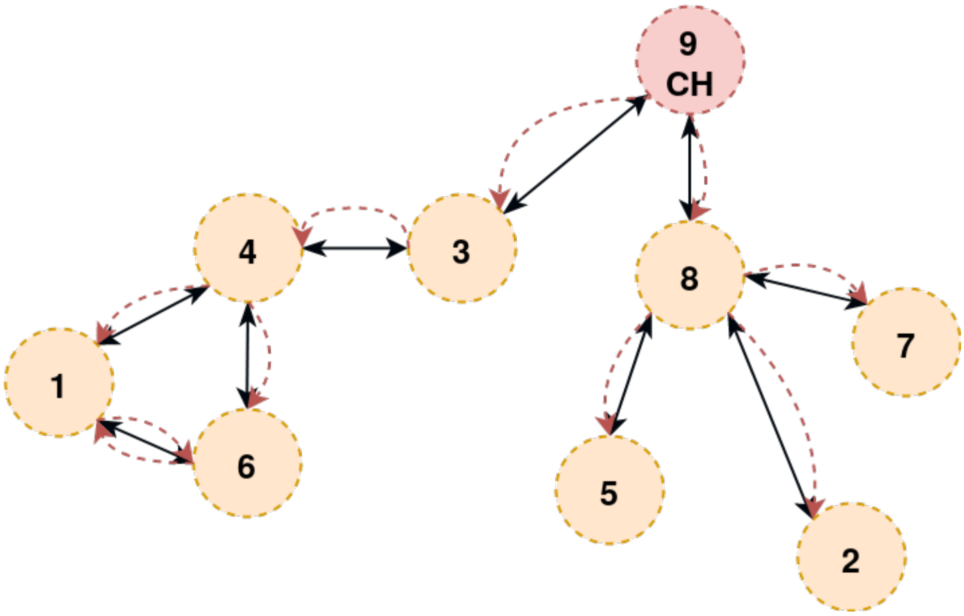

Figure 5.7: Figure show how nodes get an request for sending data to the leader.

6

Implementation

This chapter will elaborate on how we implemented the system, general implementation requirements, issues and choices.

The system is implemented in the open source programming language GO 1.9.3[1].

6.1 Distance To Other Nodes In The Network

The formula 6.1 is used to calculate the range between two points in a two-dimentional coordinate map and is used to see if a node is within a range or not, as seen in Figure 5.2.

$$d = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{6.1}$$

Figure 6.1 shows how a node contact the lookup service where the nodes position is calculated and nodes in range is returned to the node. At last, the node will try to connect to the reachable nodes.

1. https://golang.org/

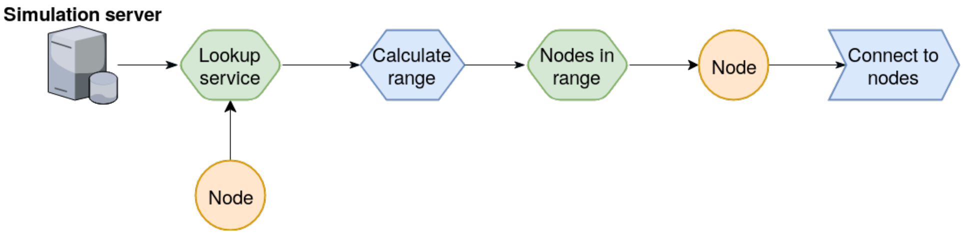

Figure 6.1: Broadcast simulation

6.2 Data Transmission

6.2.1 Cluster Head Election

6.3 Minimize Path To Leader

6.4 Data Accumulation

The collected data is stored in a map and maps in Go are not safe for concurrent use. If a map is read from and wrote to from concurrent goroutines, the access must be compromised by a synchronization mechanism. One of the most common ways to protect maps is by using mutexes.

Listing 6.1: Small Go program showing how mutexes are used when updating a map

```
/*DBStation is a strucure that contains
a map that store data at leader*/
var DBStation struct {
        sync.Mutex
        BSdatamap map[uint32][]byte
}

func sendDataToLeaderHandler() {
        DBStation.Lock()
        defer DataBaseStation.Unlock()

        ...

        DBStation.BSdatamap[sData.FP] = receivedData
}
```

7 Evaluation

This chapter describes the experimental setup and metrics used to evaluate the implemented system.

