UNIVERSIDAD AUTÓNOMA METROPOLITANA
UNIDAD AZCAPOTZALCO

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA
LICENCIATURA EN INGENIERÍA EN COMPUTACIÓN

Proyecto terminal

Implementación de una interface gráfica de un router en un sistema empotrado.

Trimestre 13-I

Luis David Avendaño Lopez
204358546

Asesor: Arturo Zuñiga Lopez

INDICE

Página

RESUMEN

El presente trabajo de investigación tiene como objetivo primordial implementar un equipo de enrutamiento en un sistema empotrado.

Para lograr este objetivo, será necesario realizar una serie de pasos que culminarán finalmente en la contrucción de un sistema operativo de proposito específico, esto es, un sistema que solo tendrá la finalidad de realizar tareas relacionadas a procesos de comunicación entre redes de computadoras.

En la actualidad es común encontrar sistemas empotrados, en casi cualquier electrodomestico o aparato electrónico, ya sea una lavadora, refrigerador, automoviles o hasta los más comunes sistemas empotrados de la actualidad, los telefonos celulares inteligentes.

Todos estos dispositivos poseen es sistema similar al que se pretende desarrollar en este proyecto, es un sistema que ocupa muy pocos recursos y que realiza tareas muy especificas, a diferencia de las computadoras de escritorio, que pueden ejecuta un sin fin de actividades.

Estos sistemas de procesamiento de datos son de gran relevancia en la vida del ser humano, realizando tareas de manera automática y sin necesidad de supervisión.

En la actualidad este tipo de equipos se han abaratado a tal grado que es posible adquirirlos por tan solo unos cientos de dolares, esto a propiciado que los procesos de manofatura de industrial, sea mas eficientes y menos costos.

En base a lo comentado con anterioridad, en este proyecto se desarrollará un router, utilizaremos una tarjeta de sistema empotrado para llevar a cabo este proyecto, la tarjeta se adquirio a muy bajo costo, en comparación a otros dispositivos,

INTRODUCCIÓN

Un sistema empotrado es una forma procesar, organizar o realizar una o múltiples tareas de acuerdo a un plan, programas o reglas predeterminadas. Este sistema ha sido definido en libros y publicaciones de muchas maneras, pero quizá la definición que más se apega al proyecto es la siguiente: “Un sistema embebido es un dispositivo que cuenta con un procesador, que posee características especiales, que permite realizar múltiples procesos dentro del sistema”.

El presente proyecto busca desarrollar un router con características similar a los disponibles en el mercado; comercializados por empresas como Linksys y D-Link, los cuales son dispositivos de mediana complejidad, que cumplen con las funciones necesarias para poder brindar un servicio de interconexión.

OBJETIVO GENERAL

Diseñar e implementar la interfaz gráfica de un router que permita administrar una red de computadoras, a través de un sistema empotrado.

OBJETIVOS PATICULARES

1. Crear los scripts necesarios para la manipulación de los servicios del router.
2. Diseñar una interfaz gráfica que permita reconfigurar los servicios del router, a través de un explorador web.
3. Programar e implementar la interfaz gráfica en un entorno web.

JUSTIFICACIÓN

Esta propuesta de proyecto terminal busca implementar un router[1] que cumpla con las funciones necesarias para administrar una red de computadoras, para ello se valdrá de un sistema operativo de libre distribución así como de las herramientas necesarias (Programas) para dicha actividad.

El hecho de realizar la instalación de un sistema operativo más “robusto” (completo) al que ya dispone de fabrica, hace que el proyecto tenga mayor relevancia lo cual se traduce en una mayor funcionalidad por parte del router. Esto supone un avance importante, debido a que se deja el camino libre para explotar con mayor facilidad las cualidades del hardware en el cual se planea trabajar.

Los alumnos de la carrera de Ingeniería en Computación, así como los de Electrónica, han realizado algunos trabajos similares, la idea es tomar algunos conceptos utilizados en proyectos realizados con anterioridad por estos estudiantes e incluirlos en el dispositivo; por lo que este proyecto desarrollará nuevas funciones que enriquecerán tanto los trabajos anteriores como el que se intenta realizar.

1 Router, enrutador o encaminador de paquetes es un dispositivo que proporcióna conectividad a nivel de red o nivel tres en el modelo OSI. Su función principal consiste en enviar o encaminar paquetes de datos de una red a otra.

Debido a lo basto del tema, este proyecto podra ser en un futuro tomado como punto de partida para algunos otros trabajos de investigación y proyectos terminales.

De las posibles líneas de investigación que se podrían seguir una vez concluido el presente trabajo, sería el diseño e implementación de mecanismos de enrutamiento más complejos. Otro podría ser el hecho de tener conexiones de forma segura vía sockets[2], implementar un firewall[3], implementar y desarrollar algoritmos de seguridad o bien de encriptación[4] de información o poner en operación un segmento DMZ[5] de alta seguridad. La figura 1 muestra la distribución física del DMZ.

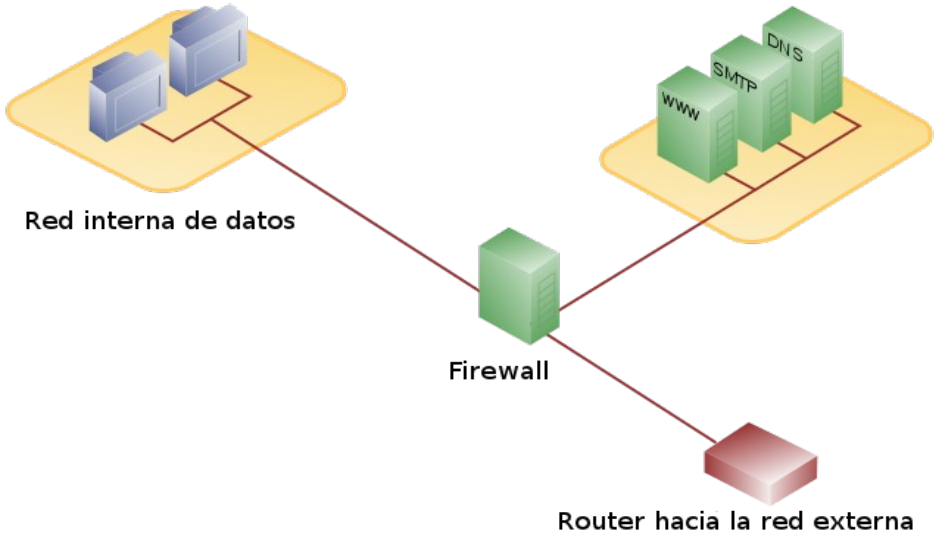


Fig. 1 Esquema de la DMZ o zona desmilitarizada

ESTADO DEL ARTE

El router o enrutador es el dispositivo encargado de transmitir y recibir paquetes de datos (figura 3), que son dirigidos de un segmento de red a otro. El router es conectado a dos o más lineas de datos que componen las dos o más subredes que el router conoce, por ende el router deberá esta equipado con al menos dos dispositivos de red independientes.

Los routers son considerados equipos de capa 3 dentro del modelo OSI[6], su función primaria es el

2 Concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

3 Mecanismo de protección de las redes de computadoras que puede estar basdado en software o en hardware, su principal función es la permitir o negar la comunicación entre dos segmentos de redes a traves de reglas que son configuradas para realizar este trabajo.

