WSimulNET - A Model for Simulating Large Wireless Networks

Short Guide – V.6.26 - Basic

Valério Rosset (Aug. 2018)

INTRODUCTION

WSimulNET is a simple and fast way to introduce the main concepts of wireless environments to (under)graduate students. The visual features of the model combined to the short learning curve of Netlogo's environment and language make this tool suitable to be used in either introductory or advanced courses. Therefore, for many experiment setups, no prior knowledge about Netlogo's programming language is needed to use this tool as they may not require modifying the source code.

With this tool one may simulate several types of Wireless Networks, although mostly the model concentrates on Wireless Sensor Networks (WSNs) perspective. The tool provides the means to observe, and measure by several metrics, the behavior of nodes communicating on a wireless environment. Also, it is possible to simulate scenarios with multiple sinks, with or without mobility. Despite these basic features, the WSimulNET can also be used to simulate more sophisticated protocols considering large scale and high density networks, given to network's practitioners and researchers very solid and realistic results.

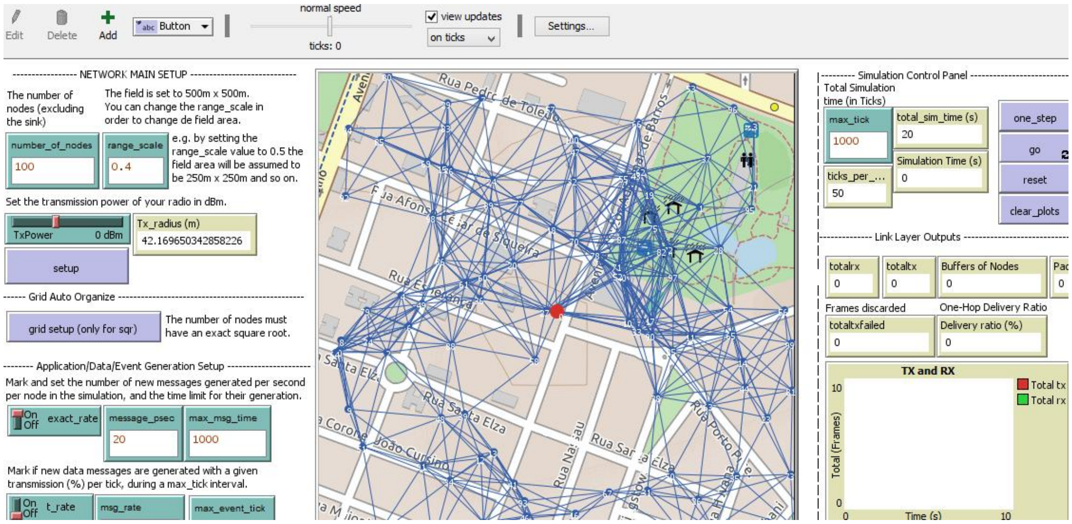

Figure 1. Overview of the tool's interface.

The model implements the CSMA/CA of IEEE 802.11-DCF, the CSMA(p-persistent) and TDMA abstraction as the medium access protocols.

In the network layer, the model implements three routing strategies:

1. **direct:** Where nodes send messages directly to a sink, if there exists a link between them.
2. **shortest-path:** Nodes proceed with multi-hop routing considering the nest hop in the shortest path to the sink. However, this strategy does not use any control message to discover the path to the sinks. The path discovery is abstracted by the use of the "network" Netlogo extension.
3. **mp-rings:** this is a adaptation of the multipath rings protocol used in the Castalia Simulator (https://github.com/boulis/Castalia). In this protocol the paths are constructed considering the shortest distance (in hops) of each node to the sinks, every node will choose the nearby sink to join and include the ID of the next hop in the shortest path towards the chosen sink in the routing table. Differently of the former protocols this is a full implemented protocol, control messages are actually used in path discovery procedure executed during the setup period. This means that, due to failures, some nodes may not find the shortest path to the chosen sink or join to the actually nearby sink. At the beginning of protocol operation the sinks broadcast *discovery path messages* to the network and sensor nodes rebroadcast the control messages to their neighbors updating the *hopcount* field in the message header. Therefore, nodes may decide on the nearby sink and the next hop to it by observing the *hopcounts* received by their neighbors. After the setup period, nodes forward data messages through the chosen path.

