DEPARTMENT OF
COMPUTER SCIENCE

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

JOANA FALCÃO DE MOURA
BsC in Computer Science

IOC: INTERNET OF COWS

TOO BE ADDED

Dissertation Plan
MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
Draft: *January 16, 2023*

DEPARTMENT OF
COMPUTER SCIENCE

IOC: INTERNET OF COWS

TOO BE ADDED

JOANA FALCÃO DE MOURA
BsC in Computer Science

Adviser: João Carlos Antunes Leitão
Assistant Professor, NOVA University Lisbon

Co-adviser: João Nuno de Oliveira e Silva
Assistant Professor, University Lisbon

Dissertation Plan
MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
Draft: *January 16, 2023*

Abstract

Resumo

Contents

List of Figures **vi**

List of Tables **vii**

Acronyms **viii**

1 Introduction **1**

1.1 Motivations 1
1.2 Problem Statement 2
1.3 Objectives 2
1.4 Structure 2

2 Related Work **3**

2.1 Gossip Protocol 3
2.1.1 History and Overview 3
2.1.2 Parameters 3
2.1.3 Strategies 3
2.1.4 Tree-based Approaches 4
2.1.5 Examples 5
2.1.6 Gossip Limitations 6
2.1.7 Discution 6
2.2 Wireless Sensor Networks 6
2.2.1 Definition 6
2.2.2 Architectures and Strategies 6
2.2.3 Gossip in WSNs 6
2.2.4 Applications 6
2.3 Sensors and Arduino 6
2.4 Cows Walking and Eating Habits 6
2.5 Existing Collars 6
2.6 Summary 6

3 Work Plan 7

Bibliography 8

Appendices

Annexes

List of Figures

List of Tables

Acronyms

CREW Concurrent Random Expanding Walkers *(p. 6)*

MON Management Overlay Network *(p. 6)*

NeEM Network Friendly Epidemic Multicast *(p. 5)*

SCAMP Scalable Membership Protocol *(p. 5)*

TCP Transmission Control Protocol *(p. 5)*

1

Introduction

This chapter will explore the motivations for this thesis development, the underlying problem that provoced the need for this dissertation, the objectives expected to achive during its development and the current document structure.

1.1 Motivations

In the Bla Bla Bla Farm, located in Bla Bla Bla, they have over xxx cows, alongside many other animals, spread throughout xxxacres. Controling that many animals in such a vaste terrain is quite a difficult task. Futhermore, the region lacks network connection, which complicates this mission even more.

The cows are kept seperated in herds depending on their ages, this means that the younger cows are not in the same herd as the older ones. This kind of seperation is quite important for them to coexist. Inside each herd they follow a hierarchical structure, having a leader that all the other cows follow.

The BLA BLA BLA Farm currently has physical fences in place to mantain the multiple herds seperated from each other and protected. However, this fences are not very persistent, which lead to an often replacement and potencial danger for the cows. In addition, since the farm has an immensive amount of land, it is reasonably strenuous to locate all cows and make sure all are healthy and safe.

Having already available some great options of collars that create virtual fences for all kinds of animals, there are still no alternative that would work for the BLA BLA BLA Farm. Mainly because of the lack of network connection available, but also do to the immensely amount of cattle at this location.

1.2 Problem Statement

1.3 Objectives

During this dissertation it is expected to be developed a fully functional animal collar, adaptable to any cow, that connects to gps and creates a virtual fence for each herd. This fence should be adjustable accordingly to the user's wish and the collar should send a vibration to the cows, when this fence area is changed, in order to get the herd to the new location created by the new virtual fence position.

The collars should provide accurate information about the cows locations as well as be highly scalable to handle all the cows informations.

1.4 Structure

The remainder of this dissertation is organized as follows:

- Chapter 2 - Related Work: includes research on existing protocols for broadcasting, focusing on Gossip Protocol, options for wireless sensor networks, available sensors and Arduino alternatives, how cows behave in a herd and lastly some possible existing collars.
- Chapter 3 - Work Plan: a description of the future work organization and explication of each work phase.

