

**MÁSTER UNIVERSITARIO EN
INGENIERÍA INDUSTRIAL**

TRABAJO FIN DE MÁSTER

***ANÁLISIS DE LOADNG COMO PROTOCOLO
DE RUTADO DINÁMICO PARA LAS
COMUNICACIONES PLC EN MT***

**DOCUMENTO N°3: MANUAL DE
USUARIO**

Alumno *Rodríguez Fernández, Jorge*
Director *de la Vega Moreno, David*
Departamento *Ing. de Comunicaciones*
Curso académico *2017/2018*

Bilbao, a 6 de Febrero de 2018

Índice

1	Anexo I: Manual de usuario	3
1.1	Introducción	3
1.2	Instalación	3
1.3	Archivos .ned.....	6
1.4	Omnetpp.ini	8
1.4.1.1	Archivos .xml.....	9
1.5	Simulación	10
1.6	Obtención de datos	15

Índice de ilustraciones

Ilustración 21 Importación	4
Ilustración 22 Selección proyecto.....	5
Ilustración 23 Estructura de carpetas	6
Ilustración 24 Conexión.....	7
Ilustración 25 Definición del canal.....	7
Ilustración 26 Omnet.ini.....	8
Ilustración 27 eventos.xml	10
Ilustración 28 Selección simulación	11
Ilustración 29 Simulación	12
Ilustración 30 Controles simulación.....	12
Ilustración 31 Consola	13
Ilustración 32 Parámetros	14
Ilustración 33 Interior router.....	15
Ilustración 34 Datos obtenidos.....	16
Ilustración 35 Gráfico de resultados	17
Ilustración 36 Exportación de datos	18

1 Anexo I: Manual de usuario

1.1 Introducción

En este documento se recoge la explicación del tanto del proceso de instalación de la herramienta desarrollada como el funcionamiento de la misma para realizar simulaciones con ella.

1.2 Instalación

Una vez instalada la versión OMNeT++ 5.0 basta con dirigirse a *File > Import* y dentro a *Existing Projects into Workspace*, y seleccionar la carpeta adjuntada a este documento llamada *routing* que contiene el software para realizar las simulaciones para LOADng