4 Proceso de codificación de mensajes (Información), para evitar la información que se transporta en una comunicación sea observada por personas que no esten autorizadas ver dicha información

5 Sub red física o lógica, que contiene los servicios informáticos expuestos de una organización, principalmente a la internet, esto con el fin de proporcionar una mayor seguridad a la red interna y prevenir una ataque cibernético desde el exterior.

6 El modelo OSI, es un modelo de red descriptivo creado por la Organización Internacional para la Estandarización, en el año 1984, es un marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones

envío/recepción de paquetes basados en el protocolo IP[7], especificamente este paquete tiene como dato principal una dirección ip origen y una dirección ip destino. El proceso mediante el cual el enrutador o router sabe hacia cual subred se dirige un paquete determinado, es llamado enrutamiento. La figura 2 muestra con un rectangulo de color morado la ubicación de los enrutadores dentro del modelo OSI, además se identifica la dirección del flujo de los datos entre cada una de las capas del modelo[7].

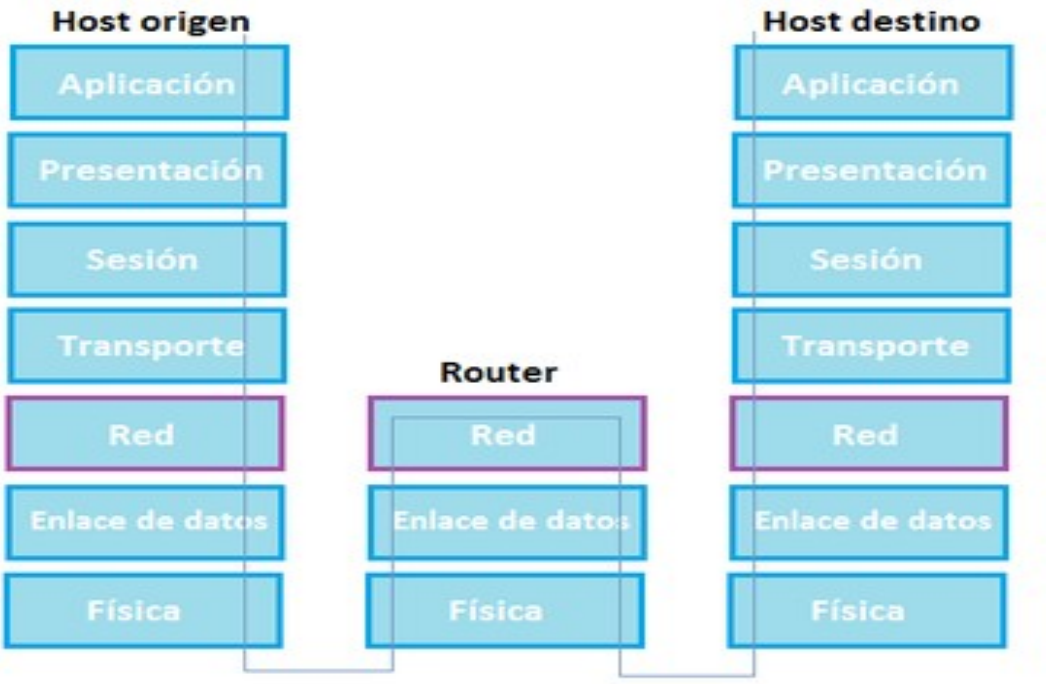


Fig. 2 Ubicación de los enrutadores en el modelo OSI

En la arquitectura de un router podemos encontrar los siguientes elementos[8]:

- Puertos de entrada: realiza las funciones de la capa física consistentes en la terminación de un enlace físico de entrada a un router; realiza las funciones de la capa de enlace de datos necesarias para interoperar con las funciones de la capa de enlace de datos en el lado remoto del enlace de entrada; realiza también una función de de búsqueda y reenvío de modo que un paquete renviado dentro del entramado de conmutación del router emerge en el puerto de salida apropiado.

- Entramado de conmutación: conecta los puertos de entrada del router a sus puertos de salida.

- Puertos de salida: almacena los paquetes que le han sido reenviados a través del entramado de conmutación y los transmite al enlace de salida. Realiza entonces la función inversa de la capa física y de la capa de enlace que el puerto de entrada.

7 Es el principal protocolo de comunicación basado en paquetes de datos, encargado tambien del encapsulamiento de los paquetes y de anexion de direcciones IP, con lo cual se puede enrutar un paquete hacia su destino final.

- Procesador de encaminamiento: ejecuta los protocolos de encaminamiento, mantiene la información de encaminamiento y las tablas de reenvío y realiza funciones de gestión de red dentro del router.

Tipos de enrutadores

Los enrutadores pueden proporcionar conectividad dentro de las empresas, entre las empresas e Internet, y en el interior de proveedores de servicios de Internet (ISP). Los enrutadores más grandes (por ejemplo, el Alcatel-Lucent 7750 SR) interconectan ISPs, se suelen llamar metro enrutador, o pueden ser utilizados en grandes redes de empresas

Conectividad Small Office, Home Office (SOHO)
Los enrutadores se utilizan con frecuencia en los hogares para conectar a un servicio de banda ancha, tales como IP sobre cable o ADSL. Un enrutador usado en una casa puede permitír la conectividad a una empresa a través de una red privada virtual segura.

Si bien son funcionalmente similares a los enrutadores, los enrutadores residenciales usan traducción de dirección de red en lugar de direccionamiento.

En lugar de conectar dispositivos locales a la red directamente, un enrutador residencial debe hacer que los dispositivos locales parezcan ser un solo equipo.

Enrutador de empresa
En las empresas se pueden encontrar enrutadores de todos los tamaños. Si bien los más poderosos tienden a ser encontrados en ISPs, instalaciones académicas y de investigación, pero también en grandes empresas.

Acceso
Los enrutadores de acceso, incluyendo SOHO, se encuentran en sitios de clientes como sucursales que no necesitan de enrutamiento jerárquico de los propios. Normalmente, son optimizados para un bajo costo.

Distribución
Los enrutadores de distribución agregan tráfico desde enrutadores de acceso múltiple, ya sea en el mismo lugar, o de la obtención de los flujos de datos procedentes de múltiples sitios a la ubicación de una importante empresa. Los enrutadores de distribución son a menudo responsables de la aplicación de la calidad del servicio a través de una WAN, por lo que deben tener una memoria considerable, múltiples interfaces WAN, y transformación sustancial de inteligencia.

También pueden proporcionar conectividad a los grupos de servidores o redes externas. En la última solicitud, el sistema de funcionamiento del enrutador debe ser cuidadoso como parte de la seguridad de la arquitectura global. Separado del enrutador puede estar un *firewall* o *VPN concentrador,* o el enrutador puede incluir estas y otras funciones de seguridad.Cuando una empresa se basa principalmente en un campus, podría no haber una clara distribución de nivel, que no sea tal vez el acceso fuera del campus.

En tales casos, los enrutadores de acceso, conectados a una red de área local (LAN), se

interconectan a través de la red dorsal o backbone[8].

El enrutamiento se realiza leyendo el encabezado de cada paquete recibido, una vez realizado este proceso el enrutador busca dentro de su configuración de enrutamiento, la tabla que más se asemeja a la dirección ip destino de ese paquete, eso genera una asociación entre la tabla de enrutamiento y la interface física del enrutador, por lo que el enrutador envía ese paquete por esta interface física a su destino.

