ANEXOS

1. Sistema Operativo Voyage.

Es una distribución de Linux que se deriva del sistema Debian, ideal para operar en dispositivos embebidos x86 tales como ALIX, WRAP, Soekris 45xx/48xx o boards Atom. Para una instalación típica tan solo se requiere de 128 MB de espacio en disco, es por esto que resulta muy apropiada para implementar un firewall, un punto de acceso inalámbrico, un gateway Asterisk/VoIP, un reproductor de música o un dispositivo de almacenamiento en red. Por estos motivos fue este el sistema operativo, en su versión 0.6.5, el seleccionado para ser instalado sobre las tarjetas ALIX usadas en este proyecto como tarjetas de enrutamiento y componentes del segmento WiFi.

2. Monitoreo de red - iPerf.

Se trata de una herramienta multiplataforma, disponible para Windows, Linux, Mac OS, entre otros, utilizada para realizar mediciones del ancho de banda disponible en la red. En el caso de Linux y particularmente en las distribuciones utilizadas en este proyecto, esta herramienta se encuentra disponible y puede ser fácilmente utilizada desde la consola del sistema.

3. Script de configuración de tarjetas ALIX.

```
#!/bin/bash

#############################################################################
## Script para configuración de colas IP en tarjeta ALIX 3
## Ing. Ivan Eduardo Hernandez
## Julian Andres Rojas
## Universidad del Cauca
## V3.0
## El presente Script se ha diseñado para implementar colas HTB sobre dispositivos de
## enrutamiento bajo la plataforma Linux, mediante la utilización de la herramienta tc del
## paquete iproute2. Las colas HTB se han planteado en torno a 4 tipos de tráfico: H_VO, H_VI,
## H_BK y H_BE. En cuanto a qué porción de ancho de banda se asigna a cada tipo de tráfico,
## se debe tener en cuenta que estos valores dependen de la demanda que exista de los
```

servicios por los usuarios que existan en la red en un caso concreto. Por lo tanto dichos
valores podrán ser modificados cada vez que se ejecute el script.
##

apagando EDCA
iwpriv ath0 wmm 0

Asiganando el bitrate del enlace WiFi
iwconfig ath0 rate 18M

UPLINK

Eliminar la configuración anterior
tc qdisc del dev ath0 root 2> /dev/null
tc qdisc del dev eth0 ingress 2> /dev/null

Ingress qdisc

Una cola en ingress para agilizar el paso de los paquetes de alta prioridad hacia la egress
qdisc.

tc qdisc add dev eth0 handle ffff: ingress

tc filter add dev eth0 parent ffff: protocol ip prio 1 u32 match ip tos 0xb8 0xff flowid :1 police rate 4900kbit burst 50kb drop

Egress qdisc

unidad=kbit
tc qdisc add dev ath0 root handle 1: htb default 10
Clase raiz ("bit" = bps, "bps" = Bps)
echo "ingrese rate (en kbit) para la clase raiz"
read rateraiz
echo "ingrese ceil (en kbit) para la clase raiz"
read ceilraiz
tc class add dev ath0 parent 1: classid 1:1 htb rate $rateraiz$unidad ceil $ceilraiz$unidad
Trafico QoS
echo "ingrese rate (en kbit) para la clase QoS"
read rateqos
echo "ingrese ceil (en kbit) para la clase QoS"
read ceilqos

```
tc class add dev ath0 parent 1:1 classid 1:12 htb rate $rateqos$unidad ceil $ceilqos$unidad
burst 17kb

#### TRAFICO H_VO
## Equivalencias de tipos de servicio según la tecnología
## DiffServ - WiMax - WiFi
## EF - UGS - AC_VO
echo "ingrese rate (en kbit) para el flujo VO"
read ratevo
echo "ingrese ceil (en kbit) para el flujo VO"
read ceilvo
tc class add dev ath0 parent 1:12 classid 1:122 htb rate $ratevo$unidad ceil $ceilvo$unidad
burst 17kb prio 0
tc qdisc add dev ath0 parent 1:122 handle 122: sfq perturb 10
tc filter add dev ath0 parent 1:0 protocol ip prio 1 u32 match ip tos 0xb8 0xff flowid 1:122

#### TRAFICO H_VI
## AF4x(1) - rtPS - AC_VI
echo "ingrese rate (en kbit) para el flujo VI"
read ratevi
echo "ingrese ceil (en kbit) para el flujo VI"
read ceilvi
tc class add dev ath0 parent 1:12 classid 1:121 htb rate $ratevi$unidad ceil $ceilvi$unidad
burst 17kb prio 1
tc qdisc add dev ath0 parent 1:121 handle 121: sfq perturb 10
tc filter add dev ath0 parent 1:0 protocol ip prio 2 u32 match ip tos 0xa0 0xff flowid 1:121

#### TRAFICO H_BK
## AF2x(1) - nrtPS - AC_BK
echo "ingrese rate (en kbit) para el flujo BK"
read ratebk
echo "ingrese ceil (en kbit) para el flujo BK"
read ceilbk
tc class add dev ath0 parent 1:12 classid 1:120 htb rate $ratebk$unidad ceil $ceilbk$unidad
burst 34kb prio 2
tc qdisc add dev ath0 parent 1:120 handle 120: sfq perturb 10
tc filter add dev ath0 parent 1:0 protocol ip prio 3 u32 match ip tos 0x20 0xff flowid 1:120
```

