Homework 4
Simulation and Performance Evaluation
University of Trento

Davide Pasetto (239941)
Deborah Aloisi (249948)

June 2024

To achieve a simulation of the Random Waypoint mobility model we modified the discrete event simulation from assignment 3. The modifications were the following:

- The events used to advance the state of the system are the *start* event and the *reached* event. The former is the first event scheduled when the system is started, it generated the starting position, the first waypoint and the velocity of every node. The latter is the event triggered when a node reaches a waypoint; it generates the next waypoint and the velocity, and schedules the next reached event.
- To read the status of the system 2 new events were introduced: the *speed* event, which returns the current speed of every node in the system, and the *transmission* event, which is used to calculate the distance between every couple of nodes and determine whether to transmit or not. Both these events are scheduled at a fixed interval.

Every simulation in this report is executed using a population of 20 nodes.

Speeds with lower limit 0

For the first simulation we set the recording interval between *speed* events to 5 seconds, and we ran 5 simulations. The results are shown in figure 1:

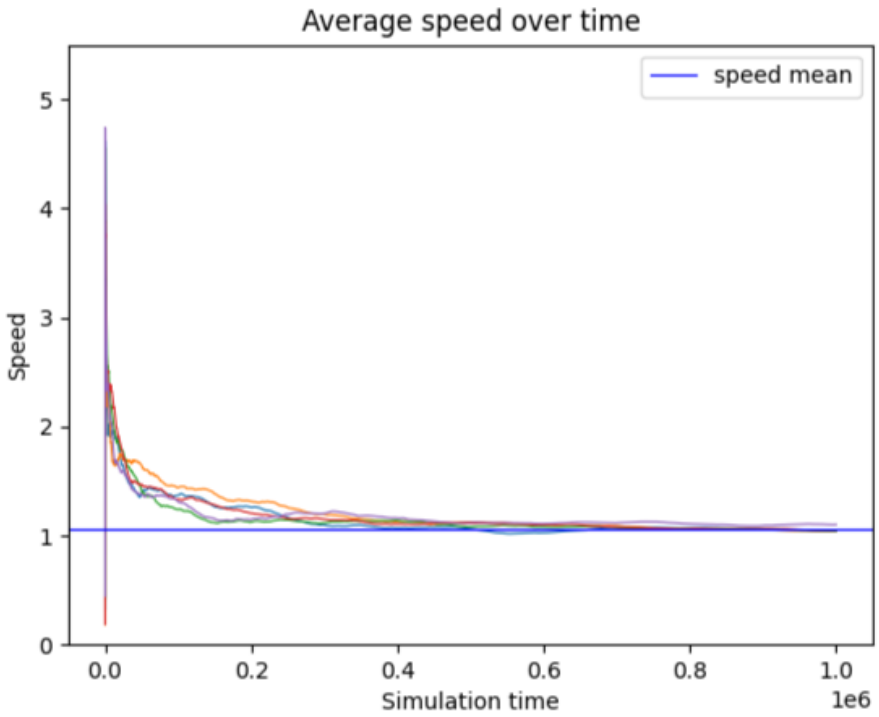
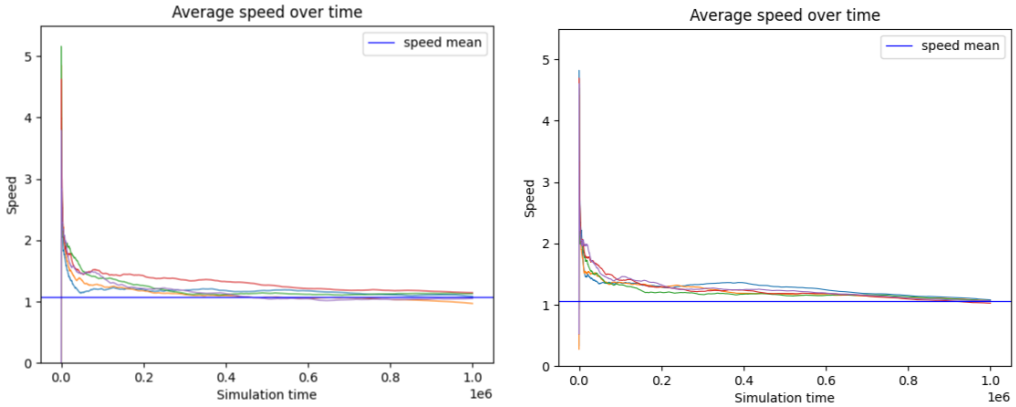


Figure 1: Simulations with 5 seconds interval, speeds between 0 and 10. Final average speed: 1.05439