Núcleo

Los enrutadores que componen la red dorsal o *backbone* se encuentran interconectandos a la capa de distribución, estos equipos tienden a ser optimizados para ancho de banda de gran capacidad.

Cuando una red de computadoras está ampliamente distribuida sin ubicación central, la función del *backbone* puede ser asumido por el servicio de WAN de dicha red, y la distribución de enrutadores se convierte en el nivel más alto[3].

Borde

Los enrutadores de borde enlazan sistemas autónomos con las redes troncales de Internet u otros sistemas autónomos, tienen que estar preparados para manejar el protocolo BGP y si quieren recibir las rutas BGP, deben poseer una gran cantidad de memoria.

Enrutadores inalambricos

En los últimos años han comenzado a aparecer enrutadores que permiten establecer una conexión entre redes fijas y móviles (Wi-Fi, GPRS, Edge, UMTS, Fritz!Box, WiMAX...) Un enrutador inalámbrico comparte el mismo principio que un enrutador tradicional. La diferencia es que éste permite la conexión de dispositivos inalámbricos a las redes mediante una conexión aerea o de radiofrecuencia.

Los enrutadores inalambricos tiene diferentes denominaciones de clase, tales como a/b/g/ y n que son asignadas en función de su velocidad de transmision y el alcance de la señal de radiofrecuencia.

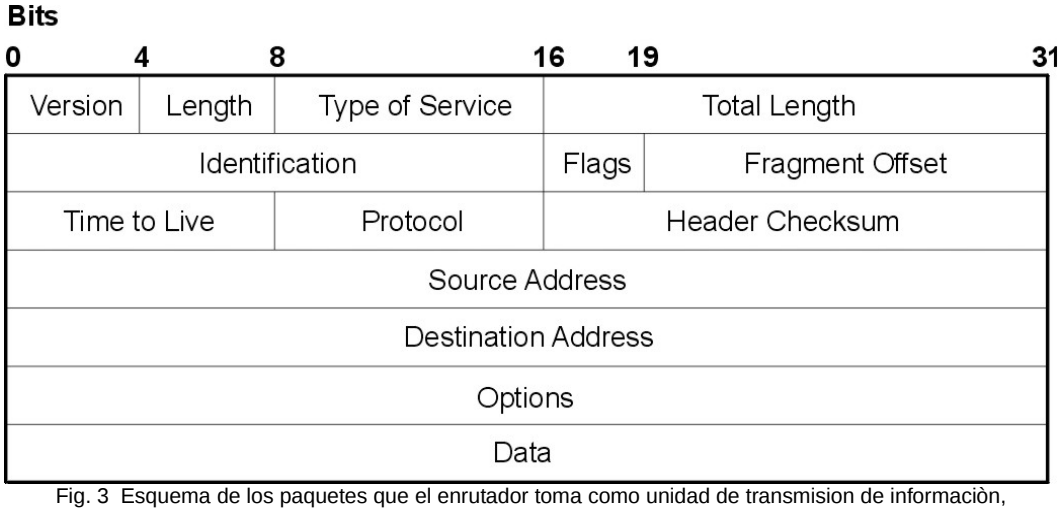


Fig. 3 Esquema de los paquetes que el enrutador toma como unidad de transmision de informaciòn, se muestra claramente los campos utlizados

8 La palabra backbone se refiere a las principales conexiones troncales de uan red, muchas veces compuesta de un gran número de routers

En la actualidad los routers utilizan protocolos de enrutamiento dínamico, esto les permite tener actualizadas las tablas de enrutamiento, lo que facilita considerablemente la administración de redes muy extensas, sin embargo el costo de tener actualizadas estas tablas así como el ahorro en la configuración, repercute directamente en el desempeño de los equipos, ya que tienen que desviar recursos de procesamiento para realizar los anuncios o envio/recepción de las tablas con las rutas de cada equipo enrutador.

En la actualidad se utlizan diferentes protocolos de enrutamiento dinámico, a continuación enumeramos algunos de ellos con sus caracteristicas mas relevantes[2]:

- RIP: (Routing Information Protocol), protocolo vector distancia, utiliza los saltos de enrutador en enrutador como métrica[9] para seleccionar un camino del origen al destino.

- RIP v.2: (Routing Information Protocol Version 2), este protocolo fue diseñado en el año 1993, se le realizaron adecuaciones y mejoras a la primera versión, lo que dio paso a la versión 2 de RIP.

- OSPF: (Open Shortest Path First), protocolo de estado del enlace, es considerado tambien un protocolo IGP o Interior Gateway Protocol.

- IS-IS: (Intermediate System - Intermediate System), protocolo de estado del enlace, es considerado tambien un protocolo IGP o Interior Gateway Protocol. Tanto OSPF como IS-IS utilizan el algoritmo de Dijkstra, para buscar el mejor camino a traves de la red.

- EIGRP: (Enhanced Interior Gateway Router Protocol), protocolo patentado por Cisco, basado en su antecesor IGRP, es un protocolo vector-distancia optimizado. Este protocolo calcula su metrica utilizando los valores de ancho de banda, retardos, confiabilidad, unidad maxima de transmision y conteo de saltos.

- BGP: (Border Gateway Protocol), protocolo utilizado en la actualizadad con mayor difución para la Internet, es el encargado del enrutamiento en la Internet. Es considerado un protocolo path-vector, que es una variante de los protocolos de vector-distancia. Este protocolo de enrutamiento deberá ir montado sobre EIGRP de lo contrario no funcionará.

Un router puede prestar multiples servicios dentro de una red de computadoras, dependiendo que tan flexible sea su sistema operativo, por ello en la actualidad, no es raro encontrar un router posea funciones que antiguamente estaban destinadas a otro tipo de dispositivos de red, gran parte de la integración de estas nuevas funciones se debe al constante avance de la industria tecnológica, lo cual trae como consecuencia que se tenga mas capacidad de almacenamiento, mayor procesamiento de datos, mayor rapidez a la hora de realizar tranferencias, etc.

Todas estas funciones aunque ventajosas, impactan directamente en el costo de los equipos comerciales, debido a este problema de costos se decidio que para este proyecto se utilizaría la versión del sistema operativo Linux que tenga la mayor cantidad de servicios disponibles y que la

9 Una métrica es cualquier medida o conjunto de medidas destinadas a conocer o estimar el tamaño u otra característica de un software o un sistema de información, generalmente para realizar comparativas, en redes es utilizada para determinar un mejor camino de un punto a otro

tarjeta pueda manejar sin comprometer la operación de la misma, ya que disponemos de recursos limitados. Con esto lograremos reducir drasticamente el total del costo de un equipo comercial de la actualidad.

OpenWrt[10] es la distribución que cumple con los minimos requerimientos para realizar el proyecto, este sistema operativo nace en el año 2004, esta distribución de Linux fue diseñada desde el principio para dispositivos de red, lo que facilita enormemente la implementación de los servicios que seran prestados por el equipo enrutador de este proyecto.

