

Large-Scale Distributed Systems

Project C - Firefly-inspired Synchronization

1 Introduction

This is **one of the three project assignment proposals**. You need to **decide before November, 19th 2015 which project you would like** to work on and notify via email the assistant and the instructor. All projects require the same amount of work. You have to do, and we will grade, only one project.

1.1 Report and Grading

You will have to deliver a report of your work to the assistant. This report will be taken into account for the final grading (see first Lesson's slides).

Your project report has to contain:

- A text report, containing the various graphs you will construct during the sessions, the associated explanations and discussions. Remember that a graph without explanation, or an assertion that is not conveyed by a graph, do not have much interest.
- Source code, formatted in a proper way with indentations, and with appropriate comments.

Format: Reports are **individual**. Your report has to be written in **English**. The deadline is set to **December, 17th, 14:00**. *Send the report to raluca.halalai@unine.ch and etienne.riviere@unine.ch.*

Use the following mail subject: **LSDS2015 - project A - FIRSTNAME LASTNAME**. Only **PDF files** for the report and ".lua", ".gpi" (gnuplot), ".sh", etc. files for the code will be accepted.¹

Grading: We will grade your progress, the correctness, and the clarity of both the code and the report. The content is what matters most, but the presentation has an impact on the final grading.

¹If you want to use another language for processing the logs, feel free to do so as long as the language is an interpreted script language: python, ruby, perl, etc.

2 Project overview

Note: This project is based on a Gossip-based protocol; hence, it requires the use of the peer sampling service. You can either use your own implementation of the Peer Sampling Service from the first assignment or the implementation of the peer sampling service that you will find on ILIAS.

Reading: You have to read the *Firefly inspired Heartbeat Synchronization in Overlay Networks* paper *before starting the implementation*. The algorithm is described in the paper. The paper is available on ILIAS and can also be downloaded from <http://members.unine.ch/valerio.schiavoni/distAlgo12/papers/fireflies.pdf>.

The problem of heartbeat synchronization is very relevant for many distributed systems. Nodes need to know when to start/finish some computation. Think about the case of cycle-based protocols, when nodes send messages to their neighbors exactly at the start of a cycle. Heartbeat synchronization differs from clock synchronization in that nodes do not care about the cycle ID nor to count. There is not the notion of absolute time.

A short video documentary of this natural phenomenon observed within a swarm of fireflies is available on YouTube at the following address: <http://youtu.be/a-Vy7NZTGos>. Finally, a talk given by Steven Strogatz about the topic: <http://youtu.be/aSNrKS-sCE0>.

3 The Peer Sampling Service provided implementation

The PSS implementation you will use for implementing Chord on Demand proposes the following API:

- `pss_init()`: initialize the PSS. When this function returns, the active thread of the Peer Sampling Service is running and maintains the view. The call lasts for some time, and returns once a few cycles of the Peer Sampling protocol have been performed;
- `pss_getPeer()`: returns a random peer from the view.

4 Suggested workplan

This section suggests a work plan for the project assignment. You should be careful to have an implementation of the simple protocol running as soon as possible and focus on the evaluation and the report. You should propose your own work plan, which should be approved by the assistant. The underlying items are suggestions.

Task 4.1: Implement the basic version of the heartbeat synchronization as proposed in the paper. In particular, implement the skeleton in such a way to support various synchronization strategies.

Task 4.2: Implement the PhaseAdvance and PhaseDelay strategies and verify their correctness.

Task 4.3: An evaluation of these two strategies of the protocol. Several metrics can be observed including, but not limited to the emission window, the relative emission window

length, the strategy overhead, etc. The paper provides detailed explanations about these metrics, as well as their graphical intuitions. One possibility is to try to reproduce some of the proposed plots (some limitations apply).

Task 4.4: Implementation and evaluation of the Adaptive Ermentrout strategy. Its performances *should* be evaluated under churn conditions. Contact the assistant to agree upon the churn conditions to evaluate the strategy.

Task 4.5: (*optional*) Finally, implement the forth strategy proposed by the paper, the Mirollo-Strogatz variant. Compare its performances against the previously studied strategies.