User Manual

Simulator of Distributed Algorithms

**1. Introduction**

The Distributed Network Simulator is a Python-based tool designed to help users simulate, visualize, and analyze distributed algorithms on various network topologies. With a user-friendly graphical interface, you can:

* Configure network parameters (number of nodes, topology, etc.)
* Upload and test your own distributed algorithms
* Visualize the network and simulation process
* Analyze simulation results and statistics

**2. Installation and Requirements**

**Prerequisites**

* Python 3.7 or higher (recommended)
* pip (Python package manager)

**Install Required Packages**

From your project directory, run:

This will install:

* PyQt5 (for the GUI)
* networkx (for network graph operations)
* numpy (for numerical operations)

**3. How to Launch the Simulator**

To start the Distributed Network Simulator:

Open a terminal and navigate to your project directory:

Run the simulator:

This will launch GUI for configuring and running simulations

Or:

Choose the in your IDE and run it.

**4. Using the GUI to Configure and Run a Simulation**

When you launch the simulator, a graphical interface will appear. Here’s how to use it step by step:

**A. Main Menu Overview**

The main window allows you to configure all simulation parameters before running.

* You will see options for:
* Number of Computers
* Topology (preset options or custom)
* ID Type
* Delay
* Display (Text or Graph)
* Root selection
* Logging level
* Algorithm file upload
* Topology file upload

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

**B. Step-by-Step Configuration**

1. **Set the Number of Computers**

* Enter the desired number in the "Number of Computers" field.

1. **Choose Topology**

* Select a preset topology (Random, Clique, Line, Tree, Star) from the dropdown.
* Or select "Custom" and upload your own topology file (.txt).

1. **Set ID Type**

* Choose how node IDs are assigned: Random, Sequential, or Custom.

1. **Set Delay**

* Choose the message delay model: Constant, Random Constant, or Random.

1. **Choose Display Type**

* "Text": Outputs results in text format.
* "Graph": Opens a visualization window (recommended for smaller networks, up to 100 nodes).

1. **Select Root**

* Choose how the root node is selected: No Root, Min ID, Random, or Custom.

1. **Set Logging Level**

* Choose the amount of detail in logs: Short, Medium, or Long.

1. **Upload Algorithm File**

* Click "Upload Python File" and select your algorithm script (.py).
* The script must define  and  functions (see next steps for details).

1. **Upload Topology File (if needed)**

* Click "Upload Topology File" and select your topology file (.txt).
* Required if you selected "Custom" for Topology, Root, or ID Type.
* Note that if you uploaded topology file the following will be set on custom:
  1. Topology
  2. ID Type
  3. Root
  4. Number of computers??

1. **Submit**

* Click "Submit" to save your configuration and start the simulation.

**C. Notes**

* If you select "Graph" display and have more than 500 computers, you will be prompted to switch to "Text" output for performance reasons.
* All fields must be filled before submitting.
* Errors or missing fields will be highlighted by the GUI.

**5. File Formats: Algorithm and Topology Files**

To run custom simulations, you may need to provide your own algorithm and/or topology files. Here’s how to format them:

**A. Algorithm File (.py)**

Your algorithm file must define two functions:

)

**Example:**

*import* simulator.computer *as* computer

*from* simulator.communication *import* Communication

*from* simulator.config *import* NodeState

*from* utils.logger\_config *import* logger

def mainAlgorithm(*self*: computer.Computer, *communication*: Communication, *\_arrival\_time*, *message*: str = None):

*if* self.state != NodeState.TERMINATED:

        communication.send\_to\_all(self.id, "running a broadcast", \_arrival\_time)

        self.color = "#7427e9"

        self.state = NodeState.TERMINATED

def init(*self*: computer.Computer, *communication*: Communication):

*if* self.is\_root:

        logger.info("%s is root", self.id)

        communication.send\_to\_all(self.id, "running a broadcast")

        self.color = "#000000"

        self.state = NodeState.TERMINATED

**B. Topology File (.txt)**

A topology file describes the network structure. It must follow this format:

* **IDs List:** Comma-separated node IDs.
* **Number of Computers:** Total number of nodes.
* **Root ID:** The root node (if applicable).
* **Edges:** List of edges as (source,target) pairs.
* **Input:** (Optional) Initial input values for nodes.