El desarrollo de OpenWrt fue impulsado inicialmente gracias a la licencia GPL , que obligaba a todos aquellos fabricantes que modificaban y mejoraban el código, a liberar éste y contribuir cada vez más al proyecto en general. Poco a poco el software ha ido creciendo y se encuentran características implementadas que no tienen muchos otros fabricantes de dispositivos comerciales para el sector no profesional, tales como QoS[11], VPN[12] y otras características que dotan a OpenWrt y hacen de un dispositivo enrutador, un sistema realmente potente y versátil. OpenWrt no sólo puede ser utilizado como enrutador, sino que ademas puede realizar funciones tales como las de servidores de archivos, nodos P2P[13], servidores de webcams, firewall[14] o puertas de acceso VPN.
No obstante antes del año 2004 esta distribución de Linux no había sido liberada para el uso de cualquier persona, OpenWrt que esta basado en la robusta distribución Debian Linux, fue especificamente construido para los equipos de la empresa Linksys cuyas arquitecturas de dispositivos son los Broadcom de la serie BCM63xx, estos dispositivos dieron origen a los enrutadores Linksys de la serie WRT54g mostrado en la figura 4 y sus derivados.

10 OpenWrt se describe como una distribución de Linux especializada en firmware, para dispositivos empotrados.
11 (**Q**uality **o**f **S**ervice, en inglés) son las tecnologías que garantizan la transmisión de cierta cantidad de información en un tiempo dado (throughput). Es especialmente importante para ciertas aplicaciones tales como la transmisión de vídeo o voz.
12 (**V**irtual **P**rivate **N**etwork en inglés) es una tecnología de red que permite una extensión segura de la red local sobre una red pública o no controlada.
13 Una red **p**eer-**t**o-**p**eer, (**P2P**, por sus siglas en inglés) es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red.
14 cortafuegos (firewall en inglés) es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas. Se trata de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

Aunque los enrutadores Linksys de la serie WRT54g estan dotados de esta distribución del sistema operativo, no tienen habilitados todos lo servicios disponibles, ya sea por que no fueron precargados o por que no se tiene forma de configurarlos a traves de su interface web.

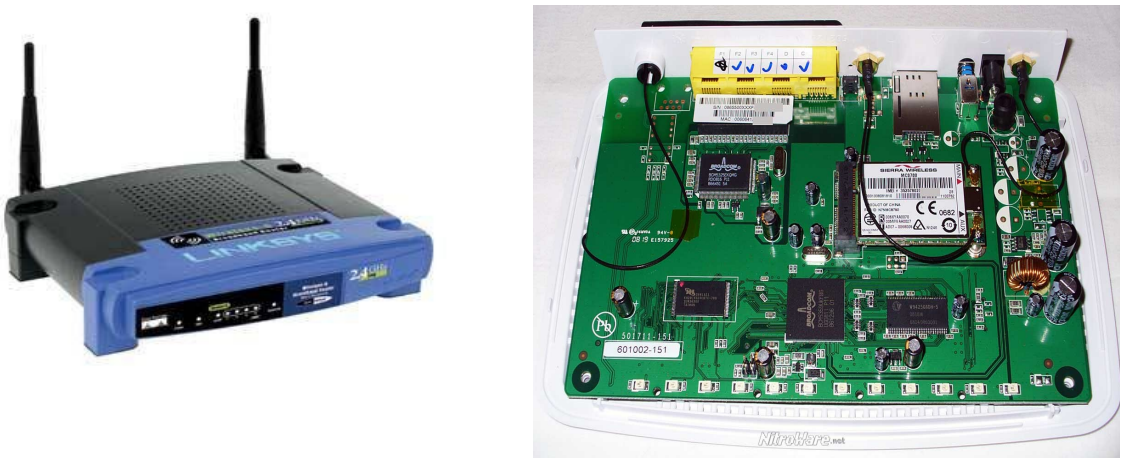


Fig. 4 Enrutador linkSys Series Wrt54g y placa madre BroadCom

Una ves liberada la primera versión del OpenWrt se comenzo a dar soporte a las diferentes plataformas en las que hoy en dia puede funcionar, algunas de ellas son las siguientes:

- AMCC/IBM PPC40x
- AMCC/IBM PPC44x
- Atheros AR231x/AR5312
- Atheros AR71xx/AR7240/AR913x Atmel AVR32
- Broadcom BCM63xx
- Marvell Orion
- RDC 321x
- TI AR7
- Broadcom BCM947xx/953xx
- Infineon/ADMtek ADM5120
- Ingenic XBurst
- Intel IXP4xx
- Lantiq GPON/XWAY
- Marvell Kirkwood
- Mikrotik RouterBoard 532
- RMI/AMD AU1x00
- x86

El mercado de las telecomunicaciones dispone de un buen número de compañias que proveen a sus clientes de una gran variedad de dispositivos de enrutamiento que van desde los más sencillos con prestaciones modestas, hasta los dispositivos más complejos y con un muy alto rendimiento en funcionamiento.

Cisco Systems (NASDAQ: CSCO) con sede en San José, California, se perfila como la más grande en el ramo. Seguida muy de cerca por su competidor directo Juniper NYSE: JNPR que tambien se encuentra en el ramo de las telecomunicaciones.

Actualmente Cisco mantiene una división de routers para redes domesticas, pequeñas y medianas empresas , esta gama de productos, no posee altas prestaciones, por lo que se encuentran limitados a la hora de realizar ciertas actividades especificas de un router, ejemplo de ello seria, que algunos protocolos de enrutamiento no estuvieran presentes, o que algunas funciones tales con SSL, QoS o VPN estuvieran limitadas o de igual forma ausentes.

Es de señalar que mientras Cisco se enfoca en equipos que tienen muy alto rendimiento, poseen sus propios protocolos y tienden hacia el lado del enrutamiento y la conmutación, Juniper apuesta más con equipos que brinden protección contra los ataques así como la detección de intrusos.

Hasta este punto se han descrito, las ventajas y desventajas de estos equipos a nivel de hardware, por tal motivo será necesario que tambien se revise lo conceniente al equipamiento lógico, esto involucra directamente al sistema operativo, parte importante del funcionamiento de estos dispositivos.

Cisco después de adquirir la compañia Linksys, opta por mantener el equipamiento lógico, como se manejaba antes de que Linksys fuera absorbida por Cisco, se tiene una distribución de linux llamada OpenWrt desarrollada expresamente para funcionar el tarjetas de compañias como BroadCom, Asus, Atheros, AVR, etc.

Inicialmente este sistema no estaba liberado para la comunidad en general, pero debido a diversos factores, se decide que el año 2004 sea lanzada a la comunidad el sistema operativo OpenWrt con la version 7.02.

Cisco con esto evita tener que equipar a estos dispositivos, de su sistema operativo, el IOS[15], esto conlleva a la ventaja, no se tiene que realizar ingenieria alguna para poder adaptar los sistemas de IOS de Cisco ya existentes, a los nuevos dispositivos de gama baja o bajo rendimiento.

Juniper sin embargo desarrolla dispositivos que vienen dotados de su sistema operativo llamado JunOS[16], este sistema fue desarrollado por ellos, y esta basado en una versión de unix freeBSD que es un clon de la version BSD Unix[4].

Dentro de los aspectos más importantes del sistema operativo JunOS tenemos las siguientes característcas:

- Experiencia en el área de enrutamiento y de protocolos de enrutamiento.
- Software de alta modularidad.

15 **I**nternetwork **O**perating **S**ystem, es el software utilizado en la gran mayoría de routers y switches de Cisco Systems . IOS es un paquete de funciones de enrutamiento, conmutamiento, trabajo de internet y telecomunicaciones que se integra estrechamente con un sistema operativo multitarea.

16 Sistema operativo que usa Juniper Networks, esta basado en la sitribucion FreeBSD, que a su ves deriva de un clon de Unix.

