

ANEXO C

INSTALACIÓN DE HERRAMIENTAS SOFTWARE

En el presente anexo se muestra la forma de instalación de las herramientas software más importantes, utilizadas en el desarrollo del proyecto.

1 **OPNET Modeler [1]**

A continuación se presentan los pasos necesarios para la instalación del software simulador de redes OPNET Modeler 11.5 sobre un sistema operativo Windows XP Service Pack 3.

1. Instalar Visual Studio (para este proyecto se utilizó la versión 2006, la cual se recomienda usar para esta versión del simulador)
2. Instalar los tres componentes que vienen incluidos en el paquete del simulador OPNET Modeler, los cuales son:
 - Modeler_115.exe
 - Modeler_docs.exe
 - Models_115.exe
3. Configurar las variables de entorno del PC o máquina virtual en el que se esté instalando el simulador, para el correcto funcionamiento del compilador.

A continuación se muestran las distintas variables que se deben configurar y sus respectivos valores. Para ingresar a las variables de entorno se debe hacer clic derecho en Mi PC y escoger la opción Propiedades, luego hacer clic en la pestaña Opciones Avanzadas y finalmente, clic en la opción Variables de Entorno.

Variable1: Path

Valor:

C:\Visual6\Common7\IDE;C:\Visual6\VC\BIN;C:\Visual6\Common7\Tools;C:\Visual6\Common7\Tools\bin;C:\Visual6\VC\PlatformSDK\bin;C:\Visual6\SDK\v2.0\bin;C:\Windows\Microsoft.NET\Framework\v2.0.50727;C:\Visual6\VC\VCpackages;C:\Visual6\SDK\v2.0\Bin;C:\Windows\Microsoft.NET\Framework\v2.0.50727;C:\Visual8\VC\bin;C:\Visual8\Common7\IDE;C:\Visual6\VC\vcpackages;

Variable2: INCLUDE

Valor:

C:\Visual6\VC\ATLMFC\INCLUDE;C:\Visual6\VC\INCLUDE;C:\Visual6\VC\PlatformSDK\include;C:\Visual6\SDK\v2.0\include;

Variable3: LIB

Valor:

C:\Visual6\VC\ATLMFC\LIB;C:\Visual6\VC\LIB;C:\Visual6\VC\PlatformSDK\lib;C:\Visual6\SDK\v2.0\lib;

Variable4: LIBPATH

Valor:

C:\Windows\Microsoft.NET\Framework\v2.0.50727;C:\Visual6\VC\ATLMFC\LIB;

Variable5: NetSamplePath

Valor: C:\Visual6\SDK\v2.0;

Variable6: DevEnvDir

Valor: C:\Visual6\Common7\IDE;

Variable7: FrameworkDir

Valor: C:\Windows\Microsoft.NET\Framework;

Variable8: FrameworkSDKDir

Valor: C:\Visual6\SDK\v2.0;

Variable9: FrameworkVersion

Valor: v2.0.50727;

Variable10: VCBUILD_DEFAULT_CFG

Valor: Debug^|Win32;

Variable11: VCBUILD_DEFAULT_OPTIONS

Valor: /useenv;

Variable12: VCINSTALLDIR

Valor: C:\Visual6\VC;

2 ns-3 [2]

En este apartado se muestran los pasos a seguir para la instalación del software simulador de redes ns-3 versión 10.0.

A continuación se muestra la instalación de los distintos paquetes que se requieren para soportar opciones comunes de ns-3. Para este caso en particular, se muestra la instalación de estos paquetes sobre un sistema operativo Ubuntu. Para otras distribuciones de Linux el requerimiento de los paquetes puede cambiar. El core de ns-3 requiere una instalación de gcc/g++ 3.4 o superior, y python 2.4 o superior.