You can find more examples in the  directory.

**Example:**

IDs List:

1,2,3,4,5,6,7,8,9,10

Number of Computers:

10

Root ID:

1

Edges:

(1,2),(1,3),(2,4),(2,5),(3,6),(3,7),(4,8),(4,9),(5,10)

Input:

[height,weight]

1:[11]

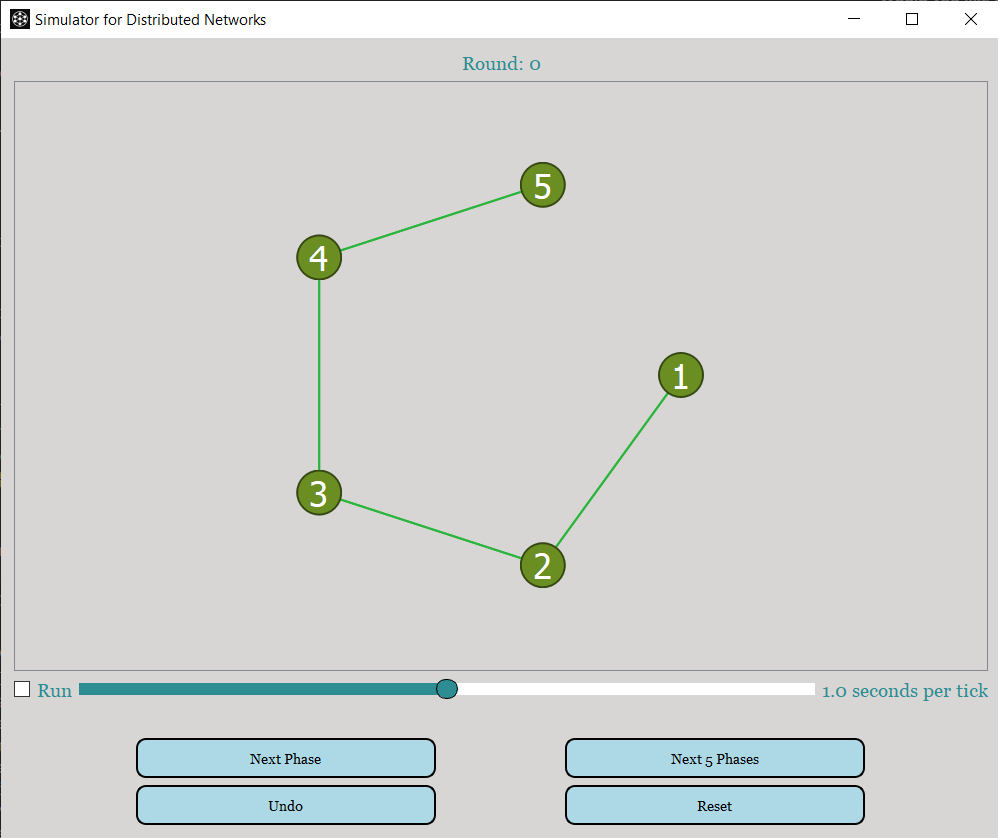
**6. Running and Interpreting Simulation Results**

After you configure your simulation and click "Submit," the simulator will execute your chosen algorithm on the specified network.

Here’s what to expect and how to interpret the results:

**A. During the Simulation**

* **Progress Window:**

If you selected "Graph" as the display type, a visualization window will show the network and the state of each node as the simulation progresses.