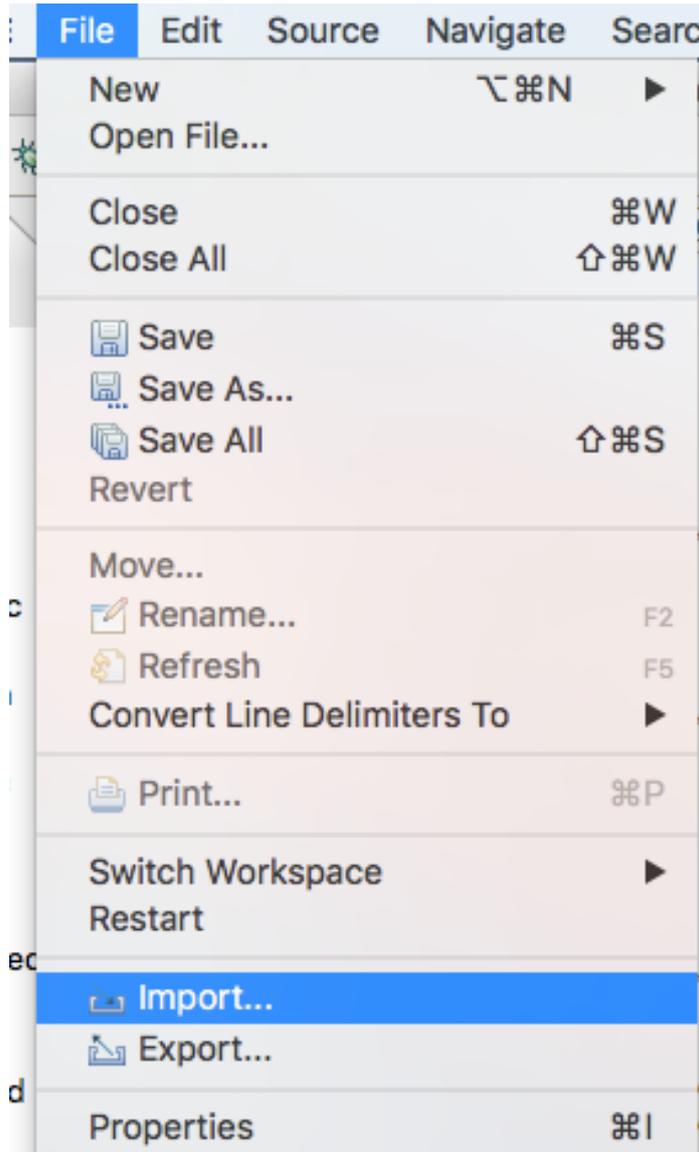


Ilustración 21 Importación

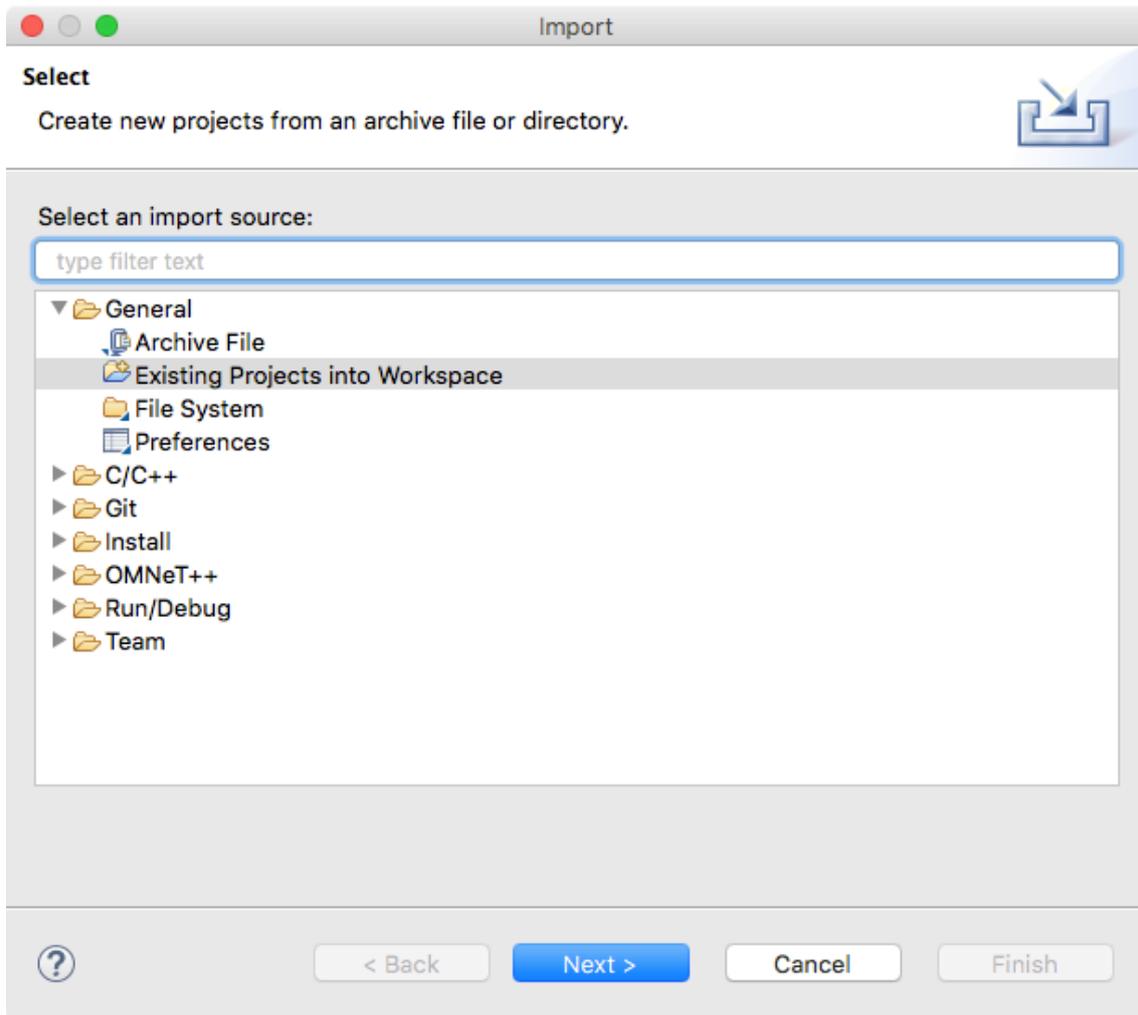


Ilustración 22 Selección proyecto

Una vez importado el proyecto se mostrará en la parte izquierda del programa en el *Project Explorer*, con lo que finaliza el proceso de instalación.

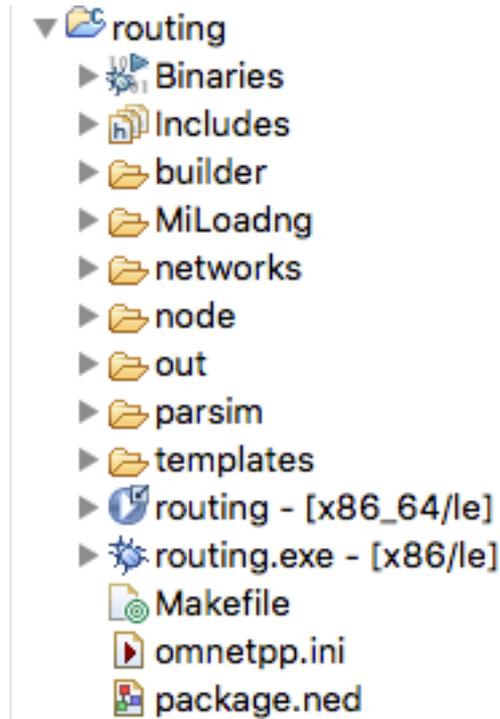


Ilustración 23 Estructura de carpetas

1.3 Archivos .ned

Dentro de la carpeta MiLoadng se encuentran todas las topologías creadas para la realización de este documento. Para la creación de una nueva topología es recomendable utilizar una de las ya creadas y añadir o eliminar conexiones o routers hasta obtener la topología deseada. Basta con copiar y pegar los routers para añadir más o suprimir para borrar. Aquí también se puede modificar la estética de los elementos haciendo clic derecho sobre ellos y accediendo a sus propiedades.

