



**Máster en Redes de Telecomunicación
para Países en Desarrollo**

**ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN**

PROYECTO FIN DE MÁSTER

**Estudio de un esquema de WiFi-bonding sobre
dispositivos empotrados con SO GNU/Linux para
redes WiFi de topología lineal para zonas rurales
aisladas de países en desarrollo**

**Autor: Geovana Navarro Mamani
Tutor: Francisco Javier Simó Reigadas
Co-tutor: Ignacio Foche Pérez**

Curso académico 2011/2012

ACTA DEL TRABAJO FIN DE MÁSTER

DATOS DE ESTUDIO DEL MÁSTER

ESTUDIOS CURSADOS: MÁSTER EN REDES DE TELECOMUNICACIÓN PARA PAÍSES EN DESARROLLO		
CURSO ACADÉMICO: 2011 / 2012		
CONVOCATORIA:	Ordinaria <input type="checkbox"/>	Extraordinaria <input type="checkbox"/> Especial de Finalización <input type="checkbox"/>

DATOS DEL ALUMNO

APELLIDOS: Navarro Mamani		NOMBRE: Geovana
DNI / Pasaporte: 5525574	E-mail: geovana.n@gmail.com	Teléfono: 626383269

TÍTULO DEL TRABAJO FIN DE MÁSTER

Estudio de un esquema de WiFi-bonding sobre dispositivos empotrados con SO GNU/Linux para redes WiFi de topología lineal para zonas rurales aisladas de países en desarrollo

DIRECTOR/ES (Obligatorio)

DNI	NOMBRE Y APELLIDOS	UNIVERSIDAD/INSTITUCIÓN
	Francisco Javier Simó Reigadas	Universidad Rey Juan Carlos
77.340.673E	Ignacio Foche Pérez	Centro Nacional de Biotecnología (CSIC)

MIEMBROS DEL TRIBUNAL

ACTÚA EN CALIDAD DE

	Presidente/a
	Vocal/es
	Secretario/a
	Suplente
	Suplente
	Suplente

Reunido el Tribunal de Evaluación con fecha, ACUERDA otorgar al alumno la calificación global de

Indicar, en su caso, si se propone la concesión de la mención Matrícula de Honor.

EL PRESIDENTE/A	EL SECRETARIO/A	VOCAL/ES
Fdo:	Fdo:	Fdo:

Copyright ©2012 Geovana Navarro Mamani. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Agradecimientos y reconocimientos

Después de varios meses de un intenso trabajo, por fin este PFM ha llegado a su fin; esto me llena de alegría porque implica el final de una etapa de mi vida donde he experimentado diferentes sensaciones como: desesperación; cuando los resultados que se obtenía no eran satisfactorios, alegrías; después de haber resuelto alguna falla en el PFM, ilusión; en que este trabajo tenía que llegar a su fin. Lo mejor que me llevo de este PFM es el aprendizaje obtenido

En primer lugar te agradezco a ti Dios, por ayudarme a terminar este proyecto, gracias por darme la fuerza y el coraje para finalizar este PFM, por darme la vida, por estar conmigo en cada momento de mi vida. Por cada regalo de gracia que me has dado.

A Javier Simó, por toda su valiosa ayuda y tutela para la realización de este trabajo; También por su amistad, consejos y recomendaciones. Gracias por ser mi tutor.

A Javier Simó, por la enseñanza brindada durante el máster y por ser el tutor de este proyecto. Muchas gracias!!!

A Ignacio Foche Pérez, que ha sido mi co-tutor, pues si él esto no hubiera sido posible; Nacho, toda mi gratitud y reconocimiento por todo el apoyo que me brindaste, por tus cosejos, tu paciencia, por las palabras de ánimo “por hacerme sentir que si podía terminar este PFM” y por las observaciones, siempre oportunas.

A los profesores del Máster COMPAD, por todo el conocimiento transmitido, por su amistad y apoyo en todo momento y a Universidad Rey Juan Carlos, por abrirme sus puertas.

A mis padres: Mario y Rogelia por su apoyo, sus consejos, hacerme sentir que todo se puede lograr con esfuerzo y dedicación. Aún en la distancia senti su amor.

A mis hermanos, abuelos, tios y al resto de mi familia, gracias por las palabras de ánimo y su cariño incondicional. En especial a mis tios Rosmery, Reina y Teo, por brindarme ese cariño de hogar y hacer que me sienta como parte de su familia.

A Ivan, a Miriam, a Juanita, a Salatiel, a Liz...y a tantos compañeros del máster que han tenido un hueco especial en mi vida durante el máster.

A Borja, Nacho Sequi, Mari Carmen, María, Erminia y Freddy por brindarme su amistad y hacer que mi estancia en Madrid sea mas placentera. Muchas gracias a todos...

Índice general

I. INTRODUCCIÓN	1
1. Organización del documento	3
2. Antecedentes, contexto y justificación del trabajo	5
3. Marco Teórico	11
3.1. Fundación EHAS	11
3.1.1. La red WiLD en el río Napo (Perú)	12
3.1.2. La red Wi-Fi del Cuzco (Perú)	12
3.2. Software libre y licencias de software	14
3.3. Sistema Operativo GNU/Linux	19
3.3.1. Arquitectura del Sistema Operativo Linux	19
3.3.2. Funciones de intercambio entre el espacio de usuario y el espacio de kernel	20
3.4. Controlador MadWifi	21
3.4.1. Capa Abstraccion del Hardware de MadWiFi	21
3.4.2. Modos Operativos	22
3.4.3. Cifrado	22
3.4.4. Super A/G	22
3.4.5. Controlador Ath5k	22
3.5. Bonding	23
3.6. Enrutamiento avanzado y control de tráfico en Linux	24
3.6.1. Iptables	24
3.6.2. Iproute2	24
4. Estado del arte de balanceo de carga en redes inalámbricas	27
II. OBJETIVO DEL PROYECTO	29
5. Objetivo del proyecto	31
5.1. Objetivo General	31
III. MÉTODOS Y MATERIALES	33
6. Métodos	35
6.1. Requisitos de selección de la solución adecuada para el balanceo de carga	37
7. Materiales	39
7.1. Materiales de hardware	39
7.1.1. Hardware para el escenario de experimentación	39
7.2. Materiales de software	39
IV. RESULTADOS	41
8. Selección de componentes de la estación Wifi	43
8.1. Hardware de la estación WiFi	43
8.1.1. Placas de cómputo	43

8.1.2. Módulo WiFi (802.11a)	43
8.1.3. Antenas dipolo	44
8.2. Software de la estación WiFi	45
8.2.1. Sistema Operativo GNU/Linux Voyage 0.6.5	45
9. Diseño de la red	47
9.1. Componentes y configuración del diseño de Red	47
9.2. Parámetros de configuración en común para las soluciones	49
9.2.1. Tasa de transmisión y paquetes a transmitir	49
9.2.2. Estandar y canales de radiofrecuencia a utilizar	49
9.2.3. Duración de la prueba	49
9.2.4. Cantidad de pruebas a realizar	51
9.2.5. Escenario para medir las pruebas del desarrollo del piloto del laboratorio	51
9.3. Script para preparar el escenario del piloto del laboratorio	51
9.4. Script Automatización de las pruebas de tráfico para el piloto del laboratorio . .	51
9.5. Criterio para las mediciones del trabajo	53
9.6. Criterio para medir QoS de la técnica del balanceo escogido	53
10. Solución de WiFi-Bonding	55
10.1. Desarrollo del piloto del laboratorio del bonding	55
10.1.1. Parámetros de configuración del piloto del laboratorio de la solución bonding	55
10.1.2. Configuración de bonding	55
10.2. Fases de pruebas para llegar a la solución final de bonding	55
10.2.1. Primera prueba de bonding	56
10.2.2. Segunda prueba de bonding	57
10.2.3. Tercera prueba de bonding	59
10.2.4. Cuarta prueba de bonding en redes ethernet	59
10.2.5. Quinta prueba de bonding	60
10.3. Automatización de configuración de la solución bonding	61
10.4. Ejecución de las pruebas de la solución bonding	61
10.4.1. Primer escenario, enviando los paquetes en un sólo flujo	63
10.4.2. Segundo escenario, enviando los paquetes en dos flujos	63
10.5. Pruebas de bonding en los diferentes modos	63
10.5.1. active-backup o 1	63
10.5.2. balance-xor o 2	64
10.5.3. broadcast o 3	64
10.5.4. 802.3ad o 4	64
10.5.5. balance_tbl o 5 y balance_al o 6	66
11. Solución de balanceo de carga a nivel de red	67
11.1. Desarrollo del piloto del laboratorio del balanceo de carga a nivel de red	67
11.1.1. Parametros de configuración del piloto del laboratorio de la solución del	67
balanceo de carga a nivel de red	67
11.1.2. Configuración del balanceo de carga a nivel red	67
11.2. Fases de pruebas para llegar a la solución final del balanceo de carga a nivel de red	67
11.2.1. Primera prueba del balanceo de carga en el nivel de red - Shorewall	67
11.2.2. Segunda prueba del balanceo de carga a nivel de red - configuración manual	68
11.2.3. Tercera prueba del balanceo de carga a nivel de red	68
11.2.4. Cuarta prueba del balanceo de carga a nivel de red	68
11.3. Automatización de configuración de la solución del balanceo de carga a nivel de red	69
11.4. Ejecución de prueba de la solución balanceo de carga a nivel de red	70
11.4.1. Primer escenario, enviando un sólo flujo	70
11.4.2. Segundo escenario, enviando los paquetes en dos flujos	70
12. Resultados de experimentación	73
12.1. Primer escenario, enviando un sólo flujo.	73
12.1.1. Comparativa del retardo	73
12.1.2. Comparativa de la variación del retardo	75
12.1.3. Comparativa del caudal	75

12.1.4. Comparativa de la pérdida de paquetes	76
12.2. Segundo escenario, enviando en dos flujos.	77
12.2.1. Comparativa del retardo	77
12.2.2. Comparativa del variación de retardo	78
12.2.3. Comparativa del caudal	79
12.2.4. Comparativa de la pérdida de paquetes	81
12.3. Conclusiones de las comparativas de las dos soluciones.	81
13. Cronograma y presupuesto	83
13.1. Cronograma del proyecto	83
13.2. Costes de la investigación	83
13.2.1. Coste de recurso humano	83
13.2.2. Coste de materiales	83
V. CONCLUSIONES Y LÍNEAS FUTURAS	87
14. Discusión de los resultados	89
15. Conclusiones	91
16. Líneas futuras de desarrollo e investigación	93
BIBLIOGRAFÍA	93
ANEXOS	97
A. Automatización de configuración del prototipo para bonding y el enrutamiento de control de tráfico	101
B. Automatización de las pruebas de tráfico	111
C. Resultados de bonding en modo 2	113
D. Resultados de bonding en modo 3	115
E. Resultados de bonding en modo 1	117
GNU Free Documentation License	119

Índice de figuras

2.1. Promedio de Asistencia Oficial para el desarrollo	6
2.2. Cantidad de líneas fijas y móviles	7
2.3. Cantidades de usuarios de internet	7
2.4. Abonados de banda ancha, fija y móvil	8
3.1. Esquema técnico de la red Rio Napo en 2009	13
3.2. Topología detallada de un segmento de la red Napo.	14
3.3. Red Wi-Fi de Cuzco	15
3.4. Arquitectura Mesh de una red Wi-Fi EHAS	16
3.5. Mapa conceptual del Software Libre	17
3.6. Categorías de software según las libertades de sus licencias	18
3.7. Arquitectura del SO Linux	20
3.8. bonding	23
6.1. Metodología del proyecto	36
8.1. ALIX 2d2 de PC Engines	44
8.2. Wistron CM9	44
9.1. Escenario real del desarrollo del prototipo del laboratorio	48
9.2. Escenario de configuración inicial de experimentación	48
9.3. Análisis espectral de dos interfaces que transmiten simultáneamente	50
9.4. Diagrama de flujo de la automatización de las pruebas de tráfico	52
10.1. Esquema de configuración de la técnica bonding	56
10.2. Tasa de transmisión de interface ethernet: 10Mbps	59
10.3. Diagrama de flujo de bonding	62
10.4. Bonding: Paquetes de TX y RX en un flujo	63
10.5. Bonding: Paquetes de TX y RX en 2 flujos	64
10.6. Paquetes transmitidos en modo 2	65
10.7. Pérdida de paquetes en modo 2	65
10.8. Caudal obtenido en modo 2	66
11.1. Esquema de configuración del balanceo de carga a nivel de red	68
11.2. Diagrama de flujo del balanceo de carga a nivel de red.	69
11.3. Iptables: Paquetes TX y RX en un flujo	71
11.4. Iptables: Paquetes TX y TR en dos flujos	71
12.1. Comparativa del retardo en un flujo	74
12.2. Comparativa de la variación del retardo en un flujo	75
12.3. Comparativa del caudal en un flujo	76
12.4. Comparativa de los paquetes perdidos en un flujo	77
12.5. Comparativa de delay en dos flujos	78
12.6. Comparativa de la variación de retardo en dos flujos	79
12.7. Comparativa del caudal en dos flujo	80
12.8. Comparativa de la pérdida de paquetes en dos flujo	81
13.1. Cronograma del proyecto	84

Acrónimos

ONU	Organización de las Naciones Unidas
PFM	Proyecto Fin de Máster
AL	América Latina
MIB	Management Information Base
ONGD	Organización No Gubernamental para el Desarrollo
GBT	Grupo Bioingeniería y Telemedicina
UPM	Universidad Politécnica de Madrid
SGR	Sistema de Gestión de Red
TIC	Tecnologías de la Información las Comunicaciones
WiFi	Marca de la WiFi Alliance que certifica equipos que cumplen con el estándar IEEE 802.11
WiLD	WiFi for Long Distances
ISM	Industrial, Scientific and Medical
VHF	Very High Frequency
HF	High frequency
ITU	International Telecommunication Union (Unión Internacional de las Telecomunicaciones)
ITU-D	Sector de telecomunicaciones para el desarrollo de la UIT (ITU)
IEEE	Institute of Electrical and Electronics Engineers
OSI	Open System Interconnection
FSF	Free Software Foundation
SL	Software Libre
OS	Operating System (Sistema Operativo)
GPL	General Public License
QoS	Quality of Service
DSCP	Differentiated Services Code Point
ODM	Objetivos del milenio
CDM	Cumbre del milenio
DDM	Declaración del milenio
CMSI	Cumbre Mundial sobre la Sociedad de la Información
MIMO	Multiple-Input Multiple-Output
HAL	Hardware Abstraction Layer
CAL	Channel Abstraction Layer
MAC	Medium Access Control
WLAN	Wireless Local Area Network
PHY	Physical layer
CSMA/CA	Carrier Sense Multiple Access/Collision Avoidance
ISM	Industrial, Scientific and Medical
OFDM	Orthogonal Frequency - Division Multiplexing
UCC	Universidad del Cauca del Colombia
PUCP	Pontificia Universidad Católica del Perú
URJC	Universidad Rey Juan Carlos
HR	High Rate

Índice de figuras

DSSS Direct-Sequence Spread Spectrum

DSL Digital Subscriber Line

ORAS Organismo Andino de Salud

D-ITG Distributed Internet Traffic Generator

TSC Teoría de la Señal y Comunicaciones

Parte I.

INTRODUCCIÓN

1. Organización del documento

La organización del documento y un breve resumen de cada capítulo se presenta a continuación:

- **Capítulo 1: Organización del documento.** Se presenta la organización del documento.
- **Capítulo 2: Antecedentes, contexto y justificación del trabajo.** Se describen las motivaciones que dieron origen a la búsqueda de soluciones tecnológicas de bajo costo para conseguir una mayor disponibilidad de la red y un mayor ancho de banda, para redes inalámbricas en países en vías de desarrollo. Justifican al lector el trabajo desarrollado.
- **Capítulo 3: Marco Teórico.** Se presentan los principales conceptos que intervienen en la comprensión de la solución de los enlaces redundantes con carga compartida en redes inalámbricas. Además de una revisión teórica breve, con el objetivo de permitir a lectores no familiarizados con el mundo del software libre.
- **Capítulo 4: Estado del arte de balanceo de carga en redes inalámbricas.** Se detalla la revisión de los estudios relevantes sobre el balanceo de carga en redes inalámbricas.
- **Capítulo 5: Objetivo del proyecto.** Donde se define el objetivo que se pretende alcanzar con el proyecto.
- **Capítulo 6: Métodos de trabajo.** Se planean la metodología que se llevó a cabo para desarrollar la solución para el balanceo de carga en redes inalámbricas.
- **Capítulo 7: Materiales de desarrollo.** Se plantean los materiales de hardware y software que se usa en el desarrollo de este proyecto.
- **Capítulo 8: Selección de componentes del nodo WiFi** Se describe que la selección de los componentes de un nodo WiFi y que software se va a usar dentro del nodo.
- **Capítulo 9: Diseño de la red** Se plantea el diseño inicial de la red, parámetros en común que usan las dos soluciones, parámetros de QoS que van a mediar los resultados.
- **Capítulo 10: Solución WiFi-Bonding.** Se presenta el diagrama de configuración de la solución de balanceo de carga a nivel de enlace, los parámetros que se necesitan para la configuración, fase para encontrar la solución de bonding, el script del auto-configurado de la solución y el análisis del comportamiento de las interfaces de los nodos.
- **Capítulo 11: Solución del balanceo de carga a nivel de red.** Se presenta el diagrama de configuración de la solución del balanceo de carga a nivel de red, los parámetros que se necesitan para la configuración, fases para encontrar la solución del balanceo de cargas a nivel de red, el script de auto-configuración de la solución final y el análisis del comportamiento de las interfaces de los nodos.
- **Capítulo 12: Resultados de experimentación.** Se presenta las comparativas y el análisis de los resultados de las dos soluciones.
- **Capítulo 13: Cronograma y presupuesto.** Se presenta el cronograma de trabajo, los costos del personal y costo de materiales que se usa en la experimentación.
- **Capítulo 14: Discusión de los resultados.** Se presenta al lector los límites de esta investigación.
- **Capítulo 15: Conclusiones.** La conclusión final del trabajo.
- **Capítulo 15: Líneas futuras de desarrollo e investigación.** Se proponen las líneas futuras para el desarrollo o investigación sobre el balanceo de carga.
- **ANEXOS:** Los scripts de autoconfiguración de las dos soluciones que se presentan en este proyecto, resultados obtenidos en las pruebas.

2. Antecedentes, contexto y justificación del trabajo

En la época actual, la sociedad en su conjunto se apoya en las Tecnologías de la Información y las Comunicaciones (TIC), de manera directa o indirecta, para cumplir su propósito con mayor eficiencia.

Estamos en el siglo XXI, era de la Internet, la inteligencia artificial, las comunicaciones por satélite, fibras óptica de banda ancha, móviles, servicios y aplicaciones sobre internet; y según Naciones Unidas, también es el momento de erradicar de la pobreza, o al menos éste era el propósito inicial de la Organización de Naciones Unidas (ONU) cuando propuso realizar la Cumbre del Milenio en el año 2000 [Gui07], donde representantes de 189 países se reunieron con el objetivo de planificar de manera colectiva un conjunto de metas y compromisos para erradicar la pobreza extrema y convertir el desarrollo en una realidad para todos hacia el 2015. Los representantes de estos países, suscribieron un documento denominado la "Declaración del Milenio"(DDM). La DDM se encuentra dividida en ocho objetivos, que a su vez contienen diferentes metas. En este PFM se enfatiza en el siguiente Objetivo de Desarrollo del Milenio (ODM) firmado en la citada DDM, y en algunas de sus metas [dlNUpeDb]:

- **Objetivo 8: Fomentar una asociación mundial para el desarrollo.**

META: Atender las necesidades especiales de los países menos desarrollados, los países sin litoral y los pequeños estados insulares en vías de desarrollo.

META: Continuar desarrollando un sistema comercial y financiero abierto, basado en reglas establecidas, predecible y no discriminatorio.

META: Lidiar en forma integral con la deuda de los países en vías de desarrollo.

META: En cooperación con el sector privado, hacer más accesible los beneficios de las nuevas tecnologías, especialmente las de información y comunicaciones.

El trabajo de este PFM se centra en este objetivo, porque tiene un énfasis especial en las Tecnologías de la Información y Comunicación (TIC). Debido a la posibilidad de aprovechar de forma transversal los beneficios de las TIC en muchos sectores, mejorar el acceso a las TIC puede contribuir al cumplimiento de otros ODM.

A 5 años de 2015, fecha límite para alcanzar los Objetivos de Desarrollo del Milenio, con motivo de la cumbre de Nueva York de septiembre de 2010, se efectuó la Cumbre de los Objetivos de Desarrollo del Milenio, donde se revisaron los logros obtenidos hasta ese momento [dlNUpeD11]. En el área de TIC que nos compete, se revisó el indicador correspondiente al objetivo 8, que arrojó resultados alentadores (ver figura 2.1), pues las ayudas siguen aumentando, aunque no se llega al promedio del 2005, donde se tuvo una subida relevante, como vemos en la figura 2.1. Estos son los indicadores que se revisaron respecto al objetivo 8:

- **Promedio de Asistencia oficial para el desarrollo (AOD) de los países en vías de desarrollo, 2000-2010 (miles de millones de dólares constantes de 2009 y dólares corrientes)**

Como podemos apreciar en la figura 2.1, en el año 2010 existe una leve subida de un 0,32 % respecto al año anterior. La ayuda para la asistencia humanitaria, la ayuda bilateral para los programas y proyectos de desarrollo aumentó un 5,9 % en términos reales, gracias al compromiso del 2005 en la Cumbre de Gleneagles del Grupo de los Ocho (G-8) [oDD] y en otros foros [dlNUpeDa][dlNUpeD11], donde los donantes se comprometieron a incrementar el apoyo a los países en vías de desarrollo.

- **Tasa de cantidad de abonados de líneas telefónicas fijas y abonados de teléfonos móviles por cada 100 habitantes, 1995-2010**

La tecnología móvil hoy en día es la TIC con mayor crecimiento, y así lo podemos ver en la figura 2.2. En los datos proporcionados por [dlNUpeD11] podemos ver, que a finales del

2. Antecedentes, contexto y justificación del trabajo

Asistencia oficial para el desarrollo (AOD) de los países desarrollados, 2000-2010 (miles de millones de dólares constantes de 2009 y dólares corrientes)

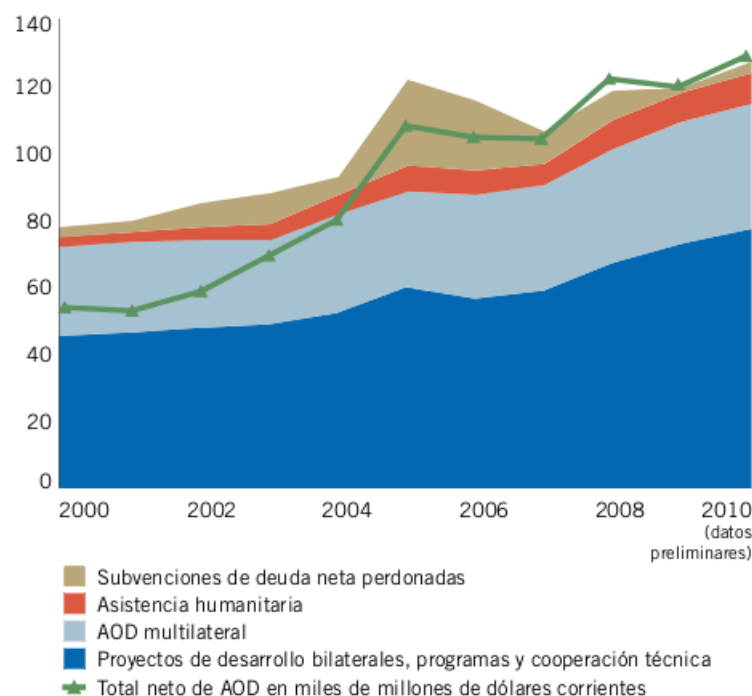


Figura 2.1.: Promedio de Asistencia Oficial para el desarrollo

2010, en los países en vías de desarrollo, los hogares con teléfono fijo en las zonas rurales suponen un 10 %, frente al 75 % de telefonía móvil. La penetración de los teléfonos móviles ha llegado casi a un 76 % en todo el mundo, pero la telefonía fija (alámbrica) es apenas del 1 %. Las comunicaciones móviles tanto en zonas urbanas como en zonas rurales, no se limitan a los servicios de voz; también pueden soportar datos que multipliquen los efectos beneficiosos en las zonas de intervención, como por ejemplo, mejoras en salud, educación o técnicas productivas, con el consecuente impacto sobre las condiciones de vida en los países en vías de desarrollo, contribuyendo al cumplimiento de otros ODM.

■ Tasa de la cantidad de usuarios de Internet por cada 100 habitantes, 1995-2010

La cantidad de usuarios de Internet sigue en crecimiento. Los datos proporcionados sobre el grado de penetración de la red en países en vías de desarrollo llegaba a 21 % a finales de 2010. [dINUpeD11] Pero en los países menos adelantados era de un 3 % a finales de 2010. [dINUpeD11] En las zonas rurales de los países en vías de desarrollo, el acceso a internet es muy reducido. Para solventar este problema, en muchos lugares, se han creado puntos de acceso comunitarios. [Uni10] Cada vez más países en desarrollo están procediendo a la creación de instalaciones públicas de acceso a Internet en las zonas rurales, generalmente gracias a la financiación procedente de las contribuciones de los fondos de acceso universal. Los datos disponibles sugieren que en muchos países en vías de desarrollo, la población rural utiliza Internet desde emplazamientos públicos. Especialmente en América Latina, donde se han adoptado políticas que promueven la creación de centros públicos de acceso a Internet. [Uni10]

■ Tasa de abonados de banda ancha, fija y móvil, por cada 100 habitantes, 2000-2010

A finales de 2010, la penetración de la banda ancha fija en los países en vías de desarrollado era de un 4,4 %. La banda ancha móvil se ha convertido en una gran alternativa. Desde el 2005 hasta el 2010 tiene un buen crecimiento, y se espera que crezca más rápidamente en los próximos años. Ya en el 2005 habían 50 países que lo comercializaban, pero en el 2010 había 143 países que ofrecían el servicio de banda ancha móvil.

Cantidad de líneas telefónicas fijas y abonos a teléfonos celulares por cada 100 habitantes, 1995-2010

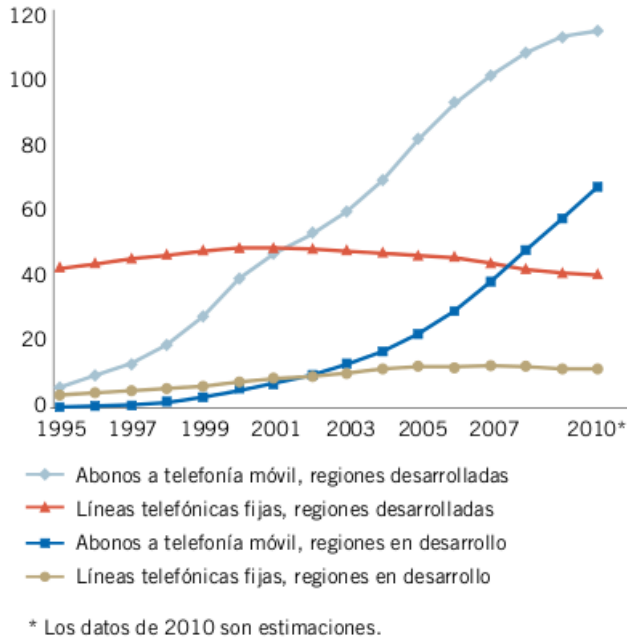


Figura 2.2.: Cantidad de líneas fijas y móviles

Cantidad de usuarios de Internet por cada 100 habitantes, 1995-2010

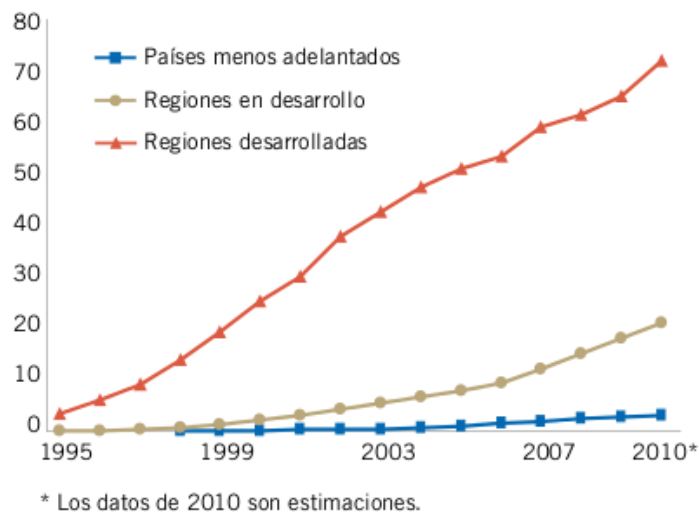


Figura 2.3.: Cantidades de usuarios de internet

2. Antecedentes, contexto y justificación del trabajo

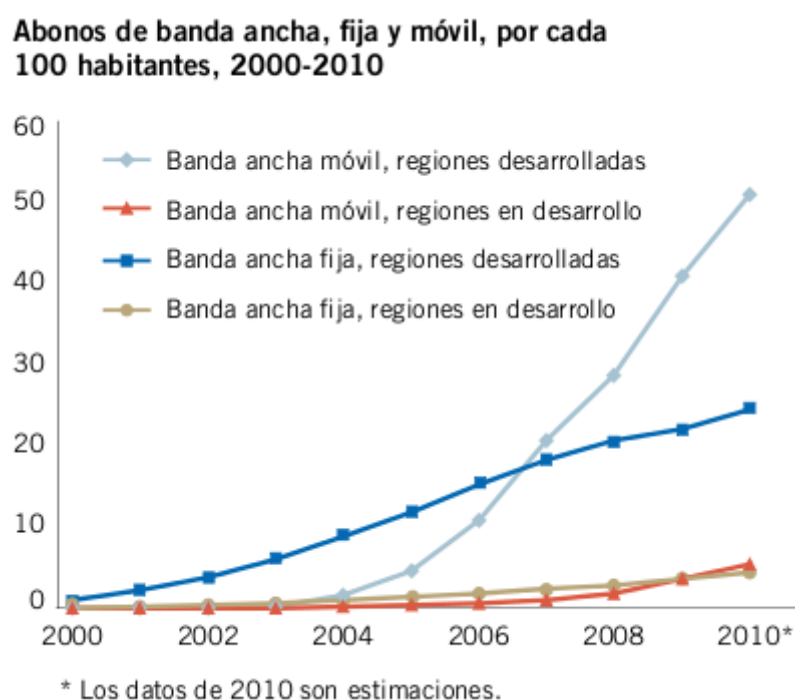


Figura 2.4.: Abonados de banda ancha, fija y móvil

Los datos presentados anteriormente, sobre el desarrollo de las TIC, muestran un leve crecimiento en las inversiones de millones de dolares para fomentar proyectos TIC en países en vías de desarrollo. Las tecnologías móviles son las que presentan mayores avances. A finales de 2010, el 90 % de la población mundial disfrutaba de cobertura de red móvil celular [dlNUpeD11], con una alta penetración en países en vías de desarrollo. El acceso a internet de los usuarios en los países en vías de desarrollo mostraba, según los mismos datos, un crecimiento relativamente lento, y la banda ancha móvil se había ya convertido en la alternativa a la banda ancha fija. Los informes de la International Telecommunication Union (ITU) sobre las TIC [Uni10] muestran que las TIC son una herramienta muy importante para impulsar el desarrollo de los países en vías de desarrollo, porque se puede ejecutar en sitios distantes, donde hay falta de carreteras e infraestructura de provisión de energía, y junto a una escasa población que se basa a una economía de subsistencia. Generalmente, en estos lugares, los operadores de telecomunicaciones no realizan inversiones porque no vislumbran retorno de inversión para estas zonas.

En este contexto, muchas entidades de cooperación para el desarrollo utilizan la ayuda de los donantes en la construcción de infraestructura de telecomunicaciones en estas zonas, especialmente en escuelas, centros de salud, instalaciones municipales, y quioscos. Muchos de estos proyectos emplean conexiones vía satélite, como es el caso proyecto de Aulas Virtuales, que se ejecuto en Bolivia por la ONG Visión Mundial [Mun], llevando conexión satelital a varias zonas rurales en el año 2008. La conexión satelital tiene un coste fijo alto, que se vuelve insostenible económicamente al finalizar la ayuda de los donantes. Para reducir estos costos, existen otras alternativas como VHF o WiFi para largas distancias (WiLD o WiFi for Long Distances). Se han ejecutado muchos proyectos con estas tecnologías, como por ejemplo el proyecto de Tele-salud Rural Pangui-Loja-Yacuambi "TUTUPALY-[Mor10], en Ecuador. Estas soluciones pueden aliviar el problema económico, pues aunque los costos iniciales son más altos debido a la implementación de la infraestructura adicional, estos costes iniciales son pagados por los donantes generalmente. No obstante estos proyectos tienen que enfrentar otros problemas, después de que la ayuda de los donantes se ha concluido. Uno de estos problemas es el de la sostenibilidad, y es que la tasa de éxitos de los últimos años en este aspecto es muy reducida, como lo revelan algunos estudios [B.02][pC] [Avg09] [Avg08] [BG10]. En algunos casos, los indicadores de fracaso por falta de sostenibilidad se elevan hasta un 80 %, como lo podemos ver en la tesis [BG10]. El gran reto que tienen las TIC, reside en lograr la sostenibilidad desde varios frentes, como son

el económico, el financiero, el institucional, el tecnológico y el educativo, respetando los factores sociales y culturales de cada intervención.