- Politicas de control y enrutamiento flexible.
- Sistema operativo independiente del hardware, cosa que el IOS de Cisco no tiene.
- Acceso al *kernel*[17] JunOS a traves del un shell de tipo Unix.

El futuro de los routers apuesta por equipos avanzados en capacidades, funciones y servicios tales como el streaming, sistemas VOIP, sistemas NAS, etc. Estos equipos seran capases de almacenar grandes volumenes en su interior, asi como prestar los servicios de: servidores de contenidos para el hogar, servidores de descargas, servidores web, sistema de copias de seguridad, etc.

Muchos de los fabricantes de los routers miran constantemente a otras disciplinas, en busca de nuevas formas de optimizar su funcionamiento, tal es el caso de la Universidad de Florida[9]. La academia de ciencias de la computación de esa Univeridad pretende implementar un modelo de comunicación basado en el comportamiento de una colonia de hormigas, pudiendo así, modelar a traves de inteligencia artificial, las comunicaciones entre redes y de esa manera eficientar y mejorar los modelos que ahora se aplican.

17 Es el software que constituye la parte más importante del sistema operativo, ademas es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

METODOLOGÍA

La metodología que se siguio para alcanzar los objetivos, que fueron fijados con anterioridad para este proyecto, es la siguiente:

- Se realiza una investigación previa para saber en que otros proyectos, se ha utilizado el equipamiento, tanto físico como lógico del proyecto, ademas de tomar en cuenta los comentarios, sugerencias y notas de las fuentes que documentaron a la hora de realizar sus proyectos.

- Se establecen de antemando tres posibles soluciones para llevar a cabo de manera más facil, ordenaday sencilla la instalación y manipulación del router, aún cuando no se presentan como las mejores opciones.

- Se establece una cuarta opción, que sustenta mejor la comunicación y funcionalidad, e incluso se utiliza en equipos que se encuentran en producción en la actualidad.

- Se lleva a cabo el registro de los pasos, de la cosntrucción de los sistemas operativos y paqueteria que incluira la version final, en una bitacora con interface web, esto debido a que en el transcurso de la cosntrucción se hicieron notar varios detalles que son de gran relevancia si se quiere construir una nueva version a partir del buildroot de OpenWrt.

- Se construyen multiples versiones del sistema operativo, estas fueron probadas en la placa NGW100 para saber si cumplian con los requerimientos minimos para cubrir los objetivos de este proyecto.

    - Se fijan las caracteristicas minimas que debera tener la imagen del *kernel* para poder arrancar la placa desde los dispositivos secundarios.

- Se realizan las siguientes pruebas de de rendimiento al equipo ya corriendo la ultima versión del *kernel*:
    - Pruebas al *firewall* (IPTABLES[18]), que incluyen ataques de tipo DDoS, escaneo de puertos, etc
    - Prueba de protocolos de enrutamiento, que incluyen pruebas con equipos Cisco y con otra tarjeta del mismo tipo con el mismo sistema operativo.
    - Pruebas a los servicios.

- De lo anterior se puede decir que el resultado fue positivo, en general y cumplen con las expectativas de este proyecto.

18 Herramienta de firewall que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red (NAT) para IPv4 o mantener registros de log.

DESARROLLO

INVESTIGACIÓN PREVIA AL DESARROLLO

Para poder realizar este proyecto, se llevo a cabo el estudio de factibilidad de diversas tarjetas de desarrollo, dentro de los factores que fueron tomados en cuenta para poder establecer cual era la óptima, se consideraron los siguientes puntos[5].

- Capacidad de procesamiento.
- Características físicas
- Precio.
- Consumo energetico.
- Permiso/Tiempo importación.

Depues de revisar algunas de las opciones, entre los diferentes fabricantes de sistemas empotrados, se llega a la conclusión, que este proyecto, lo realizariamos con la tarjeta NGW100 de la empresa Atmel[19], cuyo microcontrolador es el AVR32[20].

Un de los puntos fuertes, y por los que fue elegida la tarjeta NGW100, es que posee la capacidad de iniciar un sistema operativo desde un dispositivo externo, ya sea este un disco duro de tipo usb, desde el slot de la memoria SD[21] o via servidor de tftp[22] a traves de las tarjetas de red que posee el equipo.

El analisis previo al desarrollo del proyecto consta de los siguientes puntos, que fueron tomados en cuenta para brindarnos un panorama amplio del funcionamiento de la misma:

- Arquitectura de la tarjeta
- Sistema de arranque (Boot)
- Variables de entorno
- Sistema operativo
- Sistema de archivos
- Memorias SD

Arquitectura de la tarjeta

Como se menciono previamente la tarjeta que fue elegida para realizar este proyecto terminal, fue la NGW100 de la compañia de microcontroladores y microprocesadores Atmel, esta tarjeta esta dotada de un microcontrolador propietario de Atmel, que es el AT32AP7000, tiene una velocidad de procesamiento de 150Mhz, ademas de contar con 8 Mb de memoria flash, 8 Mb de memoria flash, 32 Mb de memoria MD SDRAM, 2 tarjetas de red y una ranura de expansion de memorias SD, la figura 5 muestra el esquema de la arquitectura de la tarjeta.

El microcontrolador es de 32 bits, la versión es la Rev2, la arquitectura es de tipo RISC[5], big endian, este microcontrolador , cuenta con 15 registros de trabajo. El set de instrucciones de este

19 Es una compañía de semiconductores, fundada en 1984. Su línea de productos incluye microcontroladores.

20 Es una familia de microcontroladores de tipo RISC del fabricante estadounidense Atmel.

21 **S**ecure **D**igital (SD) es un formato de tarjeta de memoria inventado por Panasonic.

22 Siglas de **T**rivial **f**ile **t**ransfer **P**rotocol(Protocolo de transferencia de archivos trivial)

microprocesador consta de instrucciones compactas de 16 bits e instrucciones extendidas de 32 bits, muchas de estas son instrucciones especializadas no disponbles en otras arquitecturas tal como MIPS32, ARMv5 o ARMv6 ISA, el diseño de este procesador fue pensado para hacer de este un dispositivo extremadamente eficiente y con una muy buena respuesta.

AT32AP7000 implementa una unidad manejadora de memoria o (MMU) y un moderno y flexible controlador de interrupciones, con lo cual es capaz de soportar un moderno sistema operativo o un sistema operativo de tiempo real.

Este microcontrolador incluye un basto set de instrucciones DSP y SIMD, especificamente para aplicaciones de tipo multimedia y de telecomunicaciones.

La tarjeta tambien incorpora una memoria SRAM de alta velocidad y de acceso seguro, una controladora de memoria SDRAM (memoria volatil), una compaq flash, una multimedia flash (MMC), Secure Digital (SD)-card, SmartCard, una memoria NAND flash de Atmel, y ademas para realizar transferencias masivas de información entre la memoria central y los dispositivos de almacenamiento secundario y viceversa, la NGW100 cuenta con una controladora DMA, que evita el acceso inecesario de los datos a traves del microcontrolador.

Posee una interface de medio independiente para controlar la tarjeta de red 10/100 Ethernet MAC, asi como una implementación en hardware de una maquina virtual de java, la cual permite ejecuciones del código Java a muy alta velocidad. AVR implemento instrucciones en hardware, reutilizando los flujos de datos RISC, lo cual hace que la ejecución de los programa en Java sean por mucho, más eficientes que en otras arquitecturas.