Para la realización de las conexiones solo hay que seleccionar en la paleta de elementos que aparece a la derecha la conexión en la que entre paréntesis aparezca (MiLoadng) e ir uniendo los routers de la manera deseada.

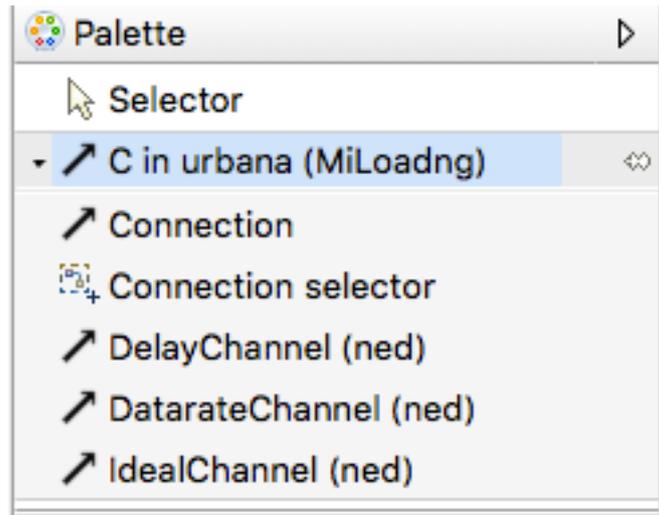


Ilustración 24 Conexión

Esta conexión es bidireccional y tiene de manera predefinida un delay de 1ms, un datarate de 1Mbps y no está marcado como enlace débil. Si se quieren cambiar estos valores hay que dirigirse al source de la topología y modificar la parte que corresponde a la definición del canal.

```

channel C extends DatarateChannel
{
    delay = 1ms;
    datarate = 1Mbps;
    bool weak = default(false);
}
  
```

Ilustración 25 Definición del canal

Tras crear la geometría de la red, no hay que olvidar añadir el módulo ScenarioManager que será el encargado de leer el documento .xml donde más adelante se añadirán los eventos que sucederán durante la simulación y que obligarán a recalcular las rutas al protocolo.

Una vez creada la topología a simular el siguiente paso es el de establecer los criterios de la simulación, para ello hay que dirigirse al archivo

omnetpp.ini, que se encuentra en *Miloadng*, ahí es donde se configuran las simulaciones.

1.4 Omnetpp.ini

Siguiendo los comentarios propios de este documento, basta con rellenar el documento según los comentarios allí mostrados.

```

[General]
#network = topo1
sim-time-limit = 1000s #tiempo de duracion de la simulacion
#repeat = 10 #numero de simulaciones a realizar

#####definicion de los parametros de la aplicacion que genera paquetes#####
**.appType = "App" #nombre de la aplicacion
**.app.packetLength = 1024 bytes #tamaño de los paquetes en bytes
**.destAddresses = "0 1 2 3 4 5 6 7 8 9" #destinos de los paquetes separados por espacios
**.sendIaTime = exponential(1000000s) # intervalo de generación de los paquetes (distribucion exponencial)

#####definicion de los parametros del protocolo LOADng#####
**.routing.timeStore = 300000s #tiempo que las rutas son validas (valor predefinido)
**.routing.timeToLive = 100000s #tiempo que espera el nodo por el RREP (valor predefinido)
**.routing.cleanretry = 100000s #tiempo tras que el se reinician los reintentos (valor predefinido)
**.routing.retry = 3000000 #numero de reintentos hasta tomar una ruta como inalcanzable (valor predefinido)
**.routing.criterio = 0 #criterio para seleccionar la mejor ruta (0 valor predefinido)
# 0 ---> Menor numero de saltos
# 1 ---> Menor delay (el que primero llega)
# 2 ---> Evitar enlaces debiles

#####definicion de los eventos#####
*.scenarioManager.script = xmldoc("eventos.xml") #apunta al archivo .xml que contiene los eventos añadir eventos en el

#si se desea modificar un nodo en concreto en vez de todos por igual
#sustituir los ** por el nombre del nodo
#Ej: Loadngnet.Router[2].routing.timeStore = 10s solo en nodo 2 tendra el timeStore en 10s

#####colas#####
**.queue[*].frameCapacity = 0 #tamaño de la cola, (0 sin limite)

#####obtencion de datos#####
**.hopCount.result-recording-modes = +histogram
**.Router[0..16].routing.hopCount.result-recording-modes = -vector
  
```

Ilustración 26 Omnet.ini

Lo primero es seleccionar la topología a simular mediante el apartado *network* junto con el tiempo que durará la simulación con *sim-time-limit* (se puede omitir este apartado si se quiere simular sin límite de tiempo).

