

[ANEXO 1]

[Instalación alternativa de OpenMCU v2.2.5]

Una vez descargadas las librerías de compilación necesarias como se explico en capítulo de metodología y los siguientes paquetes de instalación: *ptlib-2.10.1.tar.gz*, *h323plus-1.23.0.tar.gz* *applications-1.23.0.tar*, se procede con los siguientes comandos:

```
cd /home/NombrePC/ Desktop  
sudo su  
cp ptlib-2.10.1.tar.gz /usr/src  
cp h323plus-1.23.0.tar.gz /usr/src  
cp applications-1.23.0.tar.gz /usr/src  
cd /usr/src
```

Luego se debe instalar PTLib para ello se ejecuta:

```
cd /usr/src  
tar xvf ptlib-2.10.1.tar.gz  
cd ptlib-2.10.1  
./configure  
make  
make install  
PTLIBDIR=/usr/src/ptlib-2.10.1  
export PTLIBDIR
```

Inmediatamente se debe continuar con OpenH.323plus

```
cd /usr/src
```

```
tar zxvf h323plus-1.23.0.tar.gz  
cd h323plus-1.23.0  
./configure  
make bothdepend  
make bothnoshared
```

Como se explico los comandos `make bothdepend` y `make bothnoshared` darán la versión óptima y a prueba de errores, luego se continúa con:

```
cd plugin  
make  
make install  
PTLIBPLUGINDIR=/usr/local/lib/opal-1.23.0/codecs  
export PTLIBPLUGINDIR  
OPENH323DIR=/usr/src/h323plus-1.23.0  
export OPENH323DIR  
LD_LIBRARY_PATH=/usr/src/ptlib-2.8.3/lib:/usr/src/h323plus-1.23.0/lib  
export LD_LIBRARY_PATH
```

Para Marzo de 2011 el paquete de aplicaciones, en el directorio de OpenMCU el paquete `h323.cxx` venía con errores. Esto ocasiona que el compilador entre en un ciclo infinito y provoca el cese de procesos en el sistema operativo por sobrecarga de memoria. En los momentos actuales este problema esta corregido por el equipo del OpenH.323plus pero no dan garantías. Uno de los errores hallados se encontraba en la línea 216, donde se debe borrar la línea completa para corregir el problema y el otro error se repara comentando la línea 300, para así resolver el lazo infinito. Para conseguir los citados errores se recurrió a la comparación del archivo `h323.cxx` con sus similares en otros proyectos tales como Ekiga y Opal. Una vez resuelto el problema se continúa con:

```
cd /usr/src  
tar zxvf applications-1.22.0.tar.gz  
cd applications-1.22.0  
cd openmcu
```

make debugnoshared // (o make optnoshared)

Si se desea utilizar la OpenMCU con el códec H.264 se necesita mover para instalar los archivos *_pwplugin.so en */usr/local/lib/PWLib* o establecer la ruta a los plugins de códec. El comando para lo segundo es el siguiente:

```
cd /usr/local/lib/opal-1.22.0/codecs/video  
cp h264_video_pwplugin_helper /usr/local/lib/codecs/video  
cd /usr/local/lib/codecs/video  
cp h264_video_pwplugin.so /usr/local/lib/opal-1.22.0/codecs/video  
cd /usr/src/h323plus-1.22.0  
ls  
cd lib  
cp libh323_linux_x86_.so.1.22.0 /usr/local/lib  
cd /usr/local/bin
```

y se ejecuta dentro de el directorio anterior */sbin/ldconfig*. Luego se ejecuta *openmcu*, seguidamente de:

```
PTLIBPLUGINDIR=/usr/local/lib/opal-1.22.0/codecs  
export PTLIBPLUGINDIR  
openmcu -d  
openmcu -xc
```

[ANEXO 2]

[Modificación Manual de los Orígenes del Software en Ubuntu]