2

Related Work

2.1 Gossip Protocol

2.1.1 History and Overview

The gossip protocol, as the name indicates, was created based on how gossips are propagated in social groups. In a gossip protocol, nodes in a network send the information, randomly, to other nodes in the same network, similar to how a gossip is spread between members in a social group[1].

The gossip protocol is known as a highly scalable and resilient approach to implement reliable broadcast. This protocol is based on every participant propagating their messages collaboratively with all the members of their group.

This process starts when a node desires to propagate some piece of information to the other members of his network. This node will send his message to t nodes, chosen randomly, (t being a parameter called *fanout*, which is better explaned in the subsection 2.1.2). When the receiving nodes obtain the message for the first time, they will do the same as the previous node had done and resend the message to t, randomly chosen nodes. If a node receives the same message twice, it will discard it. When this happens, which may occour quite often, since the nodes are unaware of which nodes have already received a message, there is redundancy.

However, since neither node knows who has received each message and who has sent a message to whom, each node will have to keep a log of all messages that he has already received.

2.1.2 Parameters

Fanout:

Maximum Rounds:

2.1.3 Strategies

The Gossip Protocol may be executed following different approaches[5]:

Eager push approach: As soon as a node receive a message for the first time, it sends it to t randomly selected nodes. This approach consumes a great amount of bandwidth, considering it leads to multiple copies of the same messages are delivered to each target node.

Pull approach: Regularly, nodes inquire each other on new messages they've recently receive. If they adquire information about a message they haven't receive yet, they will request it expecifically from that node. This approach leads to higher values of latency, derivated from the extra round trip needed to obtain the messages.

Lazy push approach: When a node receive a message for the first time, it will only broadcast to its neighbours a small fraction of the message, a identifier, per example a hash of the message. If the neighbour never receive the given identifier it will request the rest of the message. As in the pull approach, there will be a higher value of latency.

Besides the diferences in latency and bandwidth previously mentioned, there is another important distinction between the eager push approach and the pull and lazy push approaches. Considering that the eager push approach sends the entirety of each message immediatly after receiving it, the nodes do not need to manitain a copy of these messages, contrarly to the other two approaches that may need to resend these messages later. This leads to a higher memory requirement for these approaches[6].

By combining the approaches studied above, we can get better results, obtaining a better latency/bandwidth tradeof. This are two of the studied combined approaches[2]:

Eager push and pull approach: This method is divided between two distinct phases. The first phase consists on using the eager push approach to disseminate messages straightly to the nodes in the network. The second phase uses the pull approach to recover the lacunas that might have occoured during the first phase of this method. This approach reduces the amount of redundancy in comparinse with the eager push approach, without decreasing its performance. It will, however, endure a higher level of latency due to the pull phase.

Eager push and lazy push approach: It is used the eager push approach to a subset of nodes. Then it uses the lazy push approach on the remaing subset of nodes to recover the lacunas and guarantee the reliable of the method.

2.1.4 Tree-based Approaches

Tree-based broadcasting methods have a small message complexity, however, they are not particularly resilient to faults. On the other hand, gossip protocols, as mentioned earlier in section 2.1.1, are known for their resilience, but have a high message complexity[7].

In order to obtain a small message complexity and high reliability, it was considered combining both these methods.

With this protocol we obtain the nodes organized in a tree structure formate, where each node knows to whom foward its messages. To achieve this structure we have many approaches, per example the PlumTree protocols:

PlumTree protocol This protocol uses eager push and lazy push gossip, previously explaned in the section refsubsec:strategies. It separates the nodes in the network in two subsets of randomly selected nodes. The first subset of nodes uses the eager push protocol to disseminate the messages, while the other uses the lazy push protocol. The links that the eager push method uses to propagate the messages are chosen to create a randomized broadcast network using a tree-based structure. While the links used during the lazy push gossip are used to ensure the reliability of the method when nodes fail and potencially heal the broadcast tree when needed[7].

