Apéndice C: Instalación y simulado de OMNeT++

Este apéndice ofrece al usuario una guía para la instalación de OMNeT++ y se puede encontrar de igual forma contenido en la Tesis de nombre Modelado de restauración de rutas en redes MPLS utilizando OMNeT++, realizada por José Galdino García Fierro. Así mismo, se presentan algunas soluciones a problemas relacionados con la instalación de este programa y de su plataforma de simulación conocida como INET *Framework*.

C.1 Instalación de OMNeT++

Previo a la instalación de OMNeT++, es necesario instalar el programa empleado para la compilación del mismo programa, de la plataforma de simulaciones INET *Framework* y de las propias simulaciones. En este trabajo de tesis, se empleó el programa Microsoft Visual C++ 8.0 (incluido en la paquetería de Microsoft Visual Studio 2005). El sitio de OMNeT se recomienda no usar la versión 6.0 de Microsoft Visual C++ ya que la compilación de las plataformas de simulación son propensas a fallar.

Una vez instalado el programa de Microsoft Visual C++, se procede a instalar el programa de OMNeT++. Este programa se descarga de forma gratuita del sitio de OMNeT++, esto si el programa es utilizado para propósitos académicos y sin fines de lucro; de lo contrario es necesario obtener una licencia para la versión comercial del programa llamado OMNEST. Actualmente está disponible en el sitio de Internet la versión 3.3 del programa. Una vez descargado, se ejecuta el programa. Entre los componentes a instalar, se recomienda instalar la versión completa ya que instala ejemplos de simulación que pueden ser útiles posteriormente. En la ubicación de instalación del programa, se recomienda instalarlo en la dirección predeterminada (C:\oppsim\omnetpp). En caso de querer cambiar la ubicación, el fólder destino no debe incluir espacios ni signos como "++". OMNeT++ brinda también una herramienta para la documentación de los programas, para lo cual necesita un programa llamado Ghostscript, el cual se puede descargar de forma gratuita pero una versión limitada. Si

se desea tener la versión completa del programa es necesario pagar una cuota. Finalmente, en el proceso de instalación se pide elegir el tipo de librerías de simulación con las que trabajará en conjunto con Microsoft Visual C++. Ya que se está empleando la versión 8 de Microsoft Visual C++, se elige la opción "vc80-release". Finalmente se elige instalar el programa, habiendo configurado lo anterior mencionado, y se estará listo para utilizar OMNeT++ en la simulación de redes de comunicaciones.

C.2 Instalación de la plataforma de simulaciones INET *Framework*

Si se desea emplear el INET Framework para el desarrollo de simulaciones o consulta de las mismas, es necesario descargarlo del sitio de OMNeT++. Esta plataforma de simulaciones se descarga de forma gratuita. INET Framework se descarga en un formato ".tar.gz". Este tipo de archivos indica que se utilizó un programa compresor de archivos para poder bajar la plataforma de simulaciones. Un problema que se tiene con el explorador Microsoft Internet Explorer es que éste cambia la extensión del archivo, por lo cual se recomienda cambiar manualmente la extensión a ".tgz" después de haber descargado la plataforma. Programas como WinZip o WinRAR con capaces de descomprimir este tipo de archivos. El programa que se usó para descomprimir este archivo fue el "7-Zip File Manager". Este programa descomprimió el archivo sin ningún problema.

Una vez descomprimido el archivo, se envía este a la carpeta donde se instaló el programa OMNeT++ (C:\oppsim\). Colocado el fólder en la ubicación antes mencionada, se leen los archivos "INSTALL" y "README" que ayudarán a instalar la plataforma de simulaciones. En el archivo "INSTALL" se muestran los pasos a seguir para la instalación de la plataforma. El primer paso establece que es necesario verificar la correcta instalación de OMNeT++. Para verificar lo anterior, se abre el Command Prompt y se teclea el comando "nedtool". En la Figura C.1 se muestra el texto que despliega el Command Prompt para este comando. Posteriormente, se pide ingresar el comando "nmake" para verificar si Microsoft