En caso de no poder acceder al servidor de actualización <http://archive.linux.duke.edu/ubuntu/> *lucid* se puede modificar el archivo */etc/apt/sources.list*, haciendo *nano/etc/apt/sources.list* y luego copiando y pegando el texto a continuación:

```
# deb cdrom:[Ubuntu 10.04 LTS _Lucid Lynx_ - Release i386 (20100429)]/ lucid main  
restricted  
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to  
# newer versions of the distribution.  
deb http://archive.linux.duke.edu/ubuntu/ lucid main restricted
```

```
deb-src http://archive.linux.duke.edu/ubuntu/ lucid main restricted
## Major bug fix updates produced after the final release of the
## distribution.
deb http://archive.linux.duke.edu/ubuntu/ lucid-updates main restricted
deb-src http://archive.linux.duke.edu/ubuntu/ lucid-updates main restricted
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://archive.linux.duke.edu/ubuntu/ lucid universe
deb-src http://archive.linux.duke.edu/ubuntu/ lucid universe
deb http://archive.linux.duke.edu/ubuntu/ lucid-updates universe
deb-src http://archive.linux.duke.edu/ubuntu/ lucid-updates universe
## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://archive.linux.duke.edu/ubuntu/ lucid multiverse
deb-src http://archive.linux.duke.edu/ubuntu/ lucid multiverse
deb http://archive.linux.duke.edu/ubuntu/ lucid-updates multiverse
deb-src http://archive.linux.duke.edu/ubuntu/ lucid-updates multiverse
## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
# deb http://us.archive.ubuntu.com/ubuntu/ lucid-backports main restricted universe
# multiverse
# deb-src http://us.archive.ubuntu.com/ubuntu/ lucid-backports main restricted universe
# multiverse
## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
```

```
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu lucid partner
# deb-src http://archive.canonical.com/ubuntu lucid partner
deb http://archive.linux.duke.edu/ubuntu/ lucid-security main restricted
deb-src http://archive.linux.duke.edu/ubuntu/ lucid-security main restricted
deb http://archive.linux.duke.edu/ubuntu/ lucid-security universe
deb-src http://archive.linux.duke.edu/ubuntu/ lucid-security universe
deb http://archive.linux.duke.edu/ubuntu/ lucid-security multiverse
deb-src http://archive.linux.duke.edu/ubuntu/ lucid-security multiverse
```

Una vez copiada y pegada, se ejecuta el comando `apt-get update`, y ya se tienen los repositorios educacionales de Estados Unidos, los cuales poseen todas las librerías de compilación, construidas correctamente.

[ANEXO 3]

[Consideraciones Importantes sobre el controlador de compilación MAKE]

Cómo ejecutar Make

La forma más simple es llamar a make sin parámetros. Esto construye la primera regla encontrada en un fichero llamado makefile o, mejor, Makefile. Pero podemos querer usar otro fichero *makefile* u otro compilador, u otras opciones de compilación; o podemos querer ver simplemente qué ficheros están anticuados sin que se recompilen. Mediante el uso de parámetros y opciones a make, podemos hacer estas cosas y muchas otras.

Parámetros para especificar el *makefile*

Mediante la opción '-f' seguida del nombre del fichero *makefile*. Si no se usa esta opción, make, en Unix, busca el *makefile* en un fichero que se llame makefile y, si éste no existe, Makefile. Ejemplo:

```
make -f editor.mk
```

La extensión mk es arbitraria; podría ser cualquiera, o no haber.

Parámetros para especificar el objetivo

En un *makefile* pueden existir varios objetivos. A veces un paquete de programas se compone de varios, y el primer objetivo que se pone es uno que los agrupa a todos, de forma que al llamar a make sin parámetros se reconstruye todo; pero puede ser que se quiera por alguna razón construir solamente uno; o puede que haya un objetivo para

borrar lo sobrante en el directorio, o para imprimir los fuentes, y no sea el primero del *makefile*.

Para ello, simplemente al llamar a make se le dice qué objetivo se desea. Por ejemplo, supóngase que el *makefile* contiene al principio:

PHONY: all

all: size nm ld ar as

Si no se especifica ningún objetivo, se hará all y por tanto todas sus dependencias; pero si sólo se quiere construir el programa size, se hará:

make size

Parámetros para no ejecutar las órdenes

El *makefile* le dice a make si un objetivo está anticuado y cómo actualizarlo. Pero a veces no se desea esto. Para ello están las siguientes opciones:

-t Con *te* de «tocar» (por el programa de Unix touch). Se marcan los objetivos como actualizados pero sin cambiarlos. En otras palabras, make finge compilar los objetivos pero realmente no los modifica.