Additionally, in opposition to other gossip protocols, with tree-based gossip the connections first made by the eager push propagating will remain until it is detected a failure. This will allow us to use TCP connections, which will provide extra reliability and failure detection.

2.1.5 Examples

Throughout many years there have been proposed numerous gossip-based protocols. During this section we will discuss some of them:

Scalable Membership Protocol (SCAMP): Contrarly to many other gossip-based protocols, with scamp it is proposed that the individual nodes have a randomized partial view of the global members in the network, leading to a fully descentralized system. This is quite an important advantage for large scale groups, since it requires a significant amount of memory and generates a lot of network traffic to maintain the system's overall consistency in extensive groups. Additionally, the scalable membership protocol is also compelling for its natural increase and reorganization of the partial view of the system when new nodes are added to the network. Having this partial view around *log n* nodes (being *n* the number of overall nodes in the network) [4].

Network Friendly Epidemic Multicast (NeEM): One of the biggest problems in most gossip-based protocols is when the network gets congested and, subsequentially, the messages get lost. NeEM uses Transmission Control Protocol (TCP) to disseminate the messages and resolve this problem, with the usage of its ineherant flow and congestion control mechanisms. In order to maintain the protocol's stability, NeEM uses a buffer management technique that utilizes different approaches to discard messages on overflow. It also includes the knowledge about the messages' types in order to ensure that the buffer retains enough space and bandwidth is used to better fit each request[8].

Concurrent Random Expanding Walkers (CREW): Crew is a gossip-based protocol designed to minimize the messages dissemination speed. This is acquired by maintaining in cache the information about the already established connections, which will reduce the latency of reopening a TCP connection[3].

Bayeux:

Scribe:

Management Overlay Network (MON):

2.1.6 Gossip Limitations

2.1.7 Discution

2.2 Wireless Sensor Networks

2.2.1 Definition

2.2.2 Architectures and Strategies

2.2.3 Gossip in WSNs

2.2.4 Applications

2.2.4.1 ZebraNet

2.2.4.2 Wireless Tracking

2.3 Sensors and Arduino

2.4 Cows Walking and Eating Habits

2.5 Existing Collars

2.6 Summary

3

Work Plan

Bibliography

[1] J. Carlos and A. Leitão. *Gossip-based broadcast protocols*. 2007. (Visited on 2023-01-09) (cit. on p. 3).

[2] N. Carvalho et al. "Emergent structure in unstructured epidemic multicast". In: *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*. IEEE. 2007, pp. 481–490 (cit. on p. 4).

[3] M. Deshpande et al. "Crew: A gossip-based flash-dissemination system". In: *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*. IEEE. 2006, pp. 45–45 (cit. on p. 6).

[4] A. J. Ganesh, A.-M. Kermarrec, and L. Massoulié. "Scamp: Peer-to-peer lightweight membership service for large-scale group communication". In: *International Workshop on Networked Group Communication*. Springer. 2001, pp. 44–55 (cit. on p. 5).

[5] R. Karp et al. *Randomized Rumor Spreading*. www.icsi.berkeley.edu, 2000. URL: `https://www.icsi.berkeley.edu/icsi/node/2087` (visited on 2023-01-14) (cit. on p. 3).

[6] J. Leitao. "Topology Management for Unstructured Overlay Networks". In: *Technical University of Lisbon* (2012) (cit. on p. 4).

[7] J. Leitão, J. Pereira, and L. Rodrigues. *Epidemic Broadcast Trees*. 2007. (Visited on 2023-01-15) (cit. on pp. 4, 5).

[8] J. Pereira et al. "Neem: Network-friendly epidemic multicast". In: *22nd International Symposium on Reliable Distributed Systems, 2003. Proceedings.* IEEE. 2003, pp. 15–24 (cit. on p. 5).