7.1 Experimental Setup

All experiments were done on a Lenovo ThinkCenter with the following specifications:

- Intel® CoreTM i5-6400T CPU @ 2.20GHz × 4
- Intel® HD Graphics 530 (Skylake GT2)
- 15,6 GiB memory and 503 GB disk
- Ubuntu 17.04 64-bit with gcc V6.3.0 compiler and GO 1.9.3

7.2 Experimental Design

7.2.1 CPU Measurements

7.2.2 Memory Measurements

7.2.3 Network (Socket) Usage

7.2.4 Network Lifetime?

7.3 Results

7.3.1 CPU Usage

7.3.2 Memory Usage

7.3.3 Network Usage

7.3.4 Network Lifetime?

8

Discussion

This chapter discusses our approach, experience, how we solved the problem and why we chose the solution we ended up with..

8.1 Availability of nodes in the system

8.2 Cluster Head Election

8.2.1 Cluster Head Calculation/Rating

8.3 Path To Leader

8.4 Data Transmission

8.4.1 Data Accumulation

8.5 Base Station Access

9 Conclusion

In this thesis, we have implemented a system/prototype...

Our experiments showed that the system ...

10

Future Work

11

Appendix

Bibliography

[1] W. R. Heinzelman and A. Chandrakasan and H. Balakrishnan, *Energy-efficient communication protocol for wireless microsensor networks*, 2000, in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 10 pp. vol.2-.*

[2] Z. Han and J. Wu and J. Zhang and L. Liu and K. Tian, *A General Self-Organized Tree-Based Energy-Balance Routing Protocol for Wireless Sensor Network*, 2014, in *IEEE Transactions on Nuclear Science Vol.61, Nr.2, pp. 732-740.*

[3] S. Lindsey and C. S. Raghavendra, *PEGASIS: Power-efficient gathering in sensor information systems*, 2002, in *Proceedings, IEEE Aerospace Conference Vol.3, pp. 3-1125-3-1130.*

[4] A. K. Mishra and R. Kumar and J. Singh, *A review on fuzzy logic based clustering algorithms for wireless sensor networks*, 2015, in *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), pp. 489-494.*

[5] Indranil Gupta and D. Riordan and Srinivas Sampalli, *Cluster-head election using fuzzy logic for wireless sensor networks*, 2005, in *3rd Annual Communication Networks and Services Research Conference (CNSR'05), pp. 255-260.*

[6] Maryam Sabet and Hamid Reza Naji, *A decentralized energy efficient hierarchical cluster-based routing algorithm for wireless sensor networks*, 2015, in *AEU - International Journal of Electronics and Communications Vol.69, Nr.5, pp. 790 - 799.*

[7] W. B. Heinzelman and A. P. Chandrakasan and H. Balakrishnan, *An application-specific protocol architecture for wireless microsensor networks*, in *IEEE Transactions on Wireless Communications Vol.1, No.4, October 2002, pp. 660-670.*

[8] Demers, Alan and Greene, Dan and Hauser, Carl and Irish, Wes and Larson, John and Shenker, Scott and Sturgis, Howard and Swinehart, Dan and Terry, Doug, *Epidemic algorithms for replicated database maintenance*, in *ACM: Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, 1987, pp. 1–12*.

[9] Chen, Bai and Zhang, Yaxiao and Li, Yuxian and Hao, Xiao-Chen and Fang, Yan, *A Clustering Algorithm of Cluster-head Optimization for Wireless Sensor Networks Based on Energy*, in *Journal of Information and Computational Science, Vol.8, 2011*.

[10] M. Tong and M. Tang, *LEACH-B: An Improved LEACH Protocol for Wireless Sensor Network*, in *J2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 2010, pp. 1-4*.

[11] Juang, Philo and Oki, Hidekazu and Wang, Yong and Martonosi, Margaret and Peh, Li Shiuan and Rubenstein, Daniel, *Energy-efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet*, in *ACM: SIGARCH Comput. Archit. News, Vol.30, No.5, December 2002, pp. 96–107*.

[12] Jelasity, Márk and Voulgaris, Spyros and Guerraoui, Rachid and Kermarrec, Anne-Marie and van Steen, Maarten, *Gossip-based Peer Sampling*, in *ACM Trans. Comput. Syst., Vol.25, No.3, August 2007*.

[13] Draves, Richard and Padhye, Jitendra and Zill, Brian, *Routing in Multi-radio, Multi-hop Wireless Mesh Networks*, in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, 2004, pp.114–128*.