Exercises: The Bully Algorithm

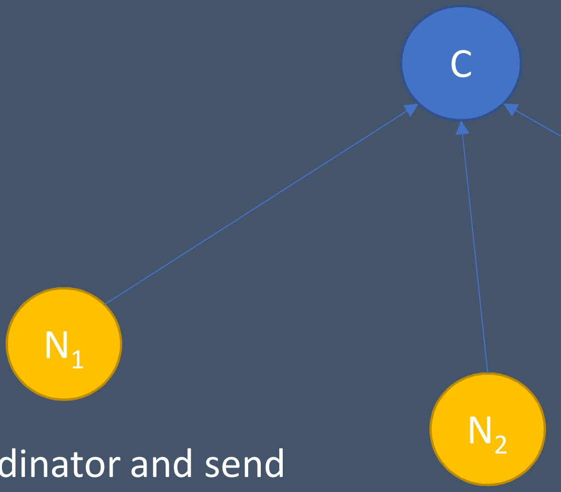
- Today's goal: A working implementation of the Bully Algorithm

Bully algorithm

- A group of processes $\{P_1, P_2, \ldots, P_N\}$ needs to elect a coordinator. We assume that processes are ordered in ascending order by their IDs $\mathrm{id}(P_k) = k$
- A process k announces that it will hold an election by means of a ELECTION message and sends it to all processes above it $P_{k+1}, \ldots, P_N$
- If some process $j > k$ answers, then it takes over and P_k waits for the result
- If no process replies, then P_k has won. It will be the new coordinator and announces it by means of a COORDINATOR message to all other processes

Exercises: The Bully Algorithm

- DS_Examples/election/bully_exercise.py



1. Coordinator starts and waits for connections

2. Nodes join coordinator and send periodic heart beats

3. As soon as coordinator stops responding, a new coordinator will be elected using the Bully Algorithm

Exercises: The Bully Algorithm

- DS_Examples/election/bully_exercise.py

1. Implement the `check_coordinator` function. It will send a heartbeat message to the current coordinator and start an election if the coordinator is considered to be down (after 5 sec. inactivity)

   The coordinator returns a copy of its list of current processes, which will be stored by the process, too

   Hint : Use `socket.setsockopt(zmq.RCVTIMEO, timeout)` to set a timeout upon receive. When timeout is exceeded, an exception ZMQError.Again will be raised

Exercises: The Bully Algorithm

- DS_Examples/election/bully_exercise.py

2. Implement the `start_election` function. This function will send a request to all processes with higher ID. If the request is acknowledged, the process waits until the end of the election
   - First step: go through the local process list
   - If reply is received: election will be held by someone else
   - If no reply is received: process becomes coordinator

Exercises: The Bully Algorithm

- DS_Examples/election/bully_exercise.py

3. Implement the `notify_new_coordinator` function. This function will send a notification to all processes that a new coordinator was elected.

   Send the new coordinator ID along with the message