-n Con *ene* de «no operar». Se ven todas las órdenes que make haría pero no se ejecutan realmente. Esta opción es muy útil para no «meter la pata» y para probar con otras opciones.

-q Con *cu* de *question* («inquirir»). Silenciosamente se mira si los objetivos están ya actualizados, pero no se ejecutan las órdenes en ningún caso; osea, no se produce compilación ni salida. El único efecto es devolver un código de *status* al *shell*.

La opción ‘-n’ es realmente muy útil. Recuérdese y utilícese.

Parámetros para cambiar macros

Un parámetro que contenga el signo igual especifica el valor de una macro.

La orden `make macro=valor`

cambia el valor de la macro *macro* a *valor*. Todas las definiciones de esa macro existentes en el *makefile* no se tienen en cuenta. Por ejemplo, se puede hacer que la compilación se efectúe con el compilador de GNU y las opciones de dar toda clase de avisos extra y generar una tabla de símbolos para el depurador mediante:

```
make CXX=g++ CXXFLAGS="-Wall -g"
```

Resumen de parámetros y opciones para make

`-make [-f fichero] [opción. . .] [variable=[valor] . . .] [objetivo . . .]`

objetivo El objetivo que va a construirse. Por omisión, el primero del *makefile*.

macro= Anular la macro especificada.

macro=valor Dar el valor especificado a la macro nombrada; preferencia sobre definiciones que hubiera en el *makefile*.

`-f fichero` El fichero especificado se convierte en el *makefile*. Por omisión, en Unix se buscan *makefile* o *Makefile*.

`-i` Idéntica a un objetivo `.IGNORE`; se soslayan todos los errores de órdenes.

`-k` Continúa tras un error tanto como sea posible, intentando otros objetivos que no dependen de aquel donde se ha producido el fallo.

`-n` Muestra las órdenes que se ejecutarían, pero no las ejecuta realmente.

`-p` Muestra la base de datos (reglas implícitas y macros predefinidas) que resultade leer el *makefile*.

`-q` Sólo devuelve un *status*: cero (éxito) si el objetivo a construirse estaba actualizado; no-cero si no.

`-r` Elimina todas las reglas implícitas y la lista de sufijos. Equivale a escribir un objetivo `.SUFFIXES` sin dependencias.

`-s` Con *ese* de silencio. No muestra las órdenes al ejecutarse. Equivale a un objetivo `.SILENT`

`-t` Actualiza los ficheros pero sin modificarlos realmente.

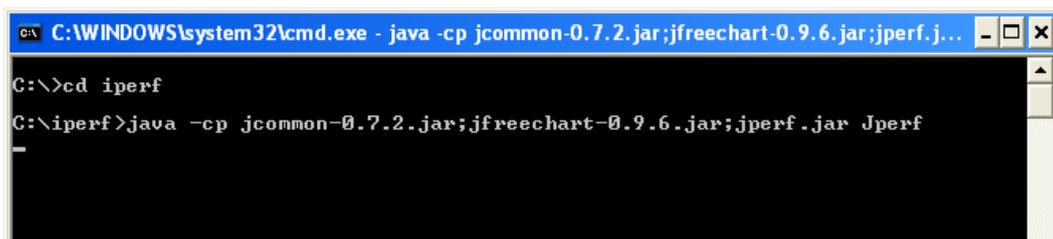
ANEXO 4]

[Manual de uso del Jperf/Iperf en Windows]

Antes que nada, para usar la versión gráfica del Jperf se debe descargar e instalar el Iperf y además tener el JAVA instalado. Para instalar Iperf en Windows podemos seguir este enlace de la web: <http://www.noc.ucf.edu/Tools/Iperf/> para así descargar el