- **Requerimientos mínimos para C++:**

```
sudo apt-get install gcc g++ python
```

- **Requerimientos mínimos para Python:**

```
sudo apt-get install gcc g++ python python-dev
```

- **Mercurial. Paquete necesario para trabajar con ns-3:**

```
sudo apt-get install mercurial
```

- **Bazaar. Paquete necesario para correr uniones con Python desde el árbol de desarrollo de ns-3:**

```
sudo apt-get install bazaar
```

- **Depurador:**

```
sudo apt-get install gdb valgrind
```

- **Soporte de *GNU Scientific Library (GSL)* para modelos de error WiFi más acertados:**

```
sudo apt-get install gsl-bin libgsl0-dev libgsl0ldbl
```

- **Analizadores léxicos *flex* y *bison*. Requerido por *Network Simulation Cradle (NSC)*:**

```
sudo apt-get install flex bison
```

- **Gcc-3.4. Requerido por algunos NSC:**

```
sudo apt-get install g++-3.4 gcc-3.4
```

- **Para leer trazas de paquetes pcap:**

```
sudo apt-get install tcpdump
```

- **Soporte para bases de datos para marcos de trabajo estadístico:**

```
sudo apt-get install sqlite sqlite3 libsqlite3-dev
```

- **Versión basada en XML del almacén de configuración (requiere versión 2.7 o posterior):**

```
sudo apt-get install libxml2 libxml2-dev
```

- **Sistema de configuración basado en GTK:**

```
sudo apt-get install libgtk2.0-0 libgtk2.0-dev
```

- **Para experimentar con máquinas virtuales y ns-3:**

```
sudo apt-get install vtun lxc
```

- **Soporte para el programa de chequeo de estilo de código:**

```
sudo apt-get install uncrustify
```

- **Doxygen y documentación relacionada:**

```
sudo apt-get install doxygen graphviz imagemagick
```

```
sudo apt-get install texlive texlive-pdf texlive-latex-extra  
texlive-generic-extra texlive-generic-recommended
```

- **Soporte para el visualizador ns-3 pyviz:**

```
sudo apt-get install python-pygraphviz python-kiwi python-  
pygoocanvas libgoocanvas-dev
```

- **Soporte para el módulo de flujo abierto (requiere algunas librerías boost):**

```
sudo apt-get install libboost-signals-dev libboost-  
filesystem-dev
```

Después de haber instalado las librerías anteriores ya se puede hacer la descarga del simulador ns3 desde la página oficial. <http://www.nsnam.org/>. Una vez descargado, se abre un terminal y se dirige hacia donde se descargó ns-3; comúnmente debe ser un archivo .tar.bz2, el cual se debe descomprimir y compilar con las siguientes líneas:

```
tar -xvf ns-allinone-3.10.tar.bz2
```

```
cd ns-allinone-3.10
```

```
./build.py
```

Una vez hecho esto, se ingresa en la carpeta ns-3.10 y se valida la instalación de la siguiente manera,

```
cd ns-3.10
```

```
./test.py
```

Una vez aparezca lo siguiente en la pantalla, ns-3 estará instalado,

Anexo C *INSTALACIÓN DE HERRAMIENTAS SOFTWARE*

```
Waf: Entering directory `~/media/Installations/ns-allinone-3.10/ns-3.10/build'
```

```
Waf: Leaving directory `~/media/Installations/ns-allinone-3.10/ns-3.10/build'
```

```
'build' finished successfully (2.010s)
```

```
PASS: TestSuite lte-bearer
```

```
PASS: TestSuite lte-propagation-loss-model
```

```
PASS: TestSuite wimax-fragmentation
```

```
PASS: TestSuite lte-phy
```

```
...
```

```
PASS: Example examples/tcp/tcp-nsc-lfn
```

```
PASS: Example examples/wireless/wifi-ap.py
```

```
PASS: Example src/contrib/flow-monitor/examples/wifi-olsr-flowmon.py
```

```
168 of 168 tests passed (168 passed, 0 skipped, 0 failed, 0 crashed, 0 valgrind errors)
```

3 LIVE555 [3]

Los siguientes pasos se deben seguir para la instalación del servidor multimedia LIVE555 sobre un sistema operativo Ubuntu de Linux, además del programa VLC, el cual permite una fácil conexión con el servidor LIVE555.

1. Instalar libmad

MAD (MPEG Audio Decoder) es un decodificador de audio MPEG de alta calidad.