Estas son algunas de las caracteristicas más importantes de la arquitectura del equipo, para una revisión más profunda se anexa en los entregables el documento completo emitido por Atmel.

Sistema de arranque y sistema de archivos

El sistema de arranque esta constituido por el programa Das-UBoot[23] desarrollado por DENX Software Engineering, es el principal gestor de arranque de sistema empotrados en la actualidad.
Soporta los siguientes sistemas de archivos:

- Cramfs
- ext2
- ext3
- ext4
- fat
- fdos
- jffs2
- reiserFS
- ubifs

23 Das U-Boot(Universal Bootloader) es un boot loader para un varios tipos de arquitecturas de computadores, incluyendo PPC, ARM, AVR32, MIPS,x86, 68k, Nios, and MicroBlaze. Es software libre liberado bajo la licencia GNU.

Blockdiagram

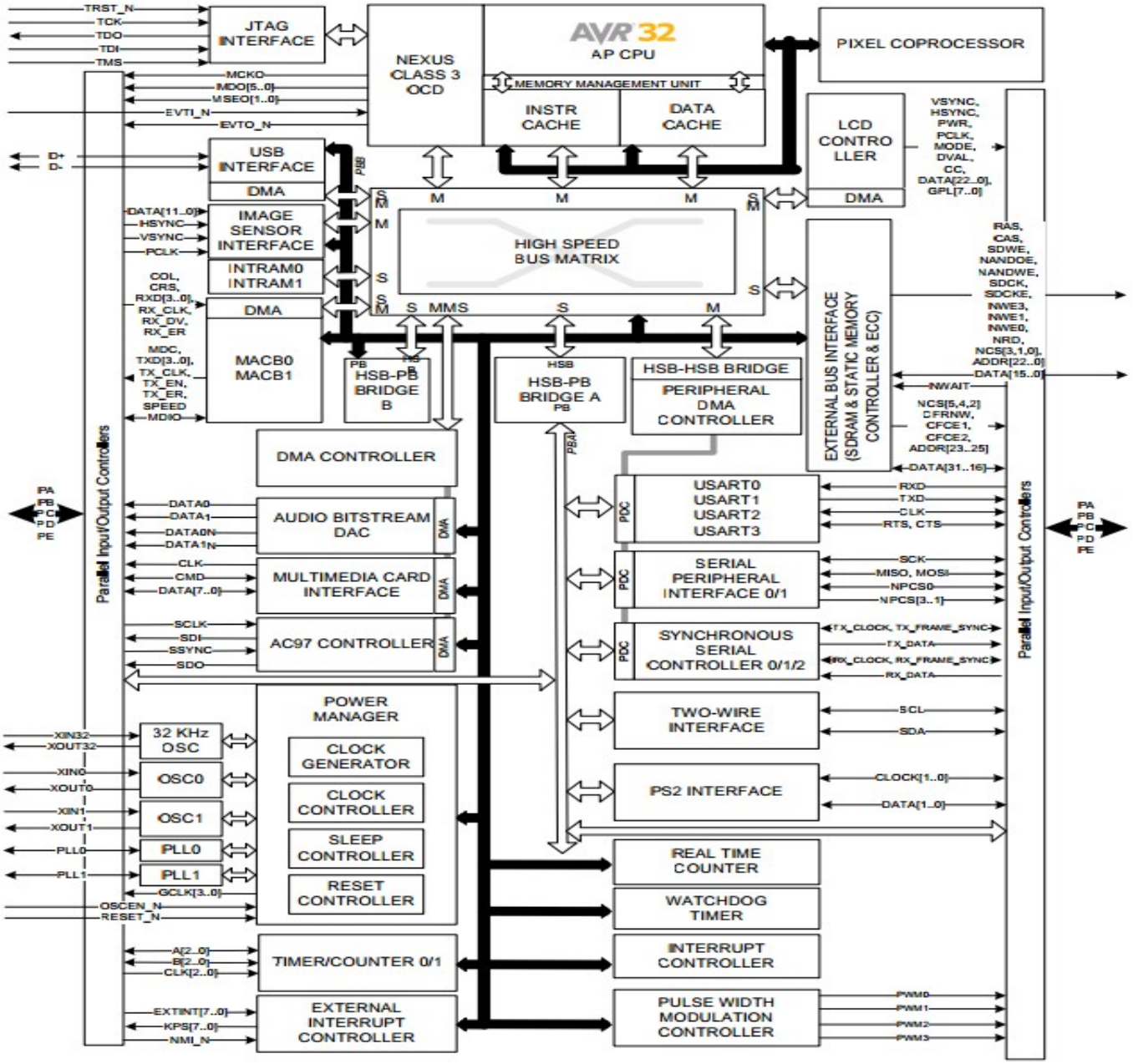


Fig. 5 Diagrama de bloques de microcontrolador Atmel AVR32.

Variables de entorno y sistema operativo

Esta tarjeta viene precargada con una version reducida del sistema operativo Linux AVR 2.6.23-rc7, basada en busybox[24], para realizar la carga por default esta tarjeta le pasa al kernel las siguentes lineas a traves del UBoot:

- setenv bootcmd 'fsload;bootm'
- setenv bootargs 'console=ttyS0,115200n8 console=tty0 root=/dev/mtdblock1 rootfstype=jffs2'

Memorias SD

El primer problema con el que se enfrenta este proyecto es el de usar una memoria de tipo SD menor a 256 Mb debido a un problema que se tiene con el gestor de arranque de la NGW100, el U-Boot. Este problema no se corrigió hasta la versión
U-Boot 2008.10 (Apr 16 2009 – 10:33:49), que es el parche que fue liberado para poder utilizar memorias de mayor capacidad, y aun así teniendo como limite memorias de 2Gb.

Se anexa una copia del parche de U-boot, en el disco de entregables del proyecto.

Aún cuando se encontro solución al problema de capacidad de las memorias, la tarjeta ngw100 de Atmel, rechaza ciertas marcas de memorias SD, aunque cumplan con los parametros de capacidad. A continuación se presenta un breve listado de las tarjetas SD que se desempeñan de manera correcta con la tarjeta de desarrollo del proyecto.

- No-name MMC 32 MB (supplied with Nokia phones)
- TwinMOS MMC 32MB
- No-name HP SD 64MB (supplied with a camera)
- Kingston SD 128 MB (stk1000 & **NGW100**)
- SanDisk SD 128 MB (stk1000)
- SanDisk SD 512 MB (**ngw100**)
- SanDisk SD 1.0 GB (**ngw100**)
- SanDisk SD 2.0 GB (**ngw100**)
- Kingston Elite Pro SD 256 MB
- Viking SD Interworks 256 MB
- TwinMOS Ultra-X SD 512 MB
- Transcend microSD 2GB (stk1000)
- Platinum 1GB SD (**NGW100**)

24 BusyBox proporciona una gran variedad de herramientas de Unix en un solo ejecutable. Se ejecuta en una variedad de entornos POSIX como Linux (incluyendo Android), FreeBSD y otros, como los núcleos de propiedad, a pesar de que muchas de las herramientas que ofrece están diseñados para trabajar con interfaces proporcionadas por el núcleo Linux.

Debido a estas restricciones, se tuvieron que adquirir memorias de diversas capacidades de la marca SanDisk, para realizar las pruebas necesarias y asi establecer cuales de ellas se prestaban para realizar el trabajo de carga del sistema operativo.