HOW IT WORKS

The turtles act as network nodes of a wireless network or events, the links (between turtles) indicate the radio connectivity between network nodes, every pair of nodes sharing the same link is able to communicate to each other. However, this model supports only one network interface per node and all nodes share the same configuration.

The model considers that links are symmetric and the communication range is calculated based on some chosen parameters. The transmission radius (R) of a node is given by Eq.1:

```
R = 10 ^ ((1/(pathLossExponent*10))*((TxPower) -(rxSensibility)-pld0) (Eq.1)
```

The Pld0 is assumed to be the free space path loss (based on the Okumura model) considering the distance d0 = 1m, and is given by the Eq.2:

```
PL(d0) = 20 * log(distance_d0)(10) + 20 * log(frequency_radio)(10) + 92.45     (Eq.2)
```

These parameters may vary according to the radio technology used and the medium, and they must me adjusted properly to get more realistic results.

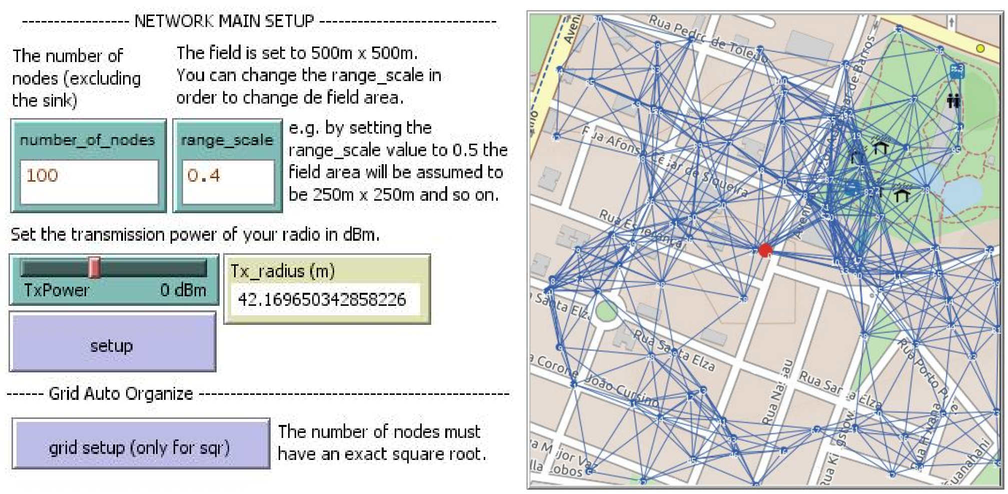

Figure 2. Main Setup Panel setting a random 100 nodes network with the TXPower set to 0 dBm and the respective view of the network created.

In the main setup panel one may choose the number of network nodes, except the sinks, used in the simulation. Also, it is possible to determine the TXPower of the radio device, by observing the communication range (TxRadius), see Figure 2.

HOW TO USE IT

Since the model simulates the many aspects of a WSN there are several parameters to be set. The setup panels are essentially separated considering parameters related to the application, network, link, radio model, energy and simulation control.

Dealing with the parameters

Application parameters:

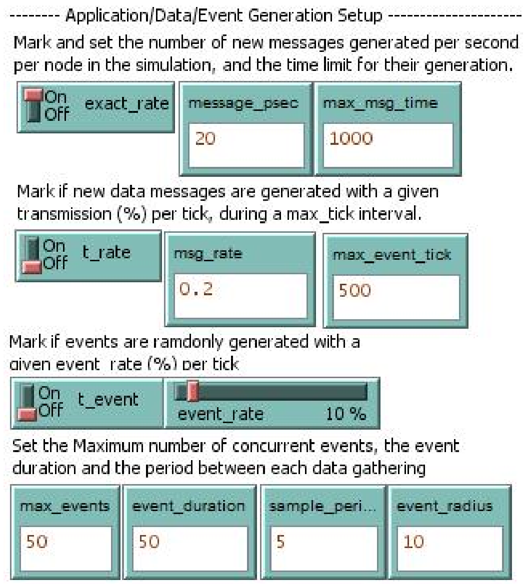

Figure 3. The *Application/Data/Event Generation Setup* panel.

In the *Application/Data/Event Generation Setup* section (Figure 3) one may find three options for data message generation (switch buttons), namely: *exact_rate*, *t-rate* and *t-event*. These three buttons can be set to *On* at the same time. However, it is preferable to use them apart.