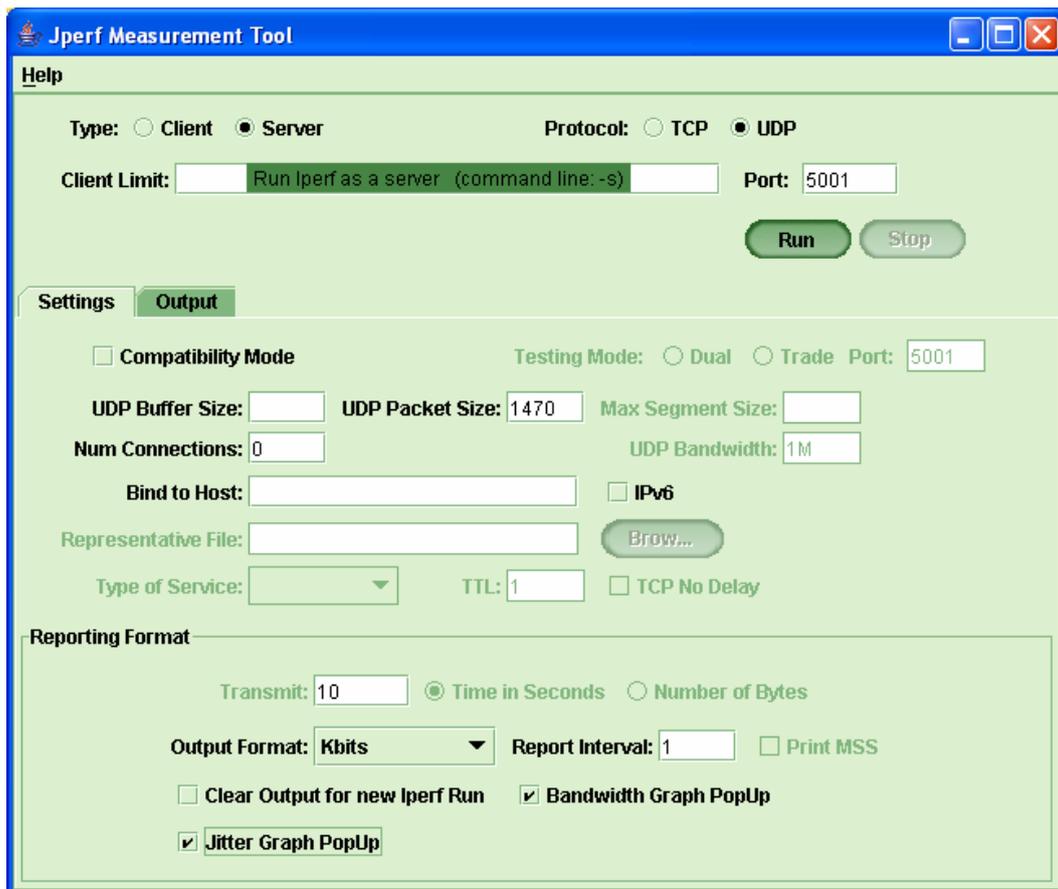
ejecutable iperf.exe. Una vez descargado se le oprime doble click al icono de Iperf y se permite la instalación.

Para empezar Jperf, se pulsa el botón de inicio y se sigue la siguiente secuencia Inicio→ejecutar→cmd. Desde acá se ingresa al directorio en el que iperf se ha instalado y se introduce el siguiente comando: "*java -cp jcommon-0.7.2.jar;jfreechart-0.9.6.jar;jperf.jar Jperf*" y luego clic en Aceptar.



```
C:\WINDOWS\system32\cmd.exe - java -cp jcommon-0.7.2.jar;jfreechart-0.9.6.jar;jperf.j...
C:\>cd iperf
C:\iperf>java -cp jcommon-0.7.2.jar;jfreechart-0.9.6.jar;jperf.jar Jperf
```

Para recibir tráfico de otra computadora a la computadora en la que se está trabajando, se selecciona *Server*.



También debe seleccionarse Jitter Graph PopUp en el apartado de formato de reporte. Luego se pulsa run cuando se quiere iniciar el experimento. Así queda este ordenador configurado como servidor.

Ahora en la computadora cliente, para enviar tráfico al servidor, se selecciona *Client*. En seguida se escribe la dirección del servidor al que se quiere enviar tráfico de datos. Ahora se configura el tipo de tráfico que queremos inyectar. Si se posee un archivo específico para la transmisión se le da a *Browser* y luego se escoge el archivo, si solo se quiere inyectar tráfico aleatorio se deja en blanco. Luego se decide el tamaño del paquete UDP, en el campo *UDP Bandwidth*.

En *Testing mode* se puede tener tráfico en ambos sentidos escogiendo *Dual*. Si *trade* está seleccionado la prueba se hará en un sentido y luego en el otro. Por último se selecciona el tiempo de prueba en el campo: *Transmit*.

Luego se presiona run, y así comienzan las pruebas.