Se investigo y se llego a la conclusión que el mejor sistema de archivos para este proyecto será el ext2[25], esto se decidió de esta manera, para evitar el desgaste de la memoria-SD, ya que este sistema de archivos es de tipo ***No Journaling***[26], lo que implica que la memoria no será accedida, a no ser que sea extrictamente necesario, con ello se incrementará su periodo de vida.

Una vez con la información reunida, y con los elementos mínimos necesarios para realizar el proyecto, comenzamos con la revición del software, esto incluye la versión y distribución de Linux[27] a utilizar, las utilerias, programas y servicios.

25 Second Extended Filesystem o "segundo sistema de archivos extendido" es un sistema de archivos para el kernel Linux. Fue diseñado originalmente por Rémy Card.

26 El **journaling** es un mecanismo por el cual un sistema informático puede implementar transacciones. También se le conoce como «registro por diario». Tambien conocido como sistema de archivos transaccional.

27 Linux es un núcleo libre de sistema operativo basado en Unix. fue concebido por el finlandés, Linus Torvalds, en año 1991.

PREPARACIÓN DE LAS MEMORIAS SD

Se adquirieron 4 memorias SD de la marca SanDisk para poder montar el sistema operativo, la tarjeta de desarrollo NGW100 documenta en su página oficial[1] que tiene la función de arrancar el sistema operativo desde las memorias secundarias. Las capacidades de la memorias adquiridas son: 128Mb, 512Mb, 1Gb y 2Gb. Se formatearon y se cosntruyeron los sistemas de archivos en todas las memorias con los programas fdisk, gparted, mkfs.ext2. Y para la revisión de la integridad del sistema operativo, se utilizo el programa e2fsck.

Una vez formateadas las memorias se procedio a descargar la imagen de la dirección electrónica de The Ångström Distribution[2] para realizar la carga del sistema operativo, esta página de internet permite generar versiones precompiladas del kernel del sistema operativo Linux para poder ejecutar el software de manera rapida, y sin necesidad de realizar el tedioso proceso de compilación via un BuildRoot[28], para ello se eligen dentro de las opciones, el tipo de arquitectura, el gestor de dispositivos, el controlador de inicio, y uno de los puntos medulares, el formato de la imagen del sistema operativo.

Ångström Distribution Linux muestra en la figura 5, claramente que soporta el microcontrolador AVR32, por lo que en reiteradas ocaciones y por se la opción más facil.

Una vez construido el archivo de imagen del kernel, se graba en la memoria SD para realizar la prueba de carga del sistema operativo,será necesario pasarle al kernel las instrucciones para que realice dicha actividad, en realidad son dos cadenas que se comportan como variables de entorno, y que son las que toma en cuenta UBoot a la hora de buscar la imagen del sistema operativo en las memorias secundarias.

Estas instrucciones son las siguientes

- set bootcmd '*mmcinit;ext2load mmc 0:1 0x10300000 uImage;bootm*'
- set bootargs '*console=ttyS0 root=/dev/mmcblk0p1 rootwait*'

Despues de realizar los pasos mencionados, se prueba la imagen que ha sido generada a traves de la página de internet mostrada en la Figura 5, intentando ejecutar el kernel de linux, pero no ha funcionado de manera correcta ya que en el momento que se estaba cargando el sistema operativo se emitieron multiples mensajes de error que son visualizados a traves de programa minicom[29] que se conecta a la NGW100 en modo de consola.

Los kernels compilados fueron guardados en una base de datos para analizar posteriormente, la razon por la cual no funcionaron.

Es muy posible, que estos mensajes esten relacionados con la falta de ciertos componentes de software, tales como controladores y programas que se ejecutan en el momento de arranque, que son necesarios a la hora de intentar inicializar el dispositivo NGW100 desde la memoria SD.

---

28 Buildroot es un conjunto de archivos Makefile y parches, para generar un sistema completo de Linux embebido. BuildRoot puede generar un conjunto de herramientas de compilación cruzada, un sistema de archivos raíz, una imagen del kernel y una imagen de gestor de arranque.

29 Programa basado en texto, para el control y emulacion de terminal, la interface mas comun de uso de este programa es el serial.

Los mensajes que en reiteradas ocaciones aparecian a la hora de intentar cargar el Ångström Linux, es el siguiente:

VFS Kernel panic - not syncing: VFS: Unable to mount root fs on unknown-block(1,0)

***** Unable to read "/boot/uImage" from mmc 0:1 *****
Booting image at 10200000 ...
Bad Magic Number

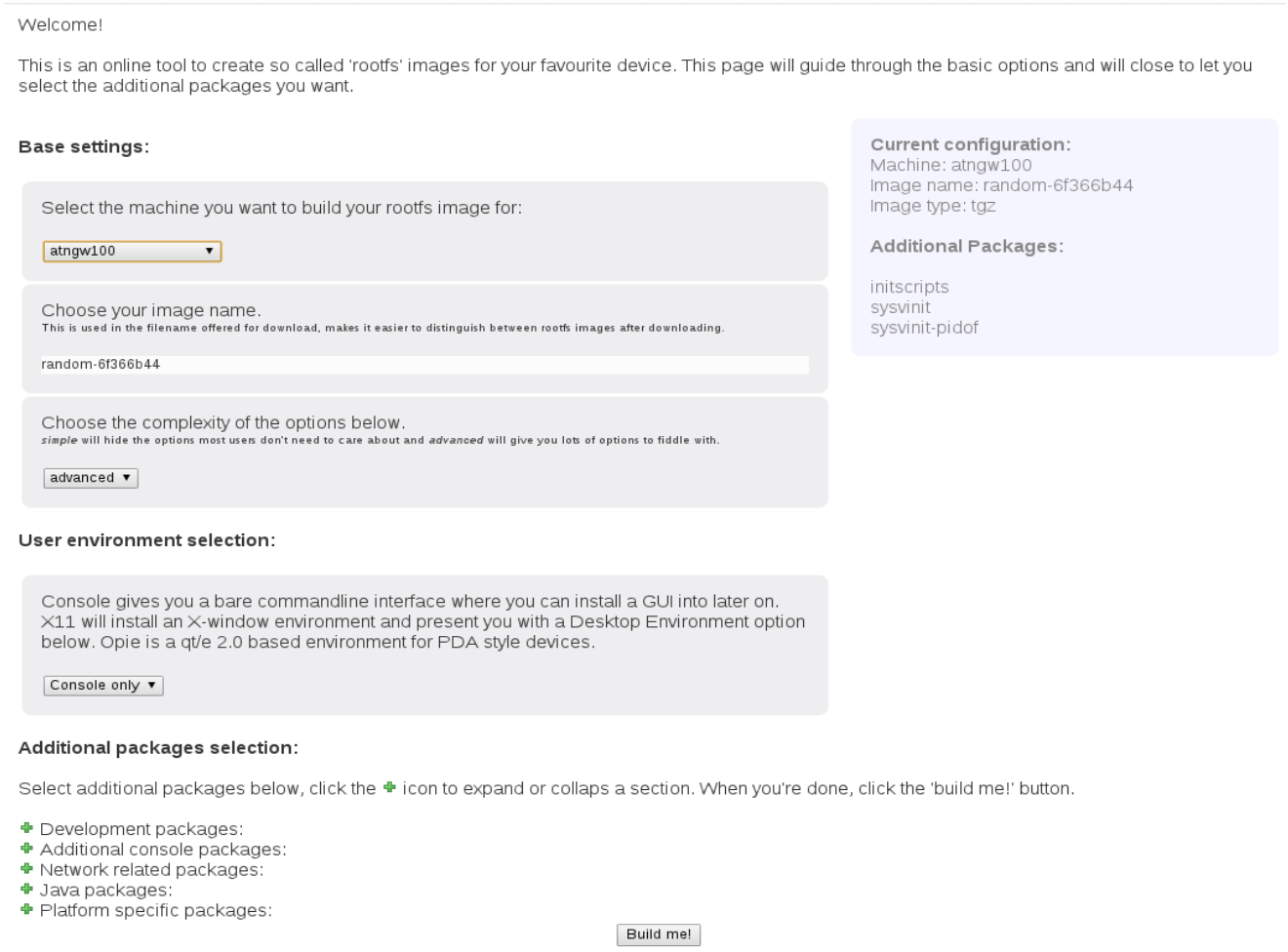


Fig. 6 Interface de la página de internet de The Ångström Distribution, constructor de imagenes de kernel en linea.