1. If the *exact_rate* is set to *On* data messages will be generated by nodes with the rate (number of messages per second) defined in the *message_psec* input during the period (in seconds) defined in the *max_msg_time* input.
2. If the *t_rate* is set to *On* data messages will be generated by nodes with the rate (number of messages per tick) defined in the *msg_rate* input during the period (in ticks) defined in the *max_event_tick* input.
3. If the *t-event* is set to *On* data messages will be generated by nodes when they detect a event nearby (a red triangle turtle), the events are generated with the rate defined in the *event_rate* chooser during the period (in ticks) defined in the *max_event_tick* input. For this latter option is is also possible to set some features of events: the maximum number of concurrent events that may appear in the world (max_events input); each event created will remain in the map by a period (in ticks) defined by the *ma_duration* input; each node will sense the environment looking for any event in the *event_radius* radius every period of *sample_period* ticks.

Network parameters:

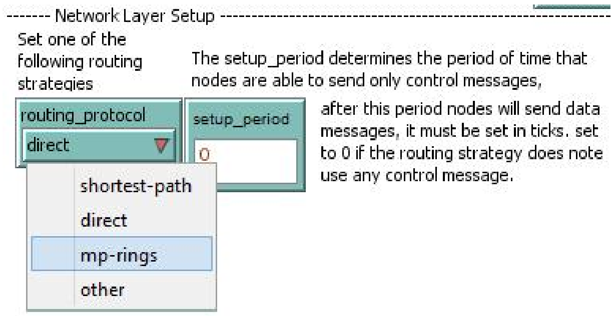

Figure 4. The *Network Layer Setup* panel.

In the *Network Layer Setup* section one may choose the routing protocol used in simulation as well as the size of the *setup_period* (in ticks) used by the protocol to exchange control messages, before starting the data generation by nodes (Figure 4). Is this model, only **mp-rings** actually generates and sends control messages, the others will not send any message during the *setup_period* interval.

Link parameters:

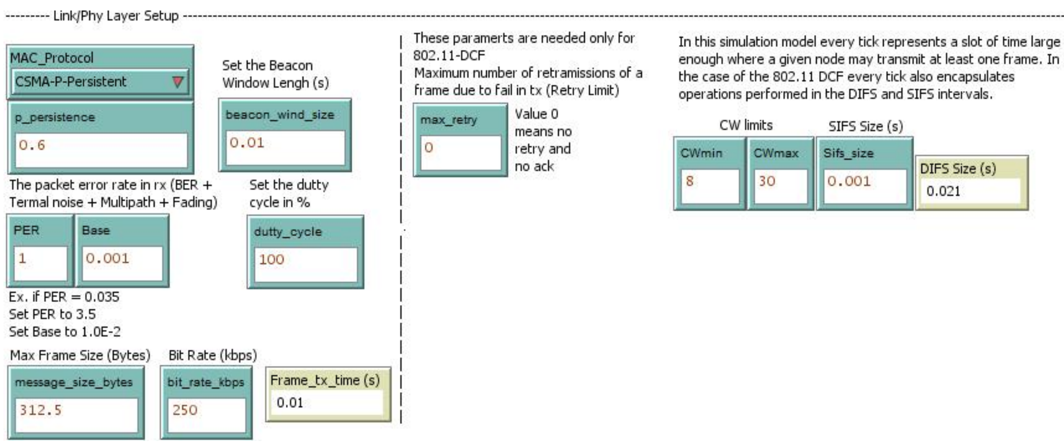

Figure 5. the *Link/Phy Layer Setup* panel.

In the *Link/Phy Layer Setup* section one may choose the MAC protocol used in simulation. The model provides three MAC strategies: the CSMA/CA strategy of IEEE 802.11-DCF including the backoff procedure, the CSMA(p-persistent) and a TDMA abstraction (Figure 6).

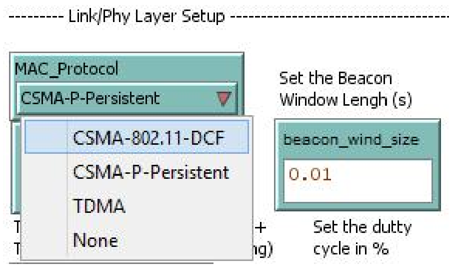

Figure 6. The implemented MAC protocols.