El siguiente paso es el de caracterizar el submódulo *App*, el encargado de la creación de paquetes, hay que determinar el tamaño de los paquetes, sus

posibles destinos y el intervalo de creación de ellos. Hacer esto según los comentarios ahí añadidos.

Ahora toca configurar los parámetros del protocolo LOADng, dado que para la red de media tensión hemos supuesto que la rutas serán válidas hasta que se detecte una desconexión, los valores aquí puestos son muy altos para evitar el borrado de rutas por expiración pero sería posible modificar estos datos para una simulación más semejante al protocolo real. Es aquí también donde se selecciona el criterio para la creación de la ruta óptima ya sea un menor número de saltos, menor delay o evitar enlaces débiles.

El último apartado a cumplimentar sería el del `scenarioManager.script`, ahí se ha señalar al documento `.xml` donde estarían los eventos que sucederán en la red y que se encargará de ejecutar el módulo `ScenarioManager`.

1.4.1.1 Archivos .xml

Aquí es donde se configuran los eventos que sucederán durante la simulación. En `MiLoadng` se encuentra el documento `eventos.xml` donde se especifican los tipos de eventos que se pueden simular y la forma de hacerlo.

<!--

etiquetas disponibles para relizar eventos:

iniciar con <scenarioy finalizar con </scenarioy

instante del evento definido con <at t="2"> y finalizado con </at> siendo 2 el instante en segundos

Posibles eventos:

1)modificar parametro de un nodo

ej: <set-param module="Router[0].routing" par="cleanretry" value="3s"/>

el nodo 0 pasa a tener en cleanretry el valor de 3s

par posibles: timeStore (s); timetolive (s); cleanretry (s); retry;

2)desconectar o conectar conexion de un nodo

ej: <set-channel-attr src-module="Router[0]" src-gate="port\$0[0]" attr="disabled" value="true"/>

<set-channel-attr src-module="Router[1]" src-gate="port\$0[0]" attr="disabled" value="true"/>

el gate 0 del nodo 0 pasa a estar desconectado y el gate 0 del nodo 1 tambien para desconectar en los 2 sentidos en la simulacion se veran en color rojo las conexiones que no funcionen (consultar en simulacion los gates)

3)modificar parametro de una conexion

ej: <set-channel-attr src-module="Router[0]" src-gate="port\$0[0]" attr="datarate" value="2Mbps"/>

la conexion del gate 0 del nodo 0 pasa a tener un datarate de 2Mbps

ej: <set-channel-attr src-module="Router[0]" src-gate="port\$0[0]" attr="weak" value="true"/>

la conexion del gate 0 del nodo 0 pasa a ser un enlace debil

attr posibles: datarate (Mbps); delay (s); weak (bool);

-->

Ilustración 27 eventos.xml

Al igual que sucedía con las topologías, lo más sencillo es utilizar uno de los documentos .xml ya creados y modificarlo para realizar la simulación deseada. Una vez creado, recordar rellenar el apartado `scenarioManager.script` en `omnetpp.ini` con el nombre del .xml para utilizarlo en la simulación.

1.5 Simulación

Ahora que tanto la topología como la configuración de simulación ya están listas, ya se puede proceder a su ejecución, para ello hay que acceder a la pestaña `run > run configuration`. Aquí tendremos que seleccionar la configuración de la simulación antes creada y proceder a su ejecución.