Visual C++ (MSVC) se encuentra en la ruta. En un incio, al ingresar "nmake", el cmd informará al usuario que el comando no es reconocido. Para que sea reconocido, es necesario correr el archivo "vcvars32.bat" que se encuentra en la carpeta de Microsoft Visual Studio. Una vez ejecutado el archivo antes mencionado, se ingresa el comando "nmake" y esta vez será reconocido por el cmd. El proceso de ejecutar el archivo "vcvars32.bat" resulta tedioso. Por lo cual se recomienda crear un archivo "make.bat" que incluya la ruta donde se encuentra el archivo antes mencionado. El contenido del archivo "make.bat" es el que se muestra a continuación:

```
@echo off
Cd/d%~dp0
Call "C:\Program Files\Micorsoft Viusual Studio
8\VC\vcvars32.bat
Nmake –f Makefile,vc*
```

Lo anterior en caso de tener instalado Microsoft Visual C++ 8.0. Una vez creado el archivo "make.bat" se recomienda colocarlo en el fólder donde fue instalado OMNeT++. En la Figura C.2 se muestra el reconocimiento del comando "nmake" usando el archivo "make.bat".

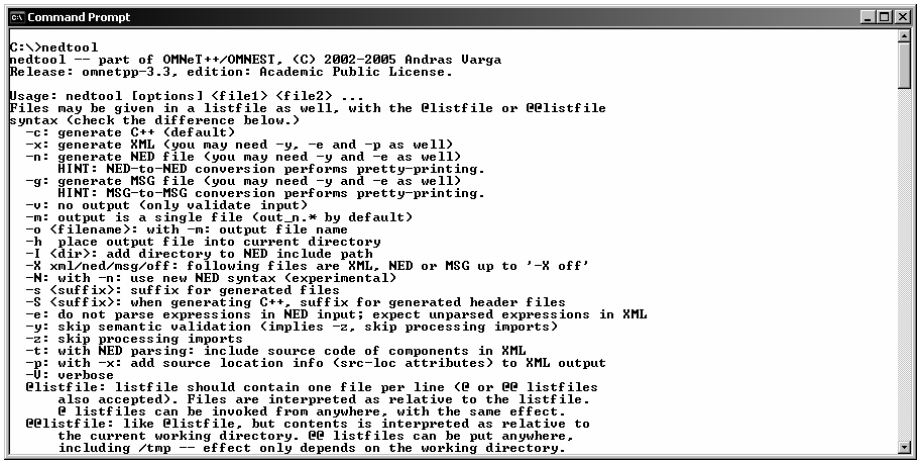

Figura C.1. Texto desplegado en el cmd al ingresar el comando "nedtool"

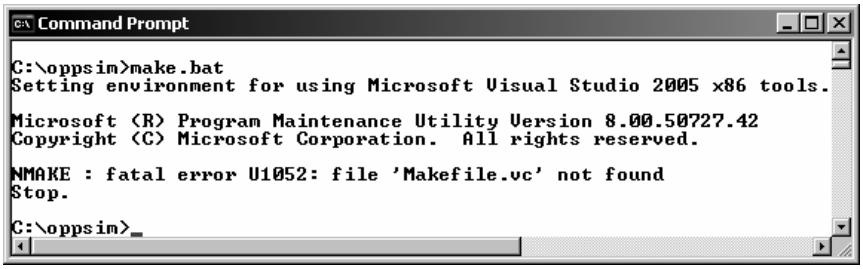

Figura C.2. Texto desplegado en el cmd al ingresar el comando "nmake" incluido en el archivo "make.bat"

Una vez reconocidos los comandos "nedtool" y "nmake", en el archivo INSTALL se pide editar el archivo "inetconfig.vc" y modificar la ruta indicada en la línea con la palabra OMNETPP_ROOT. De acuerdo a como se ha indicado la instalación, la línea se debe modificar por lo siguiente:

```
OMNETPP_ROOT=   C:/oppsim/omnetpp
```

Una vez cambiada la línea, se ejecuta el archivo "makemake.cmd" para crear los archivos "Makefile.vc". Usando el Command Prompt, se accede a la ubicación del INET Framework y se ingresa el siguiente comando:

```
nmake –f makefile.vc depend
```

Y finalmente se ingresa el comando:

```
Nmake –f Makefile.vc
```

Para construir las simulaciones. El anterior comando es también utilizado para la compilación de los archivos ".cpp / .cc". Una vez creados los archivos, sin ningún mensaje de error en el cmd, se accede al fólder "Examples" en el cmd y se teclea:

```
rundemo.bat
```

Para visualizar los ejemplos de simulación provistos por el INET Framework.

