Lab. 2: Programming and deploying distributed algorithms

Nov. 2021

This lab is carried out in groups of two. Each group chooses an algorithm to implement among those listed below. Each group must choose a programming language (Python or Go) and a communication paradigm (socket or REST).

The distributed algorithms to implement are the following:

1. Gnutella
2. Freenet
3. Shortest paths (synchronous version)
4. Shortest paths (asynchronous version)
5. Maximal Independent Set
6. Minimum Spanning Tree

In case the algorithm is implemented using sockets

Use the same yaml file formats used in Lab 1.

The program must be called as follow:

distributed_alg num_port neighbour-x.yaml INIT

or

distributed_alg num_port neighbour-x.yaml WAIT

INIT: means that the node starts execution without waiting for the first message to be received.

WAIT: means that the node waits for the first message to be received before starting to send messages.

num_port is the port used by your program (socket server).

neighbour-x.yaml is the neighbour file to be read by the node.

In case your algorithm is implemented using REST

The REST server "routes" must be defined before programming starts. It is recommended to use a tool such as APIDOC.

You are asked to write the *distributed_alg* program (which represents one of the algorithms mentioned above) and deploy it on a network of computers using a shell script.