TRAFICO H_BE
BE - BE - AC_BE
echo "ingrese rate (en kbit) para el flujo BE"
read ratebe
echo "ingrese ceil (en kbit) para el flujo BE"
read ceilbe
tc class add dev ath0 parent 1:1 classid 1:11 htb rate $ratebe$unidad ceil $ceilbe$unidad burst 17kb prio 3
tc qdisc add dev ath0 parent 1:11 handle 11: sfq perturb 10
tc filter add dev ath0 parent 1:0 protocol ip prio 4 u32 match ip tos 0x00 0xff flowid 1:11

DOWNLINK

Eliminar la configuración anterior
tc qdisc del dev eth0 root 2> /dev/null
tc qdisc del dev ath0 ingress 2> /dev/null

Ingress qdisc

tc qdisc add dev ath0 handle ffff: ingress

TRAFICO H_VO
EF - UGS - AC_VO

tc filter add dev ath0 parent ffff: protocol ip prio 1 u32 match ip tos 0xb8 0xff flowid :1 police rate $ceilvo$unidad burst 17kb drop

TRAFICO H_VI
AF4x(1) - rtPS - AC_VI
tc filter add dev ath0 parent ffff: protocol ip prio 2 u32 match ip tos 0xa0 0xff flowid :1 police rate $ceilvi$unidad burst 17kb drop

TRAFICO H_BK
AF2x(1) - nrtPS - AC_BK
tc filter add dev ath0 parent ffff: protocol ip prio 3 u32 match ip tos 0x20 0xff flowid :1 police rate $ceilbk$unidad burst 17kb drop

TRAFICO H_BE
BE - BE - AC_BE

```
tc filter add dev ath0 parent ffff: protocol ip prio 4 u32 match ip tos 0x00 0xff flowid :1 police
rate $ceilbe$unidad burst 17kb drop

## ########################################################## ################

echo "La configuracion realizada es:"
tc -s -d qdisc show dev ath0
tc -s -d class show dev ath0
tc -s -d filter show dev ath0
iwpriv ath0 get_wmm
iwlist ath0 rate
ip route show
```

4. Generador de tráfico D-ITG.

Por sus siglas: “Distributed Internet Traffic Generator”, es una aplicación capaz de generar tráfico a nivel de paquetes, simulando una conexión real a internet.

Esta cuenta con una serie de herramientas que entre otras cosas permite elegir el tipo de tráfico que se va a generar, el protocolo (IPv4 o IPv6) mediante el cual se construirán los paquetes, entre otras configuraciones, con lo cual se logran simular flujos de información con distintas prioridades de QoS.

Por otro lado, por medio de esta aplicación, también se pueden realizar mediciones en cuanto a jitter, delay, troughput y packet loss sobres las transmisiones realizadas a partir de los distintos flujos de tráfico generados. Esta información es presentada por la aplicación a través de la consola del sistema de manera organizada y fácil de leer. Es por esto, que esta herramienta resulta muy interesante e importante en el desarrollo de este proyecto, pues es a partir de esta que se generará toda la data a ser analizada para corroborar la validez de la propuesta de integración planteada.

