

Dissemination and Memberships Protocols

Francisco Nunes*
fl.nunes@campus.fct.unl.pt
MIEI, DI, FCT, UNL

Frederico Lopes†
fmc.lopes@campus.fct.unl.pt
MIEI, DI, FCT, UNL

Octávio Freire‡
o.freire@campus.fct.unl.pt
MIEI, DI, FCT, UNL

ACM Reference Format:

Francisco Nunes, Frederico Lopes, and Octávio Freire. 2020. Dissemination and Memberships Protocols. In *The Projects of ASD - first delivery, 2020, Faculdade de Ciências e Tecnologia, NOVA University of Lisbon, Portugal*. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 ABSTRACT

There are different membership and dissemination protocols being developed and used around the world for many years. In this work we did some experimental evaluation to compare how different dissemination protocols (*EagerPushGossip* and *PlumTree*) would perform with the same membership protocol (*Cyclon*). The goal was to find the best combination of parameters and protocols to achieve the highest reliability and lowest latency.

- 150-200 words (tem 62)
- 1º parágrafo -> contexto do que vai ser apresentado e objetivos
- 2º parágrafo -> contributions and main conclusions

2 INTRODUCTION

This project was developed for the Algorithms and Distributed Systems course unit taught by Professor João Leitão. The goal of this project was to implement and compare different unstructured overlay networks with different epidemic broadcast protocols and find the best one or the best ones that when working together, deliver the lowest latency and the highest reliability.

The project was developed under the internal *BABEL* framework that was created by Pedro Fouto, Pedro Ákos Costa and Professor João Leitão at NOVA LINCS. We also used the research computational cluster of DI and NOVA LINCS to conduct experiments for our project, it allowed us to measure the reliability and latency of the protocols that we implemented.

- Should provide the context and goal of the work.
- In terms of context it starts by a broad context that becomes more and more specific as you advance in paragraphs (in the context

*Student number 50774. Francisco was responsible for doing *EagerPushGossip* and *Plumtree* protocols.

†Student number 42764. Frederico was responsible for doing *Cyclon* protocol.

‡Student number 50451. Octávio was responsible for doing *PlumTree* protocol and the report.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ASD20/21, 2020, Faculdade de Ciências e Tecnologia, NOVA University of Lisbon, Portugal

© 2020 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

you have to also motivate the reader in relation to the relevance of the work) •It then presents the goal of the work and the main contributions/results

•It usually ends with a short paragraph that indicates the structure of the document after this point.

3 RELATED WORK

In this project we used some epidemic broadcast protocols that have been developed before such as the *EagerPushGossip* and *PlumTree* and also a membership protocol called *Cyclon*.

3.1 Cyclon

FALAR SOBRE CYCLON

3.2 EagerPushGossip

FALAR SOBRE EagerPushGossip

3.3 PlumTree

FALAR SOBRE PlumTree

- Discusses previous work that is relevant to understand the contribution or results presented in the current paper.
- May be divided in sub-sections to present existing works in groups that make it easier for the reader to have a quick overview of existing works.
- In your case you might want to discuss: i) overlay protocols; ii) gossip-based broadcast; iii) experimental studies on the use of these technologies (that are similar to your own work)

4 IMPLEMENTATION

The implementation of the membership and dissemination protocols were done in java with support of the *babel* framework. We followed the pseudo-code structure of the protocols that were present in XXXXXX papers.

- Here you will want to discuss the implementation you did, present pseudo-code, or correctness arguments.
- Sub-divide this into the main components of your solution. • In the start of the section, provide a quick overview of the section (to assist the reader in understanding the organization).
- When discussing a solution, it is good to start with a high-level overview, and only after that present technical details.