Explanation of the buttons:

* 1. Next phase
  2. Next 5 phases
  3. Undo
  4. Reset
* **Console/Log Output:**

The simulator logs important events, errors, and statistics to the console and to output.txt in your project directory.

2025-06-12 15:37:47,131 - INFO - Starting the simulator

2025-06-12 15:42:54,641 - INFO - file name is pathtoproject /Project/topologyFiles/line.txt

2025-06-12 16:04:21,205 - INFO - Topology file pathtoproject/topologyFiles/line.txt parsed successfully.

2025-06-12 16:04:21,208 - INFO - Computer 1 is changing is\_root to False

2025-06-12 16:04:21,208 - INFO - Computer 1 is changing color to olivedrab

2025-06-12 16:04:21,208 - INFO - Computer 1 is changing inputs to {}

2025-06-12 16:04:21,208 - INFO - Computer 1 is changing outputs to {}

2025-06-12 16:04:21,208 - INFO - Computer 2 is changing is\_root to False

2025-06-12 16:04:21,208 - INFO - Computer 2 is changing color to olivedrab

2025-06-12 16:04:21,208 - INFO - Computer 2 is changing inputs to {}

2025-06-12 16:04:21,208 - INFO - Computer 2 is changing outputs to {}

2025-06-12 16:04:21,208 - INFO - Computer 3 is changing is\_root to False

2025-06-12 16:04:21,208 - INFO - Computer 3 is changing color to olivedrab

2025-06-12 16:04:21,209 - INFO - Computer 3 is changing inputs to {}

2025-06-12 16:04:21,209 - INFO - Computer 3 is changing outputs to {}

2025-06-12 16:04:21,209 - INFO - Computer 4 is changing is\_root to False

2025-06-12 16:04:21,209 - INFO - Computer 4 is changing color to olivedrab

2025-06-12 16:04:21,209 - INFO - Computer 4 is changing inputs to {}

2025-06-12 16:04:21,209 - INFO - Computer 4 is changing outputs to {}

2025-06-12 16:04:21,209 - INFO - Computer 5 is changing is\_root to False

2025-06-12 16:04:21,209 - INFO - Computer 5 is changing color to olivedrab

2025-06-12 16:04:21,209 - INFO - Computer 5 is changing inputs to {}

2025-06-12 16:04:21,209 - INFO - Computer 5 is changing outputs to {}

2025-06-12 16:04:21,209 - INFO - Computer 3 is changing is\_root to True

2025-06-12 16:04:21,210 - INFO - ID 1 has input height with value 11

2025-06-12 16:04:21,568 - INFO - Computer 1 is changing distance to inf

2025-06-12 16:04:21,568 - INFO - Computer 2 is changing distance to inf

2025-06-12 16:04:21,568 - INFO - Computer 3 is changing distance to 0

2025-06-12 16:04:21,568 - INFO - Computer 3 is changing parent to 3

2025-06-12 16:04:21,569 - INFO - Computer 3 is changing color to blue

2025-06-12 16:04:21,569 - INFO - Computer 4 is changing distance to inf

2025-06-12 16:04:21,569 - INFO - Computer 5 is changing distance to inf

2025-06-12 16:04:21,569 - INFO - \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2025-06-12 16:04:21,569 - INFO - Initialization phase completed, starting synchronous rounds

2025-06-12 16:04:21,569 - INFO - Current round: 0

2025-06-12 16:04:21,570 - INFO - Computer 3 is changing state to NodeState.TERMINATED

2025-06-12 16:04:21,570 - INFO - Computer 3 is changing state from active to terminated

2025-06-12 16:04:21,570 - INFO - Current round: 1

2025-06-12 16:04:21,570 - INFO - Computer 2 is changing parent to 3

2025-06-12 16:04:21,570 - INFO - Computer 2 is changing distance to 1.0

2025-06-12 16:04:21,570 - INFO - Computer 2 is changing color to red

2025-06-12 16:04:21,570 - INFO - Computer 2 is changing state to NodeState.TERMINATED

