# BRITE to OMNeT++ *.ned

Input: Directorio que albergue directorios a su vez de topologías, este debe incluirse en el directorio donde se encuentre él programa. Toma el archivo *RyuFileEdges.txt* de cada topología para conocer el número de enlaces y que nodos conforman cada enlace. También toma el archivo *RyuFileNodes.txt* de cada topología para conocer la posición de cada nodo sobre el plano donde se simulará la red. Si se modifica el nombre de los ficheros habría que modificar a su vez las strings constantes que lo contienen (línea 445 y 452 dentro de la función `int Parser char *, char *);` ).

Output: Genera un directorio llamado "Omnetpp_workspace", el cual será nuestro espacio de trabajo durante las simulaciones de las distintas topologías. Dentro de este directorio, genera distintos directorios uno por cada topología parseada con el nombre de esta. Dentro de cada directorio de las distintas topologías, genera tres archivos necesarios para lanzar la simulación:

- omnetpp.ini
- nodo.cc (Genérico, no aporta funcionalidad alguna. La funcionalidad será implementada por el protocolo Amaru.)
- `configTopo.ned` (Archivo fundamental del parseado. En él se reflejarán los datos para construir el esqueleto de la topología parseada, número de nodos, latencia, régimen binario, enlaces…)

Además, genera tres archivos adicionales más, para que, a la hora de importar las distintas topologías en el espacio de trabajo, reconozca el directorio de cada topología como un proyecto (.cproject, .oppbuildspec, .project), los dos primeros son contantes para todos los proyectos por lo que se incluye una copia en directorio del programa y se van copiando a los distintos directorios de topologías, el ultimo se genera con el programa ya que varía(mínimamente) en función de la topología.

Funcionamiento: El programa pregunta por el nombre del directorio raíz de las distintas topologías, por defecto usará *"Topologias_Brite"*. Cuando obtiene el nombre del directorio raíz, hace un barrido para contar todos los directorios disponibles dentro del directorio raíz.

Sabiendo ya al número de topologías que va a tener que parsear, hará una reserva dinámica de memoria para un array bidimensional que guardará los nombres de los distintos directorios que albergan topologías.

Sabiendo ya las distintas rutas a cada archivo *RyuFileEdges.txt* y *RyuFileNodes.txt* de cada topología, se empieza a parsear, haciendo uso de las funciones:

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

Función principal del parseado, hace llamadas funciones secundarias para generar los archivos *.ini, *.ned, *.cc, y los archivos de proyecto (.cproject, .oppbuildspec, .project).

Funciones secundarias de *int Parser()* :

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

Genera el archivo *.ini para la posterior simulación.

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

Genera el archivo *.cc para la posterior simulación.

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

Genera el archivo *.ned, que describirá la topología, se apoya en tres funciones para extraer y escribir datos de las distintas partes del archivo *.ned, cabecera, número de nodos y posicionamiento en el plano, numero de enlaces y características de estos.

- `void GenFileSim (char *);`

Genera los archivos necesarios para que OMNeT++ detecte que se trata de un proyecto.