El código que se debe introducir en una consola de Linux es el siguiente:

```
Ubuntu% wget http://ovh.dl.sourceforge.net  
/sourceforge/mad/libmad-0.15.1b.tar.gz
```

```
Ubuntu% tar xvfz libmad-0.15.1b.tar.gz
```

```
Ubuntu% cd libmad-0.15.1b
```

```
Ubuntu% ./configure --prefix=/usr/local
```

```
Ubuntu% make
```

```
Ubuntu% sudo make install
```

2. Instalar libdca

Libdca es una librería gratuita para audio DTS.

```
Ubuntu% wget
```

```
http://download.videolan.org/pub/videolan/libdca/0.0.5/libdca-0.0.5.tar.bz2
```

```
Ubuntu% bzip2 -d libdca-0.0.5.tar.bz2
```

```
Ubuntu% tar xvf libdca-0.0.5.tar
```

```
Ubuntu% cd libdca-0.0.5
```

```
Ubuntu% ./configure --prefix=/usr/local
```

```
Ubuntu% make
```

```
Ubuntu% sudo make install
```

3. Instalar libmpeg2

Libmpeg2 es una librería gratuita para flujos MPEG.

```
Ubuntu% wget http://libmpeg2.sourceforge.net/files/libmpeg-0.5.1.tar.gz
```

```
Ubuntu% tar xvfz libmpeg2-0.5.1.tar.gz
```

```
Ubuntu% cd libmpeg2-0.5.1
```

```
Ubuntu% ./configure --prefix=/usr/local
```

```
Ubuntu% make
```

```
Ubuntu% sudo make install
```

4. Instalar TagLib

TagLib es una librería para la lectura y edición de información de muchos formatos populares de audio (MP3, FLAC, etc.).

```
Ubuntu% wget
http://developer.kde.org/~wheeler/files/src/taglib-1.5.tar.gz

Ubuntu% tar xvfz taglib-1.5.tar.gz

Ubuntu% ./configure --prefix=/usr/local

Ubuntu% make

Ubuntu% sudo make install
```

5. Instalar live555

Live555 es una configuración de librerías para flujo multimedia (RTP/RTCP, RTSP, SIP).

```
Ubuntu% wget http://www.live555.com/liveMedia/public/live555-
latest.tar.gz

Ubuntu% tar xvfz liv555-latest.tar.gz

Ubuntu% cd live

Ubuntu% ./genMakefiles linux

Ubuntu% make
```

6. Ahora, se usa APT para buscar todas las librerías faltantes que necesita VLC:

```
Sudo apt-get install libavc1394-dev libraw1394-dev libdc1394-
dev libdvdrread-dev libdvdnv-dev libdvdcss2-dev libfaad-dev
libtwolame-dev liba52-dev libvcdinfo-dev libiso9660-dev
libcddb2-dev libflac-dev libschrödinger-dev libogg-dev
libvorbis-dev liblua5.1-0-dev libgnomevfs2-dev libtag1-dev
libqt4-dev
```

7. VLC

Las fuentes estables pueden ser descargadas como comprimidos desde la página VLC download.

```
Ubuntu% wget  
http://download.videolan.org/pub/videolan/vlc/0.9.9/vlc-  
0.9.9.tar.bz2
```

```
Ubuntu% bzip2 -d vlc-0.9.9.tar..bz2
```

```
Ubuntu% tar xvf vlc-0.9.9.tar
```

8. Agregar X264 y Live555 a la mezcla

```
Ubuntu% cd vlc-0.9.9
```

```
Ubuntu% cp -r $YOUR_X264_BUILD_DIR extras
```

```
Ubuntu% cp -r $YOUR_LIVE555_BUILD_DIR extras
```

9. Configurar VLC

```
Ubuntu% ./configure --prefix=/usr/local --with-x264-  
tree=extras/x264 --with-live555-tree=extras/live --enable-  
release --enable-switcher --enable-shout --enable-dc1394 --  
enable-dv --enable-dvdread --enable-v4l --enable-pvr --enable-  
gnomevfs --enable-vcdx --enable-faad --enable-twolame --enable-  
real --enable-realrtsp --enable-flac --enable-tremor --enable-  
tarking --enable-theora --enable-ogg --enable-vorbis --enable-  
a52 --enable-dca
```

