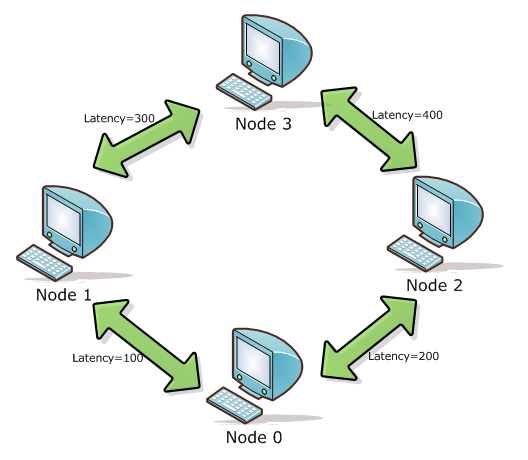
Homework1

Question1: Bidirectional links and no loss

The first topology we build upon our implementation is illustrated in the figure below.



Four nodes are connected in the way to form a ring network. The links are with no loss rate and has various latencies in milliseconds.

We obtain the following results by printing the current time when handling the FloodDoneEvent in the application component on each node:

Node 0 -> Done flooding message: "hello" at 1202054597421

Node 1 -> Done flooding message: "hello" at 1202054600508

Node 2 -> Done flooding message: "hello" at 1202054601501

Node 3 -> Done flooding message: "hello" at 1202054599421

One can see that node 0 finishes first, followed by node 3, node 1 and node 2 are among the last respectively, the order in which the food is done at each process is within our expectation.

Node 0 initializes the flooding by sending the flood message to node 1 and node 2, when node 1 sees the message it floods forward to node 0 and node 3, so does node 2. As soon as node 0 receives message back from node 1 and node 2, it finishes the job. The time spent in the transmission for node 0 to complete is 200\*2 = 400ms.

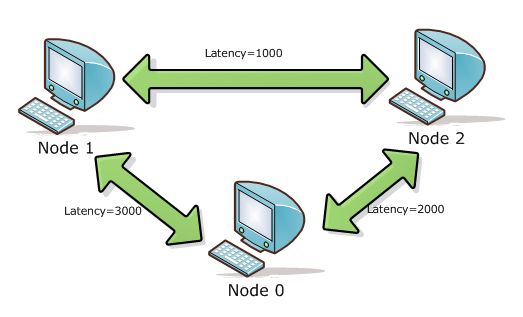
Node 3 sees the message from node 1 first (and flood it to node 1 and node 2), immediately followed by the same message from node 2, then it is done, the time spent in the transmission for node 3 to complete is 200+400 = 600ms.

Then node 1 gets the message from node 3 just before node 2 does, and they are done. The time spent in the transmission for node 1 to complete is 100+300\*2 = 700ms, whilst for node 2 is 200+400\*2 = 1000ms.

Notice that time calculated above counts from the start of the initial flooding by node 0, and the sequence of the time matches the order of termination.

If we assume the calculation and processing time on each node is more or less the same, than the interval between transmission times also matches the interval between system-times printed out.

The second topology is shown as following, three nodes are connected together, and each node has two links to the other two nodes respectively.



The result is as follows:

Node 0 -> Done flooding message: "it works" at 1202060768421

Node 1 -> Done flooding message: "it works" at 1202060765479

Node 2 -> Done flooding message: "it works" at 1202060766431

The order can be sorted as node 0, node 1