Una institución que está buscando soluciones a estos problemas es la Fundación Enlace Hispano Americano de Salud (EHAS), cuyo propósito es promover el uso adecuado de las TIC en la salud a través de la telemedicina en las zonas rurales remotas de los países en vías de desarrollo. Cuenta con más de catorce años de experiencia en instalación de redes inalámbricas de telecomunicaciones. Uno de los proyectos que ejecutó en el año 2007, se encuentra en la selva amazónica en el Perú, y es una red de telemedicina situada en el río Napo, al norte del país. La red de salud instalada en las orillas del río Napo cuenta con 17 puestos y centros de salud que están interconectados entre sí (usando una topología de red lineal), desde la ciudad de Iquitos hasta Cabo Pantoja, en la frontera con Ecuador. Cada uno de los nodos de la red tiene una separación de entre 20 y 60 Km, y la red recorre 515 Km. Estos sistemas de comunicación proporcionan un throughput aproximado de 7 Mbps extremo a extremo. En cada centro de salud se tienen instalados varios servicios como: telefonía IP, videoconferencia, chat y acceso a Internet. El detalle sobre dicha red se proporciona en el apartado 3.1.1 de este proyecto. Estos centros se beneficiaron de tener la posibilidad de acceder a un diagnóstico con segunda opinión, y una mejor coordinación en los procesos de salud, tanto en casos urgentes como no urgentes. Por otro lado, los técnicos de enfermería a cargo de algunos centros de salud rurales, antes del despliegue de las redes tenían que viajar alrededor de 3 días a su centro de referencia para temas de formación, coordinación o gestión, dejando ese tiempo el centro sin atención, gracias a las TIC instaladas por la fundación EHAS, se pueden realizar gran parte de las tareas antes mencionadas a distancia y se disminuyen en gran medida los viajes de los técnicos de salud [BG10][Per10].

Desde el punto de vista de las infraestructuras de telecomunicaciones, en la red del río Napo, se presentan fundamentalmente dos situaciones a resolver:

- **Baja disponibilidad de la red**, se presenta una baja estabilidad en el sistema y su consecuente pérdida del servicio. Ésto es debido a que el servicio que presta un nodo depende de la conectividad de todos los nodos anteriores (pues la topología de la red es lineal y la salida a internet se encuentra en un extremo). Es necesario resaltar dos situaciones problemáticas asociadas a la topología lineal: **Fallo completo de la red**, cuando falla el nodo principal o de acceso, o cualquier nodo anterior a uno dado en la cadena; y también que el **último nodo ofrece peor disponibilidad**, dado a que su probabilidad de fallo es la combinación de las probabilidades de fallo de cualquier nodo de la red.
- **Insuficiente ancho de banda**, como consecuencia de la aparición de nuevos servicios, algunos con gran consumo de caudal de red, que demandan constantemente mayor capacidad de tráfico a la red. La situación con respecto a esto problema es también más crítica en el último nodo, que por su estructura lineal es el que cuenta con menor ancho de banda disponible extremo a extremo en sus comunicaciones.

Por lo anterior, este PFM enfoca la solución del sistema en la configuración de enlaces paralelos entre los nodos, de tal manera que se constituyan como:

- **Enlaces redundantes**, que mejoren la robustez del sistema, disminuyendo la probabilidad de la caída del servicio, al no depender de la caída de cualquier enlace individual, sino de la caída simultánea de los 2 enlaces paralelos que unen cada par de nodos.
- **Enlaces con carga compartida**, que busquen doblar el ancho de banda entre nodos y, por consiguiente, duplicar la capacidad de tráfico en toda la red.

3. Marco Teórico

3.1. Fundación EHAS

La Fundación EHAS (Enlace Hispano Americano de Salud) es una institución sin ánimo de lucro cuyo fin es promover el uso apropiado de las Tecnologías de la Información y la Comunicación (TIC) para mejorar los procesos de salud en zonas rurales aisladas de países en desarrollo [dS].

Para ello plantea cuatro grandes líneas de acción:

1. La investigación y el desarrollo de tecnologías de comunicación y sistemas de acceso e intercambio de información adaptadas a las zonas rurales de países en desarrollo es decir tecnologías de bajo coste.
2. El asesoramiento, desarrollo y evaluación de protocolos de actuación para la mejora de los procesos de atención de salud en las zonas rurales, con especial atención en los relacionados con la salud materno-infantil.
3. El diseño y la ejecución de proyectos de cooperación para el desarrollo que permitan validar tanto la tecnología como los protocolos de actuación anteriores.
4. El desarrollo de actividades de formación, difusión, transferencia e incidencia política para promover el uso adecuado de las TIC en el sector salud rural de países en desarrollo.

La Fundación EHAS trabaja desarrollando proyectos TIC, para su uso en establecimientos de salud en áreas rurales y aisladas de países en vías de desarrollo, con el objetivo de mejorar los procesos de atención sanitaria e impactar positivamente en la salud de las personas [BG10]. Desde hace varios años ha desarrollado sus actividades en Perú, Colombia, Cuba y Ecuador. Inicia sus actividades en el año 1997, cuando un equipo de estudiantes del Grupo Bioingeniería y Telemedicina (GBT) de la Universidad Politécnica de Madrid (UPM) y de la ONGD Ingeniería Sin Fronteras, comienzan a investigar en el diseño de sistemas y servicios de comunicación apropiados a las necesidades del personal sanitario rural de los países de América Latina (AL). A raíz de estos trabajos se diseña y ejecuta el Programa Enlace Hispano Americano de Salud (EHAS), que en octubre del 2004 se constituyó en fundación sin ánimo de lucro, teniendo como patronos a la UPM y a la ONGD Ingeniería Sin Fronteras. En 2008 se amplió el patronato con la Universidad del Cauca de Colombia (UCC), Pontificia Universidad Católica del Perú (PUCP) y Universidad Rey Juan Carlos (URJC). Estos programas de telemedicina de la Fundación EHAS pretenden contribuir a la mejora de los sistemas públicos de asistencia sanitaria, en las zonas rurales de AL. En el año 1999 arranca el Subprograma EHAS en Perú, con la ayuda de la Sección de Electrónica de la Pontificia Universidad Católica del Perú, posteriormente se crea el Grupo de Telecomunicaciones Rurales en dicha institución, constituyéndose como contraparte tecnológica de EHAS en Perú. Entre el 2000 y 2002 se puso en marcha un proyecto piloto en la provincia de Alto Amazonas del departamento de Loreto en Perú, cuyo objetivo fue implementar una solución de comunicaciones de bajo coste y evaluar su impacto.

Dicho proyecto involucra al Hospital Provincial de la capital Yurimaguas, y a 40 establecimientos de salud divididos en dos categorías: centros de salud y puestos de salud. La selección de la provincia de Alto Amazonas se llevó a cabo debido a que es una región de selva baja idónea para probar los sistemas de comunicación en VHF (primer producto del Programa EHAS); esta zona es muy extensa y sin carreteras (el 95 % de los establecimientos de salud solamente se pueden acceder por el río); y tiene fuertes carencias en infraestructura de telecomunicaciones (sólo dos establecimientos de salud contaban con línea telefónica).

Desde ese momento, la Fundación EHAS ha desarrollado numerosos proyectos que plantean el uso de las TIC para la mejora de los sistemas públicos de salud en las zonas rurales más aisladas de América Latina; de estos proyectos el más destacado es la red de comunicaciones instalada a lo largo del río Napo en Perú. La Fundación ha seguido desde entonces una política de investigación de nuevas herramientas de transmisión de voz y datos para entornos de difícil

3. Marco Teórico

acceso, obteniendo durante el transcurso de los años sistemas mejorados para la tecnología de radio VHF/HF y actualmente se trabaja en tecnologías WiFi 802.11a/b/g/n que usan las bandas ISM de 2.4 y 5 GHz con tecnología WiLD (WiFi adaptada a muy largas distancias) [BG10],[dS].

Actualmente, la Fundación EHAS se encuentra investigando nuevas posibilidades que otras tecnologías inalámbricas pueden ofrecer, tales como: el estándar 802.11e y la tecnología certificada WiMAX, 802.16. Asimismo, se diseñó un plan de sostenibilidad en: lo financiero, tecnológico, social y humano, para proyecto del río Napo [BG10].

3.1.1. La red WiLD en el río Napo (Perú)

Este apartado se centra en el proyecto de la red de telemedicina instalada a lo largo del río Napo (Perú) por la fundación EHAS, en esta se prestan servicios de comunicaciones de voz y de datos de banda ancha en 17 puestos y centros de salud, que nos muestra la figura 3.1. Dicha red fue instalada en dos etapas diferentes:

- **La primera etapa**, en el año 2007, en el marco del proyecto PAMAFRO "Proyecto de control de Malaria en las zonas fronterizas de la Región andina" del Organismo Andino de Salud (ORAS), dio conectividad a 12 puestos y centros de salud, entre las localidades de Tacsha Curaray y Cabo Pantoja del Perú.
- **La segunda etapa**, en el año 2008, formó parte del proyecto "Mejora de condiciones de salud de la población materno-infantil a través del uso apropiado de las Tecnologías de la Información y comunicaciones (TIC) en los puestos de salud del Río Napo (Perú)", desarrollado por la Fundación EHAS conjuntamente con el Ayuntamiento de Madrid, conectando otros 5 puestos de salud entre las localidades de Tacsha Curaray e Iquitos del Perú.

La red completa puede verse en la figura 3.1, donde vemos que cada uno de los nodos está separado entre 20 y 60 Km, y la red total alcanza aproximadamente los 515 Km, son 17 puestos y centros de salud interconectados. Para la conectividad entre los nodos se utiliza la tecnología de comunicaciones WiFi (estándar IEEE 802.11) modificada para largas distancias. Esta tecnología necesita línea de visión entre los extremos de cada enlace para este rango de distancias, como también una mayor potencia de transmisión, los repetidores han sido instalados en torres ventadas de gran altura (hasta 90 metros), la altura necesaria para sortear la curvatura terrestre manteniendo al menos el 60% de despeje en la primera zona de fresnel. La figura 3.2 muestra un segmento de la red troncal a lo largo de todo el río Napo. El acceso a la red de los establecimientos de salud se realiza mediante los enlaces de distribución, que utilizan la tecnología WiFi convencional. Las estaciones clientes constan generalmente de un punto de trabajo con ordenador e impresora, además de un teléfono VoIP, que permite las comunicaciones de voz sin coste entre todos los centros de salud de la red del río Napo. Este sistema de comunicaciones proporciona conectividad de banda ancha, es decir, superior a 1 Mbps. Existen dos puntos de acceso a Internet para la red río Napo: una conexión DSL en Iquitos y una conexión satelital en Santa Clotilde. Los servicios de datos que funcionan en esta red son todos los que puede proporcionar una red de banda ancha con acceso a Internet: correo electrónico, mensajería instantánea, gestión de la red, sistemas de información remota (basados en Web y bases de datos), videoconferencias, transmisión de audio e imágenes médicas para consulta remota, navegación Web y acceso a Internet. Los únicos emplazamientos que no cuentan con ordenador y sólo disponen de servicio de telefonía IP son Copal Urco y Túpac Amaru en el Perú, ya que no labora personal sanitario.

3.1.2. La red Wi-Fi del Cuzco (Perú)

Este apartado se centra en el proyecto de la red de telemedicina instalada en las Provincias de Quispicachis y Acomayo del Departamento de Cuzco (Perú) por la fundación EHAS en el 2006 [Gar], en esta se prestan servicios de comunicaciones de voz y de datos de banda ancha en 12 puestos y centros de salud a través de una red inalámbrica WiFi que usa 7 repetidores, que nos muestra la figura 3.3. En este proyecto se usa arquitectura Mesh¹ de una red WiFi EHAS

¹Red Mesh: Es una red que se forma "de manera espontánea" sin ninguna infraestructura, constituyendo mallas que conectan un cierto número de nodos entre sí, y en que cada nodo puede ser al mismo tiempo estación de trabajo de un usuario, encaminador o pasarela

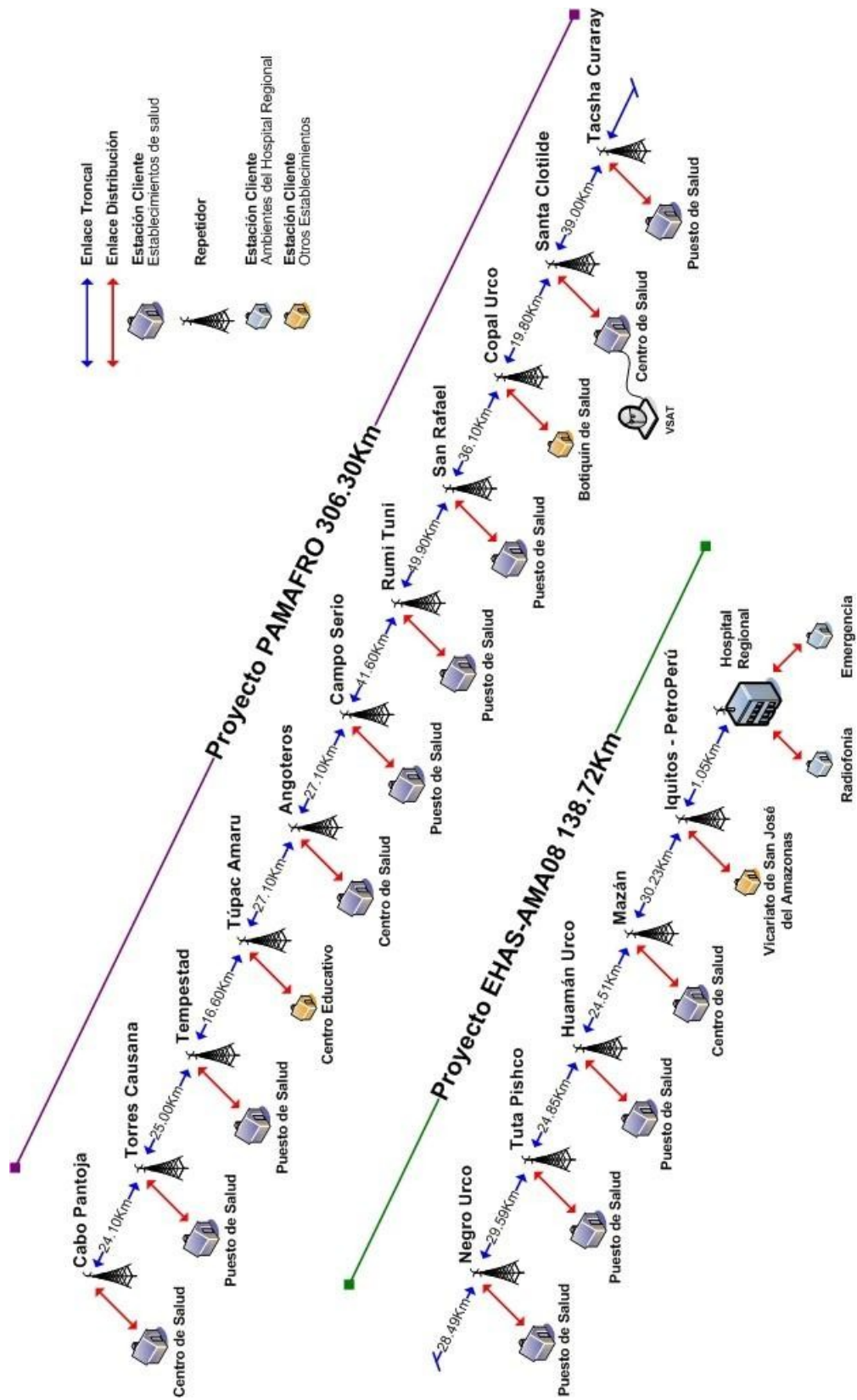


Figura 3.1.: Esquema técnico de la red Rio Napo en 2009

3. Marco Teórico

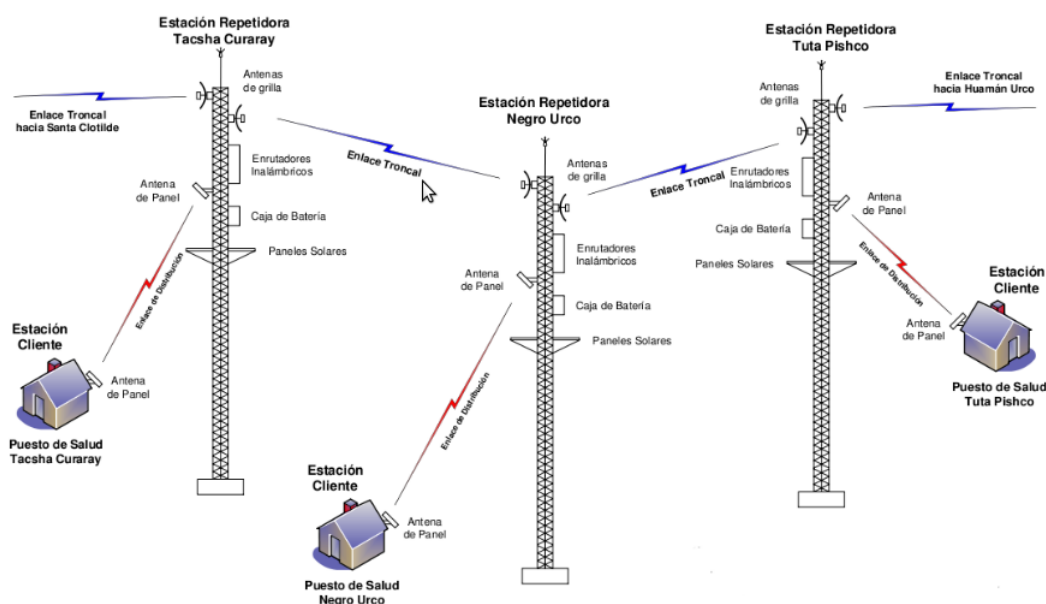


Figura 3.2.: Topología detallada de un segmento de la red Napo.

como podemos ver la siguiente figura: 3.4. Cada uno de los nodos está separado hasta 40 Km. Para la conectividad entre los nodos se utiliza la tecnología de comunicaciones WiFi (estándar IEEE 802.11) modificada para largas distancias.

Con este proyecto se consiguió integrar las tecnologías VSAT y Wi-Fi. VSAT permitió saltar la barrera de construir costosas infraestructuras de comunicaciones y ahorrar tiempo de instalación de equipos en zonas donde la solución cableada resulta imposible. Las estaciones cliente constan generalmente de un punto de trabajo con ordenador e impresora, además de un teléfono VoIP, que permite las comunicaciones de voz sin coste entre todos los centros de salud de la red de Cuzco. Los servicios de datos que funcionan en esta red son todos los que puede proporcionar una red de banda ancha con acceso a Internet: correo electrónico, mensajería instantánea, gestión de la red, sistemas de información remota (basados en Web y bases de datos), videoconferencias, navegación Web y acceso a Internet. Por otro lado, la tecnología WiFi permitió la interconexión de telecentros ubicados en comunidades cercanas.

3.2. Software libre y licencias de software

Software Libre (SL) es un concepto que atiende a la “libertad” con la que cuentan los usuarios para: ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. El SL es una cuestión de libertad que tienen los usuarios respecto al software. Más precisamente, la Free Software Foundation (FSF) [Foua], gran precursor del SL a nivel mundial, indica que el SL son aquellos programas que cumplen cuatro libertades esenciales:

- *Libertad de ejecutar el programa con cualquier propósito (libertad 0).*
- *Libertad de estudiar cómo funciona el programa, y cambiarlo para hacer lo que desees (libertad 1). El acceso al código fuente es una precondición para esto.*
- *Libertad de redistribuir copias de forma que puedas ayudar a tu vecino (libertad 2). Esto le permite ofrecer a toda la comunidad la oportunidad de beneficiarse de las modificaciones. El acceso al código fuente es una condición necesaria para ello.*
- *Libertad de distribuir copias de sus versiones modificadas a otros (libertad 3). Haciendo esto puedes dar a toda la comunidad la oportunidad de beneficiarse de las modificaciones. Es necesario contar con el código fuente.*

Las libertades 1 y 3 requieren acceso al código fuente para estudiar y modificar el software.

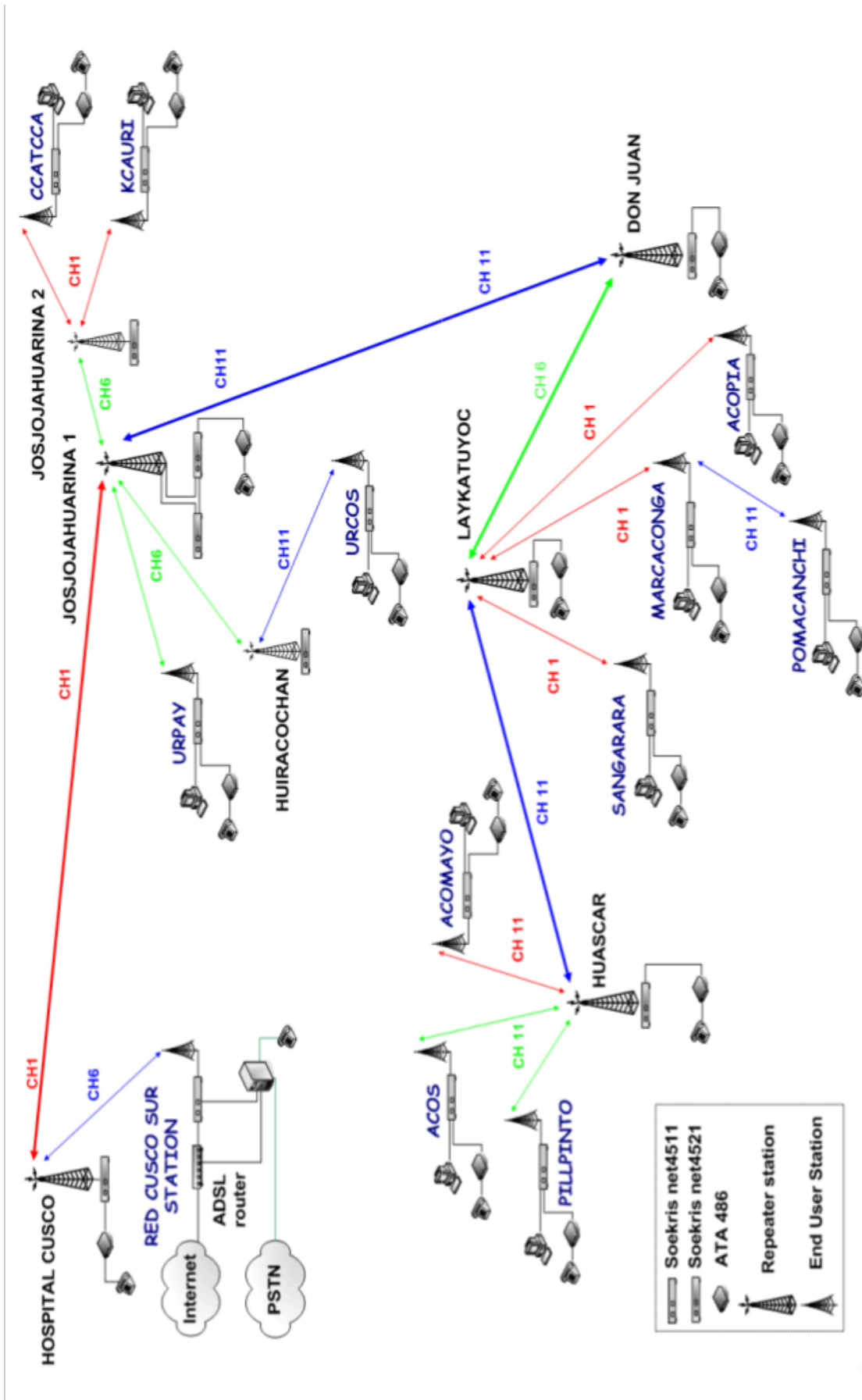


Figura 3.3.: Red Wi-Fi de Cuzco

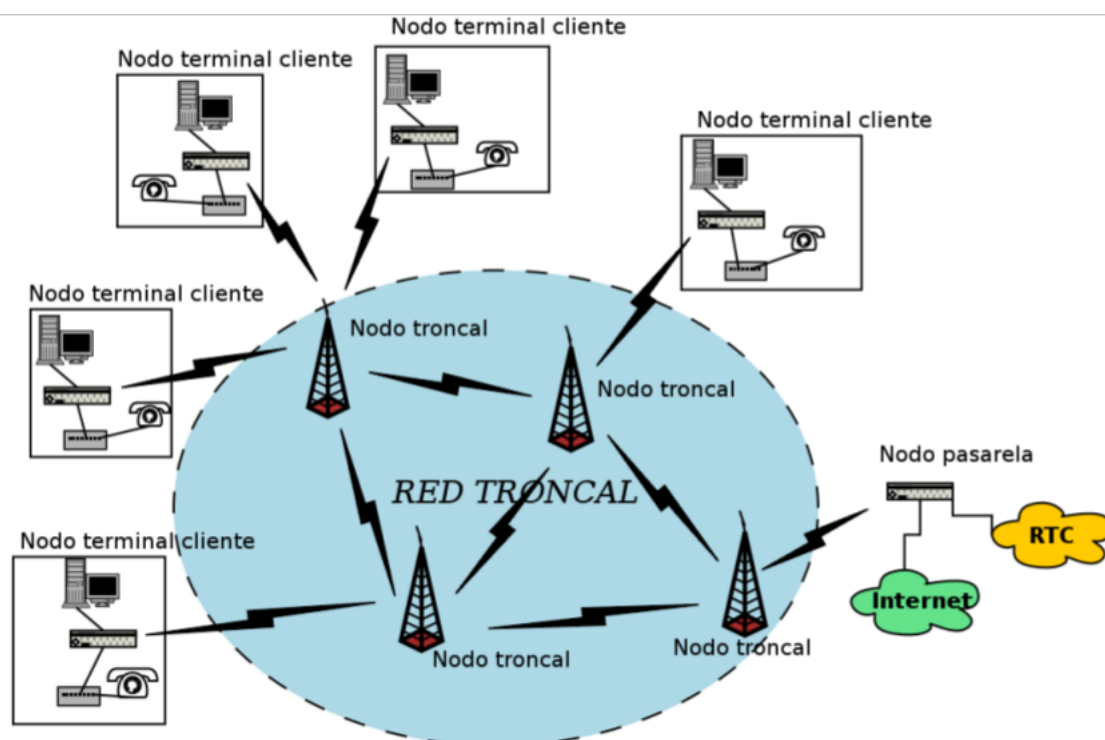


Figura 3.4.: Arquitectura Mesh de una red Wi-Fi EHAS

Un programa se considera SL si los usuarios cuentan con todas estas libertades. El usuario debe ser libre para redistribuir copias, con o sin modificaciones, ya sea gratis o cobrando una tarifa por la distribución, a cualquiera en cualquier sitio. Ser libre para hacer estas cosas significa que no necesitas preguntar ni pagar por el permiso. René Mérou, con su Mapa conceptual del SL, expresa de manera clara y visual el significado de Software Libre y cómo aporta a la humanidad la aparición de este concepto de SL como contraposición al software privativo (ver Figura 3.5).

La FSF nace de la mano de Richard Stallman, considerado el fundador del movimiento del SL [Sta]. Entre los años 1960 y 1970, el software se consideraba un elemento agregado, que los vendedores de las grandes computadoras aportaban a sus clientes para permitirles usarlas. En ese tiempo, los desarrolladores de software, compartían entre sí los programas libremente y sin ningún tipo de condiciones. A finales de los años 1970 esto empezó a cambiar, porque las compañías comenzaron a imponer condiciones a los usuarios, con el uso de licencias. Con la llegada de los años 1980 las computadoras comenzaron a utilizar sistemas operativos privativos (software no libre, que no puede ser modificado o mejorado por el usuario sin permiso expreso del autor) y esto forzaba a los usuarios a aceptar condiciones impuestas de las compañías para su uso. Los programas empezaron a ofrecerse en formato binario, de manera que el usuario no podía conocer como estaba hecho, o incluso mejorarlo. En 1983, Richard Stallman, miembro del Laboratorio de Inteligencia Artificial del Instituto Tecnológico de Massachussets (MIT), anunció la creación del proyecto GNU's Not Unix! (GNU), diciendo que estaba frustrado con los efectos del cambio en la cultura de la industria de las computadoras y sus usuarios. En el año de 1984 comenzó de forma efectiva el desarrollo del proyecto GNU [Pro], y en 1985 funda la FSF. Además desarrolló la definición del SL y copyleft (diseñado para proteger legalmente la libertad de usuarios y desarrolladores).

Una licencia de software es un contrato entre el desarrollador de un software sujeto a propiedad intelectual, que posee los derechos de autor, y el usuario final. Ya que el mundo de las licencias de software ha ido ganando complejidad desde 1970, cuando la industria comenzaba a emplearlas. De esta manera, podemos distinguir las diferentes categorías del software según el siguiente diagrama diseñado por Chao-Kuei [Pro](ver figura 3.6).

- **Software Libre.** Software con autorización para que cualquiera pueda usarlo, copiarlo y/o distribuirlo, que puede o no tener modificaciones, gratuitamente o mediante pago.

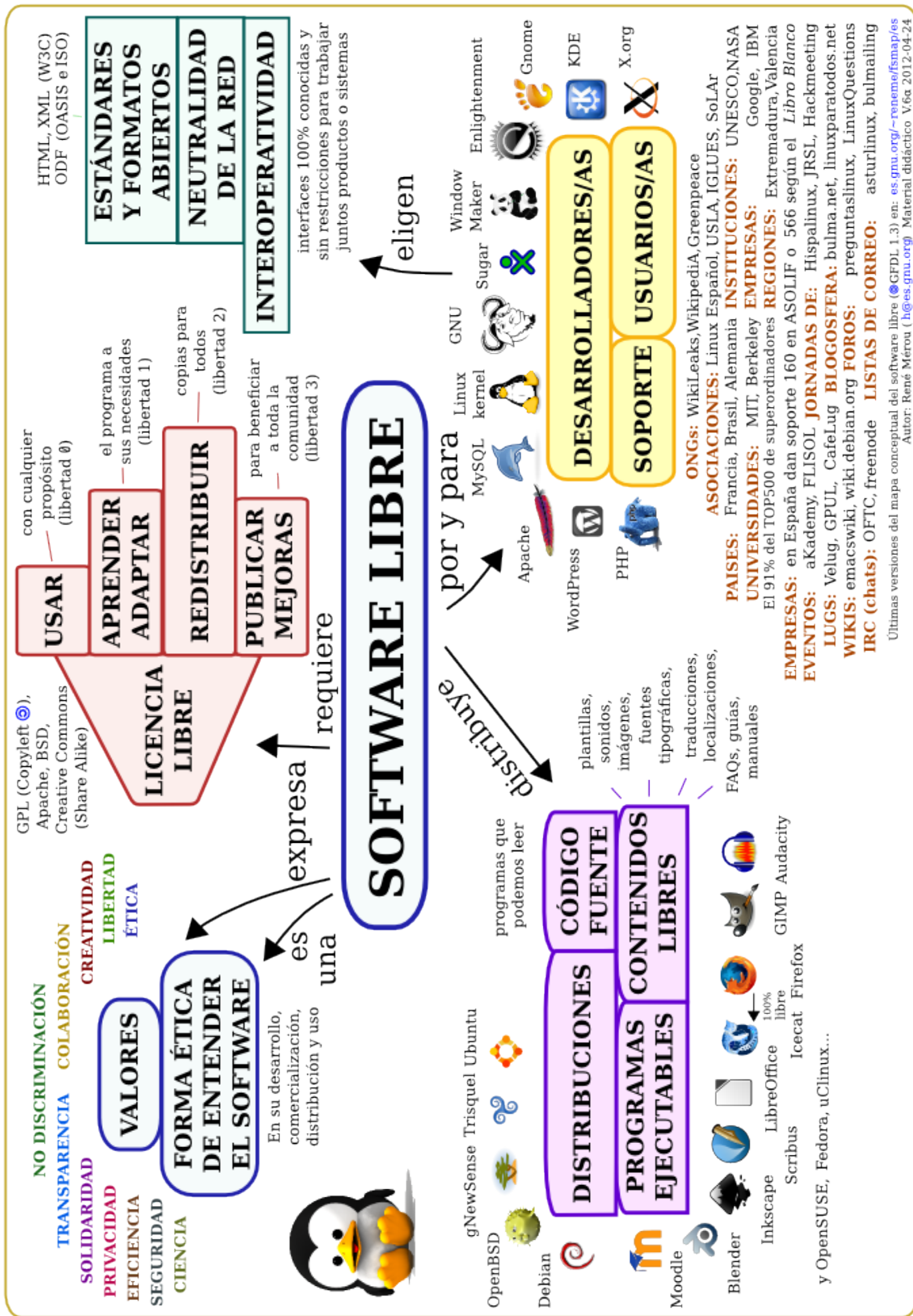


Figura 3.5.: Mapa conceptual del Software Libre

3. Marco Teórico

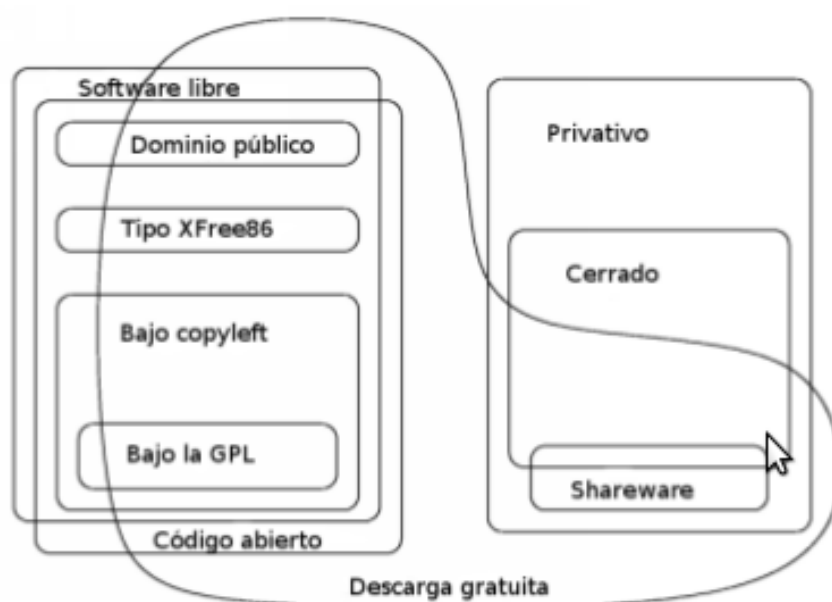


Figura 3.6.: Categorías de software según las libertades de sus licencias

Esto significa que el código fuente está disponible. Este software sigue el la máxima de “si no hay fuentes, no es software”, emitida por la FSF.