In this section some parameters, such as *beacon_win_size*, *PER-Base*, *dutty_cycle*, *message_size_bytes* and the *bit _rate_kbps*, are shared by all MAC protocols and other parameters are specific of the IEEE 802.11-DCF and CSMA(p-persistent) , see Figure 5. A short description of each parameter is presented next.

1. Common parameters

- *beacon_win_size*: the beacon window size in seconds.
- *PER-Base*: the packet error rate (PER) of fail in rx procedure, this rate should include BER, Thermal noise, Multipath and Fading. Use the *PER* abd *Base* inputs to indicate the rate. For example, if rate = 0.035, set the *PER* to 3.5 and the *Base* to 1.0E-2.
- *dutty_cycle*: indicates the ratio of activity of nodes (0-100). If 100 all nodes will perform their tasks during the total period of simulation. If less, nodes will randomly sleep with a probability of 100 - *dutty_cycle*.
- *message_size_bytes*: the maximum size of a frame including all headers and preambles. Notice that, in this model, all frames have the same size.
- *bit_rate_kbps*: the transmission rate of the network interface in kbps. This parameter is defined by the model of the radio device chosen for the simulation.

1. Specific parameters

- *p-persistence*: the value of p in the CSMA-p-persistent protocol.
- *max_retry*: the number of retransmissions attempts before dropping a frame (used in IEEE 802.11-DCF).
- *CWmin* and *CWmax*: the CWmin and CWmax of IEEE 802.11-DCF.
- *Sifs_size*: the SIFS size of IEEE 802.11-DCF.

Radio parameters:

In the *Radio Config.* section one may set the radio parameters used in the Eq1. and Eq2 to defined transmission radius of nodes (Figure 7). The “rx_sensibility” and “frequency_radio” parameters are defined in the specifications of radio devices. The other, are estimated based on the assumed simulated environment.

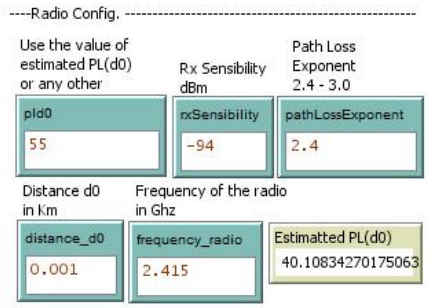

Figure 7. *Radio Configuration* panel.

Energy parameters:

In the *Energy Setup* section one may set the energy related parameters of nodes (Figure 8). In that section is possible to defined the initial energy capacity of nodes (in mAh), as well as the energy consumption (in mAh) for tx, rx, sleeping and processing tasks. Notice that this parameters may vary depending on the radio device configuration and the chosen TxPower.

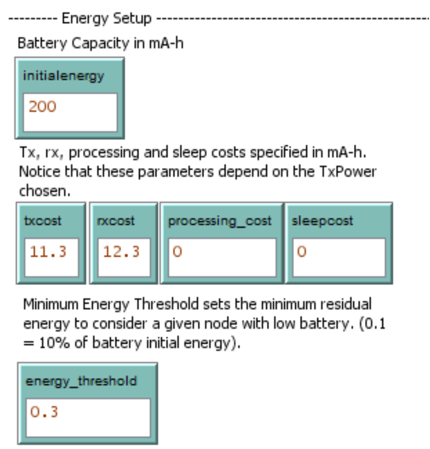

Figure 8. The *Energy Setup* panel.

Here is an example of power consumption for several device configurations described in the nRF24L01+ specification:

Tx_Power <> Tx Cost
0dBm <> 11.3mA
-6dBm <> 9.0mA
-12dBm <> 7.5mA
-18dBm <> 7.0mA

Bit_rate <> Rx Cost
2Mbps <> 13.5
1Mbps <> 13.1
250kbps <> 12.6

BER = 1.0E-3
Operation Frequency
2.4 GHz
Channel Bandwidth
1 MHz at 250 - 1000 kbps
2 Mhz at 2Mbps

Rx- Sensibility = -94
Radio Freq - 2.415

Max Frame Size - 41,125 bytes

The *energy_threshold* is the value used to define a node as a low battery state.

Simulation Control parameters

In the *Simulation Control* section the *max_tick* sets the maximum number of ticks used in the simulation (Figure 9). However, depending on the network configuration, some nodes may have messages in the buffer at the end of simulation. In this case, if the option *run_until_clear_buffers* set to "On" the simulation will continue until all tx buffers of nodes be empty. Notice that when changing the *max_tick* the *total_simulation*.