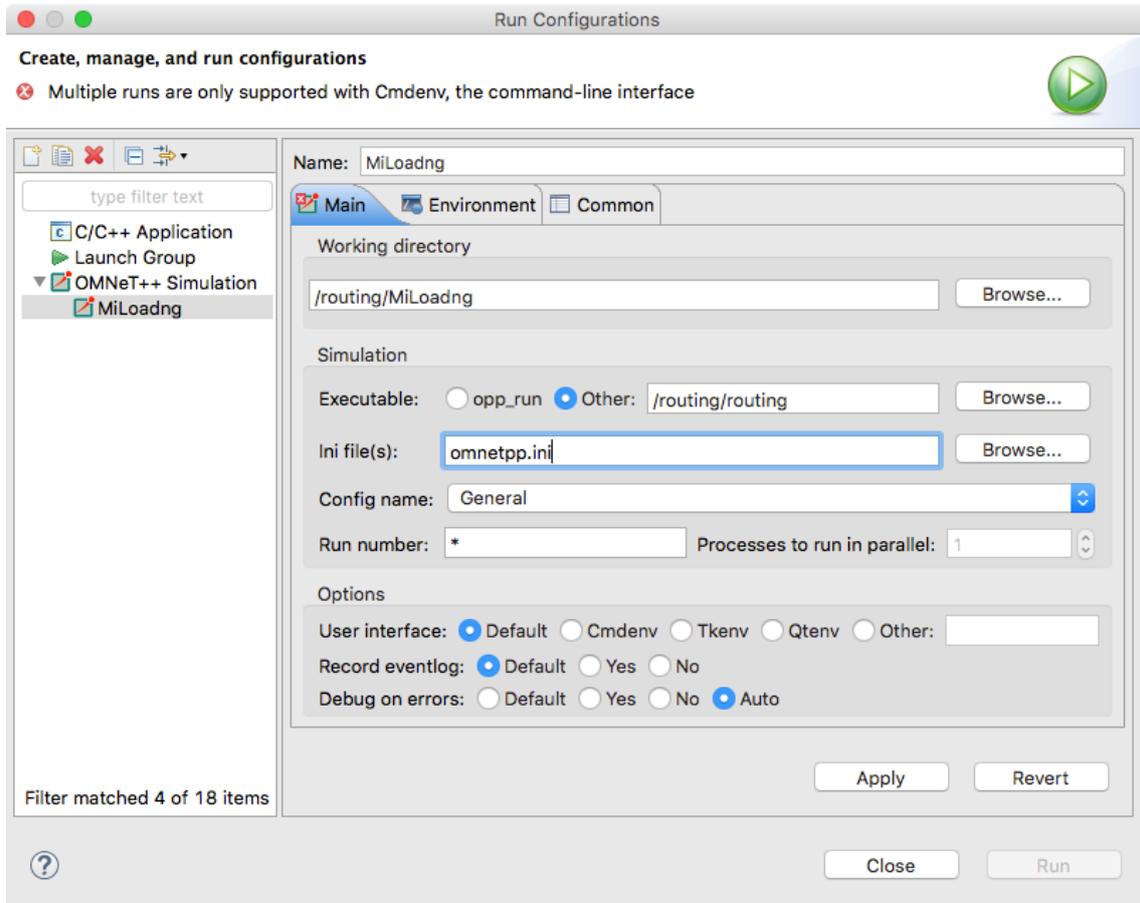


Ilustración 28 Selección simulación

Se mostrará entonces la ventana donde se realizará la simulación

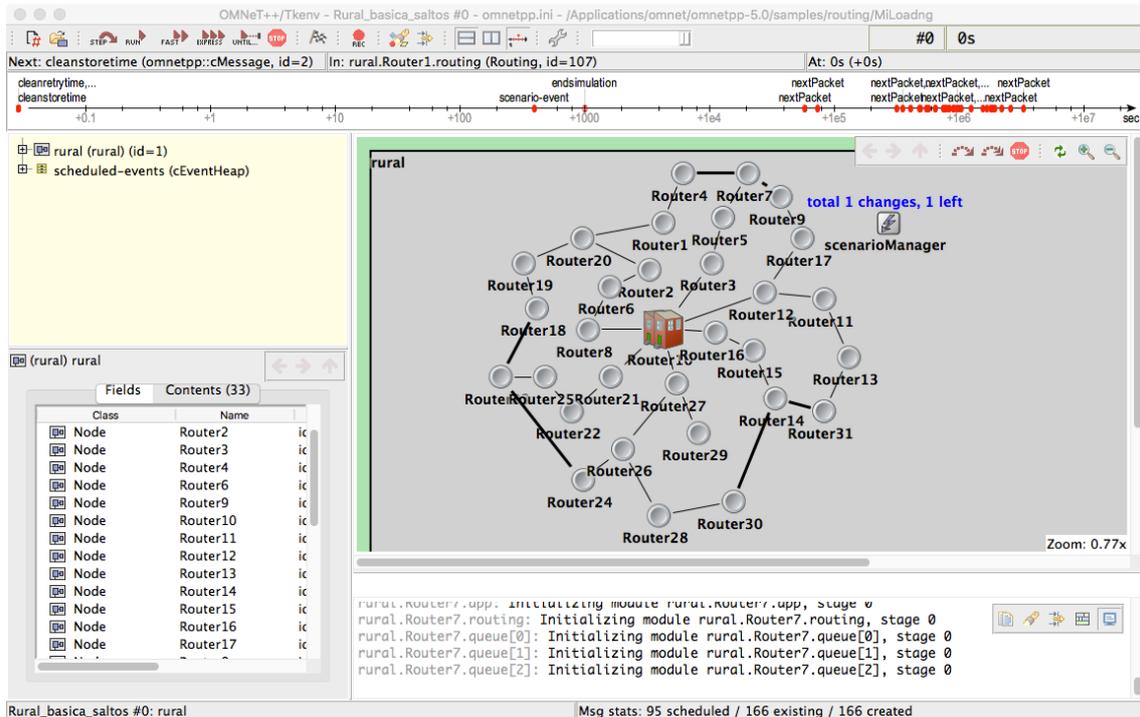


Ilustración 29 Simulación

En la parte superior se encuentran los controles de la simulación como son simulación paso a paso, run, simulación rápida, simular hasta determinado evento o instante y stop.