- **Código abierto.** Se tiene acceso al código fuente del software, pero no necesariamente significa que permita su modificación. Generalmente el código abierto acepta licencias más restrictivas que el SL. Porque SL hace referencia a las libertades de desarrolladores y usuarios, mientras que código abierto hace referencia únicamente a que esté disponible o no el código fuente.
- **Software de dominio público.** Este software no tiene derechos de autor. Es un caso especial de software libre no protegido con copyleft, por lo que algunas copias o versiones modificadas pueden no ser completamente libres. En algunos casos, un programa ejecutable puede ser de dominio público y no dar acceso al código fuente, en este caso no es SL. Pero en general, la mayor parte del SL no es de dominio público, porque está protegido por los derechos de autor, y los propietarios de esos derechos han dado el permiso legal para que cualquiera pueda utilizarlo libremente usando una licencia de SL. Debido a que tras la convención de Berna [dIPI] cualquier obra escrita (y un programa lo es) queda automáticamente bajo el dominio de derechos de autor; si se desea que un programa sea de dominio público, se deben cumplir ciertos procedimientos legales. Si un programa es de dominio público, cualquiera puede compartir el programa y sus mejoras, pero también está permitido que personas no tan cooperativas conviertan el programa modificado en software privativo.
- **Software protegido con copyleft.** Software cuya licencia garantiza que tanto él mismo como todas las copias de todas sus versiones tengan casi los mismos términos de distribución. Este tipo de licencia obliga a tener el código fuente disponible y no permite a terceros agregar requisitos adicionales (exceptuando un conjunto limitado de requisitos para aumentar su seguridad). Habitualmente las diferentes licencias copyleft son incompatibles entre sí, eso quiere decir que es imposible reutilizar código de una licencia copyleft en otro código de otro tipo de licencia copyleft. Por eso mismo, se suele utilizar una misma licencia copyleft en el mundo del SL, La Licencia Pública General de GNU (GPL) es la más extendida en este momento. Por ello, se encuentra la versión 3 [Foub].
- **Software libre no protegido con copyleft.** Software que tiene la autorización del autor para ser redistribuir y ser modificado, así como también para agregarle restricciones adicio-

nales. El problema de este tipo de software es que algunas copias o versiones modificadas del mismo podrían no ser completamente libres.

- **Software con licencia GPL.** Consiste en un conjunto específico de condiciones de distribución empleadas para proteger un programa con copyleft. Por lo tanto, el software licenciado bajo General Public License (GPL) es un subconjunto del copyleft.
- **Software con licencia permisiva.** Entre las licencias permisivas se encuentran las licencias BSD, que mantienen la protección de copyright solamente para la renuncia de garantía y para requerir la adecuada atribución de la auditoría en trabajos derivados. Estas licencias permiten utilizar el código de cualquier manera, inclusive la distribución de binarios privativos con o sin modificaciones del código.
- **Software privativo.** Software que no es SL y es sinónimo de software no libre. Software cuyo uso, redistribución o modificación no esta permitido. Este tipo de software convierte a los usuarios en dependientes de los proveedores del software, de manera total o parcial.
- **Shareware.** Software que permite redistribuir copias, pero se tiene que pagar la licencia para su uso. No se dispone de código fuente (por lo tanto no se puede modificar) y no está permitida la copia sin el pago de una licencia, incluso aunque se realice sin ánimo de lucro. Evidentemente, no es SL ni semilibre.
- **Freeware.** Generalmente se usa para denominar el software que se puede redistribuir pero no modificar, y su código fuente no está disponible. Este software no es SL.
- **Software privado a medida.** Software desarrollado a medida para un usuario, organización o compañía. El usuario lo utiliza y no lo libera al público. La mayor parte del desarrollo software suele ser software privado a medida. Si el usuario tiene plenos poderes sobre el software no necesariamente tiene por qué ser incompatible con el SL, y es que si no realiza una contribución a la comunidad por ser muy específico del usuario en cuestión, no es tan dañina la omisión de su publicación.
- **Software comercial.** Software desarrollado por un negocio y para obtener lucro de su utilización. Esta definición no excluye al software comercial de ser SL. Se puede desarrollar software para obtener lucro y sin embargo cumplir las 4 libertades del SL. Generalmente la mayor parte del software comercial suele ser privativo. Habitualmente el lucro del SL comercial se deriva de contratos de mantenimiento del software.

3.3. Sistema Operativo GNU/Linux

El proyecto GNU's Not Unix! (GNU), inició en 1984 con la ilusión de crear un sistema operativo (SO) completamente libre, fundamentándose en las características del SO propietario Unix, que con su modularidad y solidez constituían la mejor opción entre los SO de la época. Durante 7 años se fueron desarrollando todos los componentes necesarios para disponer de un SO completo (un compilador de C, utilidades de compilación, un intérprete de comandos o shell, un depurador de código, editor de texto, etc), al que únicamente le faltaba el núcleo de SO (o kernel), que es considerado como el corazón del SO y el responsable de realizar algunas tareas imprescindibles, como la gestión de los recursos, la interlocución con el hardware, la multiplexación de los accesos al hardware y ofrecer una interfaz limpia y uniforme para los programadores [Pro].

En el año 1991, un estudiante finlandés de 21 años llamado Linus Benedict Torvalds, usuario del SO MINIX (escrito desde cero por Andrew S. Tanenbaum con fines fundamentalmente educativos), escribiría una entrada en el grupo de noticias de MINIX, en la que revelaba que estaba desarrollando un SO libre como pasatiempo, y pedía a la comunidad de usuarios de MINIX realimentación para determinar qué tipo de cosas les gustaría ver implementadas en su nuevo sistema. En el año 1992 Linus Benedict acabó creando el núcleo que le faltaba al proyecto GNU y había portado a su nuevo SO las principales herramientas del proyecto GNU (bash, GCC, update, apt-get, etc). De esta manera se añadió Linux a GNU y con esto nacería el SO que hoy en día se conoce como GNU/Linux [Pro].

3.3.1. Arquitectura del Sistema Operativo Linux

La arquitectura del sistema operativo LINUX se divide en dos niveles [Jon07], como vemos en la figura 3.7, estos se describen de la siguiente manera:

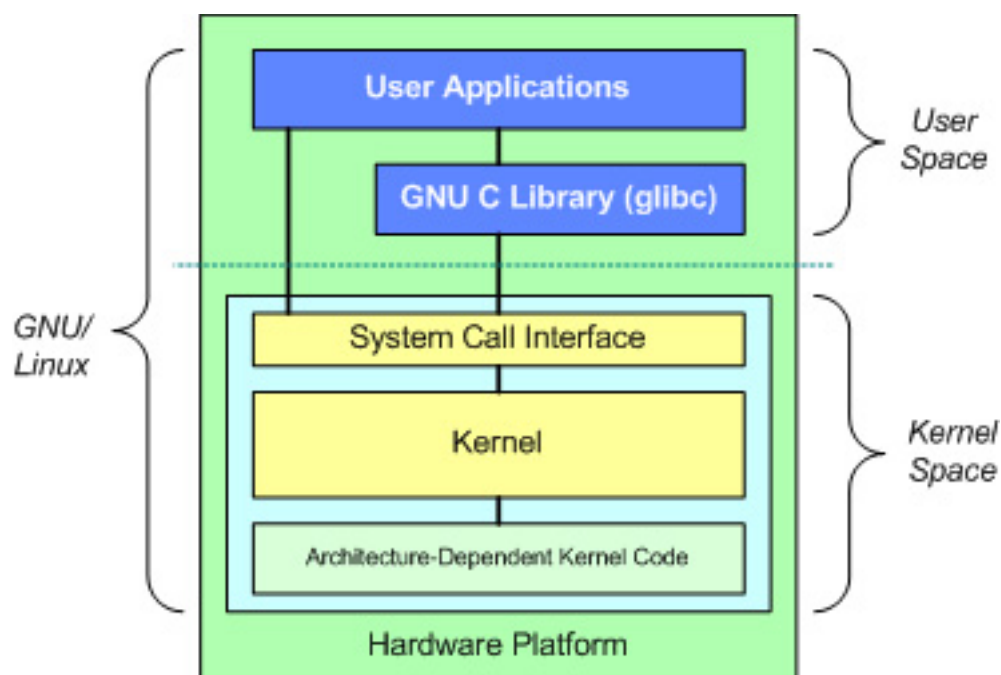


Figura 3.7.: Arquitectura del SO Linux

3.3.1.1. Espacio de Usuario

Son los programas que utiliza el usuario final, tales como: shell, gedit, kpresenter y otros. Todos estos programas residen en el espacio de usuario. Como es lógico dichos programas necesitan interactuar con el hardware del sistema, pero no lo hacen directamente, sino a través de las funciones que soporta el kernel. Cada usuario que se conecte al ordenador tendrá su propio espacio de usuario y sus permisos sobre las funcionalidades del equipo. Además Linux por defecto tiene varios usuarios "robots" para organizar mejor y de forma más segura los procesos del sistema. También tiene un superusuario llamado "root", que tiene todos los permisos del sistema.

3.3.1.2. Espacio del Kernel

También llamado "Núcleo del sistema" es el sistema central de cualquier SO. Se encarga de gestionar los diferentes procesos como la comunicación entre el hardware del computador y los diferentes programas y las tareas imprescindibles para la gestión de recursos (CD-ROM, unidad de disco, impresora, etc.), la interlocución con el hardware (memoria), la multiplexación de los accesos al hardware y ofrecer al usuario una interfaz agradable para su uso. El kernel ocupa un único espacio de dirección. La comunicación entre los procesos de usuario y el kernel, se efectúa a través de la librería de C de GNU, que provee un mecanismo de transición entre las aplicaciones y el kernel. Revisando la figura 3.7 con respecto al espacio del Kernel, encontramos tres niveles, a grandes rasgos son:

1. Llamadas al sistema, implementación de las funciones básicas tales como read o write.
2. Kernel, el código del kernel independiente de la arquitectura, este código es común a todas las arquitecturas de procesadores soportadas por Linux.
3. Código dependiente de la arquitectura, que forma lo que es más comúnmente llamado BSP (Board Support Package). Este código es específico para cada procesador y plataforma.

3.3.2. Funciones de intercambio entre el espacio de usuario y el espacio de kernel

El kernel ofrece una serie de funciones en el espacio de usuario que permiten al usuario interactuar con el hardware o software. Habitualmente en sistemas Linux, este diálogo se hace

Eventos	Funciones de usuario	Funciones de kernel
Carga de módulo	insmod	init_module
Abrir dispositivo	fopen	file operations: open
Leer dispositivo	fread	file operations: read
Escribir dispositivo	fwrite	file operations: write
Cerrar dispositivo	fclose	file operations: release
Quitar módulo	rmmmod	cleanup_module

Cuadro 3.1.: Ficheros equivalentes en espacio de usuario y kernel

a través de las funciones de lectura y escritura de ficheros, ya que en este sistema los propios dispositivos son vistos como ficheros [Wik07].

Por otro lado, en el espacio del kernel, Linux también ofrece una serie de funciones para interactuar por un lado con el kernel (bajo nivel), con los dispositivos hardware y por otro lado permite el paso de información desde el kernel al espacio de usuario [Wik07].

Es común que para cada función en el espacio de usuario que permite el uso de dispositivos o ficheros, exista su equivalente en el espacio de kernel que permita el traspaso de información del espacio de kernel al de usuario y viceversa. Esto se puede apreciar en el siguiente cuadro: 3.1

3.4. Controlador MadWifi

En el año 1992, Sam Leffler un Ingeniero de la computación, tenía la ilusión de construir una red en malla para tarjetas inalámbricas, utilizando componentes libres, pero en ese tiempo el hardware existente y soportado era demasiado restrictivo para ese propósito, de esta forma que Sam se puso en contacto con Atheros para brindar soporte a su hardware en para la comunidad del software libre. Al principio el trabajo de Leffler fue realizar un controlador para las tarjetas inalámbricas en FreeBSD, para los sistemas BSD, y en su tiempo libre empezó a portar el código del controlador a Linux. De esta manera nace **MadWiFi**, que es un tipo de controlador multibanda para interfaces de red WiFi. MadWiFi es un software desarrollado para tarjetas wireless con chipset Atheros. Sus siglas significan “Multiband Atheros Driver for Wireless Fidelity”, es decir, provee de drivers para el kernel de Linux, soportando chipset Atheros. Con el driver la tarjeta inalámbrica aparecerá como una interface de red dentro del sistema, permitiendo que se pueda configurar la tarjeta de una manera simple usando el ifconfig, iwconfig, etc. Para cargar los módulos de MadWiFi se necesita un Kernel 2.4.23 o superior y 2.6.x. en el SO Voyage Linux [mp].

El controlador MadWiFi consta de 4 partes:

- Pila de protocolos net802.11 de FreeBSD: tiene las funciones genéricas de 802.11 y las funciones callback que pueden ser remplazados por los dispositivos. Soporta múltiples opciones hardware, pero MadWiFi es solamente usado para los dispositivos Atheros.
- La parte ath: define las callbacks específicas de atheros para la capa net802.11, como también el acceso del hardware a través del HAL.
- HAL(Hardware Abstraccion Layer): Capa de abstracción de hardware, todos los accesos al hardware tienen que pasar por este componente de código fuente cerrado, el cual es mantenido por Atheros. Desafortunadamente no hay documentación para esto.
- Algoritmos de selección de tasa: diferentes algoritmos para seleccionar la mejor tasa de transmisión han sido implementados.

3.4.1. Capa Abstraccion del Hardware de MadWiFi

En esta capa se tiene un conjunto de software, desde el cual podemos acceder al hardware, que va a ser controlado. Desde la HAL se controlan varios aspectos de los dispositivos inalámbricos, como por ejemplo: potencia de transmisión, tasa de transmisión, frecuencia o canal, etc.. Por

3. Marco Teórico

ejemplo la frecuencia, ya sea una banda ISM² o no-ISM³. Las bandas no-ISM están reguladas por entes reguladores en todo el mundo, estos entes tienen leyes que restringen el uso del espectro inalámbrico en: las bandas de frecuencia y la potencia de transmisión. Los entes reguladores controlan el espectro según el dominio regulatorio y el código de país. Por esa razón Sam Leffler y Atheros acordaron que la HAL se distribuyera en formato binario y código cerrado, de manera que usuarios no autorizados no pudieran modificarla fácilmente. Ésto asegura que las configuraciones no autorizadas no sean aplicadas en los dispositivos inalámbricos.

3.4.2. Modos Operativos

Se soportan los siguientes modos operativos:

- **sta** Modo Estación, el dispositivo actúa como una típica estación cliente WLAN. Este es el modo por defecto si no se especifica otro modo.
- **ap** Punto de Acceso (AP), modo maestro. Este dispositivo actúa como Punto de Acceso para otras estaciones clientes WLAN.
- **adhoc** Ad-hoc modo IBSS. Este dispositivo está en una WLAN de igual a igual sin la necesidad de un punto de acceso.
- **ahdemo** Ad-hoc Demo. Es un modo propietario antiguo, no compatible con 802.11.
- **monitor** Este modo se puede usar para monitorizar (sniffing) tramas de red en crudo 802.11.
- **wds** Wireless Distribution System. Este modo puede ser utilizado para crear redes inalámbricas amplias mediante la vinculación de varios Puntos de Acceso.

3.4.3. Cifrado

Para asegurar las conexiones de la red inalámbrica se pueden usar diferentes tipos de cifrado, como:

- **WEP** Wired Equivalent Privacy, con claves de 40/64 bits y 104/128bits; soportado en modos: sta, ap, adhoc.
- **WPA** WiFi Protected Access; soportado en los modos: sta (a través de wpa_supplicant), ap (a través de hostapd).
- **WPA2/IEEE 802.11i** WiFi Protected Access 2; soportado en los modos: sta (a través de archivo wpa_supplicant), ap (a través de archivo hostapd).
- **IEEE 802.1X** Control de Acceso a Red basado en Puertos. Soportado en el modo: ap.

3.4.4. Super A/G

Atheros ha introducido un conjunto de características (propietarias) que han sido diseñadas para incrementar el rendimiento de la red, incluyendo la agregación de tramas grandes, compresión de hardware.

3.4.5. Controlador Ath5k

Ath5k es un controlador Linux completamente de código abierto para tarjetas inalámbricas Atheros. Está basado en MadWiFi y en OpenHAL (la HAL es abierto). Ath5k simplemente hace llamadas a las funciones del hardware directamente.

²ISM: Bandas sin licencia, se usa en el área industrial, científica o médica

³no-ISM: Bandas con licencia, se usa en el área radar civil o militar

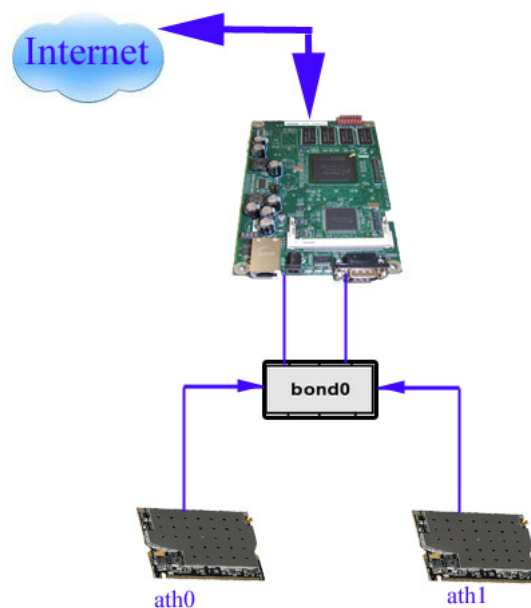


Figura 3.8.: bonding

3.5. Bonding

El controlador bonding para GNU/Linux, originalmente creado por Donald Becker, está incluido en prácticamente todas las distribuciones de GNU/Linux y es un método para la agregación de múltiples interfaces de red sobre un único interfaz lógico, como muestra la figura 3.8 y las interfaces que se agregan al bonding se les denominará interfaces esclavas. Al agregarse como esclavas las interfaces al interfaz de bonding, este último interfaz pasa a controlar la operación general de las demás, y actuar como interlocutor de éstas con el sistema operativo. De esta forma se consigue obtener redundancia y/o soportar balanceo de carga sobre las interfaces que se encuentran en dominio del bonding [TGT⁺09].

Bonding es cargado en el núcleo de GNU/Linux como un módulo con el comando “modprobe”. Debido a que existen multitud de algoritmos para determinar el comportamiento de las interfaces esclavas en la operación común gestionada por el interfaz de bonding, en función de si lo que se pretende es conseguir redundancia, backup o aumento de prestaciones, en la carga del módulo el parámetro más determinante es el parámetro “mode”, que determina dicho algoritmo. Los diferentes modos de funcionamiento determinados por el parámetro “mode” son:

- **balance-rr o 0.** Política round robin. Transmite paquetes en orden secuencial desde el primer slave disponible hasta el último realizando el cambio de un interfaz a otro en función de un algoritmo round robin. Este modo proporciona balanceo de carga y tolerancia a fallos ya que si un envío por un interfaz falla, la retransmisión puede ocurrir por otro interfaz, aumentando la probabilidad de éxito del envío en alguna de sus retransmisiones.
- **Active-backup o 1.** Política Active-Backup. Con este modo se tiene: una interfaz esclava en estado “activo”, por donde cursa tráfico y la otra en estado “pasivo”. Cuando la interfaz activa falla, la otra interfaz que está en estado pasivo toma la dirección MAC y se convierte en la interfaz activa. Este modo sólo ofrece tolerancia a fallos.
- **balance-xor o 2.** Política XOR. Para transmitir una trama lo hace basándose en el siguiente calculo: (MAC de origen XOR MAC destino) % numero de interfaces esclavas. De esta manera se realiza un balanceo de la carga con distintos destinatarios, pero tramas con un mismo destinatario son enviadas siempre por el mismo interfaz. Este modo ofrece balanceo de carga.

3. Marco Teórico

- **broadcast o 3.** Política de broadcast. Transmite todas las tramas por todas sus interfaces slave. Este modo nos ofrece tolerancia a fallos.
- **802.3ad o 4.** Se trata del estandar IEEE 802.3ad (Dynamic link aggregation). Crea grupos de agregación de tráfico que comparten la misma velocidad. Utiliza todos los slave en el agregador activo de acuerdo con la especificación 802.3ad. Es necesario que el driver de la tarjeta de red sea soportada por ethtool ⁴ y soporte el estándar IEEE 802.3ad. En este modo nos ofrece una alta disponibilidad y un aumento de velocidad.
- **balance_tlb o 5.** Transmite balanceando la carga entre los esclavos en función de la carga de cada uno de los slave. El tráfico entrante lo realiza siempre por un slave. En caso de fallo de éste, el otro slave asume toda la tarea. Es necesario que la tarjeta de red soporte ethtool.
- **Balance-alb o 6.** Balanceo de carga adaptativo, consiste en un balance-tlb, pero le añade un ajuste adaptativo a la interfaz entrante.

3.6. Enrutamiento avanzado y control de tráfico en Linux

A continuación se describen tres elementos fundamentales en la parte de enrutamiento de paquetes y sobre los cuales se soporto este PFM.

3.6.1. Iptables

En el año 1998 empezó el proyecto Netfilter/Iptables de la mano de Rusty Russell, En 1999 se fundó el Netfilter Core Team, un grupo de personas a cargo del desarrollo y mantenimiento de este proyecto. Netfilter tiene licencia GLP y fue incorporado a partir de la version 2.3 del kernel de Linux, en marzo del 2000. Iptables es una aplicación que se ejecuta en el espacio de usuario y solo puede ser ejecutado por el superusuario root.^o si se da permiso al usuario para ejecutarlo.

Iptables es la herramienta más popular del framework Netfilter. Algunas partes de netfilter se encuentran integrados en el kernel de Linux, y permite al administrador la definición de un conjunto de reglas aplicables a los paquetes IP que entran y/o salen de un ordenador para realizar las siguientes operaciones: filtrado de paquetes, seguimiento de conexiones, traducción de direcciones IP y puertos [Net].

Existen 3 conceptos importantes en iptables:

- **Reglas.** Una regla de iptables especifica una condición (características que debe cumplir un paquete para que la regla le sea aplicable) y una acción (indica lo que se hace con el paquete si cumple dicha condición).
- **Cadenas.** Una cadena es una lista ordenada de reglas.
- **Tablas.;** Una tabla de iptables contiene un conjunto de cadenas, tanto las predefinidas por el sistema, como las definidas por el usuario.

3.6.2. Iproute2

Iproute2 es un paquete de utilidades desarrollado por Alexey Kuznetsov. Este paquete es un conjunto de herramientas para el control del tráfico, es muy poderoso para administrar interfaces de red y conexiones en sistemas Linux. Iproute2 reemplaza completamente las funcionalidades presentes en ifconfig, route, y arp y las extiende, llegando a tener características similares a las provistas por dispositivos exclusivamente dedicados al encaminamiento y control de tráfico. Éste paquete esta incluido en las versiones posteriores a la 2.2 del kernel de Linux en las distribuciones de Debian y RedHat, así como en sus derivadas [HMvO⁺04].

Algunas de las funcionalidades principales que provee iproute2 son:

- **QoS (Quality of service).** Mecanismo de clasificación en el cual: a) se identifican los paquetes, b) se colocan en diferentes clases o flujos, c) políticas en donde se limita el número de paquetes, d) etiquetado de bytes que pueden ser utilizados en un stream de clasificación particular, e) decisiones de planificación.

⁴ethtool: herramienta útil para conocer el estado de nuestra tarjeta de red y modificar el comportamiento de algunas funciones generales de red como: la versión del controlador, velocidad del dispositivo, etc.,

- **Balanceo de carga.** Se asigna un peso a cada una de las interfaces, y de acuerdo a este valor se distribuye el tráfico entre las diferentes interfaces.

3.6.2.1. Configuración de encaminamiento

Encaminamiento es la función de buscar un camino entre todos los posibles en una red, donde exista conectividad. Dado que se trata de encontrar la mejor ruta posible. Dentro del funcionamiento del “balanceo de carga” a nivel de red, existen ciertas matizaciones, que no es estrictamente cierto que cada petición salga conforme a la regla establecida. Ya que en el kernel guarda el IP destino visitado y a esto se le llama cache de ruteo, donde almacenan las rutas conocidas, esto por un determinado tiempo. Eso hace que cada vez que un paquete tiene que ir a una ruta ya conocida, elija el mismo camino. Para intentar reducir ese tiempo del cache de ruteo, tocamos algunos parámetros que listan, aunque estos no son los únicos.

- */proc/sys/net/ipv4/route/gc_interval*. Parametro que indica el tiempo se debe esperar antes de notificar al *gc_timeout* que una ruta a muerto, es similar al tiempo de gracia.
- */proc/sys/net/ipv4/route/gc_elasticity*. Parametro que controla la frecuencia y comportamiento del algoritmo recolector de basuras de la caché de rutas. Indica los segundos antes de que el kernel salte a otra ruta porque la anterior ha muerto. Valor por defecto 300 segundos.
- */proc/sys/net/ipv4/route/gc_timeout*. Parametro que indica el tiempo en el que el kernel declara una ruta como muerta. En función de este parámetro y *gc_interval* ajustaremos los tiempos para que el kernel deje de utilizar una ruta cuando esta falla o reintenta utilizarla.
- */proc/sys/net/ipv4/route/max_delay*. Retrasos para vaciar la caché de rutas.
- */proc/sys/net/ipv4/ip_forward*. Parametro para activar el enrutamiento en un sistema Linux. El valor 0 significa que esta desactivado y el valor 1 que esta activo.

4. Estado del arte de balanceo de carga en redes inalámbricas

En los últimos años se han propuesto una gran cantidad de enfoques con respecto al balanceo de carga en redes inalámbricas, teniendo en cuenta las diferentes capas de modelo TCP/IP. Estas investigaciones nos ayudan a identificar y comprender los problemas comunes abordados anteriormente, sino que también, nos permiten establecer qué cuestiones están pendientes de ser resueltas.

Para esta revisión de publicaciones científicas se realizó la búsqueda genérica en Google Scholar con las siguientes palabras clave (“load balancing over wireless networks” que arroja 44600 resultados). Como los resultados obtenidos eran demasiado extensos, se decidió acotar la búsqueda utilizando las siguientes palabras clave:

La búsqueda “wireless ”multiple interface“ ”, obtiene 487.000 resultados.

La búsqueda “wireless bonding”, obtiene 22.100 resultados.

La búsqueda “802.11 bonding”, obtiene 3.250 resultados.

La búsqueda “ ”load balancing “ wireless bonding ”, obtiene 1.910 resultado.

La búsqueda “load balancing over wifi”, obtiene 8540 resultados.

La búsqueda “load balancing over 802.11”, obtiene 13400 resultados.

La búsqueda “wifi bonding”, obtiene 1290 resultados.

La búsqueda “wifi bonding madwifi”, obtiene 45 resultados.

Realizando la revisión de los artículos científicos sobre el balanceo de carga en redes inalámbricas se seleccionaron esencialmente 9 referencias que resumen las tendencias más importantes de investigación, y éstas se clasificaron según su enfoque:

- **Enfocados en Capa de Transporte** En esta capa, normalmente mente se encuentran en las redes de telefonía celular ya se realiza un balanceo de carga en la capa de transporte, como se expresan en: [DWPW07][ADYO07][WRH10], han conseguido aumentar el caudal utilizando la red celular 3G y una red Wifi dentro de un teléfono móvil. Para utilizar estos dos servicios se adiciona un nuevo diseño denominado “CTCP” que incluye mecanismos para el equilibrio de carga de tráfico en varias rutas, este diseño es añadido en la cabecera de TCP [ADYO07], y posteriormente se crearon algoritmos de control de congestión múltiple, esto para equilibrar el tráfico [WRH10].

- **Enfoque en Capa de Red** Para el balanceo de carga en esta capa, se trabaja en los protocolos de encaminamiento. Este tema del encaminamiento es muy importante en las redes Mesh y es aquí donde se encuentra la gran oportunidad del balanceo de carga desde un origen a un destino, a través de diferentes rutas.

En [iWMN11], el trabajo se centra en la gestión de las cargas de tráfico en ciertas rutas de acceso, lo que conlleva a tener resultados negativos para la red cuando uno todo el tráfico intenta salir por un solo nodo por el cache de enrutamiento. Para este problema se crean dos algoritmos: 1) descubre rutas alternativas para llegar a un destino y 2) divide el tráfico para que pueda llegar al destino usando otras rutas, para emplean un esquema round robin, donde un nodo envía cada paquete por una ruta diferente. Pero este método genera el problema, de que los paquetes llegan al destino de forma desordenada y esto genera perdidas innecesarias en el protocolo TCP.

- **Enfoque en Capa de Enlace** Aquí los trabajos se basan en que las interfaces físicas de un nodo son integradas y asociadas a una única interfaz virtual y todo el tráfico es enviado a dicha interfaz virtual. Teóricamente se esperaría sumar matemáticamente todos los anchos de banda de cada una de las interfaces físicas asociadas.

En el proyecto Net-X [CKV06] [Tri07], se trabaja con redes Mesh WiFi, en Linux con el kernel v.2.4. Se implementó una arquitectura software para soportar múltiples interfaces

4. Estado del arte de balanceo de carga en redes inalámbricas

físicas y usar múltiples canales radio. Para ello se modificaron los controladores Madwifi y Bonding. Luego en [Mü07], se intentó actualizar esta arquitectura para hacerla compatible con el kernel v2.6, pero al final se concluyó que era poco práctico porque la arquitectura del kernel v.2.4 y v.2.6 existen algunos cambios como: llamada a los dispositivos, número de particiones, etc. El proyecto Net-X la interfaz lógico, se usan para la recepción y/o transmisión de paquetes simultáneamente.

En [FG11], se usan dos radios, una exclusiva para transmitir y la otra exclusiva para recibir. Esto implicó una aplicación en el espacio de usuario y la modificación del driver inalámbrico Madwifi y Bonding del kernel de Linux. Los resultados de la simulación Ns-2 muestran un incremento sustancial en el rendimiento, comparado con la configuración estandar de un solo canal y una sola radio.

En la tesis [Sor08], de la red Willay Cuzco en el Perú, utilizan la tecnología Mikrotik y se pretendía tener redundancia y balanceo de carga en sus enlaces, se analizaron tres técnicas: Bonding Nstreme; puede trabajar en dos modos de operación bonding, el respaldo y balanceo de carga (modo round robin). En cualquier caso las interfaces no se pueden usar de forma independiente (transmiten y reciben por todas las tarjetas) aunque el caudal aumente teóricamente al doble, pero el caudal efectivo puede que no aumente demasiado debido a las colisiones de los paquetes, además no es muy estable en Mikrotik. Dual Nstreme; tiene dos nodos enlazados, cada nodo tiene dos tarjetas inalámbricas. Esta técnica usa exclusivamente un enlace para transmitir y la otra para la recibir los datos, además si una de las 4 tarjetas inalámbricas se estropea, los enlaces quedaran inutilizadas. Nstreme con OSPF; el funcionamiento es similar a la técnica dual Nstreme que crea enlaces dedicados, uno para la transmisión y la otra para la recepción de datos pero sin que estos colisionen a través del OSPF que manipula los costos de cada interfaz inalámbrica en el nivel de enrutamiento, pero además de que se estropea una de las tarjetas el otro enlace queda operativo para transmitir y recibir los datos. Finalmente se escogió la técnica de Nstreme con OSPF ya que la selección del enlace transmisor o receptor no lo hace mediante un protocolo propietario de MikroTik, además de presentar redundancia por si un enlace falla.

La conclusión de la revisión de los artículos científicos sobre el balanceo de carga en redes inalámbricas no existe mucha información, pero destaca el proyecto Net-X de la Universidad de Illinois en Urbana-Champaign donde se ha conseguido un balanceo de carga en el capa de enlace, modificando los controladores de bonding y Madwifi en redes Mesh, el código fuente de este proyecto esta disponible en <http://www.crhc.illinois.edu/wireless/netx.html>, este proyecto se desarrollo para sistemas embebidos con una versión de SO GNU/LINUX kernel 2.4. Dicho proyecto no da la ilusión de que se puede realizar el balanceo de carga en capa de enlace en SO GNU/LINUX con kernel 2.6.X. También encontramos el balanceo de carga a capa de red, donde los trabajos presentados realizar el balanceo de carga con la ayuda de los protocolos de encaminamiento.