5. Script para generar tráfico mediante D-ITG.

```
#! /bin/bash

#####################################################
#Script para pruebas con D-ITG y Octave
#Lanzamiento de flujos y Graficas automáticamente.
```

```
#Agosto 2013
#Ivan Hernandez - Julian Andres Rojas
#Universidad del Cauca
#V1.0_____
#Copiar el contenido de lanzar.zip dentro de
#la carpeta bin
#####################################################
#          DATOS DE ENTRADA
#    resultados en la carpeta /Resultados
#####################################################
echo "**********INICIA SCRIPT DITG**********"
name=Pruebaglobal4    #NOMBRE DE LA PRUEBA SE SUGUIERE MENOS DE 10 CARACTERES
t=200            #Tiempo en ms para el muestreo del grafico de Bitrate,Delay,Jitter,Packet loss
num_flujos=3               #Numero de flujos
dirrec=10.10.0.4                  # Direccion logs
splee=115    # 350% mas que el tiempo de la prueba(prueba+25sg)
dirsendremoto=10.10.0.14     ##DIRECCION USR Y PSW DEL SEND REMOTO
usr1="lab6-usuario"
psw1="M4ST3R.c0mp4d"
dirrecgest=10.10.0.15              # Direccion de gestion del pc RECV
#############################################################
dirtarget=10.10.21.3                ##DIRECCION PC DESTINO
flujos_print=[1]   #Que flujos desea imprimir,completar el vector ejm: [1 3 5] imprime flujos 1,3
y 5
flujo[1]="-T UDP -a ${dirtarget} -m RTTM -rp 4200 -sp 4202 -C 550 -c 500 -t 90000 " # flujo be
flujo[2]="-T UDP -a ${dirtarget} -m RTTM -rp 4300 -sp 4303 -C 550 -c 500 -t 90000 " # flujo be
flujo[3]="-T UDP -a ${dirtarget} -m RTTM -rp 4400 -sp 4404 -C 550 -c 500 -t 90000 " # flujo be

########################## INICIO ################################
#echo “ingrese el nombre del log con extencion .log”
#read envio
rm flujos
rm flujos.lst
sender=sender.log   ##Dejar igual
receiver=reciver.log ## Dejar igual
fecha=`date +"%F_%T_"`
declare -a flujo
flow=flujos.lst
```

```
touch $flow
for valor in 1 2 3 4
do
echo ./ITGManager $dirsendremoto "${flujo[$valor]}" "&">>flujos
done
./1_remote_conexion.sh $dirsendremoto $usr1 $psw1 /home/lab6-
usuario/Pruebas_Unicauca_Colombia_pc2/D-ITG-2.8.0-rc1/bin "./ITGSend -Q -x reciver.log -l
sender.log -X ${dirrec} -L ${dirrec}"
sleep 1
./1_remote_conexion.sh $dirrecgest $usr1 $psw1 /home/lab6-
usuario/Pruebas_Unicauca_Colombia/D-ITG-2.8.0-rc1/bin "./ITGRecv "
sleep 1
bash flujos
sleep $splee
./1_remote_conexion.sh $dirsendremoto $usr1 $psw1 /home/lab6-
usuario/Pruebas_Unicauca_Colombia_pc2/D-ITG-2.8.0-rc1/bin "./matar_proceso.sh ITG"
./1_remote_conexion.sh $dirrecgest $usr1 $psw1 /home/lab6-
usuario/Pruebas_Unicauca_Colombia/D-ITG-2.8.0-rc1/bin "./matar_proceso.sh ITG"
#./ITGSend flujos.lst -l $sender -x $receiver -X $dirrec
rm flujos
rm flujos.lst
sleep 2
./ITGDec $sender -v -j $t -p $t -b $t -d $t >resultadoRTTM$1.txt
./ITGDec $receiver >resultadoOWDM$1.txt
if `cd ../Resultados >/dev/null`
then
echo “Resultados Existe”
cd ..
else
cd ..
mkdir Resultados
fi
cd Resultados
mkdir $fecha$name
cd ../bin
mv $sender ../Resultados/$fecha$name
mv $receiver ../Resultados/$fecha$name
mv resultadoRTTM$1.txt ../Resultados/$fecha$name
```

```
mv resultadoOWDM$1.txt ../Resultados/$fecha$name
numa=$1
mv jitter.txt ../Resultados/$fecha$name/jitter$numa.txt
mv delay.txt ../Resultados/$fecha$name/delay$numa.txt
mv bitrate.txt  ../Resultados/$fecha$name/bitrate$numa.txt
mv packetloss.txt ../Resultados/$fecha$name/packetloss$numa.txt
cp ITGplot ../Resultados/$fecha$name
cd ../Resultados/$fecha$name
#octave ITGplot delay$numa.txt $flujos_print
#octave ITGplot jitter$numa.txt $flujos_print
#octave ITGplot bitrate$numa.txt $flujos_print
#octave ITGplot packetloss$numa.txt $flujos_print
rm ITGplot
clear
cat resultadoRTTM$1.txt
```