4 **ffmpeg [4]**

ffmpeg es un lenguaje de programación que brinda soluciones para la grabación y edición de flujos de audio y video. Para la instalación de ffmpeg en un sistema operativo Ubuntu de Linux, se deben seguir los pasos enumerados a continuación:

1. Obtener las dependencias

```
sudo apt-get update  
sudo apt-get -y install autoconf automake build-essential git  
libass-dev libgpac-dev \  
libsdl1.2-dev libtheora-dev libtool libva-dev libvdpau-dev  
libvorbis-dev libx11-dev \  
libxext-dev libxfixes-dev pkg-config texi2html zlib1g-dev  
mkdir ~/ffmpeg_sources
```


2. Compilación e instalación

Yasm: Ensamblador usado por x264 y ffmpeg.

```
cd ~/ffmpeg_sources
wget http://www.tortall.net/projects/yasm/releases/yasm-1.2.0.tar.gz
tar xzvf yasm-1.2.0.tar.gz
cd yasm-1.2.0
./configure --prefix="$HOME/ffmpeg_build" --bindir="$HOME/bin"
make
make install
make distclean
. ~/.profile
```

X264: Codificador de video H.264.

```
cd ~/ffmpeg_sources
git clone --depth 1 git://git.videolan.org/x264.git
cd x264
./configure --prefix="$HOME/ffmpeg_build" --bindir="$HOME/bin" --
enable-static
make
make install
make distclean
```

fdk-aac: Codificador de audio AAC.

```
cd ~/ffmpeg_sources
git clone --depth 1 git://github.com/mstorsjo/fdk-aac.git
cd fdk-aac
autoreconf -fiv
./configure --prefix="$HOME/ffmpeg_build" --disable-shared
make
make install
make distclean
```

libmp3lame: Codificador de audio MP3.

```
sudo apt-get install nasm
cd ~/ffmpeg_sources
wget http://downloads.sourceforge.net/project/lame/lame/3.99/lame-
3.99.5.tar.gz
tar xzvf lame-3.99.5.tar.gz
cd lame-3.99.5
./configure --prefix="$HOME/ffmpeg_build" --enable-nasm --disable-
shared
make
make install
make distclean
```

libopus: Codificador y decodificador de audio Opus.

```
cd ~/ffmpeg_sources
wget http://downloads.xiph.org/releases/opus/opus-1.0.3.tar.gz
tar xzvf opus-1.0.3.tar.gz
cd opus-1.0.3
./configure --prefix="$HOME/ffmpeg_build" --disable-shared
make
make install
make distclean
```

libvpx: Codificador y decodificador de audio VP8/VP9.

```
cd ~/ffmpeg_sources
git clone --depth 1 http://git.chromium.org/webm/libvpx.git
cd libvpx
./configure --prefix="$HOME/ffmpeg_build" --disable-examples
make
make install
make clean
```

ffmpeg:

```
cd ~/ffmpeg_sources
git clone --depth 1 git://source.ffmpeg.org/ffmpeg
cd ffmpeg
PKG_CONFIG_PATH="$HOME/ffmpeg_build/lib/pkgconfig"
export PKG_CONFIG_PATH
./configure --prefix="$HOME/ffmpeg_build" \
  --extra-cflags="-I$HOME/ffmpeg_build/include" --extra-ldflags="-L$HOME/ffmpeg_build/lib" \
  --bindir="$HOME/bin" --extra-libs="-ldl" --enable-gpl --enable-libass --enable-libfdk-aac \
  --enable-libmp3lame --enable-libopus --enable-libtheora --enable-libvorbis --enable-libvpx \
  --enable-libx264 --enable-nonfree --enable-x11grab
make
make install
make distclean
hash -r
```

REFERENCIAS

- [1] www.opnet.com Página web del simulador de redes OPNET Modeler
- [2] www.nsnam.org Página web del simulador de redes ns-3
- [3] www.live555.com Página web del servidor LIVE555
- [4] www.ffmpeg.org Página web del software ffmpeg