2025-06-12 16:04:21,571 - INFO - Computer 2 is changing state from active to terminated

2025-06-12 16:04:21,571 - INFO - Computer 4 is changing parent to 3

2025-06-12 16:04:21,571 - INFO - Computer 4 is changing distance to 1.0

2025-06-12 16:04:21,571 - INFO - Computer 4 is changing color to red

2025-06-12 16:04:21,571 - INFO - Computer 4 is changing state to NodeState.TERMINATED

2025-06-12 16:04:21,571 - INFO - Computer 4 is changing state from active to terminated

2025-06-12 16:04:21,571 - INFO - Current round: 2

2025-06-12 16:04:21,571 - INFO - Computer 1 is changing parent to 2

2025-06-12 16:04:21,571 - INFO - Computer 1 is changing distance to 2.0

2025-06-12 16:04:21,571 - INFO - Computer 1 is changing color to green

2025-06-12 16:04:21,572 - INFO - Computer 1 is changing state to NodeState.TERMINATED

2025-06-12 16:04:21,572 - INFO - Computer 1 is changing state from active to terminated

2025-06-12 16:04:21,572 - INFO - Computer 5 is changing parent to 4

2025-06-12 16:04:21,572 - INFO - Computer 5 is changing distance to 2.0

2025-06-12 16:04:21,572 - INFO - Computer 5 is changing color to green

2025-06-12 16:04:21,572 - INFO - Computer 5 is changing state to NodeState.TERMINATED

2025-06-12 16:04:21,572 - INFO - Computer 5 is changing state from active to terminated

2025-06-12 16:04:21,572 - INFO - Current round: 3

2025-06-12 16:04:21,572 - INFO - sync run completed

2025-06-12 16:04:21,572 - INFO - \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

2025-06-12 16:04:21,573 - INFO - Network Statistics:

2025-06-12 16:04:21,573 - INFO - Total number of computers: 5

2025-06-12 16:04:21,573 - INFO - Total number of messages sent: 4

2025-06-12 16:04:21,573 - INFO - Total number of messages received: 4

2025-06-12 16:04:21,573 - INFO - No nodes collapsed during the simulation.

**B. After the Simulation**

* **Results Location:**
* **Text Output:**

Results and statistics are printed in the terminal and saved in output.txt.

* **Graph Output:**

The visualization window will display the final state of the network. You can close the window when done.

* **What’s Included in the Output:**
* Total number of computers
* Total messages sent and received
* Collapse and reorder statistics (if applicable)
* Outputs from each node (e.g., final values, states)
* Timing information (total simulation time, network creation time, algorithm run time)