4.1 Pseudo-code

4.1.1 Cyclon:

- 1: **Interface**
- 2: **Requests**
- 3: getNeighbors();
- 4:
- 5: **Indications**

```
// Link the protocols. This should be done after creating all protocols, since
// there can be data: protocol
// communications in this step.
private static void pickBroadcastProtocol(String config, Properties props, Host myself) {
    throw new IllegalArgumentException("HandlerRegistrationException", 10Exception {
        GenericProtocol broadcast = null;
        GenericProtocol broadcastApp = null;
        try {
            if (config.toUpperCase().equals(ProtocolName.EAGER_PUSH_GOSSIP.getName())) { // EagerPushGossip
                broadcast = new EagerPushGossip(props, myself);
                broadcastApp = new BroadcastApp(myself, props, ProtocolIds.EAGER_PUSH_GOSSIP.getId());
                logger.info("Started broadcast EagerPushGossip...");
            } else if (config.toUpperCase().equals(ProtocolName.PLUMTREE.getName())) { // PlumTree
                broadcast = new PlumTree(props, myself);
                broadcastApp = new BroadcastApp(myself, props, ProtocolIds.PLUMTREE.getId());
                logger.info("Started broadcast PlumTree...");
            } else if (config.toUpperCase().equals(ProtocolName.FLOOD.getName())) { // Flood
                broadcast = new FloodBroadcast(props, myself);
                broadcastApp = new BroadcastApp(myself, props, ProtocolIds.FLOOD.getId());
                logger.info("Started broadcast Flood...");
            } else {
                throw new NullPointerException("Invalid Broadcast Protocol");
            }
        } catch (IOException | HandlerRegistrationException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        GenericProtocol membership = pickMembershipProtocol(props.getProperty("membership_protocol"), props, myself);

        // Register applications in babel
        babel.registerProtocol(broadcastApp);
        babel.registerProtocol(broadcast);
        babel.registerProtocol(membership);

        broadcastApp.init(props);
        broadcast.init(props);
        membership.init(props);
    }
}
```

Figure 1: Parametrização do protocolo de broadcast a usar, através do ficheiro config.properties.

```
// Link the protocols. This should be done after creating all protocols, since
// there can be data: protocol
// communications in this step.
private static GenericProtocol pickMembershipProtocol(String config, Properties props, Host myself) {
    GenericProtocol membership = null;
    try {
        if (config.toUpperCase().equals(ProtocolName.CYCLON.getName())) { // Cyclon
            membership = new Cyclon(props, myself);
            logger.info("Started membership Cyclon...");
            return membership;
        } else if (config.toUpperCase().equals(ProtocolName.SIMPLE_FULL_MEMBERSHIP.getName())) { // SimpleFullMembership
            membership = new SimpleFullMembership(props, myself);
            logger.info("Started membership SimpleFullMemberShip...");
            return membership;
        } else {
            throw new NullPointerException("Invalid Membership Protocol");
        }
    } catch (IOException | HandlerRegistrationException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return null;
}
```

Figure 2: Parametrização do protocolo de membership a usar, através do ficheiro config.properties.

```
6:     neighbours(n) //n is the set of neighbors of the local
    process;
7:
8:     State
9:     neigh //set of neighbors of the process (local partial
    view);
10:    N //maximum number of neighbors ;
11:    sample //Sample of neigh sent to the other process
    in last shuffle;
12:
13:
14: Upon request() Do
15:     // some comment
16:     something();
```

```
17:
18: Upon indication() Do
19:     Trigger someFunction();
20:
21: If (thing ∈ set) Then
22:     doStuff();
23: Else If (someOtherCondition()) Then
24:     thing ← newThing;
25: Else
26:     Setup Periodic Timer newTimer(t, "Fail");
27:
```

5 EXPERIMENTAL EVALUATION

- Here you will present both your methodology, results, and discuss them.
- Methodology should be detailed and present all parameters that were used. Remember that good engineering is typically based in good science: the reader should have all the information required to replicate your results on his own.
- Present plots in a clear and consistent (in the document) way. Discuss results one by one. You might want to have different sub-sections for different aspects of your evaluation.
- Many plots does not mean better evaluation. Readable and complete plots are what you should strive to obtain.

Aqui vamos mostrar alguns gráficos e testes feitos aos protocolos. Tempos de Latencia e experiencias com combinações de protocolos. Cyclon e Eager push gossip, Cyclon e Plumtree, etc..

Measure: Rate of transmission(1 broadcast every 2sec or every 200miliseconds) PayloadSize small vs big (1Kb vs 1Mb) Reliability Latency

6 CONCLUSIONS

As we saw in the sections above, there are multiple algorithms to broadcast messages and to establish unstructured overlay networks. Unfortunately there is always some trade off's that we have to choose from.

- A summary of what was presented in the paper and the main take-away lessons from you result.
- You can also present some aspects that you did not cover but that you think would be interesting (aka Future Work).
- You can also present some limitations of your work and discuss them.