Parte II.

OBJETIVO DEL PROYECTO

5. Objetivo del proyecto

5.1. Objetivo General

De acuerdo con lo expresado en los anteriores capítulos, existen multitud de dificultades en zonas rurales aisladas de países en desarrollo, que una correcta introducción de TIC puede ayudar a mitigar. Para la introducción de estas tecnologías, un problema específico de estas regiones es el de la conectividad entre equipos, debido a multitud de condicionantes geográficos y sociales. En Latinoamérica, la fundación EHAS ha desarrollado una solución de bajo coste para la conectividad inalámbrica, mediante sistemas WiFi modificados para establecer enlaces WiFi de largo alcance (decenas de kilómetros) dispuestos como repetidores en las áreas de actuación. No obstante, a medida que estas redes han ido incrementando su uso en número de usuarios y de servicios ofrecidos, los límites de esta tecnología en cuanto a capacidad han comenzado a vislumbrarse como un grave problema para la sostenibilidad futura de las redes. De esta situación surge la motivación de tratar de encontrar medios para aumentar la capacidad de las redes, que permitan a éstas seguir introduciendo usuarios y servicios, posibilitando aumentar el alcance de los beneficios de las TIC en las regiones aisladas objetivo de los proyectos de EHAS, y realizarlo de forma que pueda ser extensible a otros escenarios similares o, al menos, fácilmente adaptable. Tras el estudio del estado del arte en la extensión de esta capacidad a través de la agregación de interfaces inalámbricas a los encaminadores de tráfico que componen la red que distribuyen los servicios, destacan 2 aproximaciones cuyo estudio comparativo es algo no resuelto a día de hoy.

Por tanto, en este PFM, se decide abordar el estudio de una solución de enlaces redundantes con carga compartida, analizando dos soluciones: Bonding¹ y el balanceo de carga a nivel de red, ambas sobre tecnología WiLD.

Para lograr lo anterior, se desarrollaron los siguientes objetivos específicos:

1. Analizar algunas de las diferentes técnicas existentes, intentando definir la más adecuada.
2. Medir la QoS alcanzable por cada técnica de balanceo de carga escogida, a través de 4 parámetros (retardo, variación del retardo, caudal y pérdida de paquetes).
3. Determinar la solución más idónea para la implantación en la red que EHAS instaló en las orillas del río Napo, en la Amazonia peruana, a principios de 2007.
4. Analizar el impacto sobre los costes de instalación de la solución escogida.
5. Identificar las ventajas aportadas por la solución en materia de sostenibilidad de las redes en función de la topología de la red.

¹Bonding: Método para la agregación de múltiples interfaces de red sobre un único interfaz lógico, realizando balanceo de carga a nivel de enlace.

Parte III.

MÉTODOS Y MATERIALES

6. Métodos

Una vez que se conoce el problema, el contexto y el objetivo que se pretende resolver con este PFM, conviene detallar el método de trabajo que se seguirá para la solución del problema planteado. Como se puede apreciar (ver figura 6.1), se esboza la metodología del proyecto. Esta metodología define el transcurrir del desarrollo del proyecto. Esta investigación pasa por diferentes etapas. Se inicia de una idea, que surge de una necesidad y va madurando a través de las diferentes etapas del desarrollo. Estas etapas generan salidas que obligan a replantear los procesos, si estos no cumplen con el objetivo, lo que se torna en un bucle de I+D hasta encontrar una solución compuesta por un desarrollo experimental que permite extraer una conclusión de la investigación. Esa solución no es mas que una foto de un momento de la investigación. A continuación se describen los pasos para cada una de las fases, hasta alcanzar el objetivo que se ha propuesto.

1. **Etapla del planteamiento de la idea.** Esta es la primera etapa donde inicia la identificación de la necesidad. Estas ideas pueden surgir entre las reuniones con los posibles interesados en el proyecto. Una idea surge de una necesidad, que en el caso que se plantea, es la mejora de las infraestructuras de telecomunicaciones, que en la red del río Napo requieren de mayor estabilidad para permitir la introducción de nuevos servicios de telemedicina.
2. **Etapla análisis preliminar.** Esta etapa es considerada como un primer filtro del proyecto, donde un grupo de integrantes con experiencia y conocimientos previos analizan si la idea necesita un replanteamiento, si pasa a la siguiente etapa, o por el contrario, se desecha la idea porque no merece la pena ser tomada en consideración.
3. **Etapla diseño preliminar.** Aquí se estiman los recursos necesarios para llevar a cabo la idea, estos recursos pueden ser: humanos y materiales. En el diseño preliminar es necesario identificar los mayores riesgos del fracaso para este proyecto. Estos riesgos pueden ser aquéllos que lo sumen en un bucle infinito sin ninguna salida útil o trabajos que pueden requerir un incremento inadmisible de recursos para su superación.
4. **Asignación de recursos.** Con la idea clara, los riesgos identificados y el diseño preliminar ya terminado, llega el momento de localizar los recursos, para luego conseguir recursos económicos y de esta manera garantizar el término del proyecto. En algunos casos, se puede disponer internamente de los recursos y de esta manera dar comienzo a las siguientes fases del proyecto. Para esta investigación se cuenta con los recursos materiales, puestos a disposición del equipo de trabajo por parte del departamento de Teoría de la Señal y las Comunicaciones (TSC) de la Universidad Rey Juan Carlos, así como con los recursos humanos para realizar el proyecto, que consisten esencialmente en la autora de este PFM y los tutores del mismo.
5. **Fase de formación.** En esta fase, el personal que va a desarrollar el proyecto requerirá de un tiempo para adquirir las habilidades básicas que necesita el proyecto para el inicio del desarrollo. En esta fase se profundiza más ampliamente en el estudio del estado del arte, actualizando el diseño preliminar y revisión de los recursos requeridos, además del estudio de nuevas tecnologías. Como entrada tendremos la búsqueda de artículos científicos sobre “Balanceo de carga en redes inalámbricas”, que nos pudieran proporcionar información referente a las diferentes maneras de enfrentar el problema propuesto, a la vez que obtendríamos el estado actual de este tema de investigación reflejado en la literatura científica y una identificación de los recursos. Nos dará como resultado un equipo de trabajo cualificado para realizar la tarea, con una distribución apropiada del trabajo, y un buen nivel de interrelación en el equipo.
6. **Fase de desarrollo.** Esta es la fase en la que se realiza la configuración apropiada para el funcionamiento de una solución. Cada vez que se termina de hacer una configuración

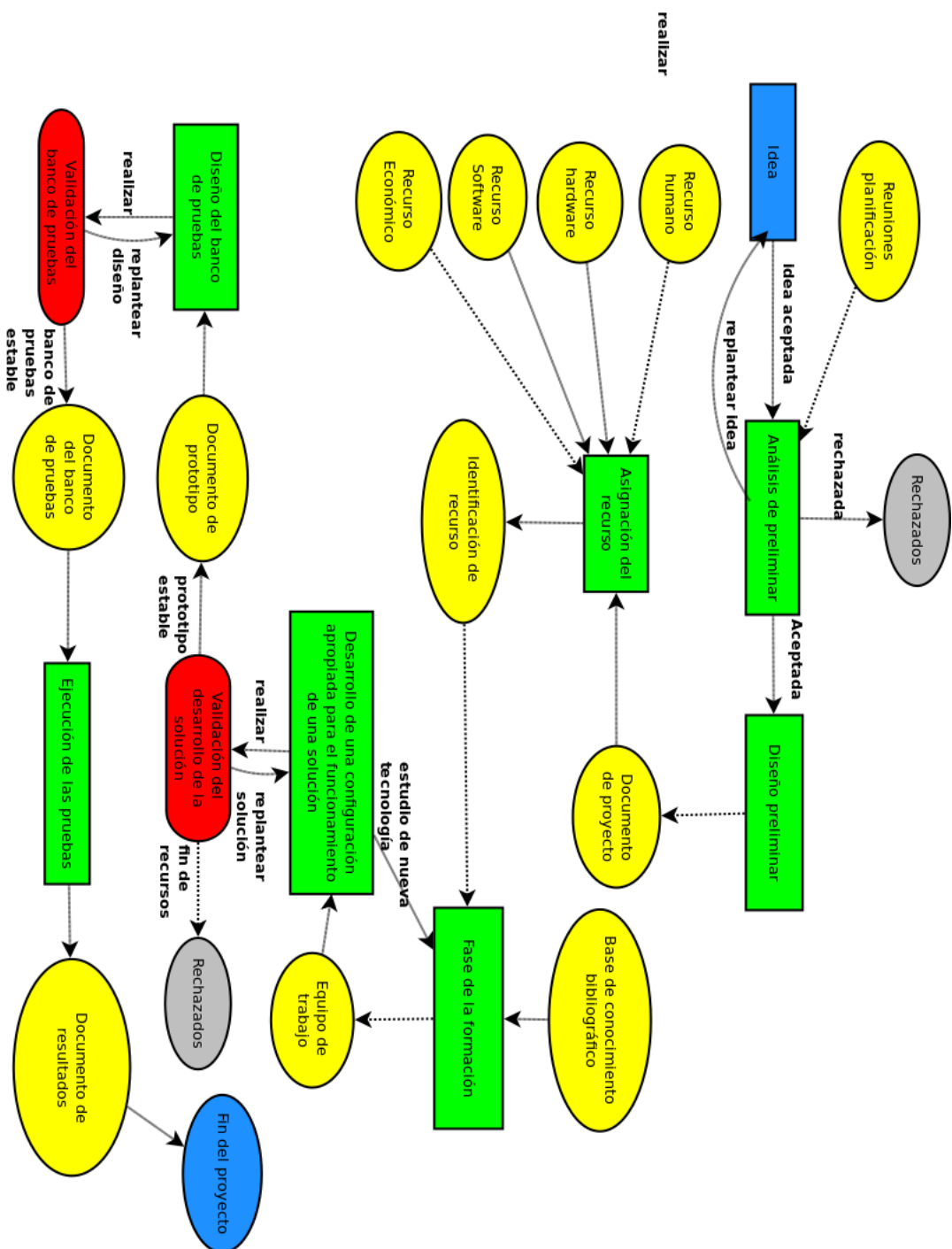


Figura 6.1.: Metodología del proyecto

6.1. Requisitos de selección de la solución adecuada para el balanceo de carga

apropiada para el funcionamiento de una solución entra a la fase de validación. Si no pasa esta fase, puede entrar nuevamente a la fase de desarrollo, donde se probarán nuevos mecanismos para encontrar la solución, o desechar la configuración probada. Todo esto hasta conseguir la configuración apropiada para esa solución.

7. **Fase de validación de desarrollo.** En esta fase se usarán ciertos parámetros de medición, para validar si la configuración es apropiada para esa solución y si puede pasar a la otra fase, o si debe replantear o desechar esa configuración.
8. **Fase de diseño de banco de pruebas.** Después de tener la configuración apropiada para una solución y validada, ahora se entrará a la fase de diseño del banco de pruebas, aquí se desarrolla el escenario de experimentación donde se automatiza el funcionamiento de la configuración apropiada de una solución.
9. **Fase de validación de banco de pruebas.** Una vez que se tiene automatizada la configuración apropiada de una solución, se tienen que realizar varias pruebas, para ver si los resultados usan todos los parámetros introducidos correctamente. Muchas veces, puede existir mucho ruido en el medio o que algún recurso de hardware no tenga buen funcionamiento. Se valida así la automatización del diseño del banco de pruebas comprobando manualmente que la configuración automática del banco de pruebas se efectúa correctamente.
10. **Fase de ejecución de pruebas.** Tras su validación se lanzan las pruebas y recogen los resultados.
11. **Fase de validación de la ejecución.** Se realiza la validación de los resultados obtenidos. Si este resultado no es acorde a lo que se busca, se desecha esa prueba y se vuelve a fases anteriores para subsanar los posibles errores, hasta que se tenga confianza en los resultados obtenidos.

A continuación detallaremos un poco más los requisitos para la selección de la solución adecuada para el balanceo de carga.

6.1. Requisitos de selección de la solución adecuada para el balanceo de carga

- La solución seleccionada tiene como requisito imprescindible el uso de software libre, ya que de esta manera se puede garantizar el futuro de la investigación.
- La solución seleccionada tiene como requisito el uso de GNU/Linux Voyage como SO, esto se debe a que en el escenario donde va a ser aplicada esta solución usa este SO, además de que la fundación EHAS trabaja bajo este SO en varios de sus proyectos.

Con estos requisitos para seleccionar la solución adecuada para el balanceo de carga.

7. Materiales

7.1. Materiales de hardware

Para poder realizar este PFM se utilizó instrumental del laboratorio de Teoría de la Señal y las Comunicaciones de la URJC, que fue necesario en cada una de las etapas del proyecto:

Analizador de espectros o WiSpy bi-banda. En las pruebas iniciales de diseño de la red inalámbrica fue necesario poder observar a nivel RF las señales presentes en las bandas de trabajo, por lo que, más allá de la detección de señales WiFi mediante una tarjeta 802.11, se empleó un analizador de espectros que observa las señales del entorno sin distinción entre señales WiFi y cualquier otra señal presente, para seleccionar aquellas frecuencias con menor presencia de interferencias de cualquier tipo.

Ordenadores portátiles. Se emplearon 2 ordenadores en el proceso de desarrollo. Uno de ellos fue empleado como estación de trabajo y en todo el desarrollo software. El otro, para algunas pruebas de validación de la interconexión de los equipos ALIX.

Cables RJ-45 y 1 hub. Necesarios para implementar los escenarios de pruebas planteados en este PFM.

7.1.1. Hardware para el escenario de experimentación

En este PFM se tiene dos estaciones WiFi, que nos sirve de enrutadores inalámbricos, que están asentados en la siguiente placa base:

Placa base. Una placa Alix, para instalar el sistema empotrado.

Compact Flash (CF). De 4 Gb, para almacenar el SO.

Tarjetas radio 802.11 a/b/g miniPCI. dos radios por placa, donde una radio trabaja como punto de acceso y la otra como estación cliente.

Antenas dipolo y pigtail. dos antenas y dos pigtail por tarjeta inalámbrica (cuatro por placa), para las pruebas de conectividad de redes 802.11. Ha sido necesario emplear antenas externas que trasladen la señal de las tarjetas inalámbricas hacia las antenas receptoras. Si bien podría haberse anulado la diversidad de las tarjetas y emplear únicamente una antena por tarjeta inalámbrica, se decidió utilizar la diversidad de las tarjetas con el fin de mejorar la señal recibida y evitar los efectos adversos que la experimentación en el laboratorio pudieran derivarse de las interferencias destructivas causadas por el multitraxecto.

7.2. Materiales de software

Además de los materiales de hardware utilizados en el apartado anterior, se utilizaron herramientas de terceros durante la realización de pruebas y procesamiento de resultados, los cuales se mencionan a continuación:

Sistema Operativo GNU/Linux Ubuntu 10.04, (PC), Voyage GNU/Linux 0.6.5 (empotrado) con el kernel 2.6.30 y Voyage GNU/Linux 0.7.5 (empotrado) con el kernel GNU/Linux 3.0.

Lenguajes de programación: Shell script (configuraciones automáticas y pruebas automatizadas).

Compiladores: gcc/gcc++ (programación sobre PC).

Versionado de código y memoria del proyecto: subversion.

Registro de eventos del sistema: syslog.

Generadores de tráfico: D-ITG e iperf.

Redirección de puertos y balanceo de carga a nivel de red: iptables y shorewall.

Redacción de documentos: LATEX y Kile.

7. *Materiales*

Driver de tarjeta inalámbrica: Madwifi 0.9.4-rc2-1+7.0-1_i386 y ath5k.

Controlador de balanceo de carga a nivel de enlace: ifenslave-v1.1.0.

Tratamiento de imágenes y diagramas: Día y Gimp.

Parte IV.

RESULTADOS

8. Selección de componentes de la estación Wifi

En este capítulo se realiza la selección de los materiales de hardware y software que intervienen para el funcionamiento adecuado de una estación WiFi, cuando este tiene enlaces redundantes y balanceo de carga entre las dos estaciones Wifi.

8.1. Hardware de la estación WiFi

Los componentes seleccionados para el sistema de comunicación entre las estaciones WiFi, tienen la tarea de gestionar, mantener y llevar acabo la comunicación entre las dos estaciones. Una estación esta compuesto de:

- Placas de cómputo.
- Módulo WiFi (802.11a).
- Antenas dipolo.

8.1.1. Placas de cómputo

Esta placa de cómputo tiene que soportar al sistema operativo GNU/Linux, por eso se ha seleccionado la placa para empotrados ALIX 2d2 de PC Engines. Esta placa dispone de todo lo necesario para albergar un sistema operativo completamente funcional. Además, esta placa es usada en el proyecto del río Napo, por lo que se tiene una experiencia previa con el uso de este dispositivo que no se dispone con otro equipamiento. Las características fundamentales de la placa de empotrados ALIX 2d2 (ver figura: 8.1)son las siguientes: CPU: 500 MHz AMD Geode LX800 (x86)

DRAM: 256MB DDR DRAM

Almacenamiento: 1 socket CF

Expansión: 2 ranuras MiniPCI

Conectividad: 2 puertos Ethernet 10/100

I/O: 1 puerto RS232, 2 puertos USB

8.1.2. Módulo WiFi (802.11a)

El módulo WiFi seleccionado para el proyecto es la radio Mini-PCI CM9 y CM9-GP de Wistron Neweb. Ambas radios son 802.11a/b/g de chipset Atheros, capaces de funcionar perfectamente con el driver madWiFi de GNU/Linux. Las especificaciones más relevantes de esta radio son las que siguen:

- **Frecuencias de trabajo:**
 - 802.11b/g → 2,312-2,497 GHz.
 - 802.11a → 4,920-6,1 GHz.
- **Chipset:** Atheros AR5213A
- **Potencia de transmisión/sensibilidad en recepción:**
 - IEEE 802.11a:
 - 17dBm/-88dBm @ 6Mbps
 - 13dBm/-71dBm @ 54Mbps

8. Selección de componentes de la estación Wifi



Figura 8.1.: ALIX 2d2 de PC Engines



Figura 8.2.: Wistron CM9

- IEEE 802.11b:
 - 18dBm/-95dBm @ 1Mbps
 - 18dBm/-90dBm @ 11Mbps
- IEEE 802.11g:
 - 18dBm/-90dBm @ 6Mbps
 - 15dBm/-74dBm @ 54Mbps

Inicialmente se empezó usando la banda 2,4Ghz en el estándar 802.11g, pero se realizo un análisis del espectro con el analizador de espectros y se detecto que dicha banda estaba muy saturada. Por lo que se decidió trabajar en la banda de 5 Ghz.

8.1.3. Antenas dipolo

Las dos estaciones con la que cuenta esta investigación, requieren de conectividad WiFi 802.11 y cada una de las estaciones dispone de 2 tarjetas inalámbricas que ofrecen conectividad entre las dos estaciones.

Para determinar el tipo de antena que se ha de emplear, se realizó un reconocimiento del medio donde se va llevar acabo esta investigación. En trabajo se realiza en el laboratorio del TSC de la URJC. Por lo cual se decide usar antenas de poca ganancia, porque esta investigación se realiza en un ambiente cerrado y esto genera reflexiones en las paredes. Antenas de mayor ganancia supondrían mayores interferencias.

Conocido el medio donde se va a realizar la investigación, y limitada la ganancia de las antenas, se procedió a seleccionar el tipo de antena adecuado para la estación WiFi, donde se decidió

usar una antena omnidireccional que proporciona una amplia cobertura y evita la necesidad de realizar un apuntamiento.

Frecuencia GHz: 2.4 / 5 GHz.

Ganancia : 5 dBi.

Potencia de Salida: 2 W

Polarización: Vertical

VSWR: <= 2.0

Ancho del haz a 3dB: E-plan 25° H-plan 360°

Dimensiones Diam x L: 12.9 x 165 mm

Peso: 33 g

Pigtail: Mini Coaxial

Conector: RPSMA MALE

Temperatura: -20 a +80°C

Humedad: 40 % 60 % RH

Direccional: No

8.2. Software de la estación WiFi

En este apartado se describe el software seleccionado para el funcionamiento de la estación Wifi, en el cual realizaremos pruebas sobre la investigación que aborda este proyecto.

8.2.1. Sistema Operativo GNU/Linux Voyage 0.6.5

El sistema operativo seleccionado desde el inicio de la investigación es GNU/LINUX Voyage que construye un entorno plenamente funcional, capaz de habilitar drivers para todos los periféricos de los que dispone la placa ALIX, además de red del río Napo esta implementado en este SO, entonces solo nos quedaba determinar que versión de Voyage Linux se iba a utilizar.

Para seleccionar la versión definitiva de SO de GNU/LINUX Voyage paso por las siguientes versiones: Voyage Linux 0.6.5, Voyage Linux 0.7.0 y Voyage Linux 0.7.5, en el proceso de encontrar una solución adecuada para el cumplimiento del objetivo de este PFM. Al final este trabajo de investigación se selecciona la versión Voyage Linux 0.6.5. porque las dos soluciones planteadas funcionan teniendo este SO.

Sobre este SO, esta instalado el paquete de bonding, además se ha realizado los script de autoconfiguración de las dos soluciones planteadas.

8.2.1.1. Instalación de bonding

Para instalar el software de bonding de la manera más sencilla dentro de las placas ALIX, solo hacen falta los siguientes pasos:

1. Dentro de las ALIX, tiene que estar en la sesión de root, y tener permisos de escritura.
2. Dentro de las ALIX, verificar si se tiene acceso a internet.
3. Ejecutar: apt-get install ifenslave-2.6.
4. En el archivo /etc/modules, se añade la línea: bonding o se instala el modulo con el comando: modprobe bonding.

Para la configuración de bonding se realizarán los pasos indicados en el capítulo 10 de la solución bonding.

8.2.1.2. Script de configuración de la solución de bonding

Aunque el desarrollo de los script tiene como fin la autoconfiguración de los escenarios en el que se va a trabajar en este PFM, es necesario configurar manualmente ciertos parámetros que se encuentran dentro de las placas ALIX y que son:

8. Selección de componentes de la estación Wifi

1. Dentro del archivo ubicado en la ruta /home/interface-bond se encuentra la configuración del modo bonding, este archivo puede ser modificado por el usuario root. Aquí se configuran los parámetros de interfaces como: dirección IP, frecuencia, modo de trabajo de la interfaz, además de unos parámetros privados, para la eliminación de los ACK.
2. Para aplicar la configuración de bonding basta con ejecutar “./home/comun bonding”

El funcionamiento de estos script se detallará en mayor profundidad en el capítulo 10 de la solución de bonding.

8.2.1.3. Script de autoconfiguración de la solución del balanceo de carga a nivel de red

1. Dentro del archivo ubicado en la ruta /home/interface se encuentra la configuración del balanceo de carga a nivel de red, este archivo puede ser modificado por el usuario root. Aquí se configuran los parámetros de interfaces como: dirección IP, frecuencia, modo de trabajo de la interface.
2. Para aplicar la configuración de bonding basta con ejecutar “./home/comun iptables”

Este script de automatización será más detallado en el capítulo 11 de la solución de balanceo de carga a nivel de red.

9. Diseño de la red

En este PFM tratamos de solucionar algunos de los problemas de una red de telemedicina implementada en el las orillas del río Napo. Esta red esta compuesta de 17 puestos y centros de salud que están interconectados entre sí (usando una topología de red lineal). Si intentamos simular esta red en un laboratorio, tendríamos mucha interferencia entre cada estación, al tiempo que requeriríamos demasiados equipos. El propósito de este PFM es conseguir enlaces redundantes y aumentar el ancho de banda. Para demostrar este comportamiento sólo nos hace falta tener dos estaciones (ver figura 9.1) que se comuniquen entre si, y si el escenario fuese más grande, se introduciría otro factor que sería el de la reutilización de frecuencias manteniendo una distancia cocanal suficiente para que no se produjesen interferencias de relevancia. Además en este capítulo se definen los parámetros de configuración común que se tienen en las dos soluciones propuestas (tasa de transmisión, canal, etc.), el criterio en el que es montado el laboratorio del piloto y el criterio en el cual se va a medir la QoS de las soluciones.

Para este PFM se ha diseñado un escenario inicial de configuración apropiada para el funcionamiento de las dos soluciones (ver figura: 9.2), donde se pueden apreciar la configuración inicial de la que se parte para el desarrollo de este PFM. Las subredes en las que se trabaja son las siguientes:

1. Subred 10.1.5.0/24. En esta subred se comunica todo el escenario, es necesaria para preparar el escenario en el cual se va a trabajar.
2. Subred 10.9.1.0/24. Se usa para sincronizar el reloj entre el equipo_1 y equipo_2.
3. Subred 10.1.6.0/24. y 10.1.7.0/24. En estas subredes se comunican el equipo_1 y la ALIX_1, fueron necesarios dos subredes porque se van a transmitir varios flujos simultáneamente que debían simular tener precedencia y destino en distintas subredes.
4. Subred 10.1.10.0/24. y 10.1.11.0/24. En estas subredes se comunican el equipo_2 y la ALIX_2, fueron necesarios dos subredes por el mismo motivo anteriormente mencionado para la ALIX_1.
5. Subred 192.168.0.0/24. En esta subred se comunican la ALIX_1 y ALIX_2 y a este enlace se le llamó GEO_0.
6. Subred 192.168.1.0/24. En esta subred se comunican la ALIX_1 y ALIX_2 y a este enlace se le llamó GEO_1.

Ahora se va a detallar cada uno de los componentes que se muestran en el escenario de configuración del inicial de experimentación, además de los parámetros comunes entre las dos soluciones planeadas sobre el balanceo de carga.

9.1. Componentes y configuración del diseño de Red

Equipo_1 Este ordenador portátil bautizado como “emisor”, es el equipo central de todo el proyecto, donde se tienen almacenadas todas las configuraciones que se van a realizar en todos los equipos. Para la comunicación de todas las configuraciones se utiliza la subred 10.1.5.0/24 y dicho ordenador tiene instalado el servidor NTP que realiza la sincronización de los relojes del escenario. Además, para las pruebas de inyección de tráfico, trabaja también en las subredes 10.1.6.0/24 y 10.1.7.0/24.

Equipo_2 Este ordenador portátil bautizado como “receptor”, nos ayuda en las pruebas de inyección de tráfico siendo el destinatario de los flujos transmitidos por “emisor”. En las pruebas de inyección de tráfico emplea las subredes 10.1.10.0/24 y 10.1.11.0/24.

9. Diseño de la red



Figura 9.1.: Escenario real del desarrollo del prototipo del laboratorio

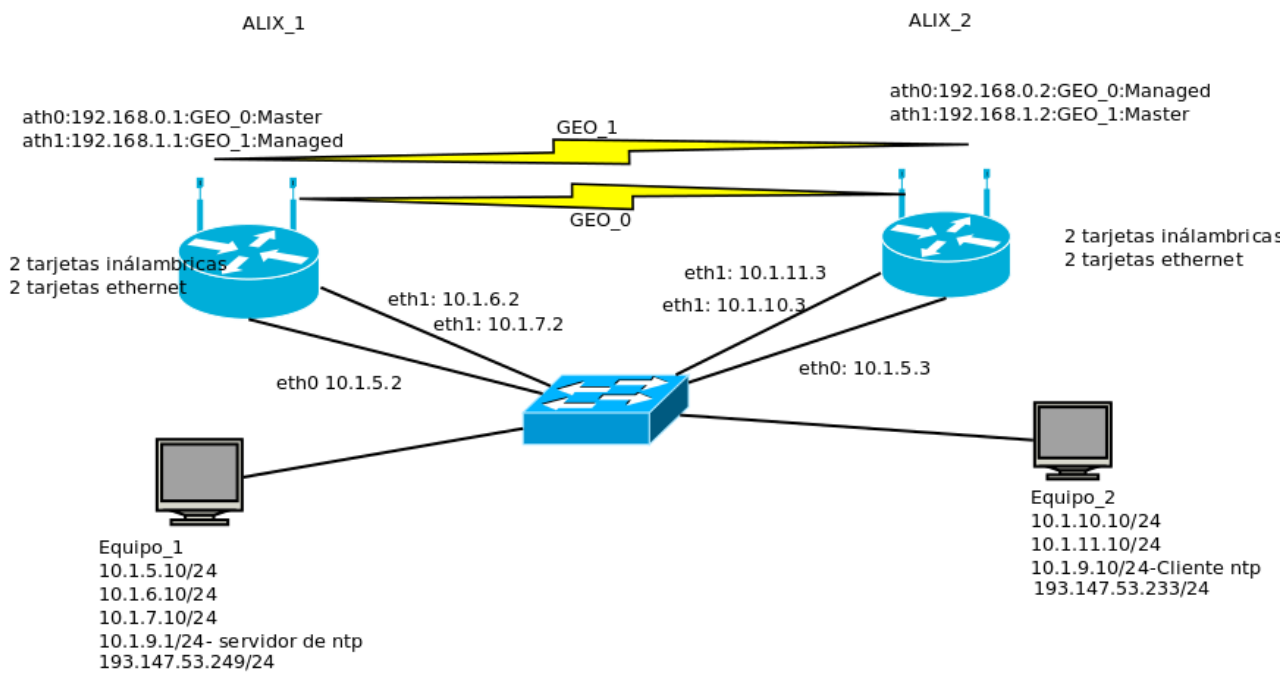


Figura 9.2.: Escenario de configuración inicial de experimentación

9.2. Parámetros de configuración en común para las soluciones

Tasa de bit teórico	Tasa de bit real	Tamaño de paquete	Paquetes a transmitir
6	5	1472	446
9	7,4	1472	660
12	9,2	1472	820
18	12	1472	1070
24	16,5	1472	1471
36	20,8	1472	1855
48	27,6	1472	2461
54	31,8	1472	2835

Cuadro 9.1.: Tasa de transmisión y paquetes a transmitir

ALIX_1 Esta estación WiFi esta compuesta por dos radios y la red trabaja en modo infraestructura. Una de la radios esta configurada como un punto de acceso (AP) y la otra como una estación cliente. En nuestro caso la interfaz ath0 actúa como un AP y tiene la dirección IP 192.168.0.1. La interfaz ath1 actúa como estación cliente, tiene la dirección IP 192.168.1.1 y se conecta al AP del ALIX_2.

ALIX_2 Esta estación WiFi al igual que el ALIX_1 esta compuesto por dos radios y la red trabaja en modo infraestructura. Una de la radios esta configurada como un punto de acceso (AP) y el otro como una estación cliente. En nuestro caso la interfaz ath1 actúa como un AP y tiene la dirección IP 192.168.1.2. La interfaz ath0, tiene la dirección IP 192.168.0.2 actúa como estación cliente y se conecta al AP del ALIX_1.

9.2. Parámetros de configuración en común para las soluciones

Los parámetros comunes que se usan en el desarrollo del piloto del laboratorio de las dos soluciones son los siguientes:

9.2.1. Tasa de transmisión y paquetes a transmitir

Para realizar las pruebas nos basamos a la siguiente tabla 9.1, pero se adiciona un 5% a la cantidad de paquetes a transmitir, de esta manera se encuentra el caudal máximo.

9.2.2. Estandar y canales de radiofrecuencia a utilizar

Se utiliza el estándar 802.11a y los canal de radiofrecuencia que utilizan las dos interfaces son: 153 y 161. Para llegar a usar el estándar y los canales mencionados hubo que superar una serie de problemas, que no se detectaron inicialmente. Al inicio del diseño del banco de pruebas, se partió usando el estándar 802.11g y los canales de radiofrecuencia que se utilizaron fue siguiendo la recomendación simple de salvaguardar 5 canales de separación y estos canales eran el 6 y el 11. Se realizaron pruebas de tráfico sobre ambos enlaces de manera simultánea y se encontró un deterioro en el tráfico, debido a interferencias entre ambos, que se verificó con el analizador de espectro. Por ello se decidió usar un salvaguarda de 7 canales y con esto se consiguió enviar datos por los dos canales como se puede ver en la figura 9.3 que es una imagen tomada del dispositivo WiSpy para analizar el espectro en las bandas ISM. En esa imagen se puede ver que existe un ruido e interferencias considerables en la banda de trabajo, por lo cual se opto por trabajar en la banda 5G (canales 153 y 161).

9.2.3. Duración de la prueba

Cada flujo que se envía, tiene una duración de 60 segundos. Este tiempo es adecuado en las pruebas porque se encuentra una estabilidad de la red en ese tiempo.

