BRITE to OMNeT++ *.ned

Input: Directory that contains directories of topologies, this should include in the directory where the program is located. Take the file *RyuFileEdges.txt* of each topology to know the number of links and which nodes make up each link. It also takes the file *RyuFileNodes.txt* of each topology to know the position of each node on the plane where the red will be simulated. If the name of the files is modified, the constant strings that contain it are modified (line 445 and 452 inside the function int Parser char *, char *););

Output: It generates a directory called *"Omnetpp_workspace"*, which will be our workspace during the simulations of the different topologies. Within this directory, it generates different directories one for each topology parsed with the name of this one. Within each directory of the different topologies, it generates three files necessary to launch the simulation.

- *omnetpp.ini*
- *nodo.cc* (Generic, does not provide any functionality, the functionality will be implemented by the Amaru protocol.)
- *configTopo.ned* (Fundamental file of the parsing, it will reflect the data to build the body of the parsed topology, number of nodes, latency, binary regime, links ...)

In addition, it generates three additional additional files, so that, when importing the different topologies in the workspace, recognize the directory of each topology as a project (.cproject, .oppbuildspec, .project), the first two are constant for all the projects so a copy is included in the directory of the program and they are copied to the different topology directories, the last one is generated with the program because it varies (minimally) depending on the topology.

Operation: The program asks for the name of the root directory of the different topologies, by default it will use *"Topologias_Brite"*. When it gets the name of the root directory, it sweeps to count all available directories within the root directory.

Knowing already the number of topologies that it will have to parse, it will make a dynamic memory reserve for a two-dimensional array that will store the names of the different directories that contain topologies.

Knowing the different routes to each file *RyuFileEdges.txt* and *RyuFileNodes.txt* of each topology, it starts parsing, making use of the following functions:

- `int Parser(char *, char *);`

Main function of the parsing, calls secondary functions to generate the files * .ini, * .ned, * .cc, and the project files (.cproject, .oppbuildspec, .project).

Secondary functions of *`int Parser():`*

- `void GenOmnetppIni (char *,char *);`

  Generates the * .ini file for the subsequent simulation.

- `void GenNodeCC (char *, char *);`

  Generates the * .cc file for the subsequent simulation.

- `void GenConfigNet (char *, char *,FILE *,FILE *);`

  It generates the * .ned file, which will describe the topology, it is supported by three functions to extract and write data from the different parts of the * .ned file, header, number of nodes and positioning in the plane, number of links and characteristics of these.

- `void GenFileSim (char *);`

  Generates the necessary files for OMNeT ++ to detect that it is a project.