Ilustración 30 Controles simulación

Durante la simulación se puede ir siguiendo el desplazamiento de los paquetes de forma visual siempre y cuando la simulación sea sencilla, el seguimiento visual se va dificultando con la complejidad de la simulación. Si la simulación es muy complicada puede resultar interesante realizarla sin la ejecución visual.

En la parte de abajo se irá mostrando información sobre los eventos que suceden durante la simulación. Como son los tratamientos de los mensajes y las entradas o borrados en las tablas de enrutamiento.

```
** Event #201 t=0.106 rural.Router24.routing (Routing, id=199) on LOADng-RREQ
INFO (Routing)rural.Router24.routing: direcciones del nodo24
INFO (Routing)rural.Router24.routing: Tabla ida
INFO (Routing)rural.Router24.routing: 21424231501
INFO (Routing)rural.Router24.routing: LOADng Route Request arrived with source addr: 2 destinatio
INFO (Routing)rural.Router24.routing: hay una ruta mas corta, dejo de reenviarlo
** Event #202 t=0.106 rural.Router28.routing (Routing, id=217) on LOADng-RREQ (LOADngRREQ, id=
INFO (Routing)rural.Router28.routing: direcciones del nodo28
INFO (Routing)rural.Router28.routing: Tabla ida
INFO (Routing)rural.Router28.routing: LOADng Route Request arrived with source addr: 2 destinatio
```

Ilustración 31 Consola

En la parte izquierda se muestra información jerarquizada sobre los distintos parámetros que hay en la simulación como son valores de las conexiones, características de los routers etc... Aquí se puede consultar los valores los parámetros en tiempo real y ver cómo van variando durante la simulación. También se puede consultar aquí los siguientes eventos que van a suceder en la simulación.