6. Archivos Logs generados por D-ITG.

Durante la ejecución de los escenarios de prueba, para cada flujo de cada tipo de tráfico, se toman alrededor de 450 medidas. A continuación se presenta una muestra de los archivos “logs” generados por D-ITG:

bitrate.txt

```
Time RP-1400 RP-1200 RP-1300 Aggregate-Flow
0.0000000000000000e+00 1.1920000000000000e+03 1.1580000000000000e+03 1.1660000000000000e+03 3.5160000000000000e+03
1.9999999999999833e-01 1.1960000000000000e+03 1.1980000000000000e+03 1.2000000000000000e+03 3.5940000000000000e+03
3.9999999999999667e-01 1.1880000000000000e+03 1.1900000000000000e+03 1.1840000000000000e+03 3.5620000000000000e+03
5.9999999999998668e-01 1.1880000000000000e+03 1.1860000000000000e+03 1.1980000000000000e+03 3.5720000000000000e+03
7.9999999999999334e-01 1.1880000000000000e+03 1.1940000000000000e+03 1.1840000000000000e+03 3.5660000000000000e+03
1.0000000000000000e+00 1.2000000000000000e+03 1.1920000000000000e+03 1.2100000000000000e+03 3.6020000000000000e+03
1.1999999999999734e+00 1.2140000000000000e+03 1.2160000000000000e+03 1.1880000000000000e+03 3.6180000000000000e+03
1.4000000000000133e+00 1.2000000000000000e+03 1.1980000000000000e+03 1.2140000000000000e+03 3.6120000000000000e+03
1.5999999999999867e+00 1.2080000000000000e+03 1.2140000000000000e+03 1.1780000000000000e+03 3.6000000000000000e+03
1.8000000000000266e+00 1.1960000000000000e+03 1.1940000000000000e+03 1.2220000000000000e+03 3.6120000000000000e+03
2.0000000000000000e+00 1.2060000000000000e+03 1.2080000000000000e+03 1.1980000000000000e+03 3.6120000000000000e+03
2.1999999999999734e+00 1.2120000000000000e+03 1.2060000000000000e+03 1.2140000000000000e+03 3.6320000000000000e+03
2.3999999999999467e+00 1.1980000000000000e+03 1.2020000000000000e+03 1.2000000000000000e+03 3.6000000000000000e+03
2.6000000000000533e+00 1.2120000000000000e+03 1.2000000000000000e+03 1.1940000000000000e+03 3.6060000000000000e+03
2.8000000000000266e+00 1.1900000000000000e+03 1.2020000000000000e+03 1.2060000000000000e+03 3.5980000000000000e+03
3.0000000000000000e+00 1.2120000000000000e+03 1.2020000000000000e+03 1.1960000000000000e+03 3.6100000000000000e+03
3.1999999999999734e+00 1.1960000000000000e+03 1.2080000000000000e+03 1.2160000000000000e+03 3.6200000000000000e+03
3.3999999999999467e+00 1.2160000000000000e+03 1.2020000000000000e+03 1.2000000000000000e+03 3.6180000000000000e+03
3.6000000000000533e+00 1.2020000000000000e+03 1.2080000000000000e+03 1.2000000000000000e+03 3.6100000000000000e+03
```