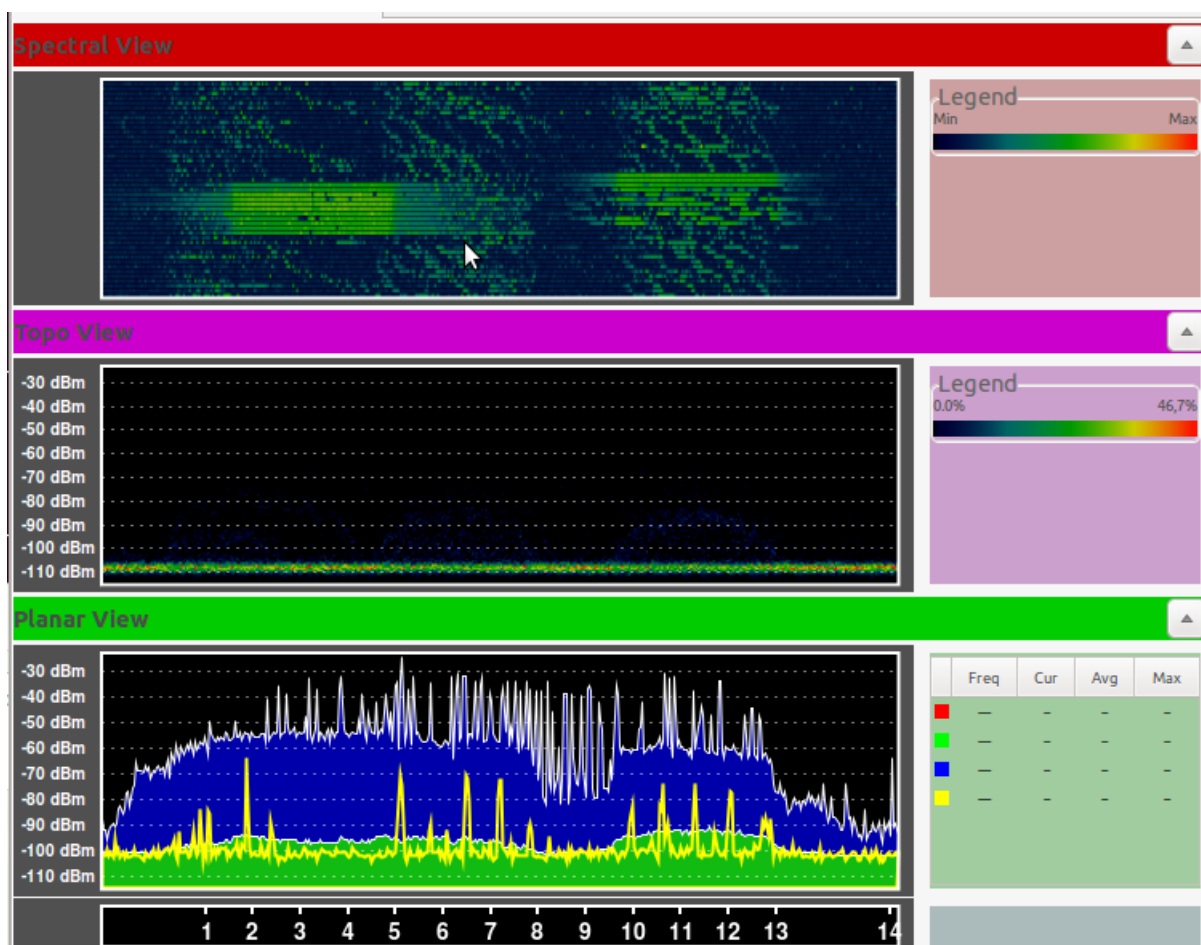


Figura 9.3.: Análisis espectral de dos interfaces que transmiten simultáneamente

9.2.4. Cantidad de pruebas a realizar

El número de repeticiones es opcional, pero en este PFM se realizó 7 repeticiones, de las cuales se saca una estadísticas por cada tasa de transmisión (modulación) y de esta manera podemos medir el comportamiento tráfico en la red. Las repeticiones se espaciaron en el tiempo, de forma que cualquier problema puntual que pudiera afectar a una de las mediciones, se distribuyera de tal forma que se redujera el impacto negativo sobre las estadísticas finales.

9.2.5. Escenario para medir las pruebas del desarrollo del piloto del laboratorio

Se presentan dos escenarios: primer escenario, enviando los paquetes en un solo flujo y el segundo escenario, enviando los paquetes en 2 flujos. Esto para cada una de las soluciones de balanceo de carga.

9.3. Script para preparar el escenario del piloto del laboratorio

Este es el primer script, que tiene que ser ejecutado para preparar el escenario del piloto del laboratorio. Este script se encuentra en el equipo Emisor. Dicho script copia todos los archivos necesarios dentro de las estaciones WiFi y los equipos de computo, para que funcione en el escenario que se usa. Este script se llama “2_prepar_placas.sh” que se puede ver en el anexo B.

9.4. Script Automatización de las pruebas de tráfico para el piloto del laboratorio

Para las pruebas de tráfico se utilizó el software D-ITG, además de tomar como criterio pasar por cada una de las tasas de transmisión y posteriormente empezar con la siguiente iteración, así sucesivamente hasta cumplir con la cantidad de pruebas que se quiere realizar. Esto nos ayudara en:

- 1.- Si una de las tasas falla en todo el proceso de la prueba. Esto significa que hay un problema, y se tiene que revisar que esta ocurriendo.
- 2.- Si una tasa falla en alguna iteración de la prueba. Esta prueba nos puede servir, con los datos que se consiguieron en las otras interacciones donde no existió problemas.

El siguiente diagrama de flujo 9.4 representa los procesos para la automatización de las pruebas de inyección de tráfico, y también se puede ver el script en el anexo B.

- **Diagrama de flujo de la automatización de las pruebas de tráfico.** Como se puede observar en la figura 9.4

Los pasos que se siguieron para esta automatización son las siguientes:

1. El script se encuentra en /pruebas/Principal.sh, que recibe 4 parámetros, por ejemplo ./principal.sh <modo><tipo><nro repeticiones><nombre_prueba>. Donde modo puede ser: bonding o iptables, tipo: flujo o multiflujo, número de repeticiones: la cantidad de pruebas que se quiere realizar y nombre_prueba, es el nombre que se va a guardar la prueba.
2. Realiza la configuración automática del modo elegido, dentro del equipo ALIX.
3. Se declara todas las tasas de bit en las que se va a trabajar.
4. Desde $i = 1$ hasta Número de repeticiones.
5. Desde la primera tasa hasta la última tasa declarada.
6. Prepara la cantidad de paquetes que va a ser transmitido.
7. Si es un solo flujo, pasa al paso 8 sino a 11
8. Paquetes totales a transmitir será el doble de paquete calculado en el paso 6.
9. Enviar los paquetes totales que se transmite.
10. La tasa de bit se modifica al siguiente valor... así hasta llegar al último valor.

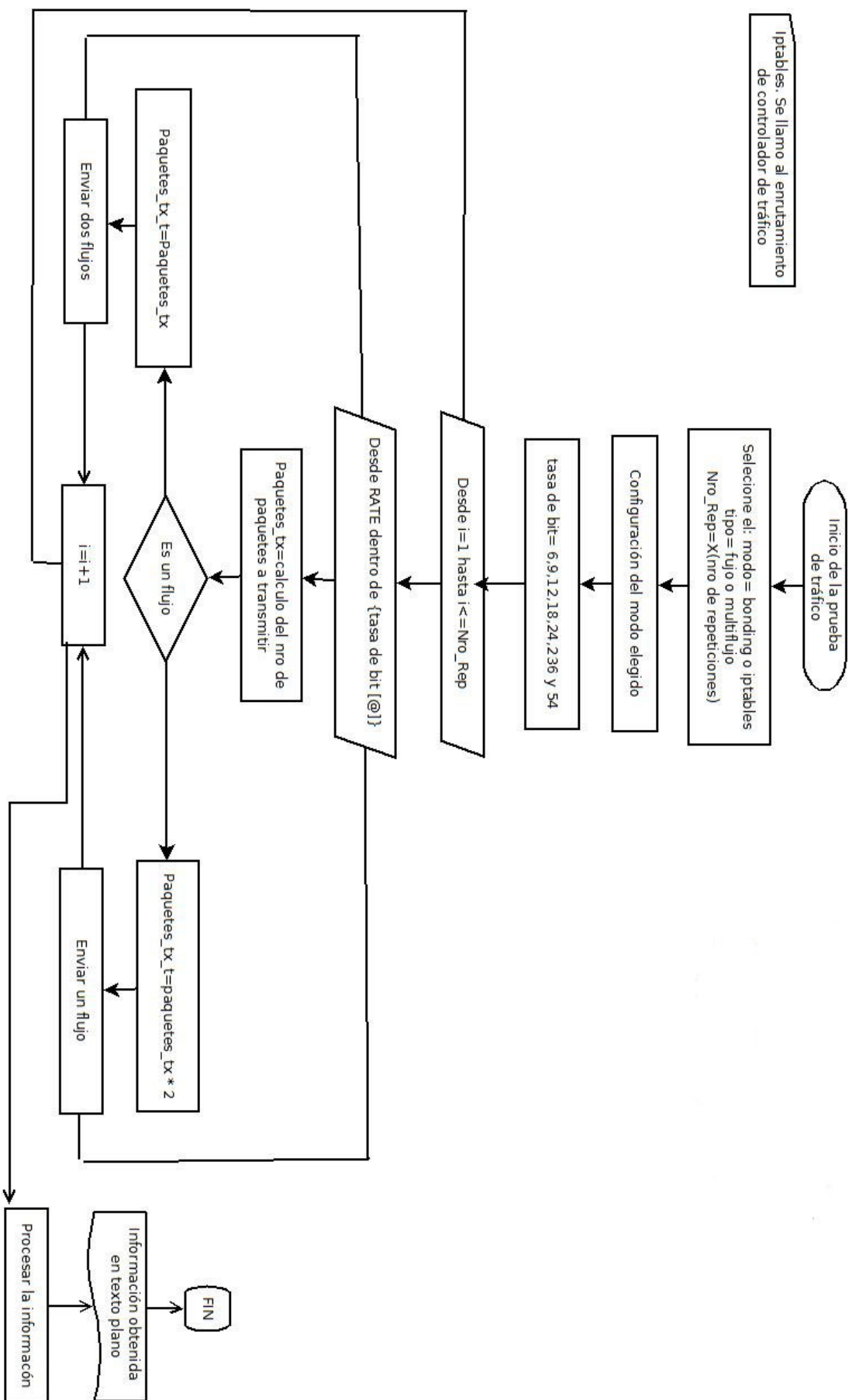


Figura 9.4.: Diagrama de flujo de la automatización de las pruebas de tráfico

9.5. Criterio para las mediciones del trabajo

11. El valor de i incrementa en uno, esto hasta terminar.
12. Procesa toda la información obtenida de las pruebas.
13. Recopila los datos que necesitamos en archivos de texto plano.
14. Solo si es de tipo multiflujo, El paquete total a transmitir es el mismo que se obtuvo en el paso 6.
15. Solo si es tipo multiflujo, Se envía los paquetes en dos flujos. Se pasa al paso 10.
16. Fin de todo el proceso.
17. procesamiento de resultados.
18. Fin de todo el proceso.

Este script nos devuelve cuatro archivos de texto plano, que tienen la siguiente información:

- a) Archivo de los paquetes transmitidos por las dos interfaces de las ALIX.
- b) Archivo de los paquetes recibidos por las dos interfaces de las ALIX.
- c) Archivo de los resultados de paquetes transmitidos por el equipo_1.
- d) Archivo de los resultados de los paquetes transmitidos por el equipo_2.

9.5. Criterio para las mediciones del trabajo

El criterio a seguir para la mediciones del trabajo, es guardar y respetar la misma polarización de la antenas, en todas las pruebas que se van a realizar. Estas mediciones son las siguientes:

1. Respetar la polarización vertical.
2. Asegurar un mismo plano de tierra para la instalación de 4 antenas.

El hecho de no cambiar la polarización, ni el plano de tierra, ni las distancias de separación entre las antenas, asegura un entorno de propagación constante para cada una de las pruebas.

9.6. Criterio para medir QoS de la técnica del balanceo escogido

Para medir la QoS de los resultados obtenidos de la técnicas de balanceo de carga en este PFM, se tomarán en cuenta los siguientes parámetros:

Retardo; es tiempo que le toma a un paquete en ir desde un punto a otro de la red.

Variación del retardo; es la variación que sufren los diferentes paquetes en sus retardos, al atravesar la red, de un punto a otro.

Caudal; es la tasa máxima que soporta la red para transportar los paquetes de un punto a otro.

Pérdida de paquetes; cantidad de paquetes que se han perdido, al atravesar el medio, de un punto a otro.

10. Solución de WiFi-Bonding

En este capítulo se desarrolla la solución de WiFi-Bonding, donde vemos el escenario en el que se trabaja, los parámetros que se usan para la configuración, las fases que se paso para encontrar la solución bonding, pruebas de los modos de bonding en redes inalámbricas, script de autoconfiguración de bonding y análisis del comportamiento de las dos interfaces que tiene una estación WiFi.

10.1. Desarrollo del piloto del laboratorio del bonding

Para el desarrollo del piloto del laboratorio del bonding se parte del escenario inicial, que se vio en el diseño de red, pero en este escenario se añade un interfaz lógico que maneja a las interfaces reales, como se puede ver en la figura 10.1, que representa la configuración gráfica del escenario de bonding.

10.1.1. Parámetros de configuración del piloto del laboratorio de la solución bonding

Configuración de los ordenadores. Estos equipos mantienen la configuración inicial que se presento en el diseño de la red.

Configuración de interfaces de las placas ALIX. Las interfaces de ALIX_1 y ALIX_2 mantienen casi toda la configuración inicial que se presento en el diseño de red. Sólo que en esté se crea un interfaz logico llamado “bond0”, este interfaz se configuro en la subred 192.168.3.0./24. La interfaz bond0 tendrá como esclavos interfaces reales que tienen las estaciones WiFi.

Configuración del interfaz lógico “bond0”. La configuración y los parámetros que se tomo en cuenta para la interface bond0 de las dos placas ALIX es la siguiente: `miimon=100`, significa que se monitoriza cada 100 ms si las tarjetas reales funcionan; `max_bonds=1`, es la cantidad de las interfaces virtuales; `mode=1`, el modo de trabajo de bonding será el de round robin.

10.1.2. Configuración de bonding

Para realizar la configuración de bonding dentro de los equipos ALIX, se siguen los siguientes pasos:

1. Ejecutar `“ifconfig bond0 192.168.3.1/24 up”`, aquí se crea y se configura la interface virtual bond0.
2. Ejecutar `“modprobe bonding miimon=100 max_bonds=1 mode=1”`, estos son los parámetros de configuración que tiene la interfaz de bonding.
3. Ejecutar `“ifenslave bond0 ath0 ath1”`, las interfaces ath0 y ath1 quedan como interfaces esclavas y en dominio de la interfaz bond0.

10.2. Fases de pruebas para llegar a la solución final de bonding

Para llegar a la solución final del balanceo de carga utilizando la técnica bonding, se realizaron una serie de pruebas hasta que el ancho de banda sea mayor a la de un enlace.

10. Solución de WiFi-Bonding

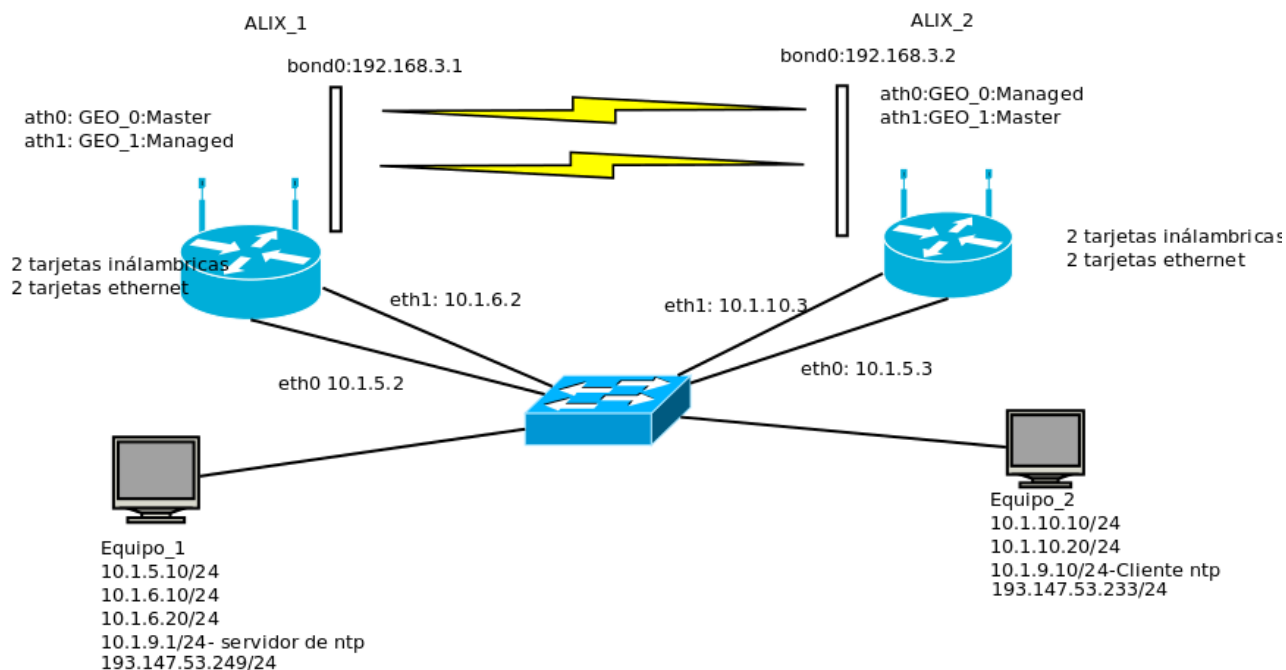


Figura 10.1.: Esquema de configuración de la técnica bonding

10.2.1. Primera prueba de bonding

La primera prueba se realizó con el SO GNU LINUX Voyage 0.7.0, donde se instaló el bonding y los resultados de la configuración inicial de bonding, se pueden observar en el archivo `/proc/net/bonding/bond0` y su contenido es el siguiente:

```
root@voyage:/home# cat /proc/net/bonding/bond0 Bonding Mode: load balancing (round-robin)
; MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0
```

```
Slave Interface: ath0
MII Status: up
Link Failure Count: 0
Permanent HW addr: 06:15:6d:55:5c:b8
```

```
Slave Interface: ath1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 06:15:6d:55:5c:b9
```

En este archivo nos muestra que el modo de trabajo de bonding es round-robin, la dirección MAC de las interfaces y se tiene habilitada la monitorización de las interfaces reales. Aparentemente la configuración de bonding es correcta pero los resultados no fueron satisfactorios. Para las pruebas de comunicación entre los enlaces se utilizó el comando “ping”, donde se observa los paquetes perdidos son aproximadamente del 80%. Esto se debe a que bonding trabaja en redes Ethernet, y estas redes suelen transmitir a 10 y 100 Mbps. Los resultados se pueden apreciar en el siguiente párrafo.

```
root@voyage:/home# ping 192.168.3.2
PING 192.168.3.2 (192.168.3.2) 56(84) bytes of data.
64 bytes from 192.168.3.2: icmp_req=4 ttl=64 time=2.11 ms
```

10.2. Fases de pruebas para llegar a la solución final de bonding

```
64 bytes from 192.168.3.2: icmp_req=8 ttl=64 time=11.6 ms
64 bytes from 192.168.3.2: icmp_req=14 ttl=64 time=2.05 ms
64 bytes from 192.168.3.2: icmp_req=18 ttl=64 time=1.72 ms
— 192.168.3.2 ping statistics —
50 packets transmitted, 11 received, 78 % packet loss, time 49014ms
```

Como el resultado de esta fase no fue satisfactorio se continuó en la búsqueda de una nueva solución.

10.2.2. Segunda prueba de bonding

Aquí se intento realizar una ingeniería inversa del proyecto NET-X, que se vio en el capítulo 4. NET-X realiza un balanceo de carga a nivel de enlace. Este proyecto esta implementado en Linux kernel 2.4.26, usan placas Soekris y las radios con chipset Atheros. Pero los equipos de cómputo que se usan en este PFM utilizan el kernel 2.6.35 y la más antigua distribución del SO Voyage de Linux, usa el kernel 2.6.15.

Para realizar esta ingeniería inversa, fueron necesarios los siguientes pasos para la compilación del kernel 2.4.26:

1. Descargar el código fuente del kernel 2.4.26 y también instalamos los paquetes fakeroot, kernel-package, binutils y gcc en el equipo_1
2. Se necesitan un compilador gcc-2.95 g++-2.95, pero fue necesario buscar diferentes mecanismos para lograr instalar esté compilador en el equipo de cómputo, ya que el computador tiene instalado el gcc-4.5 por defecto, por eso se instaló varias maquinas virtuales con diferentes versiones de SO GNU/LINUX como Ubuntu 8.04 LTS (Hardy Heron) que usa el kernel 2.6.24. En esta versión de SO se pudo instalar fácilmente el compilador gcc-2.95, también se logro instalar el compilador en Ubuntu 10.04, pero para su instalación tuvo que pasar por varios errores.
3. Una vez instalado el compilador en el gcc-2.95, se procede a preparar los paquetes de compilación del kernel 2.4.26, a pesar de los múltiples errores que aparecieron en la compilación, al final se logró crear los paquetes de instalación del kernel 2.4.26. Para esta preparación de paquetes se paso por varios errores que se detallan:
 - Preparando los paquetes de instalación del kernel 2.4.26 en el SO GNU/LINUX Ubuntu 8.04.

```
root@pepe-desktop:/usr/src/linux-2.4.26# fakeroot make-kpkg --initrd --revision=2 --
append-to-version 386-voyage"kernel-image kernel-headers
/usr/bin/make CFLAGS=D_KERNEL_ -I/usr/src/linux-2.4.26/include -Wall -Wstrict-
prototypes -Wno-trigraphs -O2 -fno-strict-aliasing -fno-common -fomit-frame-pointer
-pipe -mpreferred-stack-boundary=2 -march=i486 C arch/i386/kernel
make[2]: se ingresa al directorio '/usr/src/linux-2.4.26/arch/i386/kernel'
gcc -D_KERNEL_ -I/usr/src/linux-2.4.26/include -Wall -Wstrict-prototypes -Wno-
trigraphs -O2 -fno-strict-aliasing -fno-common -fomit-frame-pointer -pipe -mpreferred-
stack-boundary=2 -march=i486 -nostdinc -iwithprefix include -DKBUILD_BASENAME=process
-c -o process.o process.c
{entrada estándar}: Mensajes del ensamblador:
{entrada estándar}:748: Error: sufijo u operandos inválidos para 'mov'
{entrada estándar}:749: Error: sufijo u operandos inválidos para 'mov'
{entrada estándar}:895: Error: sufijo u operandos inválidos para 'mov'
{entrada estándar}:897: Error: sufijo u operandos inválidos para 'mov'
{entrada estándar}:909: Error: sufijo u operandos inválidos para 'mov'
make[2]: *** [process.o] Error 1
make[2]: se sale del directorio '/usr/src/linux-2.4.26/arch/i386/kernel'
make[1]: *** [_dir_arch/i386/kernel] Error 2
make[1]: se sale del directorio '/usr/src/linux-2.4.26'
make: *** [stamp-build] Error 2
```

10. Solución de WiFi-Bonding

- Buscando este error en buscador google, encontramos en la pagina <http://kerneltrap.org/node/5785> una solución, donde se aplica un parche de la siguiente manera:

```
root:/usr/src/linux#  
wget www.kernel.org/pub/linux/devel/binutils/linux-2.6-seg-5.patch  
root:/usr/src/linux# patch -Np1 -i linux-2.6-seg-5.patch
```
 - Después de aplicar el parche se vuelve a preparar los paquetes:

```
root@pepe-desktop:/usr/src/linux-2.4.26# fakeroot make-kpkg --initrd --revision=2 --append-to-version 386-voyage"kernel-image kernel-headers  
if [ -r System.map ]; then /sbin/depmod -ae -F System.map -b /usr/src/linux-2.4.26/debian/tmp-image -r 2.4.26-486-voyage; fi  
Version requires old depmod, but couldn't run /sbin/depmod.modutils: No such file or directory  
make[2]: *** [_modinst_post] Error 2  
make[2]: se sale del directorio '/usr/src/linux-2.4.26'  
make[1]: *** [real_stamp_image] Error 2  
make[1]: se sale del directorio '/usr/src/linux-2.4.26'  
make: *** [kernel-image-deb] Error 2
```
 - Para subsanar este error, se creo un enlace simbólico, de la siguiente manera:

```
sudo ln -s /sbin/depmod /sbin/depmod.modutils
```
4. Teniendo los paquetes del kernel 2.4.26, se procedió a instalar en una distribución que tenía el kernel 2.6.32. Pero esto no fue satisfactorio, ya que existe una cantidad de errores, como por ejemplo:
- Instalando los paquetes creados dentro las ALIX, el sistema de arranque que utilizan es LILO.

```
root@voyage:/home/linux-2.4.26# dpkg -i kernel-image-2.4.26-486-voyage_2-i386.deb kernel-headers-2.4.26-486-voyage_2-i386.deb  
Report bugs to <bug-grub@gnu.org>.  
run-parts: /etc/kernel/posrm.d/zz-update-grub exited with return code 1  
Failed to process /etc/kernel/posrm.d at /var/lib/dpkg/tmp.ci/posrm line 273.  
dpkg: error while cleaning up:  
subprocess new post-removal script returned error exit status 2  
tar: ./postinst: time stamp 2011-06-21 10:03:03 is 361963841.698826828 s in the future  
tar: ./control: time stamp 2011-06-21 10:03:47 is 361963885.6948442 s in the future  
tar: .: time stamp 2011-06-21 10:03:47 is 361963885.692566257 s in the future  
Selecting previously deselected package kernel-headers-2.4.26-486-voyage.  
Unpacking kernel-headers-2.4.26-486-voyage (from kernel-headers-2.4.26-486-voyage_2-i386.deb)  
...  
Setting up kernel-headers-2.4.26-486-voyage (2) ...  
dpkg: warning: obsolete option '--print-installation-architecture', please use '--print-architecture' instead.  
Errors were encountered while processing:  
kernel-image-2.4.26-486-voyage_2-i386.deb
```
 - Para corregir este error se instaló el module-int-tooll 3.2.2.

```
root@voyage:~# apt-get install module-int-tooll 3.2.2.  
root@voyage:/home/linux-2.4.26# dpkg -i kernel-image-2.4.26-486-voyage_2-i386.deb kernel-headers-2.4.26-486-voyage_2-i386.deb  
run-parts: /etc/kernel/posrm.d/zz-update-grub exited with return code 1  
Failed to process /etc/kernel/posrm.d at /var/lib/dpkg/tmp.ci/posrm line 273.  
dpkg: error while cleaning up:  
subprocess new post-removal script returned error exit status 2  
Errors were encountered while processing:  
kernel-image-2.4.26-486-voyage_2-i386.deb
```



```

root@voyage:~# ethtool eth0
Settings for eth0:
    Supported ports: [ TP MII ]
    Supported link modes:   10baseT/Half 10baseT/Full
                           100baseT/Half 100baseT/Full
    Supports auto-negotiation: Yes
    Advertised link modes:  Not reported
    Advertised pause frame use: No
    Advertised auto-negotiation: No
    Speed: 10Mb/s
    Duplex: Full
    Port: MII
    PHYAD: 1
    Transceiver: internal
    Auto-negotiation: off
    Supports Wake-on: pumbg
    Wake-on: d
    Current message level: 0x00000001 (1)
    Link detected: yes
root@voyage:~# █

```

Figura 10.2.: Tasa de transmisión de interface ethernet: 10Mbps

Se logro instalar el kernel 2.4.26, pero después de reiniciar el equipo, no se pudo arrancar el sistema.

Al final, se llevo a la conclusión de que no se debía proseguir indagando en esta solución. Así que se decidió seguir analizando otra solución para bonding.

10.2.3. Tercera prueba de bonding

Aquí se cambio de versión de SO a GNU/Linux voyage 0.7.5. Esta versión contiene el controlador ath5k¹ y este tiene incorporado un módulo para usar bonding en redes inalámbricas. En esta prueba bonding funciona correctamente ya que no existen los pérdida de paquetes al utilizar el comando “ping”, como se vio en la primera prueba de bonding. Pero cuando se inyecta tráfico, sólo se transmite el 50 % de los paquetes. Esta prueba se realizó con la herramienta llamada Iperf.

10.2.4. Cuarta prueba de bonding en redes ethernet

Esta prueba se realizó en redes ethernet, con el propósito de probar el funcionamiento de la técnica del bonding en SO GNU/LINUX Voyage 0.7.5. Para realizar esta prueba, se tuvo que limitar la tasa de transmisión a 10 Mbps de los dos puertos ethernet (ver figura: 10.2) usando el comando:

```

root@voyage: # ethtool -s eth0 speed 10duplex full
root@voyage: # ethtool -s eth0 speed 10 duplex half
root@voyage: # ethtool -s eth1 speed 10duplex full
root@voyage: # ethtool -s eth1 speed 10 duplex half

```

Se configura bonding en modo round robin, una vez que se tiene la configuración de bonding se empezó con las pruebas de tráfico con la herramienta Iperf, el cual muestra el siguiente resultado:

```

geovana@geovana: #iperf -s -u -S0xb8 -i1 -l1472 -fk [ 4] 0.0-60.0 sec 131873 KBytes 18006
Kbits/sec 0.131 ms 6/91744 (0.0065 %)
[ 4] 0.0-60.0 sec 4 datagrams received out-of-order
[ 3] Server Report:
[ 3] 0.0-60.0 sec 131882 KBytes 18006 Kbits/sec 0.099 ms 2/91743 (0.0022 %)

```

¹Ath5k: Es un controlador Linux completamente de código abierto para tarjetas inalámbricas con chipset Atheros. Está basado en MadWiFi y en OpenHAL.

10. Solución de WiFi-Bonding

[3] 0.0-60.0 sec 3 datagrams received out-of-order.

El ancho de banda obtenido en esta prueba es aproximadamente de 18 Mbps. Estos resultados muestran que el funcionamiento de bonding es correcto.

10.2.5. Quinta prueba de bonding

En esta sección se trabajó en el SO GNU/LINUX Voyage 0.6.5, ya que las pruebas del balanceo de carga a nivel de red funcionó en esa versión de Linux, además que la fundación EHAS ha ido trabajando en esta versión de SO.

Se buscó más información sobre el balanceo de carga con bonding y se encontró con la solución de la distribución Zeroshell, que indica que se tiene que unificar la MAC de las interfaces antes de que quede en dominio del controlador bonding. La unificación de la MAC se resuelve el problema, pero existe un truco, que se detalla en los siguientes pasos:

- 1.- Obtener la MAC de la interfaz que trabaja en modo managed.
- 2.- La MAC del modo master tendrá el valor de la MAC del modo managed.

Esta prueba se utilizó una tasa de transmisión de 6 Mbps por enlace y para la inyección de tráfico la herramienta Iperf. En el siguiente párrafo se muestra los resultados de unificar la MAC.

```
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 110 KByte (default)
```

```
[ 3] local 192.168.3.2 port 5001 connected with 192.168.3.1 port 53113
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
[ 3] 0.0- 1.0 sec 1.29 MBytes 10.8 Mb/s/sec 11.396 ms 6/ 924 (0.65 %)
[ 3] 0.0- 1.0 sec 432 datagrams received out-of-order
[ 3] 1.0- 2.0 sec 1.30 MBytes 10.9 Mb/s/sec 33.135 ms 6/ 932 (0.64 %)
[ 3] 1.0- 2.0 sec 460 datagrams received out-of-order
[ 3] 2.0- 3.0 sec 1.30 MBytes 10.9 Mb/s/sec 50.237 ms 6/ 930 (0.65 %)
[ 3] 2.0- 3.0 sec 459 datagrams received out-of-order
[ 3] 3.0- 4.0 sec 1.29 MBytes 10.8 Mb/s/sec 47.292 ms 4/ 922 (0.43 %)
[ 3] 3.0- 4.0 sec 458 datagrams received out-of-order
[ 3] 0.0- 4.5 sec 5.76 MBytes 10.8 Mb/s/sec 12.272 ms 2033/ 4112 (49 %)
[ 3] 0.0- 4.5 sec 2033 datagrams received out-of-order
```

La prueba es satisfactoria, porque se transmitió aproximadamente 11 Mbps. Pero sigue existiendo un problema al usar el comando “ping”, que sigue registrando pérdidas de paquetes, esto se debe a que el tiempo del ACK en redes ethernet es menor que en redes inalámbricas. Por esta razón se suprimió el ACK en 802.11. con la ayuda del MadWiFi y se utilizó la siguiente configuración en todas las interfaces que están en dominio del bonding dentro del archivo /home/interface-bond.

```
up iwpriv ath0 noackpolicy 0 0 1
up iwpriv ath0 noackpolicy 1 0 1
up iwpriv ath0 noackpolicy 2 0 1
up iwpriv ath0 noackpolicy 3 0 1
```

En esta política de la eliminación de ACK el primer valor representa al valor del clase de acceso (en inglés Access class (AC)) que se puede ver en la tabla 10.1, el segundo valor siempre es 0 y el tercer valor es el Noack deseado (0 o 1).

Finalmente se ha conseguido usar la técnica bonding con el SO GNU/LINUX Voyage 0.6.5 en este PFM.

AC	Number	Descripción de la clase de acceso
0	BE	Best Effort
1	BK	Background
2	VI	Video
3	VO	Voice

Cuadro 10.1.: Access class (AC) Values

10.3. Automatización de configuración de la solución bonding

Aquí realizamos el script de la automatización de configuración de la solución de bonding. El siguiente diagrama de flujo (ver figura 10.3) representan los procesos de configuración de la técnica de bonding que se realizó dentro de los equipos ALIX:

Para configurar del piloto del laboratorio de bonding se encuentra dentro de las ALIX y sigue los siguientes pasos:

1. Se ejecuta el script /home/comun, que recibe parámetro (es el modo en el que va a trabajar el Equipo ALIX, en este caso será bonding).
2. Copia el archivo /home/bond a /etc/network/interface. El archivo bond tiene la configuración de las interfaces inalámbricas, además de la eliminación de los ACK.
3. Reiniciar el servicio de red
4. Si el modo de trabajo una interface es master y de la otra managed, pasa al siguiente punto. Sino se vuelve al paso 2.
5. Se busca la MAC de la interfaz que trabaja en modo managed.
6. Se cambia la MAC de la interfaz que trabaja en modo master por la MAC de managed.
7. Se elige el modo (estos modos se les puede ver en la sección: 3.5) en el que va a ser aplicado bonding.
8. Configuración del el equipo en modo bonding, esta es una función llamada configuración_bonding() y se encuentra en /home/comun.
9. Mensaje de la finalización de la configuración. Este script se lo puede ver en el anexo A.