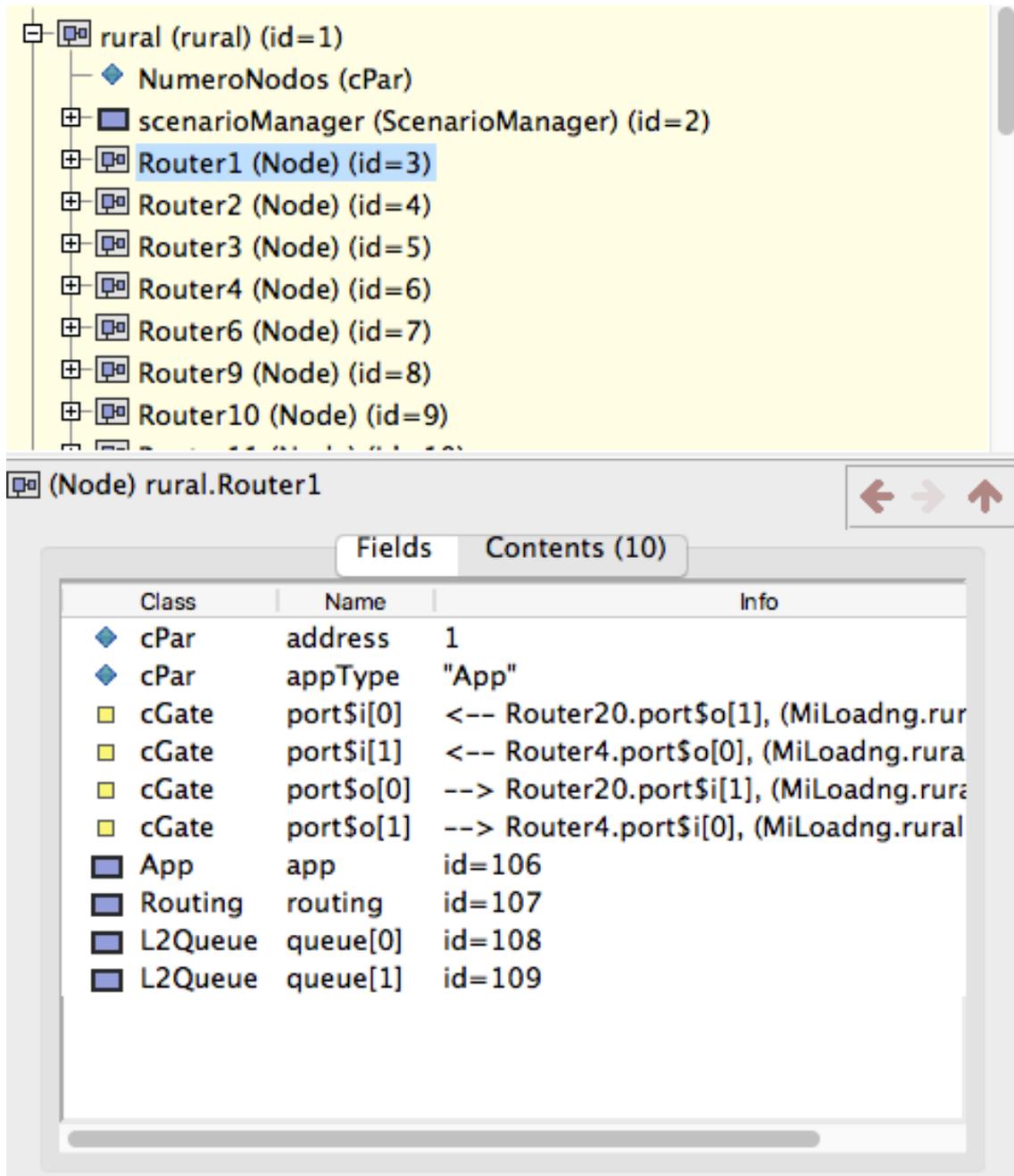


Ilustración 32 Parámetros

Finalmente recalcar que en todo momento durante la simulación se pueden consultar las tablas de enrutamiento de cada router simplemente haciendo doble click sobre el router que se quiera observar, en su interior sobre el submódulo *routing* se irán actualizando dinámicamente las tablas de enrutamiento mostrando el origen del mensaje, su destino, el puerto por el

que ha llegado, el número de saltos que ha dado hasta llegar al router actual y la cantidad de enlace débiles por los que ha pasado.

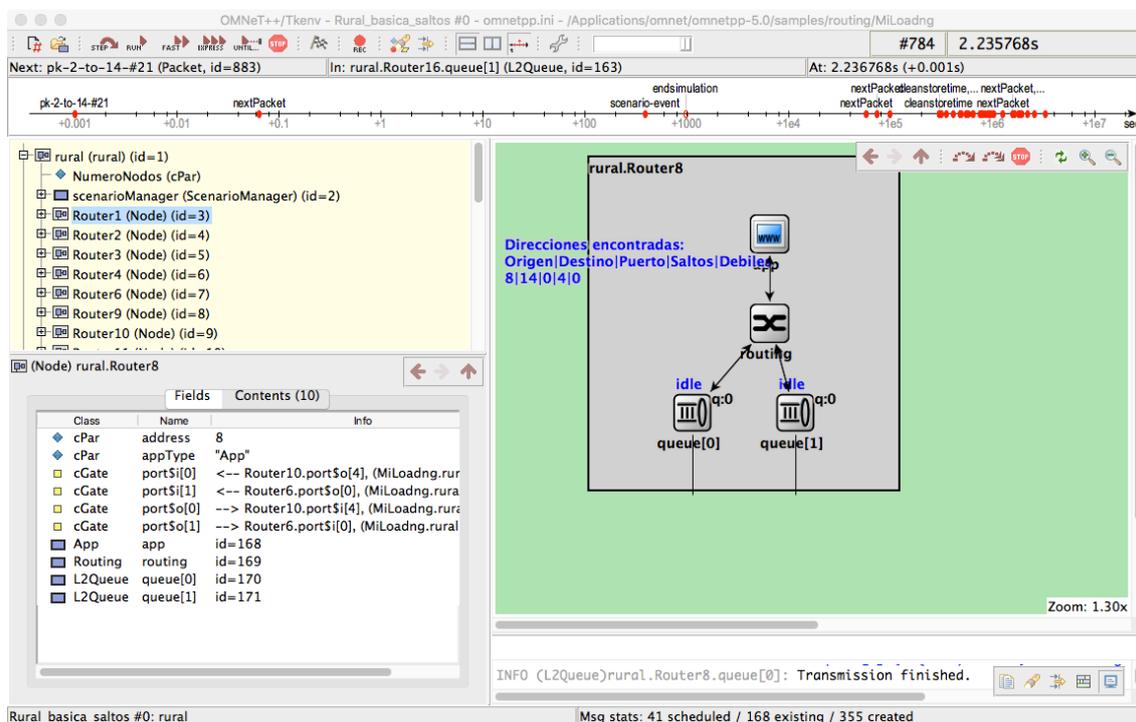


Ilustración 33 Interior router

1.6 Obtención de datos

Tras finalizar la simulación, se habrá creado una carpeta llamada *results* en la que aparecerán unos archivos *.sca*, *.vci* y *.vec* con el nombre de la topología simulada. Estos archivos son los datos recabados de la simulación anteriormente realizada.

Haciendo doble click sobre cualquiera de los archivos recién creados se nos abrirá una ventana en la que si nos dirigimos a *Browse Data* podremos ver la información de la simulación.