C.3 Compilación de archivos .cpp/ .cc y msg

Para compilar los archivos ".cpp / .cc", que contienen la descripción del comportamiento de los módulos simples declarados en el GNED, y los archivos ".msg", que definen la estructura de distintos tipos de mensajes, se utilizará el Command Prompt. Una vez creados los archivos antes mencionados, en un fólder donde se encuentra instalado OMNeT++, el INET Framework y el archivo "make.bat", se accede al fólder donde se encuentran usando el cmd. Antes de ingresar el

comando de compilación, se debe haber ejecutado el "make.bat" desde el cmd para que reconozca el comando de compilación. Habiendo ejecutado el archivo ".bat", se ejecuta el comando:

```
$ opp_nmakemake
```

Este comando crea un archivo "Makefile.vc", el cual tendrá un listado de todos los archivos que se encuentran en el fólder. Una vez creado el archivo "Makefile.vc", se procede a ingresar el comando de compilación:

```
nmake –f Makefile.vc
```

Este comando checa la sintaxis del archivo ".ned" y lo transforma en un archivo "_n.cpp". Se realiza también la compilación de cada archivo ".cpp / .cc" creado y convierte los archivos ".msg" en archivos ".cpp" y ".h". Finalmente, crea archivos ".obj" y un archivo ejecutable ".exe", que servirá para observar la simulación de la red programada.

Para más detalles respecto al proceso de compilación está el manual de usuario que ofrece OMNeT++ en su instalación.

C.4 Problemas enfrentados durante el desarrollo de la simulación

Durante la creación de los archivos de simulación, se presentaron ciertos problemas que impidieron el correcto funcionamiento de la simulación.

El primer problema al que se enfrentó fue el llamado a funciones entre distintos archivos. Se realizaron gran cantidad de llamados sin embargo no funcionaban aunque los archivos estuvieran en un mismo fólder. Se llegó a implementar un llamado a funciones que a primera vista realizaba las funciones de otros archivos; sin embargo, al correr la simulación este tipo de llamados generaba variables "basura" en cada módulo donde se realizaba el llamado. En el manual de usuario del programa OMNeT++, en el apartado 4.10, se describe el llamado a funciones entre módulos. Este llamado no genera las variables "basura" en los módulos e incluso

permite configurar dichos llamados para que incluyan o no una animación mostrada en el Tkenv usando el comando “Enter_Method()” y “Enter_Method_Silent()”, respectivamente.

Otro problema que se enfrentó, seguido el anterior, es la localización de archivos en distintas carpetas. Se tuvieron que colocar todos los archivos en una misma carpeta para que se pudieran compilar y que el llamado entre funciones se realice sin problemas. El manual de usuario de OMNeT++, en su apartado 7.2.2 y 7.2.3 explica los comandos necesarios que deben ser ingresados en el Command Prompt para permitir que los archivos estén en distintas ubicaciones (fólders).

Otro problema se presentó en el GNED. En la creación de los módulos y sus compuertas y crear al mismo tiempo el código automáticamente, este código crea el nombre de compuertas de acuerdo a un nombre predeterminado y no al configurado dentro de los módulos de GNED. Así, si se crea un módulo con nombre de compuertas “msgin” y “msgout”, el código generado automáticamente por la herramienta de OMNeT++ las nombra como “in” y “out”. GNED identifica el error al momento, sino que al momento de compilarlo y crear el archivo “.cpp” se crean incongruencias entre el nombre de la compuerta establecido en el módulo y el nombre de la compuerta declarado en el código que se generó. Esto provoca que aparezcan muchos errores de compilación, por lo cual es necesario corregir estos errores cuando el GNED debería haberlos detectado desde un principio para evitarlos.