As it is possible to see, the average speed of the nodes in the system is much smaller than the average of the distribution from which the speeds are drawn from, which is 5. This is because a slow node will take more time to reach the waypoint, thus being registered by a big number of *speed* events. On the other hand, a fast node will spend little time traveling, thus having its speed registered fewer times before changing the velocity, having a smaller impact on the total average. During the simulation we experimented with different intervals between speed registrations. We expected to have slightly smaller means with a bigger interval, as we thought more likely to miss or to underrepresent a very fast and node that travels for a small amount of time, but the simulation showed no particular trend when changing the interval, as seen in figure 2 and table 1



(a) Simulations with 1 seconds registration interval, speeds between 0 and 10.

(b) Simulations with 10 seconds registration interval, speeds between 0 and 10.

Figure 2

velocity	**registration interval**	**speed mean**
[0,10]	1	1.07564
[0,10]	5	1.05439
[0,10]	10	1.06160

Table 1

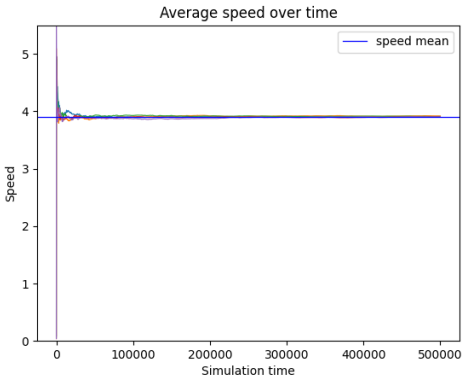
Speeds with lower limit greater than 0

When changing the speeds generation interval from [0,10] to [1,10] we noticed a sharp increase in the speed average, as seen from figure 3 and table 2

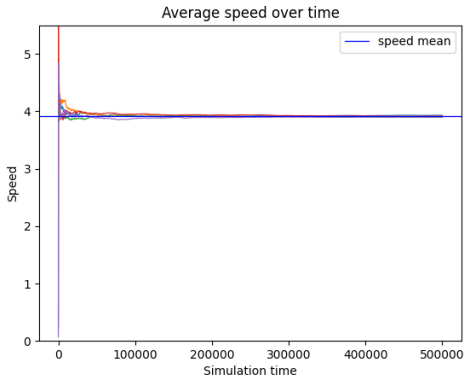
registration interval	**speed [0,10]**	**speed [1,10]**	**speed[2,10]**	**speed[4,10]**
1	1.07564	3,90836	4.97085	6.55240
5	1.05439	3,91275	4.97544	6.55186
10	1.06160	3,91631	4.96762	6.54559

Table 2

This extreme change is due to not having 0 and numbers close to 0 as possible speeds: ideally, if we had a simulation that goes on infinitely, every node will eventually generate a speed of 0, thus being indefinitely stuck and dropping the average speed to 0.



(a) Simulations with 1 seconds interval, speeds between 1 and 10.



(b) Simulations with 5 seconds interval, speeds between 1 and 10.

Figure 3

In our finite-time simulation this doesn't happens, but it is possible for a node to have a speed small enough to prevent it from reaching the next waypoint within the simulation time, therefore not having the chance to draw a different value for the velocity and dragging the average down. When we exclude this possibility by having a minimum speed greater than 0, the simulation does not have stuck nodes anymore, dramatically increasing the average speed.

Transmission

For the last point of the assignment we introduced a new event, called *transmission*, which is scheduled every second. When this event is triggered, every couple of node is checked to search for nodes within 50 meters of each other. In order to do so the position of every node is calculated, keeping in mind the velocity and the direction of the movement. If 2 such nodes are found, a transmission happens, where the involved nodes exchange a certain number of bytes of data. This exchange is bidirectional and instantaneous.

Since we have seen that the interval at which we sample the nodes speeds is negligible we chose to use a 5 seconds recording interval.

In the table we can see the comparison of the transmitted Byte's mean and throughput with different speeds intervals.

minimum speed	**maximum speed**			
	5	10	15	20
0	304.21486	315.96764	287.7592	319.9560
1	301.76928	317.23237	330.04516	342.01604
2	308.05258	320.0366	339.62717	345.48416
3	320.98969	341.17994	340.98960	349.10622

As we can see that the values are pretty constant despite the variability of the node's speed. We can conclude that the throughput of the system is around 300-340 of Byte exchanged between the nodes. The explanation is simple: if the nodes have a very low speed they will communicating more than one time during the less than 50 meter event. Instead if the node are fast they will encounter more nodes to transmit data.