Distributed System project: Global Snapshot

February 25, 2016

Ardino Pierfrancesco Alex De Biaso

In this project we were asked to implement a distributed bank application capable of transferring money between remote branches of a bank. Each bank has a TCP connection with the others banks. Each bank has to send a certain amount of money, randomly generated, to another random bank. We decided to use TCP connection because it handles automatically the retransmission and the safe delivery of a packet. The project consists in these files:

`start_bank.py`: it accepts as parameter, the IP address of the host machine. His role is to create a process for each branch of the machine declared in the file **host.list**.

`host.list`: contains the list of the branches, every host has the same file in order to have a list of every branch in the network. Every branch is a tuple composed by the IP address of the machine, the socket port of the machine and whether he has to start the global snapshot.

`bank_main.py`: this script simply start a bank instance passing to **bank.py** the Ip, the socket port and whether he has to start the global snapshot.

`bank.py`: this class is responsible for both money and snapshot handler, provides function to start the snapshot and decides the amount of money to be transfered and to which bank send the money. It contains a list of interface with each bank.

`bankinterfaceout.py`: this class provides the connection with another bank, send the money to a given bank and check whether the money has been delivered with *at most once* semantics. If transfer fails for three times, it will report to `bank.py` that somethings gone wrong and so to not withdraw the money.