10.4. Ejecución de las pruebas de la solución bonding

Después de tener los script de autoconfiguración, se empezó a realizar pruebas de tráfico ejecutando dentro del equipo_1 el archivo Principal.sh. Este archivo tiene la configuración de las pruebas de tráfico y maneja las configuraciones de las dos soluciones de la investigación. Este archivo recibe 4 parámetros: 1) nombre de la solución, 2) tipo de prueba, 3) número de repeticiones de la prueba y 4) nombre de la carpeta donde se va a almacenar todos los resultados. Para realizar las pruebas de la solución de bonding se lo se ejecuta de la siguiente manera:

```
geovana@geovana: /Pruebas$./Principal.sh bonding multifujo 7 BOND2F_2
geovana@geovana: /Pruebas$./Principal.sh bonding flujo 7 BOND1F_2
```

Después de terminar la ejecución de prueba se obtiene los resultados, luego estos resultados se lo procesa en hojas de cálculo y con ese proceso se empieza a ver los resultados finales de cada prueba. En este apartado se va a analizar el comportamiento de las interfaces cuando se realiza balanceo de carga en dos escenarios diferentes.

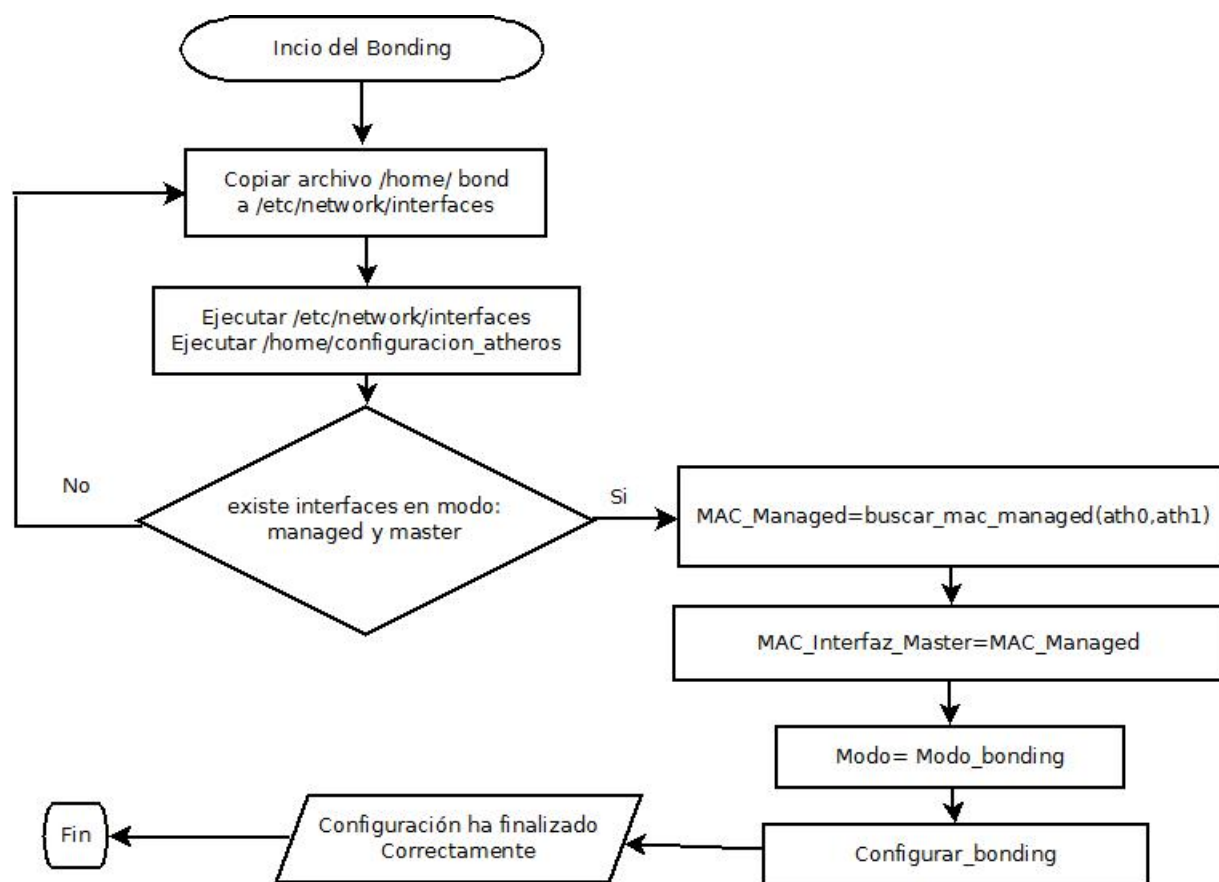


Figura 10.3.: Diagrama de flujo de bonding

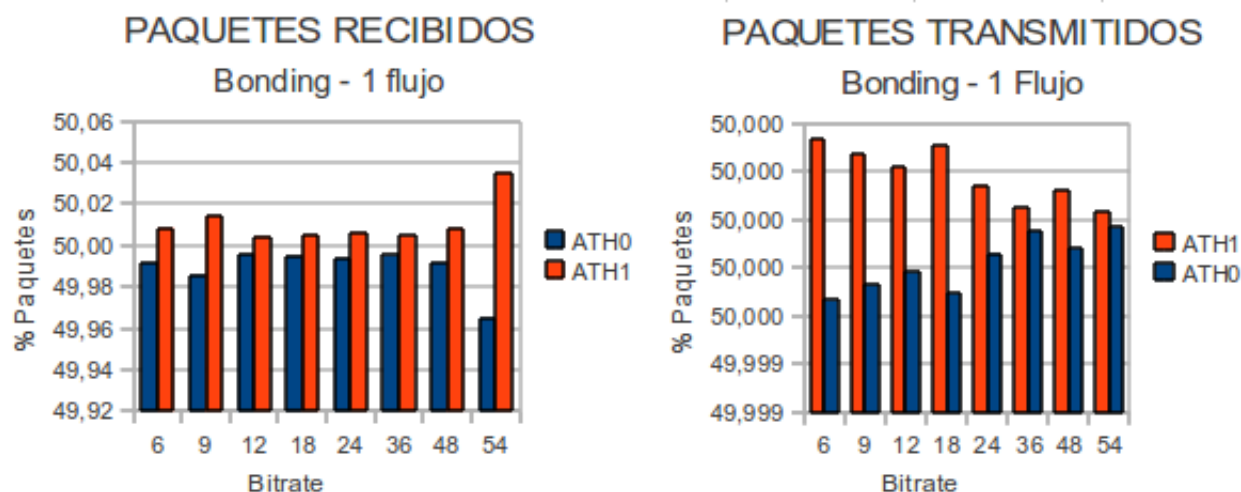


Figura 10.4.: Bonding: Paquetes de TX y RX en un flujo

10.4.1. Primer escenario, enviando los paquetes en un sólo flujo

En esta prueba los paquetes son enviados desde un origen hasta un único destino.

- Paquetes transmitidos y recibidos por las interfaces.** En esta prueba se usaron dos equipos ALIX para la transmisión y/o recepción de paquetes. En la figura 10.4 muestra que los paquetes transmitidos (TX) son del casi del 50 % para las interfaces ath0 y ath1. Los paquetes recibidos (RX) en el nodo son casi del 50 % para las interfaces ath0 y ath1, eso quiere decir que los paquetes se reparten simétricamente entre las interfaces de cada nodo. El resultado de esta prueba más detalladamente se puede ver en el anexo E.

10.4.2. Segundo escenario, enviando los paquetes en dos flujos

El envío de paquetes se realiza desde el origen hacia dos destinos, simultáneamente.

- Paquetes transmitidos y recibidos por las interfaces.** En esta prueba se transmite dos flujos de paquetes entre los dos nodos WiFi que se encuentran interconectados inalámbricamente. En la figura 10.5 se muestra que los paquetes transmitidos (TX) por las interfaces ath0 y ath1 son del 50 % y los paquetes recibidos (RX) por un nodo son del 50 % para cada interfaz, eso quiere decir que tanto la transmisión y recepción de los paquetes en las interfaces los nodos es simétrico. El resultado de esta prueba se puede ver más detalladamente en el anexo E.

En conclusión, podemos decir que bonding configurado en modo round robin funciona correctamente en el balanceo de carga entre las interfaces, tanto si envía en un sólo flujo o 2 flujos.

10.5. Pruebas de bonding en los diferentes modos

En la sección 3.5 del capítulo 3 se vio todos los modos en los que puede trabajar bonding en la red ethernet. Pero este PFM se basa en redes inalámbricas, por lo que se vio necesario probar los modos de bonding en este tipo de redes pero con las modificaciones que han sido necesarias para hacer funcionar esta solución.

10.5.1. active-backup o 1

Se intentó utilizar este modo pero no funcionó, esto fue debido a que, para hacer bonding se les asignó a cada interfaz una misma MAC, de esta manera trabajan como una sola interfaz pero no pueden trabajar en modo active backup, porque el funcionamiento de este modo está vinculado al uso de dos macs.

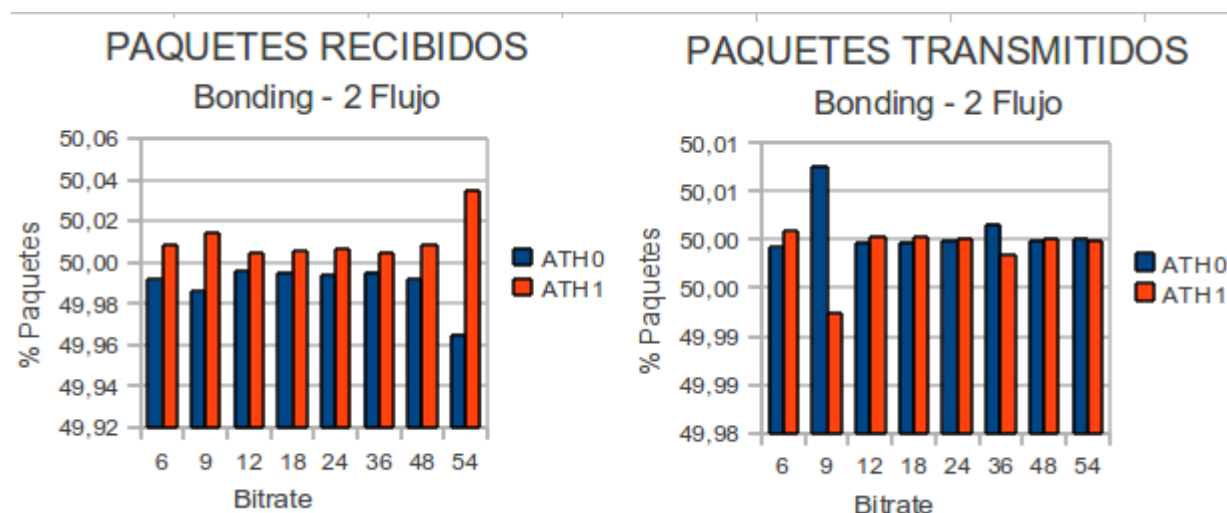


Figura 10.5.: Bonding: Paquetes de TX y RX en 2 flujos

10.5.2. balance-xor o 2

Este modo no funciona y la configuración que se utilizó fue:

1. Ejecutar `"ifconfig bond0 192.168.3.X/24 up"`
2. Ejecutar `"modprobe bonding miimon=100 max_bonds=1 mode=2"`
3. Ejecutar `"ifenslave bond0 ath0 ath1"`

Pero antes de ejecutar esa configuración se realizó la unificación de la MAC. Los paquetes fueron enviados por dos flujos (2 IP distintas) y los resultados obtenidos en esta prueba según los parámetros paquetes transmitidos, caudal y pérdida de paquetes son:

- Paquetes transmitidos. En la figura 10.6 vemos que todo el tráfico a pasado por la interface ath0.
- Caudal. En la figura 10.6 vemos que el caudal conseguido es la capacidad de un enlace.
- Pérdida de paquetes. En la figura 10.6 tenemos un 43 % a 57 % de perdida de paquetes.

Aunque se haya enviado a dos IP distintas y el nodo que recibe los paquetes tiene la misma dirección MAC en sus interfaces (ambas interfaces están en dominio de bonding) el nodo transmisor intentará enviar todos los paquetes por la misma interfaz como se ve en la figura 10.6, a consecuencia de este suceso el caudal que se consigue en esta prueba es la capacidad de un enlace y la perdida de paquetes tiene es aproximadamente del 50 %. Estos resultados pueden ser ampliados en el anexo C.

10.5.3. broadcast o 3

Este modo funciona correctamente y la configuración utilizada fue:

1. Ejecutar `"ifconfig bond0 192.168.3.X/24 up"`
2. Ejecutar `"modprobe bonding miimon=100 max_bonds=1 mode=3"`
3. Ejecutar `"ifenslave bond0 ath0 ath1"`

Los resultados de esta prueba se puede apreciar en el anexo D

10.5.4. 802.3ad o 4

En este modo no se realizaron pruebas, porque esté PFM trabaja con el estandard 802.11.

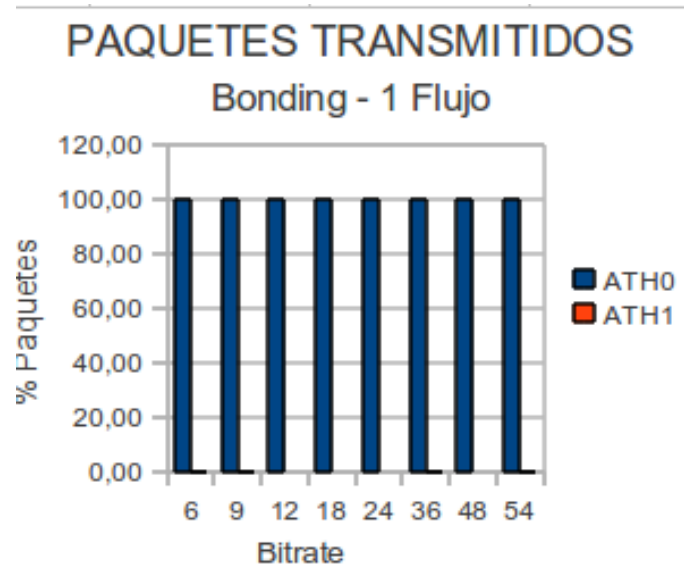


Figura 10.6.: Paquetes transmitidos en modo 2

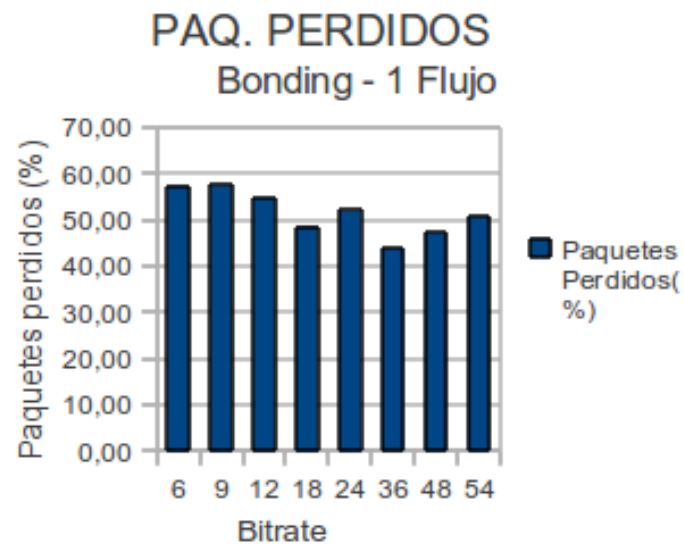


Figura 10.7.: Pérdida de paquetes en modo 2

10. Solución de WiFi-Bonding

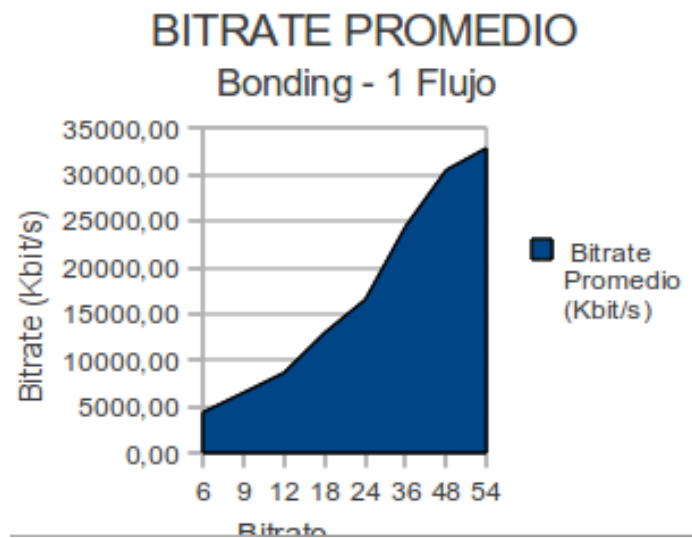


Figura 10.8.: Caudal obtenido en modo 2

10.5.5. `balance_tbi` o 5 y `balance_al` o 6

Tampoco se realizaron pruebas en el modo 5,6; porque es necesario que la tarjeta de red sea soportada por ethtool.

11. Solución de balanceo de carga a nivel de red

Este capítulo nos centramos a la solución de balanceo de carga a nivel de red, donde vamos a ver el escenario de pruebas, los parámetros de configuración, las fases que se paso para llegar a la solución final, la automatización de la solución, análisis del comportamiento de las interfaces. Pero antes de seguir hablando de este capítulo cabe recordar que este tipo de de balanceo tiene un caché de enrutamiento donde guarda todas las IP destino visitadas durante un determinado tiempo.

11.1. Desarrollo del piloto del laboratorio del balanceo de carga a nivel de red

Para el desarrollo del piloto del laboratorio del balanceo de carga a nivel red se parte del escenario inicial que se vio en el diseño de red (ver en la figura 11.1).

Las pruebas realizadas en la solución del balanceo de carga a nivel de red lo llamamos pruebas “iptables”.

11.1.1. Parametros de configuración del piloto del laboratorio de la solución del balanceo de carga a nivel de red

Configuración de los ordenadores. Estos equipos mantienen la configuración inicial que se presento en el diseño de la red.

Configuración de interfaces de las placas ALIX. Las interfaces de las estaciones WiFi mantienen la configuración inicial que se presento en el diseño de red.

11.1.2. Configuración del balanceo de carga a nivel red

Para realizar la configuración del balanceo de carga a nivel de red se ejecuta los siguientes pasos:

1. Ejecutar “ip route add default nexthop via $\{\text{ISP0}\}$ dev $\{\text{ATH0}\}$ weight 1 nexthop via $\{\text{ISP1}\}$ dev $\{\text{ATH1}\}$ weight 1.” El ISP0 y ISP1; son las IP de las interfaces del ALIX a quien se conectará. EL ATH0 y ATH1 son las IP de las interfaces locales. El weight 1; es el peso que tendrá cada una de las interfaces. En nuestro caso ambos tienen el mismo peso, eso quiere decir que el tráfico de datos se dividirá el 50% para cada interfaz.

11.2. Fases de pruebas para llegar a la solución final del balanceo de carga a nivel de red

Para llegar a la solución final del balanceo de carga a nivel de red, se realizaron una serie de pruebas hasta conseguir un caudal mayor que a la de un enlace.

11.2.1. Primera prueba del balanceo de carga en el nivel de red - Shorewall

En la primera prueba del balanceo de carga a nivel de red, se instaló el software llamado “Shorewall”¹ dentro de las dos estaciones WiFi que tenía el SO Voyage Linux 0.7.0. Este software es capaz de realizar el balanceo de carga entre dos ISP en redes inalámbricas (en nuestro caso

¹shorewall: un firewall reconocido en ambientes Linux

11. Solución de balanceo de carga a nivel de red

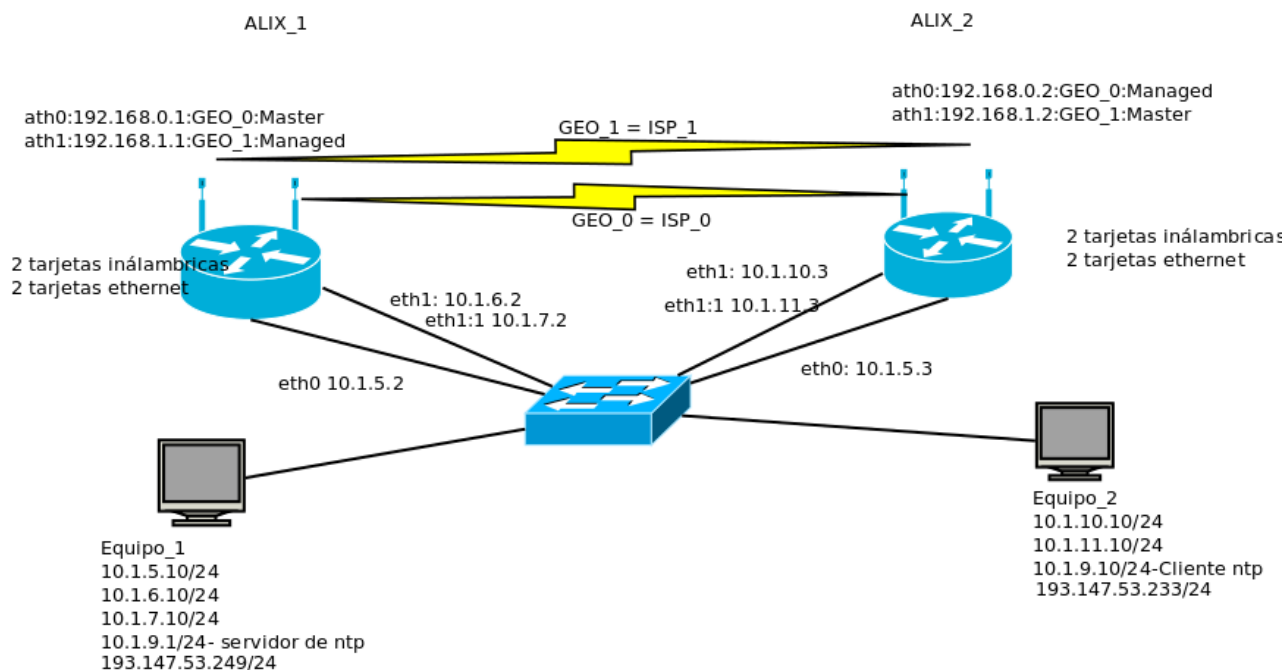


Figura 11.1.: Esquema de configuración del balanceo de carga a nivel de red

las ISP representan a las IP del estación WiFi a la que nos conectamos). Después de instalación se procedió a la configuración del balanceo de carga con dos ISP, esta configuración obtiene el balanceo de carga entre las dos interfaces, pero no se pudo duplicar el ancho de banda, el caudal que se consiguió fue la capacidad de un enlace.

11.2.2. Segunda prueba del balanceo de carga a nivel de red - configuración manual

En esta prueba se utilizó el mismo SO anteriormente mencionado, sólo que la configuración del balanceo de carga se realizó de manera manual, con la ayuda del enrutamiento avanzado (iproute) y control de tráfico en Linux (iptables). Pero el problema de duplicar el ancho de banda, como en el apartado anterior 11.2.1 se vuelve a repetir, pese a que se volvió a compilar el kernel, donde se habilitó la opción: IP advanced router -> IP: equal cost multipath. Los resultados después de la compilación fueron similares. Ya que el ancho de banda obtenido sólo fue la capacidad de un enlace.

11.2.3. Tercera prueba del balanceo de carga a nivel de red

Aquí se trabajó con el SO GNU/LINUX Voyage 0.7.5 y se realizó la configuración manual para el balanceo de carga, pero el resultado fue el mismo que en el anterior apartado.

11.2.4. Cuarta prueba del balanceo de carga a nivel de red

Aquí se usa el SO GNU/LINUX Voyage 0.6.5, donde se aumentó a 7 canales de salvaguardas (5 canales de salvaguarda es lo recomendado), con este cambio aumentó el ancho de banda que es mayor al ancho de banda de un enlace. Estas pruebas se realizaron en las bandas 2,4GHz y 5GHz. Aunque al final se optó trabajar en la banda 5GHz y usar los canales 53 y 61, porque existió un ruido considerable en el espectro.

Esta misma prueba se realizó en la versión del SO Voyage 0.7.5, pero el problema se siguió repitiendo en esta versión de SO.

11.3. Automatización de configuración de la solución del balanceo de carga a nivel de red

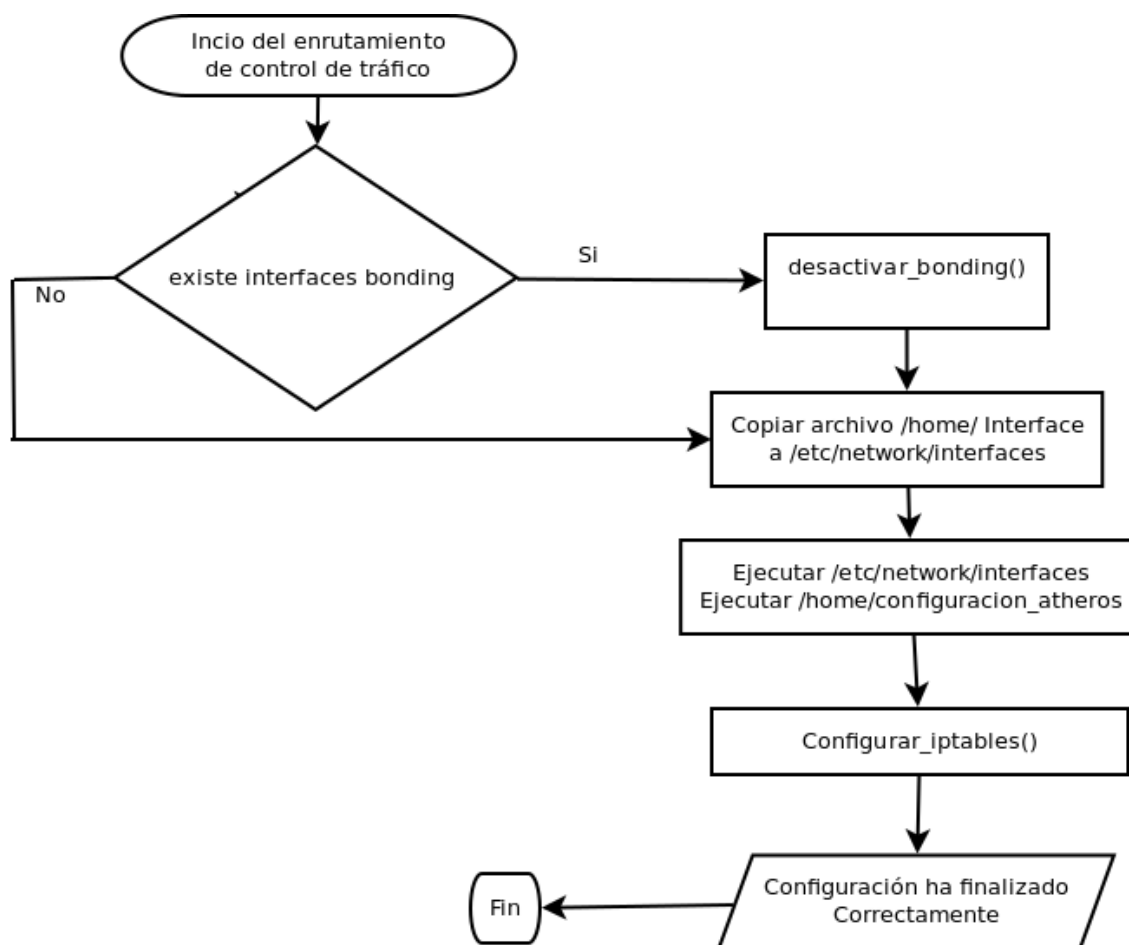


Figura 11.2.: Diagrama de flujo del balanceo de carga a nivel de red.

11.3. Automatización de configuración de la solución del balanceo de carga a nivel de red

Obtenido la solución final del balanceo de carga a nivel de red, se comienza con la automatización de la solución encontrada en balanceo de carga a nivel de red. A continuación se describe brevemente los pasos la automatización (ver figura : 11.2) de dicha solución.

1. Se ejecuta el script `./home/comun iptables`, que recibe el parámetro del modo de configuración (bonding o iptables).
2. Primeramente se verifica si tiene una interface `bond0`. Si la respuesta es afirmativa pasa al paso 3, caso contrario pasa al paso 4.
3. Eliminar la interface `bond`
4. Copia el archivo `/home/interface` a `/etc/network/interface`. El archivo `interface` tiene la configuración de las interfaces inalámbrica.
5. Reiniciar el servicio de red
6. Configuración del equipo en modo `iptables` (así llamamos al balanceo de carga a nivel de red), esta es una función llamada `configuración_iptables()` y se encuentra en `/home/comun`.
7. Finalizar proceso. Este script se lo puede ver en el anexo A

11. Solución de balanceo de carga a nivel de red

Tasas de bit	RX-ath0	RX -ath1	TX-ath0	TX-ath1
6	60,03	39,97	60,07	39,93
9	51,98	48,02	51,98	48,02
12	39,21	60,79	34,60	65,40
18	46,33	53,67	42,80	57,20
24	53,48	46,52	49,94	50,06
36	53,38	46,62	64,31	35,69
48	53,21	46,79	56,83	43,17
54	54,05	45,95	72,23	27,77

Cuadro 11.1.: Paquetes TX y RX en un flujo

11.4. Ejecución de prueba de la solución balanceo de carga a nivel de red

Después de tener los script de autoconfiguración, se empezó a realizar pruebas de tráfico ejecutando dentro del equipo_1 el archivo Principal.sh que tiene la configuración de las pruebas de tráfico y de la configuración de las dos soluciones, este archivo recibe 4 parámetros: 1) nombre de la solución, 2) tipo de prueba, 3) número de repeticiones de la prueba y 4) nombre de la carpeta donde se va a almacenar todos los resultados. Para esta prueba se lo ejecuta de la siguiente manera:

```
geovana@geovana: /Pruebas$/Principal.sh iptables multifujo 7 IP2F_2
geovana@geovana: /Pruebas$/Principal.sh iptables flujo 7 IP1F_2
```

Después de terminar la ejecución de prueba se obtiene los resultados, luego esos resultados se lo procesa en hojas de cálculo y de ese proceso se empieza a ver los resultados finales de cada prueba. En este apartado se va a analizar el comportamiento de las interfaces cuando se realiza balanceo de carga en dos escenarios diferentes.

11.4.1. Primer escenario, enviando un sólo flujo

En esta prueba los paquetes son enviados a un dirección IP destino.

- **Paquetes transmitidos y recibidos por las interfaces.** En la figura del cuadro estadístico 11.3 nos muestra que el reparto de los paquetes de transmisión (TX) en el peor de los casos es: ath0 es el 30 % y ath1 es el 70 % y el reparto de los paquetes recibidos (RX) en el peor de los casos es: ath0 es el 40 % y ath1 es el 60 %. Viendo estos datos estadísticos (ver cuadro 11.1) se puede decir que existe un balanceo de carga, pero revisando a más detalle el anexo E muestra que cada flujo enviado a una dirección IP los paquetes salen por una sola interfaz. Pero los datos estadísticos que se presentan están basados a varias pruebas, para cada una de las tasas de transmisión. Entonces se puede decir que el balanceo de carga a nivel de red se guardan las rutas a las que se a tenido acceso durante un tiempo.

11.4.2. Segundo escenario, enviando los paquetes en dos flujos

En este segundo escenario, se envía los paquetes en dos flujos a dos direcciones IP destino.

- **Paquetes transmitidos y recibidos por las interfaces.** En el cuadro estadístico 11.2 con su respectiva gráfica 11.4 nos muestra el reparto de los paquetes transmitidos (TX) y recibidos (RX) entre interfaces de dos nodos WiFi. Los paquetes TX el peor de los casos es: en ath0 es el 46 % y en ath1 es el 54 %, en el tasa de bit de 18 Mbps. y los paquetes RX en el peor de los casos se encuentra en la tasa de bit 54 Mbps. es: en ath0 49,9 % y ath1 50,04 %. Entonces se puede decir que el balanceo de carga a nivel de red enviando 2 flujos funciona correctamente.