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

All (3157 / 3157) Vectors (16 / 16) Scalars (2935 / 2935) Histograms (206 / 206)

Folder	File name	Config name	Run num	Module	Name	Count
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router30.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router28.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router26.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router27.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router10.routing	hopCount:vector	2
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router8.routing	hopCount:vector	2
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router6.routing	hopCount:vector	2
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router2.routing	hopCount:vector	3
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router15.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router16.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router2.queue[0]	error:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router24.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router23.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router18.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router19.routing	hopCount:vector	1
/routing/MiLoadng/results/	Rural_delay1_...	Rural_dela...	0	rural.Router20.routing	hopCount:vector	1

Ilustración 34 Datos obtenidos

Los datos que se obtienen de las simulaciones son vectores, con una componente que resulta ser el instante en el que se produce una entrada en tabla y su otra componente es el número de saltos que ha dado el mensaje que produce la entrada en tabla. También se recaba la información de los mensajes de error creados que fuerzan la búsqueda de una nueva ruta. Seleccionando los datos que se deseen graficar, basta con hacer click derecho sobre ellos y marcar la opción de *Plot* para mostrar así por pantalla un gráfico representativo de estos datos.

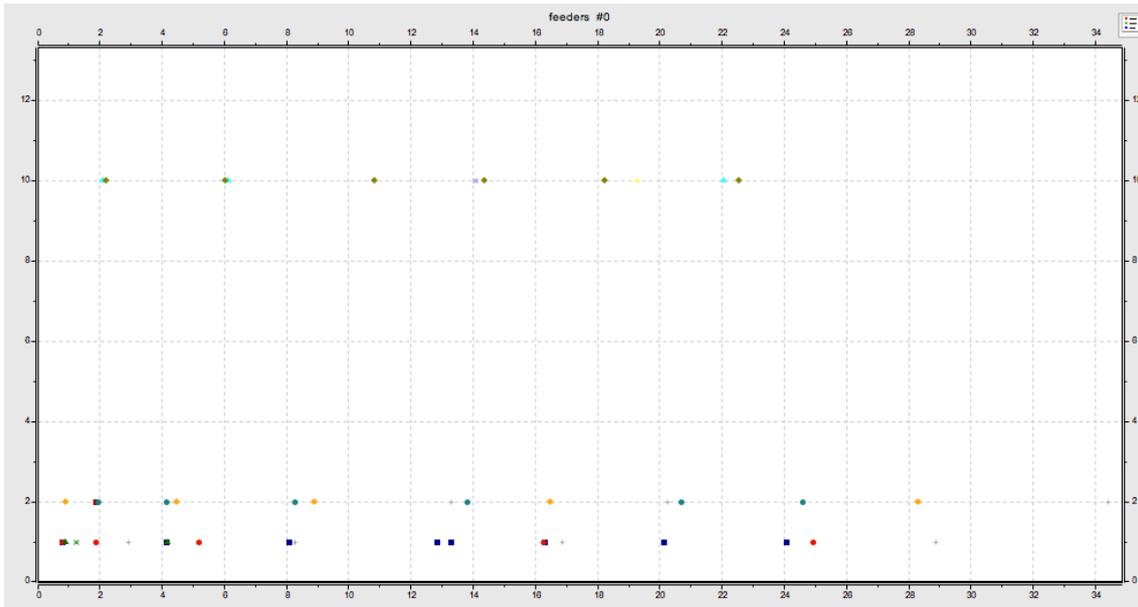


Ilustración 35 Gráfico de resultados

En el eje horizontal representa el instante de tiempo en el que se realiza la entrada en tabla de una ruta y en el eje vertical el número de saltos que produce esa entrada. Puede resultar aquí interesante la configuración de la gráfica que mejore su visualización, para ello haciendo click derecho sobre la gráfica en el apartado *Properties* se puede cambiar el color de los puntos, añadir líneas de tendencia etc...

También es posible exportar estos datos a otros formatos como CSV y Octave para su procesado con otras herramientas como Excel o Matlab. Solo hay que en el *Browse Data* en vez de seleccionar la opción de *Plot*, hay que ir al apartado *Export Data* y ahí seleccionar el formato al que exportar la información seleccionada

Browse Data

Here you can see all data that come from the files specified in the Inputs page.

Folder	File name	Config name	Run num	Module	Name	Count
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router3.routing	hopCount:vector	8
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router4.routing	hopCount:vector	5
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router6.routing	hopCount:vector	2
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router1.routing	hopCount:vector	5
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router7.routing	hopCount:vector	1
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router.routing	hopCount:vector	9
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router5.routing	hopCount:vector	1
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router2.routing	hopCount:vector	6
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router2.routing	error:vector	3
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router2.routing	error:vector	6
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router2.routing	error:vector	1
/routing/MiLoadng/results/	feeders-0.vec	feeders	0	feeders2.Router2.routing	error:vector	1

- + Add Filter Expression to Dataset...
- + Add Selected Data to Dataset...
- Export Data**
 - CSV...
 - Octave...
- Copy to Clipboard
- Set filter
- Choose Table Columns...
- Show Output Vector View

Ilustración 36 Exportación de datos