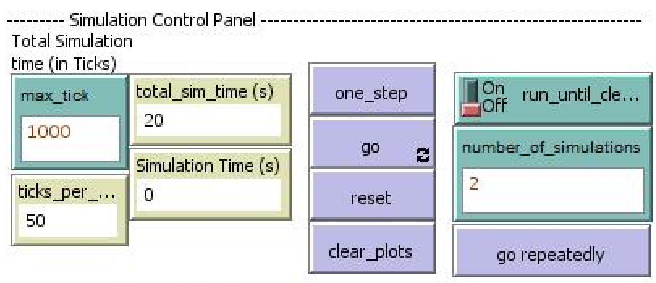

Figure 9. The *Simulation Control* panel.

A simulation is started by pressing the *go* button (or the *one-step* button for a step-by-step execution). Multiple simulations may be executed by setting the *number_of_simulation* and pressing the *go_repeatedly* button. During the simulation, output monitors and plots are updated with intermediate results.

Mobility Control parameters

In the *Mobility Control* section the *mobility_of_sink* option allows the sinks to walk through the world map. The mobility strategy is set by the *mobility_strategy* chooser. The model implements two mobility strategies: the random walk and the horizontal walk. By setting the *walk_on_streets* option "On" the sinks will only move over white (patches) lanes in the map. Sink nodes must be placed on the lanes otherwise they will get stuck.

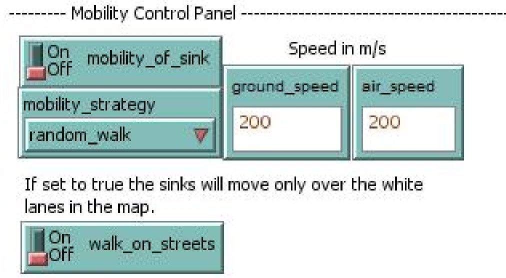

Figure 10. The *Mobility Control* panel.

The speed of nodes can be set (in m/s) in the *ground_speed* and *air_speed* inputs.

Faults

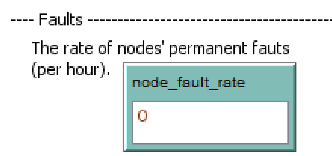

Figure 11. The Faults panel.

By setting the *node_fault_rate* one may indicate the rate of permanent faults of nodes per hour. In this case whenever a node becomes faulty it will stop to work and lose all links with other nodes. Messages in the buffers of such nodes will be lost also.

Outputs

This model presents several output information to help the users to understand their simulation results. Therefore, such outputs are organized in three main groups with their distinct set of metrics: link layer outputs, application/transport/network outputs and energy outputs, see figures 12, 13 and 14.

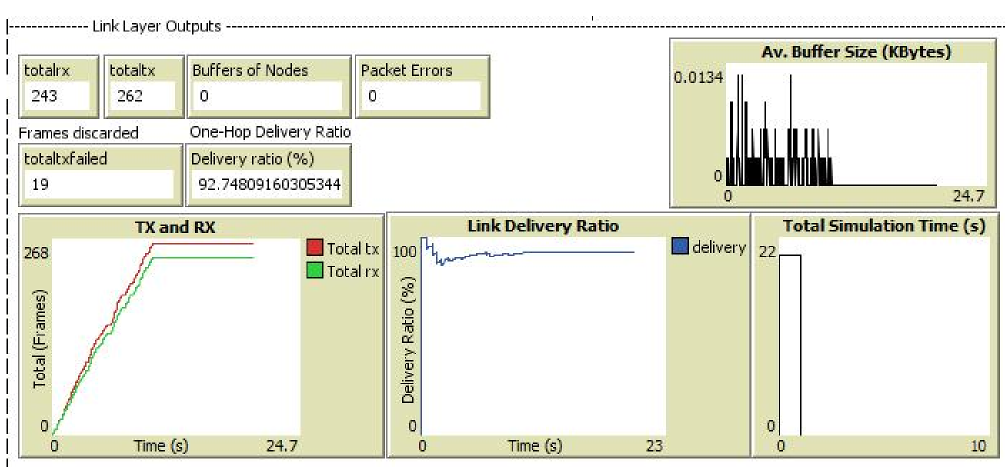

Figure 12. The *Link Layer* outputs.