11.4. Ejecución de prueba de la solución balanceo de carga a nivel de red

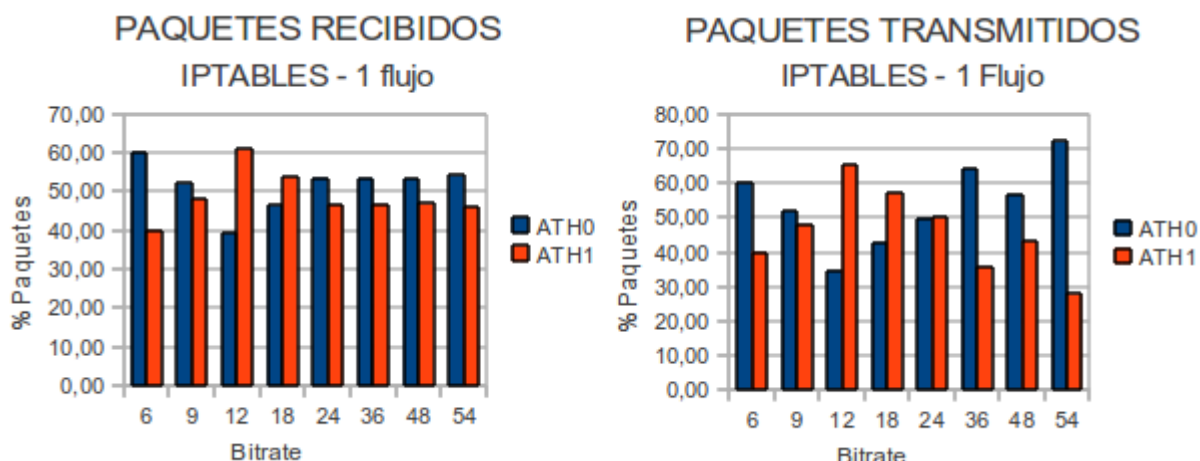


Figura 11.3.: Ipatables: Paquetes TX y RX en un flujo

Tasas de bit	RX-ath0	RX -ath1	TX-ath0	TX-ath1
6	49,88	50,12	50,82	49,18
9	50,15	49,85	50,46	49,54
12	50,32	49,68	50,76	49,24
18	47,24	52,76	46,17	53,83
24	50,10	49,90	49,48	50,52
36	50,37	49,63	49,56	50,44
48	49,24	50,76	47,25	52,75
54	49,62	50,38	48,57	51,43

Cuadro 11.2.: Paquetes TX y RX en dos flujos

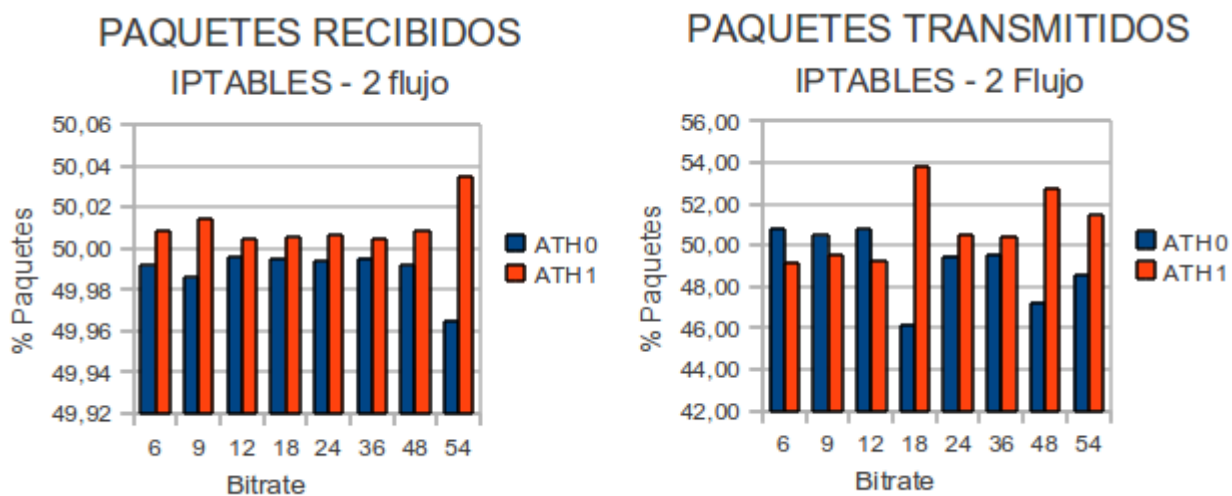


Figura 11.4.: Iptables: Paquetes TX y TR en dos flujos

11. Solución de balanceo de carga a nivel de red

En conclusión, podemos decir que el balanceo de carga a nivel de red sólo funciona de manera similar en el caso de que se envía los paquetes a 2 o más flujos de datos con diferentes direcciones IP destino. Pero si se envían paquetes a una sola dirección IP no funciona el balanceo de carga, causa del cache de enrutamiento.

12. Resultados de experimentación

En los resultados de la experimentación, debe comentarse que se analizó y probó la aplicación "Shorewall", un firewall reconocido en ambientes Linux, sin embargo en las pruebas iniciales los resultados obtenidos fueron poco satisfactorios para este propósito, además se considero como una aplicación con mayores alcances al problema que se quiere resolver y esto conlleva a su pronto descarte, esto en el balanceo de carga a nivel de red. También se tuvo en cuenta al fabricante de equipos Mikrotik, que ofrece una implementación de Bonding, y que se podría pensar que el tema de los enlaces redundantes en carga compartida ya estaba solucionado, sin embargo dada su característica de ser un sistema propietario o cerrado, que rompería con la filosofía de trabajo en sistemas abiertos y/o software libre con la que se ha trabajado hasta ahora. Y también se decidió intentar la solución con WiFi, una tecnología muy extendida, aunque ya no se considere una tecnología de punta, pero que puede proveer soluciones interesantes y suficientes para las necesidades que se tienen en los países en vías de desarrollo, y que al final le proporcionarán al usuario final los mismos servicios.

En este capítulo se presenta los resultados de las dos soluciones viables para el balanceo de carga y enlaces redundantes, que pueden ser usados en WiFi, este es el caso de bonding y balanceo de carga a nivel red con una configuración manual que lo denominamos pruebas de "iptables". Para medir estos resultados se tomarán en cuenta la QoS de las técnicas del balanceo de carga que vio en la sección 9.6 y estos parámetros son: retardo, variación del retardo, caudal y pérdida de paquetes. Estos parámetros ayudarán al PFM a medir, cuál de estas dos soluciones planteadas, es la que mejor se acomoda a los objetivos del proyecto. Para esto se ha planteado dos escenarios de pruebas que son los siguientes: el primer escenario, enviando los paquetes en un sólo flujo y el segundo escenario, enviando los paquetes en dos flujos.

12.1. Primer escenario, enviando un sólo flujo.

En este primer escenario, se analiza los resultados de los datos obtenidos entre las dos soluciones para el balanceo de carga cuando se envía los paquetes en un sólo flujo. Los datos a analizar se ven en los siguientes apartados son la comparativa de los resultados de las dos soluciones propuestas. Además estos resultados se pueden visualizar en el anexo E, donde se guarda los resúmenes de todos los resultados de las diferentes pruebas de este PFM.

12.1.1. Comparativa del retardo

Aquí realizamos el análisis de la comparativa del retardo cuando los paquetes son enviados a una sola dirección IP, esta comparativa se realizó con las dos soluciones de balanceo de carga. En el cuadro 12.1 y su respectiva figura 12.1 muestran la comparativa del retardo en función de la tasa de transmisión aplicada. El retardo según la tabla presentada muestra que el peor resultado se encuentra en la tasa de transmisión de 6 Mbps en ambas soluciones, en el caso de bonding es 0,152065 segundos y en iptables es 0,523228 segundos. El retardo según la tabla presentada muestra que el mejor resultado se encuentra en la tasa de transmisión de 54 Mbps en ambas soluciones, en el caso de bonding es 0,011864 segundos y en iptables es 0,071754 segundos. En este caso la solución bonding tiene el mejor de los resultados frente a la solución de balanceo de carga a nivel de red que tiene casi el doble o más retardo en el promedio de todas las tasas de bit. Analizando porque el retardo es mayor en la solución del balanceo de carga a nivel de red, se llega a la conclusión que es debido al cache de enrutamiento, y como en esta prueba se envió los paquetes a una sola dirección IP destino, entonces cada flujo viaja por una sola interfaz.

12. Resultados de experimentación

Tasa de bit teórico	Tasa de bit real	Promedio retardo (s) - Bonding	Promedio retardo (s) - Iptables
6	5	0,152065	0,523228
9	7,4	0,108974	0,352166
12	9,2	0,084081	0,267141
18	12	0,017028	0,182667
24	16,5	0,044817	0,141215
36	20,8	0,009991	0,097938
48	27,6	0,011331	0,071515
54	31,8	0,011864	0,071737

Cuadro 12.1.: Comparativa del retardo en un flujo

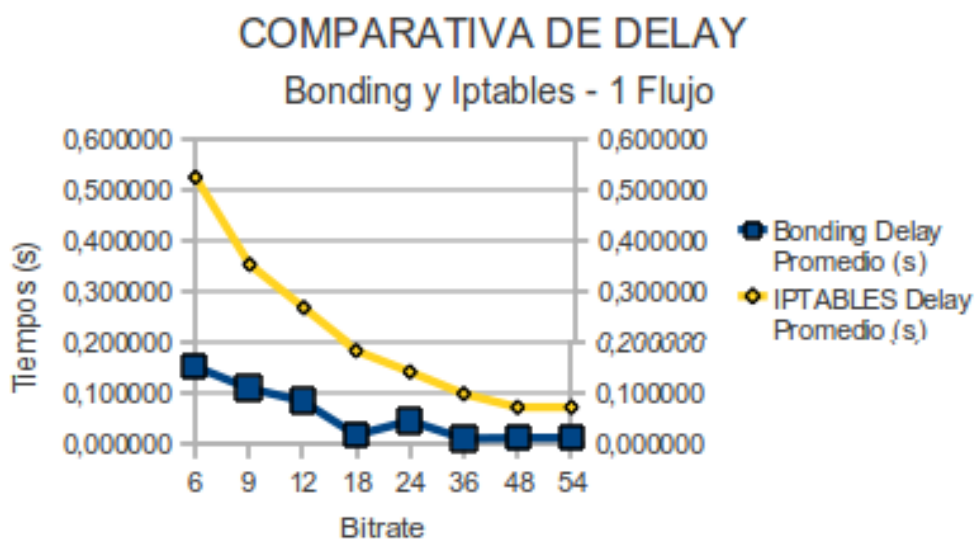


Figura 12.1.: Comparativa del retardo en un flujo

12.1. Primer escenario, enviando un sólo flujo.

Tasa de bit teórico	Tasa de bit real	Promedio variación del retardo (s) - Bonding	Promedio variación del retardo (s) - Iptables
6	5	0,259398	0,005432
9	7,4	0,172755	0,002808
12	9,2	0,138939	0,002018
18	12	0,014073	0,001327
24	16,5	0,059678	0,001035
36	20,8	0,002575	0,000716
48	27,6	0,005147	0,000532
54	31,8	0,006394	0,000556

Cuadro 12.2.: Comparativa de la variación del retardo en un flujo



Figura 12.2.: Comparativa de la variación del retardo en un flujo

12.1.2. Comparativa de la variación del retardo

En este apartado realizamos el análisis de la variación del retardo de las dos soluciones planteadas de este PFM. En el cuadro 12.2 y su respectiva figura 12.2 nos muestran la comparativa de la variación del retardo de las dos soluciones enviando los paquetes a una única dirección IP. La variación del retardo en el mejor de los casos en las dos soluciones es: en bonding es 0.002575 segundos en la tasa de bit de 36 Mbps. y en iptables es 0.00055654 segundos en la tasa de bit de 54 Mbps. En el peor de los casos la variación del retardo en las dos soluciones es: en bonding 0,259398 segundos y en iptables es de 0,0005432 segundos, ambos en la tasa de bit de 6 Mbps. Con estos datos podemos decir que solución bonding presenta mayor variación de retardo frente a la solución de balanceo de carga a nivel de red. Analizando el caso de bonding podemos decir que este método utiliza un mecanismo round robin (los paquetes son repartidos secuencialmente entre todas las interfaces en dominio de bonding) y en el caso del balanceo de carga a nivel de red es por el cache de enrutamiento (los paquetes ya conocen por que interfaz va a salir los paquetes), entonces podemos decir el balanceo de carga a nivel de red casi siempre sera mejor que bonding porque ya se conoce que ruta por donde salen los paquetes.

12.1.3. Comparativa del caudal

En este apartado nos centramos al análisis de la comparativa del caudal obtenido en las dos soluciones planteadas en este PFM. En el cuadro 12.3 con su respectiva figura 12.3 nos muestra

12. Resultados de experimentación

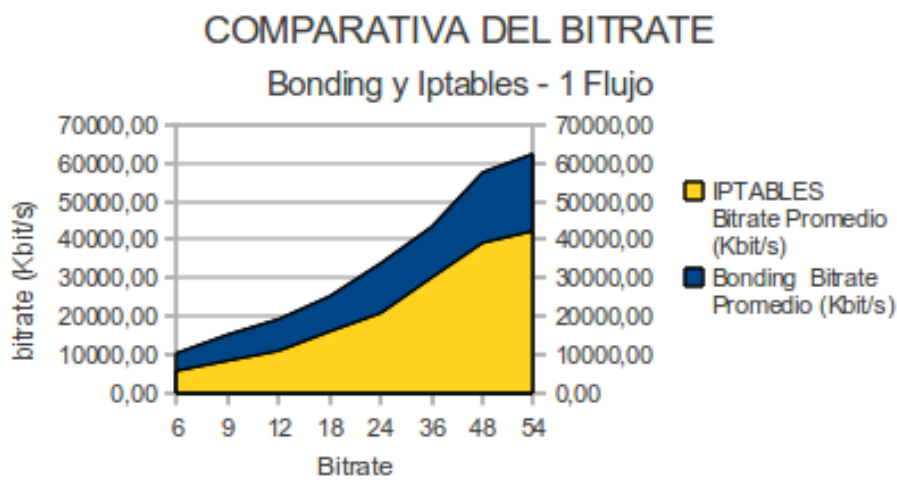


Figura 12.3.: Comparativa del caudal en un flujo

Tasa de bit teórico	Tasa de bit real	Promedio caudal (Kbps) - Bonding	Promedio caudal (Kbps) - Iptables
6	5	10162,26	5524,50
9	7,4	15120,04	8221,03
12	9,2	19153,28	10818,14
18	12	25146,04	15877,06
24	16,5	33813,31	20690,82
36	20,8	43359,65	29993,07
48	27,6	57605,50	39086,33
54	31,8	62403,81	42155,51

Cuadro 12.3.: Comparativa del caudal en un flujo

la comparativa del caudal obtenido por las dos soluciones planteadas en este PFM, enviando los paquetes a una sola dirección IP. El caudal obtenido con la solución bonding duplica el ancho de banda respecto a la columna de la tasa de bit real según la tabla 12.3, como por ejemplo en la tasa de bit de 6 Mbps se observa que se consiguió 10162,26 Kbps. (haciendo la conversión a Mbps, sería 10,16 Mbps.) y la tasa de bit real es de 5 Mbps, con esto deducimos que se logró duplicar el ancho de banda. El caudal obtenido con la solución de balanceo de carga a nivel de red más o menos es la capacidad de un enlace, según las columnas de la tasa de bit real y promedio del caudal de iptables, como por ejemplo en la tasa de bit de 6 Mbps se consiguió 5,5 Mbps y la tasa de bit real es de 5 Mbps. Analizando el caso de la solución del balanceo de carga a nivel de red que solo consiguió el ancho de banda de un enlace, podemos decir que se debe al cache de enrutamiento.

12.1.4. Comparativa de la pérdida de paquetes

En este apartado se hace el análisis de la comparativa de la pérdida de paquetes, cuando los paquetes son enviados a una IP determinada, esto para las dos soluciones presentadas en este PFM. En el cuadro 12.4 con su respectiva figura 12.4 nos muestran la comparativa de las pérdidas de paquetes de las dos soluciones en función a la tasa de bit aplicada. La pérdida de paquetes según la tabla presentada muestra que los peores resultados de las soluciones son: el caso de bonding es 4,51 % en la tasa de bit de 54 Mbps y en el caso de iptables es 46,83 % en la tasa de bit de 6 Mbps. La pérdida de paquetes con los mejores resultados de las soluciones según la tabla presentada son: en caso de bonding es 0,69 % y en iptables es 31,2 %, ambas en la tasa de bit de 36 Mbps. Analizando la solución con mayores pérdidas se debió a que los paquetes

12.2. Segundo escenario, enviando en dos flujos.

Tasa de bit teórico	Tasa de bit real	Pérdida de paquetes (%) Bonding	Pérdida paquetes (%) Iptables
6	5	2,98	46,83
9	7,4	2,51	46,72
12	9,2	0,81	43,74
18	12	0,12	36,75
24	16,5	2,37	40,14
36	20,8	0,69	31,20
48	27,6	2,51	33,82
54	31,8	4,51	35,83

Cuadro 12.4.: Tabla de la comparativa de paquetes perdidos en un flujo

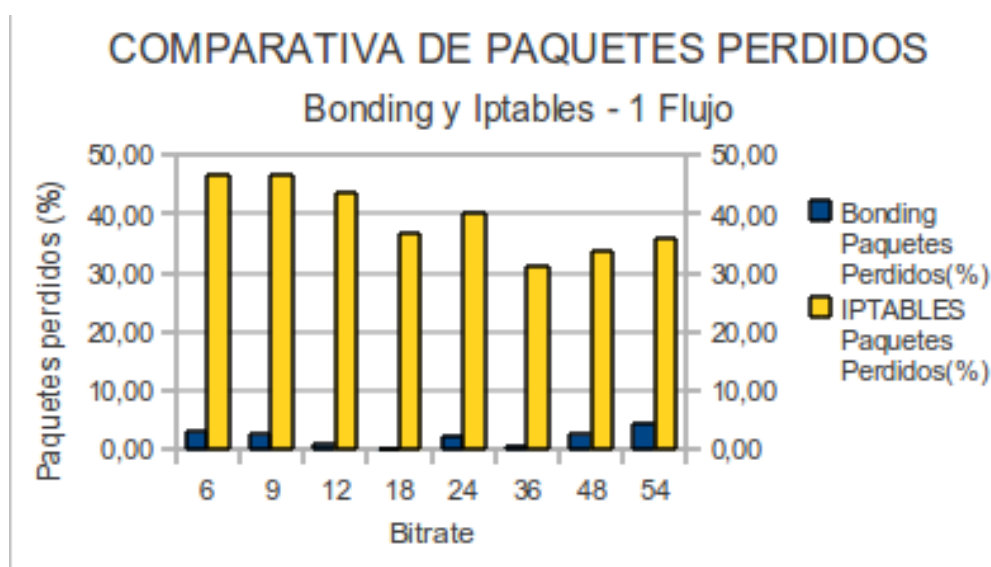


Figura 12.4.: Comparativa de los paquetes perdidos en un flujo

transmitidos intentaron salir por la misma interfaz, con esto consiguió saturar el ancho de banda de esa interfaz y los paquetes fueron desechados y nunca llegaron a su destino. Dichos paquetes salieron por la misma interfaz debido al cache de enrutamiento.

12.2. Segundo escenario, enviando en dos flujos.

En el segundo escenario, se analiza los resultados de los datos obtenidos entre las dos soluciones para el balanceo de carga cuando se envían paquetes en dos flujos. En las siguientes subsecciones se medirán los resultados de QoS en las pruebas donde se envían dos flujos a dos direcciones de IP distintos. Además estos resultados se puede visualizar en el anexo E, donde se guarda los resúmenes de todos los resultados obtenidos en este PFM.

12.2.1. Comparativa del retardo

En este apartado se realiza el análisis de la comparativa del retardo de las dos soluciones planteadas en este PFM, cuando se envían dos flujos a dos direcciones IP diferentes. En el cuadro 12.5 y su respectiva figura 12.5 nos muestran la comparativa del retardo de las dos soluciones en función a la tasa de bit aplicada. El retardo según esta tabla presentada muestra que el peor resultado se encuentra en la tasa de bit de 6 Mbps en ambas soluciones, en el caso de bonding es 0,445173 segundos y en iptables es 0,365181 segundos. El retardo según la tabla presentada muestra que el mejor resultado en ambas soluciones es: en bonding es 0,020791 segundos en

12. Resultados de experimentación

Tasa de bit teórico	Tasa de bit real	Promedio retardo (s) Bonding	Promedio retardo (s) Iptables
6	5	0,445173	0,365181
9	7,4	0,305397	0,253866
12	9,2	0,067341	0,153839
18	12	0,020791	0,106632
24	16,5	0,039877	0,092493
36	20,8	0,009420	0,065010
48	27,6	0,022562	0,060259
54	31,8	0,025014	0,059313

Cuadro 12.5.: Tabla de la comparativa del retardo en un flujo

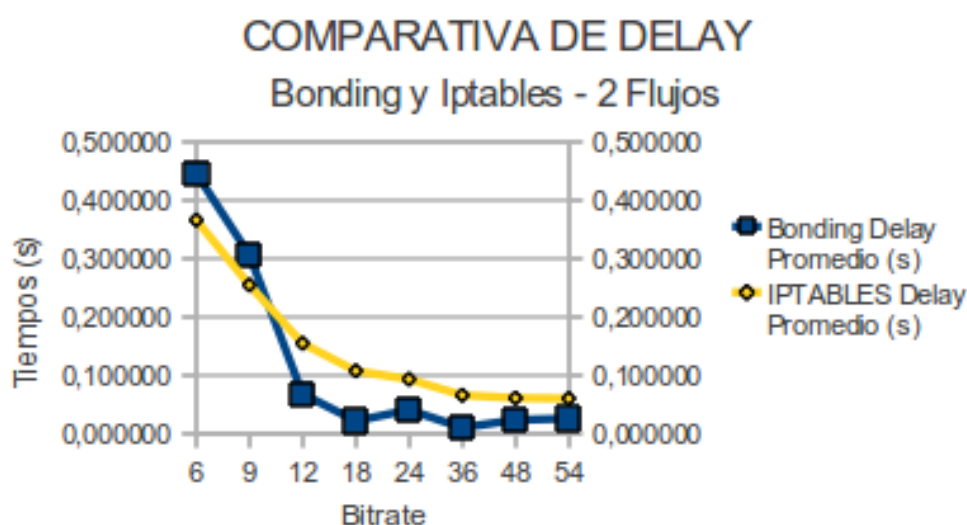


Figura 12.5.: Comparativa de delay en dos flujos

la tasa de bit de 18 Mbps y en iptables es 0,059313 segundos en la tasa de bit de 54 Mbps. Analizando los resultados de esta tabla vemos que los resultados son casi similares, en algunos casos la solución de bonding tiene menor retardo según la tasa de bit (12, 24, 36, 48 y 54 Mbps) y en otras la solución de balanceo de carga a nivel de red tiene menor resultados según la tasa de bit (6 y 9 Mbps). Entonces se puede concluir que el retardo es similar en ambas soluciones cuando se envían dos flujos a 2 direcciones IP diferentes.

12.2.2. Comparativa del variación de retardo

En este apartado realizamos el análisis de la variación del retardo de las dos soluciones planteadas de este PFM, enviando paquetes en dos direcciones IP destino (las IP son direcciones diferentes). En el cuadro 12.6 y su respectiva figura 12.6 nos muestran la comparativa de la variación del retardo de las dos soluciones enviando los paquetes a a dos direcciones IP distintas. La variación del retardo en el mejor de los casos en las dos soluciones es: en bonding es 0,001066 segundos en la tasa de bit de 36 Mbps. y en iptables es 0,000730 segundos en la tasa de bit de 54 Mbps. En el peor de los casos la variación del retardo en las dos soluciones es: en bonding 0,058417 segundos en la tasa de bit de 12 Mbps y en iptables es de 0,004610 segundos en la tasa de bit de 6 Mbps. Con estos datos podemos decir que solución bonding presenta mayor variación de retardo frente a la solución de balanceo de carga a nivel de red. Analizando el caso de bonding podemos decir que este método utiliza un mecanismo round robin (los paquetes son repartidos secuencialmente entre todas las interfaces en dominio de bonding) y en el caso del balanceo de carga a nivel de red es por el cache de enrutamiento (los paquetes ya conocen por que interfaz

12.2. Segundo escenario, enviando en dos flujos.

Tasa de bit teórico	Tasa de bit real	Promedio variación del retardo (s) Bonding	Promedio variación del retardo (s) Iptables
6	5	0,019839	0,004610
9	7,4	0,013781	0,002950
12	9,2	0,058417	0,002304
18	12	0,009094	0,001556
24	16,5	0,024338	0,001259
36	20,8	0,001066	0,000906
48	27,6	0,009955	0,000783
54	31,8	0,011671	0,000730

Cuadro 12.6.: Comparativa de la variación del retardo en dos flujos

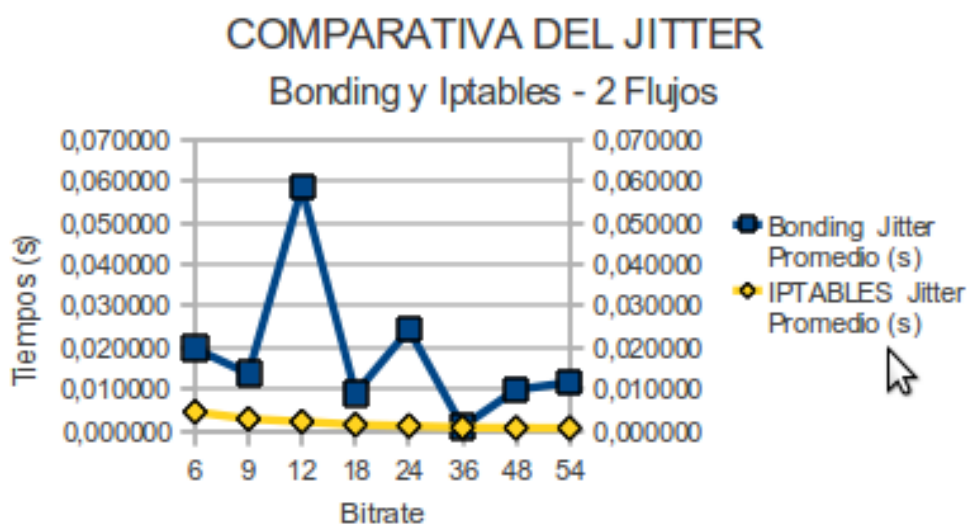


Figura 12.6.: Comparativa de la variación de retardo en dos flujos

va a salir los paquetes), entonces podemos decir el balanceo de carga a nivel de red casi siempre sera mejor que bonding porque ya se conoce que ruta por donde salen los paquetes (mientras no se borre el cache de enrutamiento).

12.2.3. Comparativa del caudal

En este apartado nos centramos al análisis de la comparativa del caudal obtenido en las dos soluciones planteadas en este PFM. En el cuadro 12.7 con su respectiva figura 12.7 nos muestra la comparativa del caudal obtenido por las dos soluciones planteadas en este PFM, enviando los paquetes a dos direcciones de IP diferentes. El caudal obtenido con la solución bonding casi duplica el ancho de banda respecto a la columna de la tasa de bit real según la tabla 12.7, como por ejemplo en la tasa de bit de 6 Mbps se observa que se consiguió 9,84 Mbps y la tasa de bit real es de 5 Mbps, con esto deducimos que se logro duplicar el ancho de banda. El caudal obtenido con la solución de balanceo de carga a nivel de red casi duplica el ancho de banda respecto a la columna de la tasa de real, según la tabla 12.7, como por ejemplo en la tasa de bit de 54 Mbps se podemos ver que se consiguió 57,49 Mbps y la tasa de bit real es de 31 Mbps, con esto se deduce que se logro aumentar el caudal en un 90%. Analizando estas dos soluciones podemos decir que ambas soluciones aumentan el caudal, cuando se envían 2 flujos a dos direcciones IP diferentes.

12. Resultados de experimentación

Tasa de bit teórico	Tasa de bit real	Promedio caudal (Kbps) Bonding	Promedio caudal (Kbps) Iptables
6	5	9845,31	9389,42
9	7,4	14555,59	13789,76
12	9,2	18954,26	17696,19
18	12	25108,28	24155,88
24	16,5	34030,26	31592,43
36	20,8	43588,20	41514,64
48	27,6	56711,09	51829,05
54	31,8	63863,75	57496,34

Cuadro 12.7.: Comparativa del caudal en dos flujo

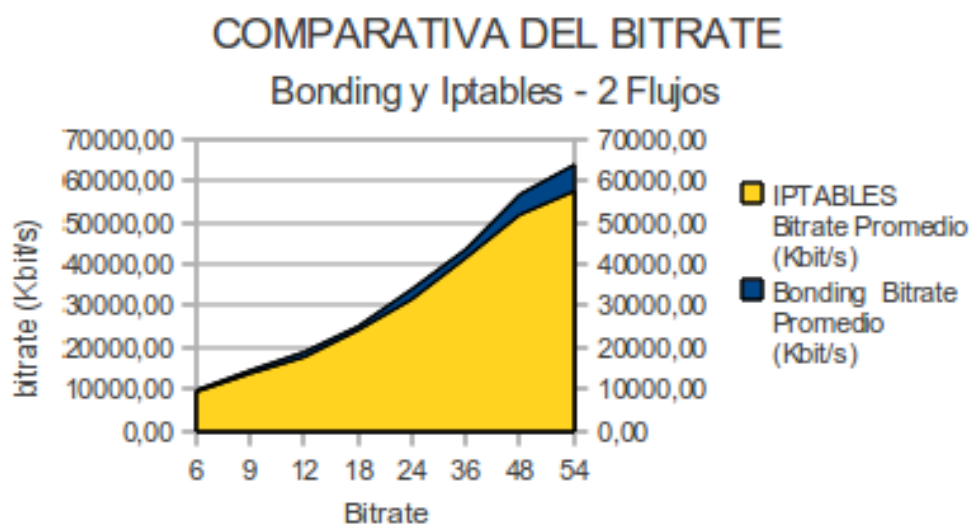


Figura 12.7.: Comparativa del caudal en dos flujo

12.3. Conclusiones de las comparativas de las dos soluciones.

Tasa de bit teórico	Tasa de bit real	Pérdida de paquetes (%) Bonding	% Pérdida de paquetes (%) Iptables
6	5	5,29	9,94
9	7,4	5,74	10,87
12	9,2	1,70	8,20
18	12	0,22	3,91
24	16,5	1,69	8,70
36	20,8	0,16	4,84
48	27,6	2,07	10,47
54	31,8	4,30	13,82

Cuadro 12.8.: Comparativa de la pérdida de paquetes en dos flujos

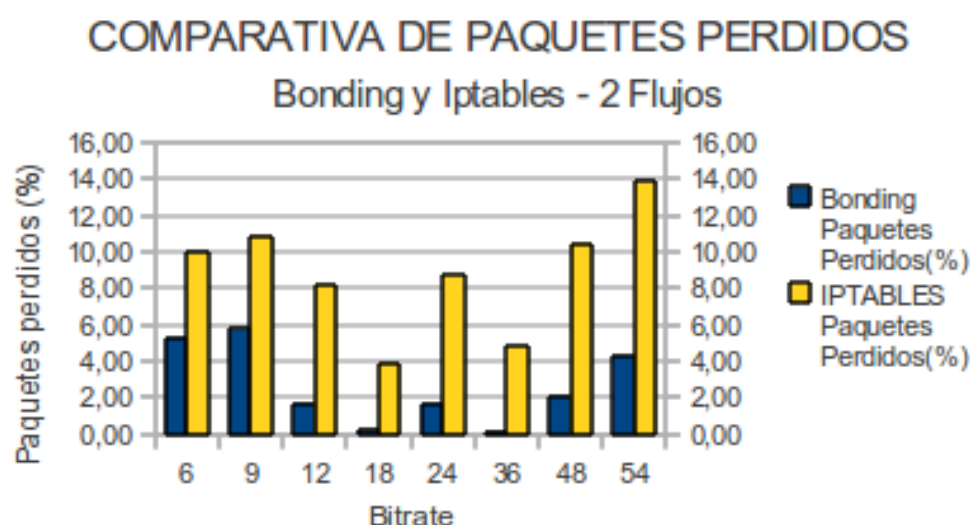


Figura 12.8.: Comparativa de la pérdida de paquetes en dos flujos

12.2.4. Comparativa de la pérdida de paquetes

En este apartado se hace el análisis de la comparativa de la pérdida de paquetes, cuando los paquetes son enviados a dos IP destino, esto para las dos soluciones presentadas en este PFM. En el cuadro 12.8 con su respectiva figura 12.8 nos muestran la comparativa de las pérdidas de paquetes de las dos soluciones en función a la tasa de bit aplicada. La pérdida de paquetes según la tabla presentada, muestra que los peores resultados de las soluciones son: el caso de bonding es 5,74% en la tasa de bit de 9 Mbps y en el caso de iptables es 13,82% en la tasa de bit de 54 Mbps. La pérdida de paquetes con los mejores resultados de las soluciones según la tabla presentada son: en caso de bonding es 0,16% en la tasa de bit de 36 Mbps y en iptables es 3,91% en la tasa de bit de 18 Mbps. La solución con mayores pérdidas es el balanceo de carga a nivel de red.

12.3. Conclusiones de las comparativas de las dos soluciones.

En todo este capítulo se ha analizado las comparativas presentadas de las dos soluciones de balanceo de carga en este PFM, midiendo los 4 parámetros de Qos (retardo, variación de retardo, caudal y pérdida de paquetes).

- En el caso del primer escenario, vemos que el funcionamiento de la solución bonding obtiene mejores resultados en tres de los parámetros consiguiendo un menor retardo, aumento del

12. Resultados de experimentación

caudal y menores pérdidas.

- En el caso del segundo escenario, vemos que el funcionamiento de las dos soluciones son similares, ambas consiguen realizar balanceo de carga. En el caso de bonding consigue un mayor caudal y menores pérdidas respecto a la otra solución. En el caso del balanceo de carga a nivel de red tiene mejor resultado en la variación del retardo, gracias al cache de enrutamiento. Pero ambas soluciones tienen similar comportamiento en el caso del retardo.

13. Cronograma y presupuesto

En este capítulo se trata de analizar el costes de la investigación y el coste de un nodo WiFi, en el cual se centra este PFM. Para estos dos propósitos se presentará un cronograma de desarrollo del proyecto.

13.1. Cronograma del proyecto

En la figura 13.1 podemos ver el diagrama de Gantt de cada una de las fases que se paso en este proyecto.