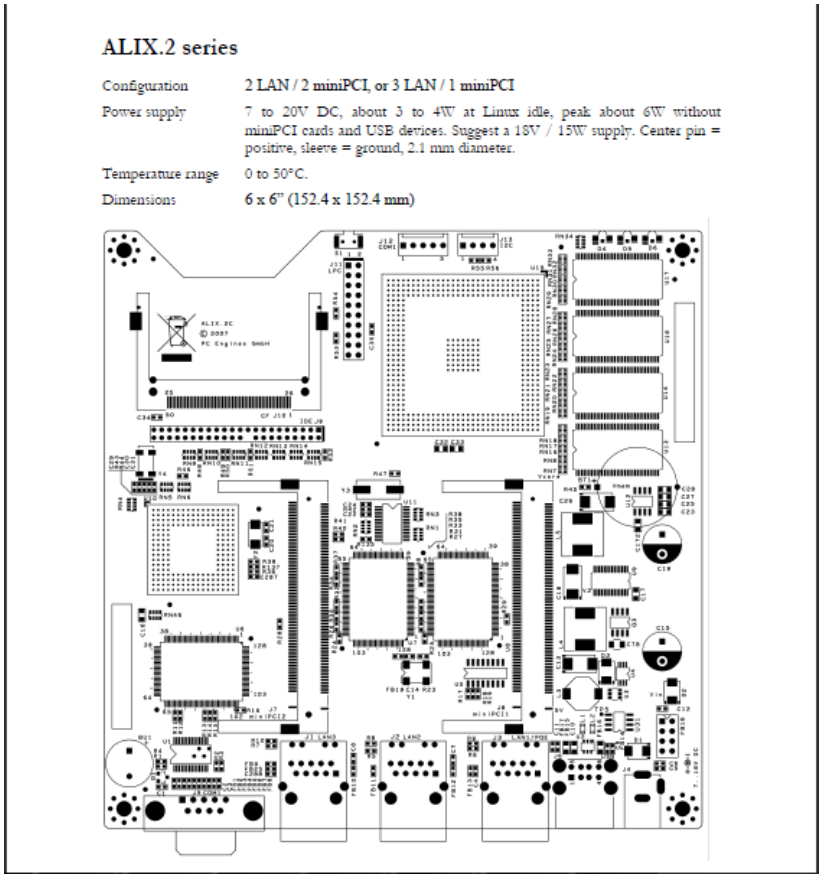
Figura 1 Mediciones de BitRate - escenario B - flujos de H_VO.

delay.txt

```
Time RP-2200 Aggregate-Flow
0.00000000000000000e+00 9.57887999999999340e-02 9.57887999999999340e-02
1.99999999999999833e-01 2.00372599999999841e-01 2.00372599999999841e-01
3.99999999999999667e-01 2.25528800000000292e-01 2.25528800000000292e-01
5.99999999999998668e-01 2.31403699999999900e-01 2.31403699999999900e-01
7.99999999999999334e-01 2.30904499999999848e-01 2.30904499999999848e-01
1.00000000000000000e+00 2.32266000000000002e-01 2.32266000000000002e-01
1.19999999999999734e+00 2.32421600000000337e-01 2.32421600000000337e-01
1.40000000000000133e+00 2.31445999999999851e-01 2.31445999999999851e-01
1.59999999999999867e+00 2.31561199999999671e-01 2.31561199999999671e-01
1.80000000000000266e+00 2.32894800000000129e-01 2.32894800000000129e-01
2.00000000000000000e+00 2.31088800000000386e-01 2.31088800000000386e-01
2.19999999999999734e+00 2.33568599999999873e-01 2.33568599999999873e-01
2.39999999999999467e+00 2.33046100000000060e-01 2.33046100000000060e-01
2.60000000000000533e+00 2.33046199999999811e-01 2.33046199999999811e-01
2.80000000000000266e+00 2.33200099999999657e-01 2.33200099999999657e-01
3.00000000000000000e+00 2.33189499999999940e-01 2.33189499999999940e-01
3.19999999999999734e+00 2.33479000000000476e-01 2.33479000000000476e-01
3.39999999999999467e+00 2.33890300000000232e-01 2.33890300000000232e-01
3.60000000000000533e+00 2.31687500000000462e-01 2.31687500000000462e-01
```

Figura 2 Mediciones de Delay - escenario A - flujo de H_VI.