Finalmente, otro problema con el que se enfrentó fue en las conexiones establecidas en el GNED. Las conexiones diseñadas de forma gráfica son colocadas en la ventana de código del GNED de acuerdo a como son creadas éstas. La asignación del número de interfaz tanto de entrada como de salida del módulo (numeradas de cero al número de compuertas que tenga el módulo) se asigna de acuerdo a cómo aparecen en la ventana de código del GNED. Así, a la primera conexión creada para un módulo se le asignará el número 0, a la segunda conexión se el asignará el número 1 y así sucesivamente. El conocer el número de la compuerta, tanto de

entrada como de salida, es indispensable para saber por dónde mandar el mensaje hacia otro módulo.

Los problemas mencionados previamente son los más importantes a los que enfrentó en la creación de la simulación deseada.

C.5 Ejemplo de simulación

La simulación que a continuación se muestra está aun incompleta, pero se pretende tenerla lista para el día de la presentación final del proyecto.

A continuación se explica un poco sobre los pasos para generar un modelo funcional en OMNeT++, con ayuda también de sus programas auxiliares:

1. Se crea un directorio de trabajo.
2. Se escribe un archivo que describa la topología de la red a crear. Este archivo, puede ser creado en cualquier editor de texto, o mediante la ayuda de GNED como se vio anteriormente (Apéndice B.
3. Una vez creado el archivo tipo .ned, se tienen que crear módulos simples, que contienen a su vez a sub-módulos que representen a ruteadores, computadoras, *switches*, *hubs*, etc. Después de esto se implementado en C++ para generar su comportamiento.
4. A continuación se crea un *Makefile* que nos ayuda a compilar y vincular nuestro programa para crear el archivo ejecutable. El siguiente comando nos ayuda con este proceso:

```
$ opp_nmakemake
```

5. A continuación se utiliza este otro comando para compilar esta simulación, este código es utilizado para sistemas Windows con MSVC:

```
$ nmake -f Makefile.vc
```

6. Hasta este punto ya se a generado un archivo ejecutable, pero hay un inconveniente, ya que este archivo no correrá debido a que necesita de un archivo tipo ini (omnetpp.ini) que indique que es lo que se quiere simular.

7. Para crear este archivo se escriben las siguientes líneas y se guarda como un archivo .ini dentro de la capeta ya creada :

```
 [General]
network = tictoc1
```

Las operaciones que le siguen a estos pasos, están aun en proceso de ser generados pero su función principal es empezar las simulaciones dentro de OMNeT++. Hay que aclarar que cuando se crean los archivos en C++ para crear módulos funcionales, la manera de programarlos en C no es de la manera "tradicional", por así decirlo, ya que existen librerías propias de OMNeT++ utilizadas para la generación de estos módulos.

Para mayor referencia de estas clases y librerías utilizadas, se puede referir a la siguiente liga, que pertenece a la página oficial de OMNeT++:

http://community.omnest.com/doc/tictoc-tutorial/classTxc1.html

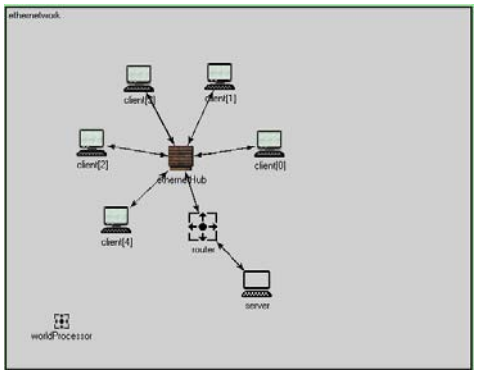

Figura C.3 Diagrama del modelo IPv6 a simularse. Consiste de una topología en estrella, con conexión a un ruteador y a un servidor de red.

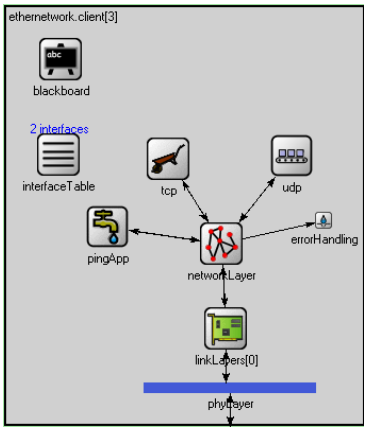

Figura C.4. Composición del módulo Cliente. Se puede observar que aquí se describe el comportamiento que presentar un nodo final, o lo que vendría siendo un usuario.