13.2. Costes de la investigación

Para el desarrollo de esta investigación se ha contado con el trabajo de 2 personas, que tienen los siguientes roles: jefe de proyecto y desarrollador.

13.2.1. Coste de recurso humano

Por acercarnos a la realidad esta investigación, se considera los sueldos por hora-hombre para cada uno de los roles de la siguiente manera:

- Hora-Hombre de Jefe de Proyecto = 11 e
- Hora-Hombre de desarrollador = 6 e

Si tomamos en cuenta los días trabajados del cronograma del proyecto (ver figura 13.1), que esos días son una aproximación de la realidad, puesto que un desarrollo de una investigación es muy variable (dependerá de la suerte que se tenga en encontrar la solución), aparte es difícil mantener al día la información del estado y avance del proyecto, para esto se considera las horas de trabajo invertidas por las fases, roles y sus respectivos costes son los que se muestran en el siguiente tabla 13.1. El coste total de la investigación en cuanto al recurso humano asciende a 7127 e, según la tabla de los costes de personal.

13.2.2. Coste de materiales

Se considera los costos de los materiales que componen un nodo WiFi en esta investigación, al valor calculado en los materiales se añadade un 10% de presupuesto adicional (otros materiales, material dañado en el proceso de la investigación). En la tabla 13.2 se presenta el coste de los materiales de un nodo.

En este proyecto se tiene dos nodos, entonces el costo de los materiales de esta investigación asciende a 438,79 e (añadido el 10% de gastos adicionales).

Sumando el coste de personal y los costos de los materiales, podemos determinar el coste de la investigación de este proyecto es de 7565,79 e aproximadamente.

13. Cronograma y presupuesto

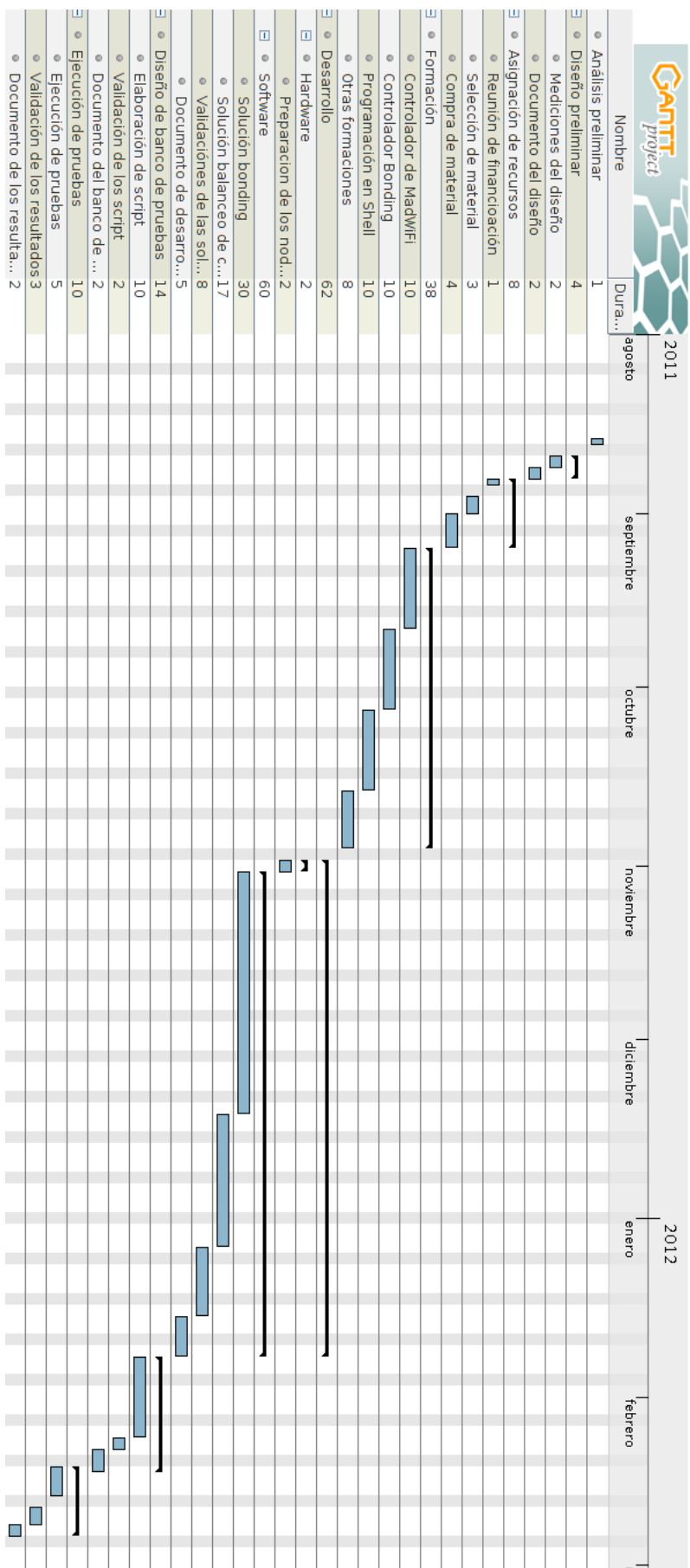


Figura 13.1.: Cronograma del proyecto

13.2. Costes de la investigación

Etapa	H-H Jefe de Proyecto	H-H Desarrollador	H-H Totales	Costes Jefe Proyecto	Costes Desarrollador
Análisis preliminar	8	0	8	88	0
Diseño preliminar	15	15	30	165	90
Asignación de recursos	20	40	60	220	240
Formación	0	300	300	0	1800
Desarrollo	50	400	450	550	2400
Validación de desarrollo	10	30	40	110	180
Diseño banco de pruebas	8	80	88	88	480
Validación de diseño	5	20	25	55	120
Ejecución de pruebas	5	50	55	55	300
Validación de diseño	6	20	26	66	120
TOTALES	127	955	1082	1397	5730

Cuadro 13.1.: Coste del personal

13. Cronograma y presupuesto

Descripción	Fabricante	Referencia	Distribuidor	Coste	Unidades	Subtotal
Placa base	PCEngines	ALIX 2D2 LX800 2LAN2, RAM 256MB	Landatel	90,9	1	90,9
Caja para el ALIX	Landatel	Caja de aluminio de interior ALIX 2D2 2 LAN Alum USB	Landatel	6,37	1	6,37
Tarjeta inalámbrica Wifi baja potencia	Wistron	Wistron Wistron CM9 WLAN 802.11a/b/g, 2 x SMT Ultra-miniature coaxial connectors	CiudadWireless	31,13	2	62,26
Compact Flash 4GB	Kingston	CF-4096 MB	Landatel	14	1	14
Antenas dipolo	Landatel	Antena dipolo DUAL 5 /2,4 GHz. 5 dBi Conector RSMA	Landatel	4,98	2	9,96
Pigtail	CiudadWireless	MMCX pigtail 2,4/5 GHz.	CiudadWireless	5,22	2	10,44
Fuente de Alimentación	Landatel	Fuente de Alimentación AC/DC 18 v. 800mA	Landatel	5,52	1	5,52
TOTAL						199,45

Cuadro 13.2.: Costes de material de un nodo WiFi

Parte V.

**CONCLUSIONES Y LÍNEAS
FUTURAS**

14. Discusión de los resultados

En el capítulo 12, se observan los resultados obtenidos en este PFM, pero existen algunos aspectos de esos resultados que se deben ser anotados:

- En el balanceo de carga a nivel de red, se utiliza una memoria caché de encaminamiento, donde se almacenan las IP destinos, durante un tiempo, esto hace que cada vez que se envíen paquetes a una dirección IP destino conocido, este elige siempre el mismo camino. Aunque esta caché de enrutamiento es configurable y en este PFM limitamos que se guarde la cache de enrutamiento durante un tiempo de 0 segundos (para emular la inexistencia de la caché), no funcionó y por esto se puede ver en el primer escenario que se envían paquetes por un camino preferente a una dirección IP destino en la solución de balanceo de carga a nivel de red.

15. Conclusiones

Este PFM ha intentado investigar, analizar y probar alternativas para enlaces redundantes en redes inalámbricas con balanceo de carga; como una manera de contribuir a la robustez y aumento del ancho de banda en redes inalámbricas que usa una topología de red lineal, tal es el caso del proyecto que se encuentra en la orillas del río Napo en el Perú, implementado por la Fundación EHAS. A continuación se resumen las fases del proyecto.

- Revisión del estado del arte en la literatura científica sobre el balanceo de carga en redes inalámbricas WiFi.
- Análisis de las alternativas y selección de las mismas, para la solución del problema.
- Diseño y construcción del prototipo de laboratorio para la prueba de alternativas.
- Diseño e implementación del banco de pruebas
- Procesamiento de los resultados de las pruebas.
- Análisis de los resultados

Luego de todo este proceso, a continuación se sintetizan las observaciones y conclusiones a los que se llegó en el presente PFM.

- La primera conclusión que se puede aportar es que se ha conseguido con éxito llegar al objetivo del proyecto.
- La solución del balanceo de carga a nivel de red, en lo referente a la caché de enrutamiento tiene la desventaja cuando la comunicación se realiza con una sola dirección IP.
- La solución del balanceo de carga a nivel de red, en lo referente a la caché de enrutamiento tiene la ventaja en el parámetro de la variación del retardo, porque sabe porque interfaz va a salir mientras la IP destino se encuentre en la caché de enrutamiento.
- La solución del balanceo de carga a nivel de red mejora su funcionamiento cuando la comunicación de los enlaces trabaja con varios flujos y distintos IP destino, según las mediciones del QoS efectuadas a los resultados de esta prueba.
- La solución de bonding, es más tolerante a fallos, por que tiene mayor robustez al usar al usar enlaces redundantes ya que la interface “bond” tiene en su dominio a las tarjetas de red real. La probabilidad de que todas las tarjetas inalámbricas fallen es mucho menor a que una de las interfaces falle. Si una interfaz falla, la comunicación de los enlaces sigue funcionando.
- La solución más adecuada, según al análisis y pruebas realizadas en el presente PFM es la solución bonding, ya que consigue un mayor caudal usando enlaces redundantes, el aumento del ancho de banda de esta solución esta entre el 95 % hasta el 100 % sobre la tasa de bit real conseguido en un enlace. Asimismo, es más tolerante a fallos y nos ofrece una mayor disponibilidad de los enlaces. El ancho de banda de solución del balanceo de carga a nivel de red con enlaces redundantes ha conseguido aumentar el caudal entre el 80 % hasta el 87 % cuando se envían datos por dos enlaces a dos direcciones IP destino. Cabe resaltar que la solución del balanceo de carga a nivel de red mejora considerablemente con el aumento de envío de más flujos a distintas direcciones IP destino, por lo que en un escenario real con múltiples comunicaciones, la mejora de bonding con respecto al balanceo de carga a nivel de red sería más ajustada.
- Implantada sobre la red del río Napo, cualquiera de las soluciones de balanceo de carga sobre redes inalámbricas redundaría en una mejora considerable del servicio, debido a la mejora de la estabilidad del sistema completo, al tiempo que podría suponer un ahorro económico importante derivado de la reducción del número de viajes de reparación urgente

15. Conclusiones

que los técnicos deben realizar (pueden programar un mantenimiento preventivo cada cierto periodo de tiempo en lugar de tener que aplicar constantemente un mantenimiento correctivo cuando un enlace falla).

- El incremento de costes que la aplicación del balanceo de carga podría tener sobre la infraestructura existente son mínimos en relación al coste de la infraestructura instalada para soportar los equipos (torres ventadas de mucha altura), por lo que su aplicación está justificada.

16. Líneas futuras de desarrollo e investigación

Se prevee que esta línea de balanceo de carga para redes inalámbricas pueda proponerse como tema de investigación en trabajos futuros. A continuación se presentan las líneas futuras de desarrollo e investigación más evidentes:

- Continuando con la misma línea de investigación, se puede investigar como hacer funcionar otros modos de bonding, como por ejemplo el modo backup, que la tecnología Mikrotik ya la tiene implementada.
- Como se vio en tercera prueba para encontrar la solución bonding se utilizó el controlador de ath5k en el que viene implementado el funcionamiento de bonding en el SO GNU/LINUX Voyage 0.7.5, donde se vio que el módulo de bonding es cargado pero solo se transmite la capacidad de un enlace. Viendo esta línea se podría investigar que es lo que ocurre en esa versión del kernel y driver.
- Cuantificar los beneficios económicos que la aplicación de esta técnica podría implicar sobre los proyectos de la fundación EHAS, en función de la topología de la red sobre la que se ha de aplicar (previsiblemente tendrá un impacto mayor en redes lineales con un único camino).
- Comparar el rendimiento de bonding sobre 802.11a/g con el de estándares más recientes como 802.11n o 802.16.

Bibliografía

- [ADYO07] S. Aust, P. Davis, A. Yamaguchi, and S. Obana. Interface status monitoring for wireless link aggregation in cognitive networks. *Information technology*, 2007.
- [Avg08] Chrisanthi Avgrou. Information systems and developing countries: Failure, success and local improvisations. *Information technology*, sep 2008.
- [Avg09] Chrisanthi Avgrou. Discourses on innovation and development in information system in developing countries research. http://www.ifip.dsg.ae/Docs/FinalPDF/FullPapers/Avgerou_Discourseson_InnovationandDevelopment.pdf, 2009.
- [B.02] Richard B. Information systems and developing countries: Failure, success and local improvisations, 2002.
- [BG10] Inés María Bebea González. *Diseño de un plan de sostenibilidad para redes de comunicaciones rurales: Estudio del caso Napo*, chapter 11. Universidad Rey Juan Carlos, 2010.
- [CKV06] Chandrakanth Chereddi, Pradeep Kyasanur, and Nitin H. Vaidya. Net-x: System extensions for supporting multiple channels, multiple interfaces, and other interface capabilities. *Information technology*, aug 2006.
- [dlNUpeDa] Programa de las Naciones Unidas para el Desarrollo. El pnud y las instituciones financieras internacionales.
- [dlNUpeDb] Programa de las Naciones Unidas para el Desarrollo. Objetivos de desarrollo del milenio.
- [dlNUpeD11] Programa de las Naciones Unidas para el Desarrollo. Objetivos de desarrollo del milenio, informe del 2011, 2011.
- [dlPI] Organización Mundial de la Propiedad Intelectual. Convenio de berna para la protección de las obras literarias y artísticas. http://www.wipo.int/treaties/es/ip/berne/trtdocs_wo001.html.
- [dS] Enlace Hispano Americano de Salud. Ehas. <http://www.ahas.org>.
- [DWPW07] Yu Dong, Dingding Wang, Niki Pissinou, and Jian Wang. Multi-path load balancing in transport layer. *Information technology*, 2007.
- [FG11] KRISTOFFER FREDRIKSSON and MATTIAS GUHL. *Multi-Channel, Multi-Radio in Wireless Mesh Networks*, chapter 5. Department of Computer Science and Engineering, 2011.
- [Foua] Free Software Foundation. Free software foundation website. <http://www.fsf.org>.
- [Foub] Free Software Foundation. General public license version 3. <http://www.gnu.org/copyleft/gpl.txt>.

Bibliografía

- [Gar] Pablo Osuna García. Ehas, diseño e instalación de una red de comunicaciones wi-fi en un área rural de Perú. <http://www.ahas.org>.
- [Gui07] Alberto Guijarro. *TIC al servicio de los Objetivos de Desarrollo del Milenio*. Colegio oficial de ingenieros de telecomunicación, 2007.
- [HMvO⁺04] Bert Hubert, Gregory Maxwell, Martijn van Oosterhout, Remco van Mook, Paul B. Schroeder, et al. Linux advanced routing and traffic control howto. <http://lartc.org/howto/>, dic 2004.
- [iWMN11] Multipath Routing in Wireless Mesh Networks. *Nagesh S and Nandiraju and Deepti S. Nandiraju and Dharma P. Agrawal*, chapter 8. Department of Computer Science and Engineering, 2011.
- [Jon07] M. Tim Jones. Anatomy of the linux kernel. http://www.ibm.com/developerworks/linux/library/l-linux-kernel/?S_TACT=105AGX59&S_CMP=GR&ca=dgr-lnxw07LKernalAnatomy, jul 2007.
- [Mor10] Carlos Rey Moreno. Actas de las 1as jornadas internacionales en investigación en tic para el desarrollo humano, jul 2010.
- [mp] The madwifi project. The madwifi project is. <http://madwifi-project.org/>.
- [Mun] Vision Mundial. Por una vida en toda plenitud para la niñez de bolivia y vsats. http://wiki.visionmundial.org.bo/index.php/Manual_de_Gesti%C3%B3n_de_VSATs.
- [Mü07] C. Müller. Implementation of a multi-channel multi-radio prototype on embedded linux. dic 2007.
- [Net] Netfilter. The netfilter.org iptables”project. <http://www.netfilter.org/>.
- [oDD] European Network on Debt and Development. Cumbre del g8 2005 en gleneagles. <http://eurodad.org/1302/?lang=es>.
- [pC] One Laptop per Child. Mission is to empower the world’s poorest children through education.
- [Per10] Ignacio Foche Pérez. *Desarrollo del un teleestetoscopio digital Bluetooth para zonas aisladas de países en desarrollo*, chapter 15. Universidad Rey Juan Carlos, 2010.
- [Pro] The GNU Project. The gnu project website. <http://www.gnu.org>.
- [Sor08] Josep Oriol Madriles Soriano. *Diseño de una red de telecomunicaciones para la interconexión de datos y telefonía*, chapter 7. Ingeniería de Telecomunicación, 2008.
- [Sta] Richard Stallman. stallman.org. <http://stallman.org>.
- [TGT⁺09] Willy Tarreau, Constantine Gavrilov, Chad N. Tindel, Janice Girouard, and Jay Vosburgh. Linux ethernet bonding driver mini-howto. <http://www.kernel.org/doc/Documentation/networking/bonding.txt>, sep 2009.
- [Tri07] Nistha Tripathi. *Rate Control Framework for Net-X*. 2007.

- [Uni10] International Telecommunication Union. *World Telecommunication/ICT Development Report 2010 MONITORING THE WSIS TARGETS*, page 65. International Telecommunication Union, 2010.
- [Wik07] Proyecto Wikimedia. Introducción a linux / arquitectura. http://es.wikibooks.org/wiki/Introducci%C3%B3n_a_Linux/Arquitectura, jul 2007.
- [WRH10] Damon Wischik, Costin Raiciu, and Mark Handley. Balancing resource pooling and equipoise in multipath transport. 2010.

Anexos

A. Automatización de configuración del prototipo para bonding y el enrutamiento de control de tráfico

El presente script se ejecuta dentro de los equipos ALIX, para realizar las configuraciones automáticas en los dos modos de bonding y enrutamiento de control de tráfico.

```
#!/bin/sh
#Script Adaptado para ALIX, este script esta dentro de las ALIX y
esta guardado en /home/comun.sh
#VARIABLES GLOBALES
#Interfaces usadas: ath0,ath1, bond0
ATH0="ath0"
ATH1="ath1"
ETH0="eth0"
ETH1="eth1"
BOND0="bond0"
PREFIX="24"
#IP de las interfaces de ethernet
IP_ETH0="10.1.6.2"
IP_ETH1="10.1.7.2 "

#IP de las interfaces inalámbricas
IP_ATH0="192.168.0.1"
IP_ATH1="192.168.1.1"

#IP de las subredes, tanto como en interfaces de ethernet e inalám-
bricas
ETH0_NET="10.1.6.0"
ETH1_NET="10.1.7.0"
ATH0_NET="192.168.0.0"
ATH1_NET="192.168.1.0"

#IP del enlace al cual me conecto. En este caso la ip inalámbrica
```

A. Automatización de configuración del prototipo para bonding y el enrutamiento de control de tráfico

de la ALIX 2

```
ISP0="192.168.0.2"
```

```
ISP1="192.168.1.2"
```

```
#VARIABLES USADAS EN BONDING
```

```
IP_BOND0="192.168.3.1"
```

```
MIIMON="100"
```

```
MAX_BOND="1"
```

```
MAC_MANAGED=
```

```
#El modo en el q trabaja el bonding. MODO_BONDING=0:round  
robin, uso: provee balanceo y alta disponibilidad. MODO_BONDING=1:  
Active-backup.
```

```
# MODO_BONDING=2: XOR, se alterna la transmisión de paque-  
tes entre los esclavos del bonding uso: provee balanceo y alta dispo-  
nibilidad.
```

```
# MODO_BONDING=3: modo broadcast, enviá todo por todos los  
esclavos.
```

```
# MODO_BONDING=4: modo Dynamic link aggregation, crea gru-  
pos que comparten la misma velocidad y duplex.
```

```
#las tarjetas deben soportar IEEE 802.3ad. uso: solo alta disponibi-  
lidad
```

```
# MODO_BONDING=5: balance-tib, balancea todo el trafico de sa-  
lida, todo el trafico de entrada es recibido por el esclavo activo.
```

```
#uso: alta disponibilidad y balanceo de carga.
```

```
# MODO_BONDING=6: balance-alb, básicamente hace lo mismo  
que el mode5, pero también balancea el trafico de entrada. #uso:  
alta disponibilidad y balanceo de carga.
```

```
MODO_BONDING="0"
```

```
DIRECTORIO_ALIX="/home"
```

```
#Funcion principal, esta funcion recibe parametros $1 = modo de  
trabajo, $2=nombre del fichero para guardar los paquetes #transmi-  
tidos y recibidos
```

```
ejecutar(){
```

```
remountrw
```

```
sleep 1
```

```
if [ $# -ne 1 ]; then
```

```
echo "ejecutar: ERROR: Ingrese en el modo q va a trabajar la prueba
```

```

(bonding, iptables, inicial)”
exit 1
fi
case $1 in
bonding)
echo ”Copiando la interface bond a /etc/network/interfaces ”
cp $DIRECTORIO_ALIX/interfaces_bond /etc/network/interfaces
sleep 3
/etc/init.d/networking restart
sleep 3
configuracion_atheros
mac_managed ${ATH0} ${ATH1}
cambiar_mac_master ${ATH0} ${ATH1}
configuracion_bonding
;;
iptables)
desactivar_bonding
echo “Copiando la interface iptables a /etc/network/interfaces ”
cp /home/interfaces /etc/network/interfaces
sleep 1
/etc/init.d/networking restart
sleep 1
configuracion_atheros
configuracion_iptables
;;
inicial)
desactivar_bonding
echo ”Copiando la interface inicial a /etc/network/interfaces ”
cp /home/interfaces /etc/network/interfaces
sleep 1
/etc/init.d/networking restart
sleep 1
configuracion_atheros
configuracion_inicial
;;
esac
remountro
}

```

```
#CONFIGURACIÓN DEL MODO INICIAL
#Este es el estado inicial de las placas alix, nos servirá para sacar el
throughput de cada enlace.
configuracion_inicial(){
remountrw
echo "_____”
echo "| Configurando en modo INICIAL |”
echo "_____”

    ifconfig eth1:1 ${IP_ATH1}/${PREFIX} up
    echo "configuracion_inicial: configurando la tabla de rutas y la cone-
xion del area local de la otra red”
    ip route add ${ETH0_NET}/${PREFIX} via ${IP_ATH0} dev ${ATH0}
    ip route add ${ETH1_NET}/${PREFIX} via ${IP_ATH1} dev ${ATH1}
    echo "configuracion_inicial: Configuracion ha sido realizada correcta-
mente”

}

#CONFIGURACIÓN DEL MODO BONDING
configuracion_bonding(){

    echo "_____”
    echo "| Configurando en modo BONDING |”
    echo "_____”
#Comprobando si se tiene bonding activo, si esta activo: eliminamos
las interfaces esclavas y
# No esta activo: levantamos las interfaces en modo bonding
# desactivar_bonding
echo "configurando_bonding: Empezando con la configuración de la
interface bond0”
ifconfig ${BOND0} ${IP_BOND0}/${PREFIX} up
modprobe bonding miimon=${MIIMON} max_bonds=${MAX_BOND}
mode=${MODO_BONDING}
ifenslave bond0 ${ATH0} ${ATH1}
echo "configurando_bonding: Bonding correctamente configurado”

    ip route add ${ETH0_NET}/${PREFIX} via ${IP_BOND0} dev
${BOND0}
```

```

ip route add ${ETH1_NET}/${PREFIX} via ${IP_BOND0} dev
${BOND0}
echo "configuracion_bonding: Configuracion ha sido realizada correc-
tamente"
}

```

#FIN DE LA CONFIGURACION DEL BONDING

```

#CONFIGURACIÓN DEL MODO IPTABLE
configuracion_iptables(){
echo "_____”
echo "| Configurando en modo IPTABLES |”
echo "_____”

```

```

echo configuracion_iptables: Configurando la tabla de rutas”
ip route add ${ETH1_NET} dev ${ETH1}
ip route add ${ETH1_NET} dev ${ETH1} src ${IP_ETH1}
ip route add ${ATH0_NET} dev ${ATH0} table 1
ip route add default via ${ISP0} table 1
ip route add ${ATH1_NET} dev ${ATH1} table 2
ip route add default via ${ISP1} table 2
ip route add ${ATH0_NET} dev ${ATH0} src ${IP_ATH0}
ip route add ${ATH1_NET} dev ${ATH1} src ${IP_ATH1}

```

```

# balanceamos la carga.
ip route add default nexthop via ${ISP0} dev ${ATH0} weight 1
nexthop via ${ISP1} dev ${ATH1} weight 1

```

```

#Para la comunicación entre las interfaces lógicas
iptables -t nat -A POSTROUTING -j SNAT -s ${IP_ATH0}/24 -o
${ETH1} -to-source ${IP_ETH1}
iptables -t nat -A POSTROUTING -j SNAT -s ${IP_ATH1}/24 -o
${ETH1} -to-source ${IP_ETH1}

```

```

echo “— Configurando parametros de auxiliares de iptables — ”
#si queremos que se produzca una nueva comprobación cada 30 se-
gundos
echo "30■ /proc/sys/net/ipv4/route/gc_interval

```

A. Automatización de configuración del prototipo para bonding y el enrutamiento de control de tráfico

```
#Indica los segundos antes de que el núcleo salte a otra ruta porque
la anterior ha muerto. Por defecto 300. echo "180 /proc/sys/net/ipv4/route/gc_elas
```

```
#Es el tiempo en el que el kernel declara una ruta como muerta.
En función de este parámetro y gc_interval ajustaremos los tiempos
para que el kernel deje de utilizar una ruta cuando esta cae o rein-
tenta utilizarla.
```

```
echo "0 /proc/sys/net/ipv4/route/gc_timeout
```

```
#Retrasos para vaciar la caché de rutas
#echo "0 /proc/sys/net/ipv4/route/max_delay
```

```
#Tamaño máximo de la caché de rutas. Las entradas viejas se pur-
gan una vez que la caché alcance este tamaño.
```

```
echo "4096 /proc/sys/net/ipv4/route/max_size
```

```
echo configuración_iptables: Configuración ha sido realizada correc-
tamente"
}
```

```
# Funciones globales, dentro de las ALIX
```

```
configurar_colas(){
echo " — Iniciando con la configurando colas AC_VO —"
/sbin/iwpriv $1 cwmin 3 0 5
/sbin/iwpriv $1 cwmin 3 1 5
/sbin/iwpriv $1 cwmax 3 0 10
/sbin/iwpriv $1 cwmax 3 1 10
/sbin/iwpriv $1 aifs 3 0 2
/sbin/iwpriv $1 aifs 3 1 2
/sbin/iwpriv $1 txoplimit 3 0 64
/sbin/iwpriv $1 txoplimit 3 1 64
/sbin/wlanconfig $1 list wme
echo " — Colas configuradas correctamente —"
}
```

```
desactivar_atheros(){
echo " — Desactivando atheros —"
/sbin/iwpriv $1 ff 0
```



```

/sbin/iwpriv $1 xr 0
/sbin/iwpriv $1 ar 0
/sbin/iwpriv $1 burst 0
echo “— Desactivado atheros correctamente —”
}
  #Otras funciones para realizar el bonding

  # testeo_iw: recibe dos parametros (ath0 y ath1), tiene que existir
una interfaz en modo managed y otra Master
testeo_iw(){
MODO_MANAGED=
MODO_MASTER=
echo ”— testeo_iw: entrando a testear las interfaces inalámbricas”
if [ $# -ne 2 ]; then
echo “testeo_iw: ERROR: Tiene que existir 2 redes tarjetas inalám-
bricas”
exit 1
fi
for ATH in $@;
do

  MODO_ATH=$(iwconfig $ATH |grep Mode | tr -s |cut -d -f2)
if [ ”$MODO_ATH-= ”Mode:Managed”]; then
MODO_MANAGED=${MODO_ATH}
elif [ “$MODO_ATH-= ”Mode:Master”]; then
MODO_MASTER=${MODO_ATH}
else
echo ”testeo_iw:Error: Es necesario que la interface este en mode=Managed
o mode=Master”
exit 1
fi
done if [ “$mode_Managed-=”Modo:Manageda ”$mode_Master-=”Mo-
do:Master”]; then

  echo ”Testeo_iw: Correcto : existe los modos: ${MODO_MANAGED},
${MODO_MASTER}”
else echo “Testeo_iw:Error: Las interface tiene que trabaje en mo-
de=Managed y la otra en mode=Master.”
exit 1

```

A. Automatización de configuración del prototipo para bonding y el enrutamiento de control de tráfico

```
fi
}
#obtenemos la mac de la interface que trabaja en modo managed.
Las interfaces utilizadas son ath0 y ath1
mac_managed(){
#al menos tiene que existir una interfaces, en nuestro caso recibe
dos parametros ath0 y ath1. Como resultado nos muestra la Mac del
equipo que trabaja en modo manager

    if [ $# -lt 1 ]; then
echo "mac_managed: Introduzca las interfaces inalambricas"
exit 1
fi

    for ATH in $@;
do MODO_ATH=$(iwconfig ${ATH} |grep Mode |cut -d " " -f11)
if [ "${MODO_ATH}" = "Mode:Managed" ]; then
MAC_MANAGED=$(ifconfig ${ATH} | grep HWaddr | cut -d " "f
11)
echo "${ATH}: ${MAC_MANAGED}"
fi
done
}
#Si la interface bond esta activo, desactivamos esa interface
desactivar_bonding(){
VAR_BOND0=$(ifconfig $BOND0 | grep $BOND0 |cut -d -f1)
echo "${VAR_BOND0}"
echo configurando_bonding: Verificando la existencia de la interface
bond e interfaces esclavas"
if test "${VAR_BOND0}" = "bond0"
then
ATH0_slave=$(ifconfig ${ATH0} | grep SLAVE | tr -s |cut -d -f5)
ATH1_slave=$(ifconfig ${ATH1} | grep SLAVE | tr -s |cut -d -f5)
if [ "${ATH0_slave}" = "SLAVE" ]; then
ifenslave -d ${BOND0} ${ATH0}
fi
if [ "${ATH1_slave}" = "SLAVE" ]; then
ifenslave -d ${BOND0} ${ATH1}
fi
fi
```

```

    ifconfig ${BOND0} down
    echo "configurando_bonding: Interface bond0 activo, pero se a des-
    activado"
    else
    echo "configurando_bonding: Interface bond0 desactivado"
    fi

}

```

```

    #Se cambia la MAC de la interface Master, con la mac obtenida
    de la interface Managed
    cambiar_mac_master(){
    for ATH in $@;
    do Modo=$(iwconfig ${ATH} |grep Mode |cut -d -f11)
    if [ "$Modo=" "Mode:Master"]; then
    ifconfig ${ATH} down
    ip link set address ${MAC_MANAGED} dev $ATH
    ifconfig ${ATH} up
    fi
    done
    }
    configuracion_atheros(){
    desactivar_atheros ${ATH0}
    desactivar_atheros ${ATH1}
    configurar_colas ${ATH0}
    configurar_colas ${ATH1}
    testeo_iw ${ATH0} ${ATH1}
    }

```

```

    # main recibe $1= modo de configuración
    ejecutar $1

```


B. Automatización de las pruebas de tráfico

Esté script, esta disponible en:

C. Resultados de bonding en modo 2

Los resultados estan almacenados en el CD en la direccion: estadistica/MODO2_1F, cuando se envía datos en una sola dirección y estadistica/MODO2_2F, cuando se envían datos a dos redes diferentes.

D. Resultados de bonding en modo 3

Los resultados estan almacenados en el CD en la direccion: estadistica/MODO3_1F, cuando se envía datos en una sola dirección y estadistica/MODO3_2F, cuando se envían datos a dos redes diferentes.

E. Resultados de bonding en modo 1

Los resultados estan almacenados dentro del CD.

Resultados para el primer escenario, del envío de paquetes por un flujo son:

Pruebas de bonding en un sólo flujo: estadistica/BOND1F/

Pruebas de iptables en un sólo flujo: estadistica/IP1F_2F

Resultados para el segundo escenario, del envío de paquetes por dos flujos son:

Pruebas de bonding en dos flujos: estadistica/BOND2F/

Pruebas de iptables en dos flujos: estadistica/IP2F_4

Resumen de las pruebas, tanto de bonding e iptables.

Resultados de la comparativa del bonding e iptables: estadistica/Resumen.ods

Dentro de éstas carpetas o archivos se encuentran todo el resumen de todas las pruebas efectuadas.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software
Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of
this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can

be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format

whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, eit-

her commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.

E. Resultados de bonding en modo 1

- I. Preserve the section Entitled “History”, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled “History” in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the “History” section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has

been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual

copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever

published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3

or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with . . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES,
with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.