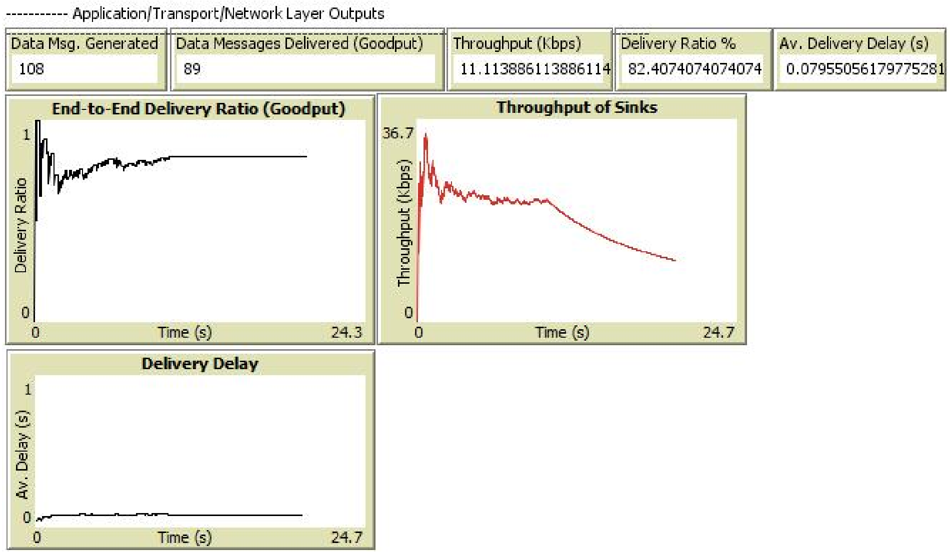

Figure 13. The App./Transport/*Network Layer* outputs.

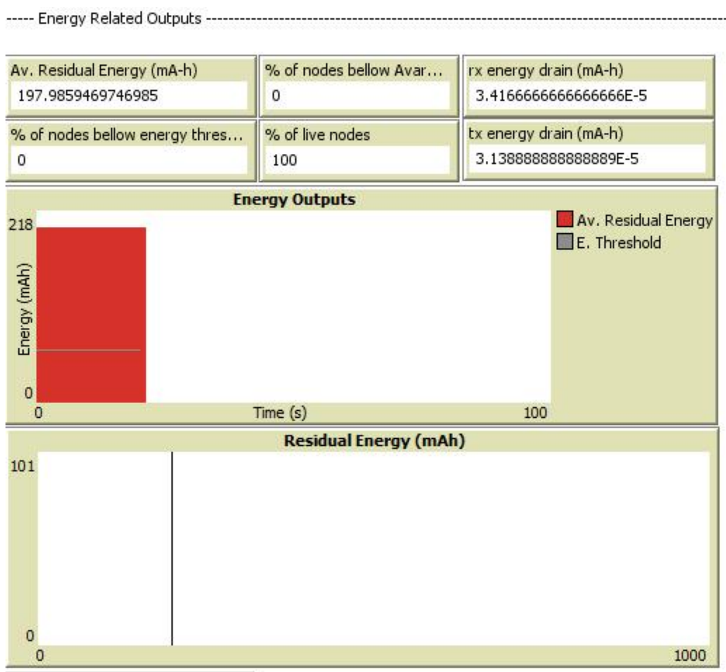

Figure 14. The *Energy* outputs.

THINGS TO NOTICE

One may run multiple simulations with the same parameters by using the button "go repeatedly". The outputs will be plot in sequence for every execution. One may execute the *random-seed <number>* command, in the command center of NETLOGO, to set the initial seed to use the same sequence of seeds in the simulations.

Multiple simulations with distinct parameters may be executed by using the Behaviorspace tool of the NetLogo environment.

THINGS TO TRY

The primarily focus of this model is to provide a way to simulate large-scale networks with minimum setup time. Therefore, one may test this model with high number of nodes in the network and verify how fast it goes in comparison to other simulation environments.

EXTENDING THE MODEL

The model is not exhaustive, there several points to be improved.

CREDITS AND REFERENCES

This model, and further improvements can be found in the author's site:
https://sites.google.com/site/valeriorosset/wsimulnet

Author: Valério Rosset (PhD), Associate Professor @ Federal University of São Paulo, Brazil.
Contact: vrosset@unifesp.br

Version: v.6.26 - Basic (08/2018)
Netlogo Version: 6.0.2