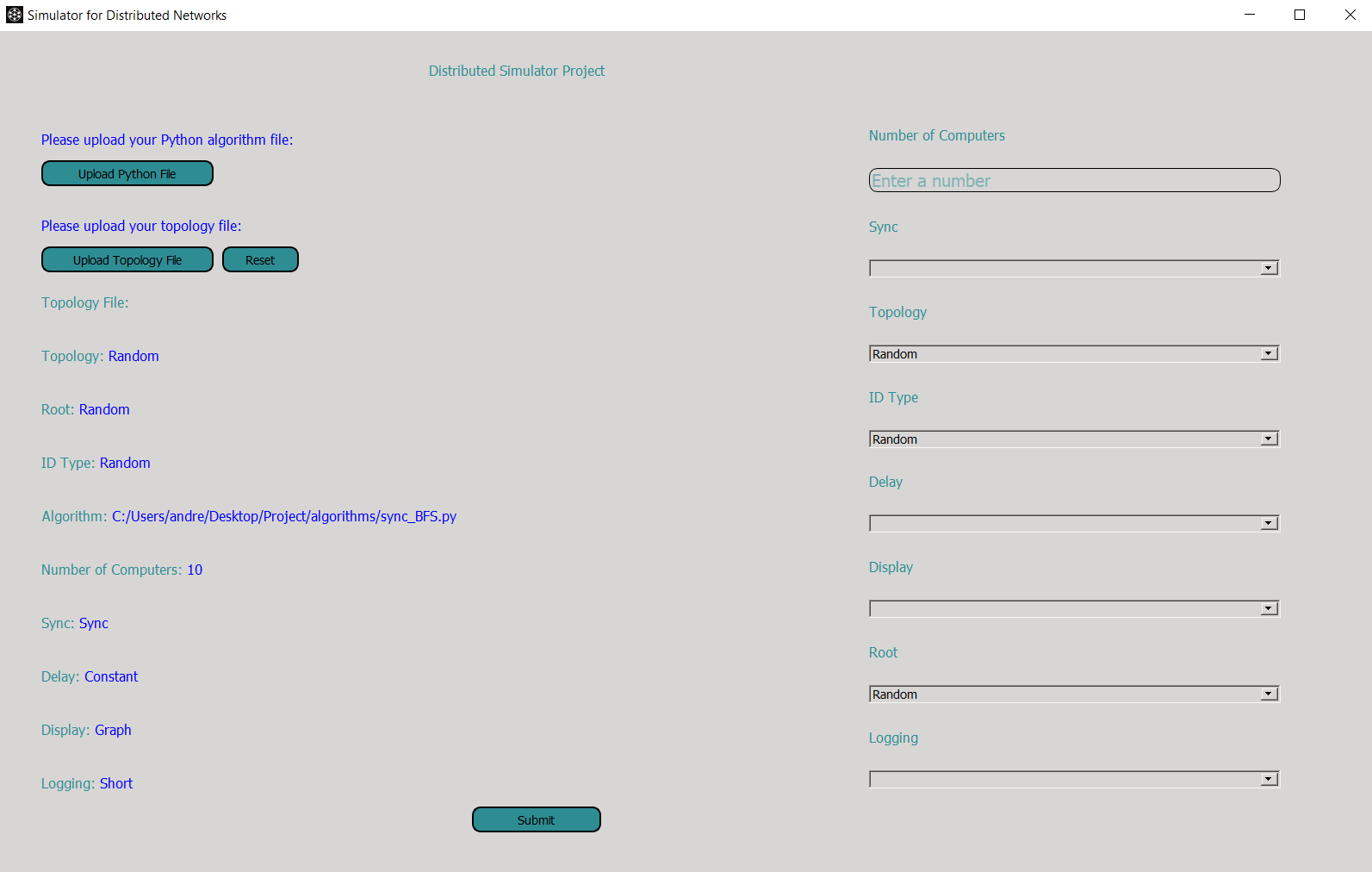
**C. Troubleshooting**

* **Missing or Incorrect Output:**
* Ensure all required fields are filled in the GUI.
* Check that your algorithm and topology files are correctly formatted.
* Review output.txt and the terminal for error messages.
* **Simulation Crashes or Freezes:**
* For very large networks, use "Text" display instead of "Graph."
* Check for errors in your custom algorithm code.

**FAQ: What are the blue settings on the left side of the main menu?**

**Question:**

What do the blue settings on the left represent (marked in a red rectangle)?



**Answer:** The blue settings on the leftside of the main menu are the saved configuration values from your  previous run of the simulator. When you launch the simulator, it automatically loads these settings as the default options for your next session. This means you do not have to reselect all the dropdown options or re-enter values each time you start the simulator—your last-used configuration is pre-filled for your convenience.

* **Where are these settings stored?**

They are saved in the  file in your project directory.

* **How does this help?**

This feature streamlines repeated experiments and makes it easy to tweak only the parameters you want to change, while keeping the rest of your previous setup intact.

* **Can I change them?**

Yes! You can modify any setting using the right-side controls. When you click "Submit," your new configuration will be saved and shown in blue the next time you open the simulator.