7. Especificación de equipos utilizados.



ALIX.2 series

Configuration	2 LAN / 2 miniPCI, or 3 LAN / 1 miniPCI
Power supply	7 to 20V DC, about 3 to 4W at Linux idle, peak about 6W without miniPCI cards and USB devices. Suggest a 18V / 15W supply. Center pin = positive, sleeve = ground, 2.1 mm diameter.
Temperature range	0 to 50°C.
Dimensions	6 x 6" (152.4 x 152.4 mm)

Figura 3 Datasheet tarjeta ALIX 2D2

1.4. System specifications

Radio parameters			
Frequency Band		5470-5725 MHz (ETSI) or 5725-5875 MHz (FCC). See ordering options.	
Modulation		OFDM IEEE 802.16-2009 - 256 subcarriers, cyclic prefix 1/4, 1/8, 1/16 or 1/32	
Supported channel bandwidth		1.75, 3.5, 7 and 10 MHz	
Adaptive modulation		BPSK, QPSK, 16QAM and 64QAM	
FEC code rate		1/2, 2/3 and 3/4 concatenated Reed-Solomon and *Viterbi*	
Maximum output power		+26 dBm (+22 dBm for 64QAM modulations)l	
Transmit power control		> 40 dB	
Duplexing method		TDD (Time Division Duplexing)	
Uplink/Downlink allocation		Programmable from 4:1 to 1:4	
Dynamic Frequency Selection		Yes	
TDD synchronization		External or internal references (10 MHz, 1pps). Requires ARBA-IDU unit	
Antenna connector		N-type, 50 ohms	
RF parameters	**Modulation**	**Sensitivity (1.75 MHz)**	**Sensitivity (10 MHz)**
	BPSK-1/2	-99.5 dBm	-92 dBm
	QPSK-1/2	-96.5 dBm	-89 dBm
	QPSK-3/4	-94 dBm	-86.5 dBm
	16QAM-1/2	-91 dBm	-83.5 dBm
	16QAM-3/4	-87.5 dBm	-80 dBm
	64QAM-2/3	-83.5 dBm	-76 dBm
	64QAM-3/4	-81.5 dBm	-74 dBm
Data traffic and Throughput			
Maximum over-the-air data rate		37.7 Mbps (64QAM-3/4, 10 MHz BW)	
Ethernet aggregated throughput	Basic	20 Mbps	
	Advanced (1)	34.4 Mbps (64QAM-3/4, 10 MHz BW)	
ARQ support		Yes, per IEEE 802.16-2009 standard - Selectable per service flow	
Simultaneous registered users	Basic	20	
	Advanced (2)	Unlimited	
Encryption		AES and 3DES	
Quality of Service (QoS)			
Supported QoS types		UGS, RTPS, nRTPS, eRTPS and BE (IEEE 802.16-2009 standard)	
Service differentiation	Layer-2	MAC source/destination address, EtherType, VLAN tag	
	Layer-3	DSCP ToS, IP source/destination address and subnet, Protocol type	
	Layer-4	TCP, UDP source/destination port range	
Differentiated service flows	Basic	One bidirectional service per user	
	Advanced (3)	Unlimited differentiated services per user	
Management and Provisioning			
Management local interfaces		Web, Command-Line Interface, RS232	
Management remote interfaces		SNMP, XML-RPC	
User and services local provisioning		XML local database	
User and services centralized provisioning		AAA Radius, LDAP, XML-RPC	
Network functionality			
Layer-2 Network functionality		Bridging (IEEE 802.1), VLAN (IEEE 802.1q)	
Layer-3 Network functionality		Static/Dynamic routing, NAT, DHCP server/client	
Supported CS		Ethernet, IPv4oEthernet, VLAN, IPv4oVLAN	
Networking modes		Bridge mode, IP routing	
Data interface		10/100 Base-T Ethernet RJ45	
Physical, Mechanical and Electrical			
Size		395 x 265 x 95 mm	
Outdoor Unit Weight		3.2 kg	
Power Supply	Basic	802.3af compliant (PoE)	
	Optional	12 or 48 Volts (separate connector for solar panel supply)	
Power Consumption		<18 Watts (full traffic conditions)	
Standards Compliance			
WiMAX		IEEE 802.16-2009 + Corrigendum IEEE 802.16-2005	
Radio		ETSI EN 301 893, ETSI EN 302 502	
Environmental		ETSI EN 300 019-1-4 C4.1E (ODU), ETSI EN 300 019-1-3 C3.2 (IDU)	

Figura 4 Especificaciones de sistema de la BS ARBA550.