

ARP - FINAL COOPERATIVE ASSIGNMENT - 1st SEMESTER - V1.4

- updates in blue -

There are N nodes, one per student, all in Azure VMs. Each node has a static public IP.

The list of IPs is known to all. Nodes can be numbered.

This task is designed, implemented and performed by a group of students (to be defined).

The task is made in *turns*. The first turn is detailed here.

REPEAT 10 TIMES

Choose randomly a node (†) (*turn leader*) that starts the turn as follows:

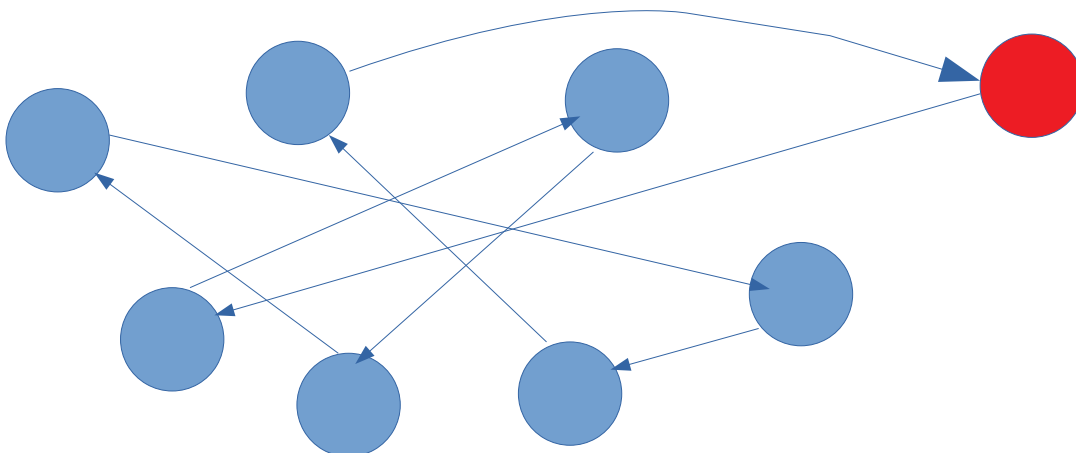
REPEAT 10 TIMES

- sorts the IP table randomly
- sends a message to the first IP of its randomly ordered table (*first node*) consisting of the name of the sending node (IP address or node number), the time of receiving (for the turn leader: 0), the time of sending
- at this point it advises its first node in the table to start execution from point 3 as *first node*
- waits for the receipt of the message from the *last* of the N nodes and measures the time elapsed in the last transmission
- the *first node* rearranges the entire table randomly and does the same as the previous one, adding the time of the last transmission to the message
- if a node knows that it has already been visited, it passes the role of first (because each node that plays as *first node* reorders the table so it could indicate as *first* a node that has already done so); in alternative, a node can refrain from sending to an already visited node
- if a node knows that all nodes have been visited, it sends to the turn leader, so the turn is completed
- the turn ends when all the nodes have been the *first* only once, have been visited once and have a list of the measures of the various preceding sections

END REPEAT (the *turn leader* repeats everything ten times)

the *turn leader* node sends the measurements and the ordered list of the sections to a central server (to be defined) with a statistic of the measurements section by section (out of the ten turns)

END REPEAT (the entire experiment is repeated 10 times, randomly choosing a *turn leader* who only has to act once).



(†) The turn leader must be chosen randomly by a sort of *vote* by all nodes. Here is a rough sketch of how it may be done (assuming that there is a common node table):

1. one node starts the vote; it can be a given node, for example the first in a node table
2. the node expresses some preference among all nodes and sends to the second node in the table
3. the second node expresses (changes, asserts...) its preference and sends to the third, and so on
4. in the end the elected is notified and becomes turn leader.

The function run by every node for the vote should be the same in all node: share its code and possibly distribute it as a library function.