PREPARACIÓN DEL ENTORNO DE DESARROLLO

El entorno de desarrollo esta constituido sobre una distribución de Linux para procesadores .x86 basdado en virtualización que consta de las siguientes caracteristicas:

Sistema operativo Linux Ubuntu version 11,10
Disco duro de 35Gb de capacidad
1.5 Gb de RAM

Aceleración VT-x/AMD-V, Nested Paging, PAE/NX
Virtualizado en VirtualBox

Para construir el sistema operativo Linux basado en AVR32 se descargo de la pagina oficial de la distrbución OpenWrt la version mas actualizada del buildroot de nombre BackFire 10.0.3 esta version de buildroot cuenta con gran variedad de soporte para hardware, por lo que es actualmente la version de este sistema operativo con mayor soporte.

BackFire soporta miltiples architecturas entre las cuales destacan las siguientes: AVR32, ARM, CRIS, m68k, MIPS, PowerPC, SPARC, SuperH, Ubicom32, x86, x86-64. Esta distribución de linux utiliza un kernel de tipo monolitico muy compacto y optimizado para dispositivos empotrados.

La primera version de OpenWrt fue liberada en el año 2004, en la actualidad BackFire es la version mas reciente de esta distribuición y fue liberda en diciembre el año 2011. BackFire ha sido traducido a mas de 20 idiomas, pero su soporte oficial se encuentra en inglés.

En la actualidad se desarrolla con una versión más reciente llamada *Attitude Adjustment* de este sistema operativo, pero por desgracia para esta tarjeta de desarrollo no se tiene soporte previsto.

El proceso de instalacion de buildroot comienza con la descarga del paquete comprimido, desde la pagina oficial de OpenWrt *https://dev.openwrt.org/wiki/GetSource*, aqui una puede elegir que version sera descagada para comenzar con el desarrollo de la imagen del kernel.

Sera necesario que de base el sistema que alojara al huesped tenga como minimo los siguientes paquetes: *subversion build-essential,* ya que estos paquetes poseen gran parte de las fuentes que seran necesarias durante el proceso de la compilacion de la imagen del kernel para arquitectura AVR32.

Una ves teniendo esos paquetes instalados, creamos un directorio específico que contendra toda la paqueteria, códigos, scripts, etc. Es conveniente que el proceso de compilación sea hecho en un disco que cuente con suficiente capacidad, ya que durante la instalación es posible que se requiera de mucho mas espacio, debido a las constantes descargas de actualizaciones, por ende se debe contar tambien con una buena conexión a la internet, ya que de no ser así el proceso de compilación podria verse frenado debido a esa situación.

Procedemos a movermos al directorio donde se descargara la version del buildroot, haciendo uso del programa svn (**Apache Subversion**), procedemos a la descarga, tecleando los siguientes comandos en el prompt de Ubuntu.

maquina-kw:~ $ ***mkdir ~/openwrt***
maquina-kw:~ $ ***cd ~/openwrt***
maquina-kw:~ $ ***svn co svn://svn.openwrt.org/openwrt/trunk/***

Despues de que temine el proceso de descarga que podria variar en tiempo, en base al ancho de banda de la conexion a internet , en el directorio especificado que en este caso el el ***/openwrt*** se genera un subdirectorio llamado ***trunk/,*** nos movemos a ese subdirectorio con la siguiente instrucción

maquina-kw:~ $ ***cd trunk***

Este directorio es muy importante, dentro de él estan todos lo elementos necesario para realizar la construcción de nuestro sistema, se puede decir que dentro de este directorio esta depositado

nuestro compilador cruzado, que consta de archivos Makefile, scripts, archivos de configuraciones, códigos fuente de paquetes, etc. Como ya se menciono, esta versión fue liberada en diciembre del 2011, pero algunos paquetes han sido actualizados a lo largo de este tiempo, por lo que antes de dar inicio propiamente a la compilación, debemos actualizar estos paquetes, para así tener las verisones más actualizadas.

Es muy recomendable realizar esta actualización, debido a que nos podemos encontrar con errores de programación comunmente conocios como *bugs* dentro de la paqueteria que quiza ya han sido resueltos. Además con esta operación nos aseguramos de tener un buildroot más actualizado, las instrucciones para inicial la actualización son las siguientes son las siguientes:

maquina-kw:~ $ ***./scripts/feeds update -a***
maquina-kw:~ $ ***./scripts/feeds install -a***

Una ves finalizado el proceso de actulización, el directorio pasa de tener unos cuatos cientos de MegaBytes a varios GigaBytes, en total se prevee que fueron instalados alrededor de 26650 archivos extras. Otra de las opciones que nos ofece la actualización es la descarga por paquete individual, proceso que se realiza con la siguiente instrucción.

maquina-kw:~ $ ./scripts/feeds install PACKAGENAME

A continuación se ejecuta el configurardor del buildroot, para poder ejecutarlo sera necesario que se instroduszan los siguientes comando, la figura 6 muestra la interfaz del comilador cruzado.

maquina-kw:~ $ ***make defconfig***	
maquina-kw:~ $ ***make prereq***	Verificación de los prerequisitos
maquina-kw:~ $ ***make menuconfig***	Menu de Configuración de buildroot

El ultimo de los comando anteriormente mencionados es el mas importante, ya que nos permite acceder a las opciones de compilación, permite decidir, el tipo de compilación, la arquitectura a la que esta destinada dicha compilación, los servicio que correran el esa version compilada, el tipo de kernel, y las aplicaciones que seran usadas en conjunto, como utilidades del sistema. Depués de realizar algun cambio sera guardada la configuración en un archivo llamado *.config,* si se necesitara regresar a la configuración original ser puede utilizar el siguiente comando, teniendo en cuenta que esta acción borrara todo lo contenido de la configuración contenida en el archivo .config por lo que quedará en el estado inicial, y será necesario reconfigurar todas las opciones para poder comenzar de nuevo con el proceso de compilación del sistema operativo.

maquina-kw:~ $ ***make clean***

Otros comandos que seran utiles durante el proceso de compilación son los siguientes:

maquina-kw:~ $ ***make distclean***
maquina-kw:~ ***$ make config***
maquina-kw:~ ***$ make***

Se debe tener especial cuidado con el comando ***make distclea,*** ya que si se ejecuta se debe estra consciente que seran borrados absolutamente todas las configuraciones, descargas, compilaciones anteriores etc.

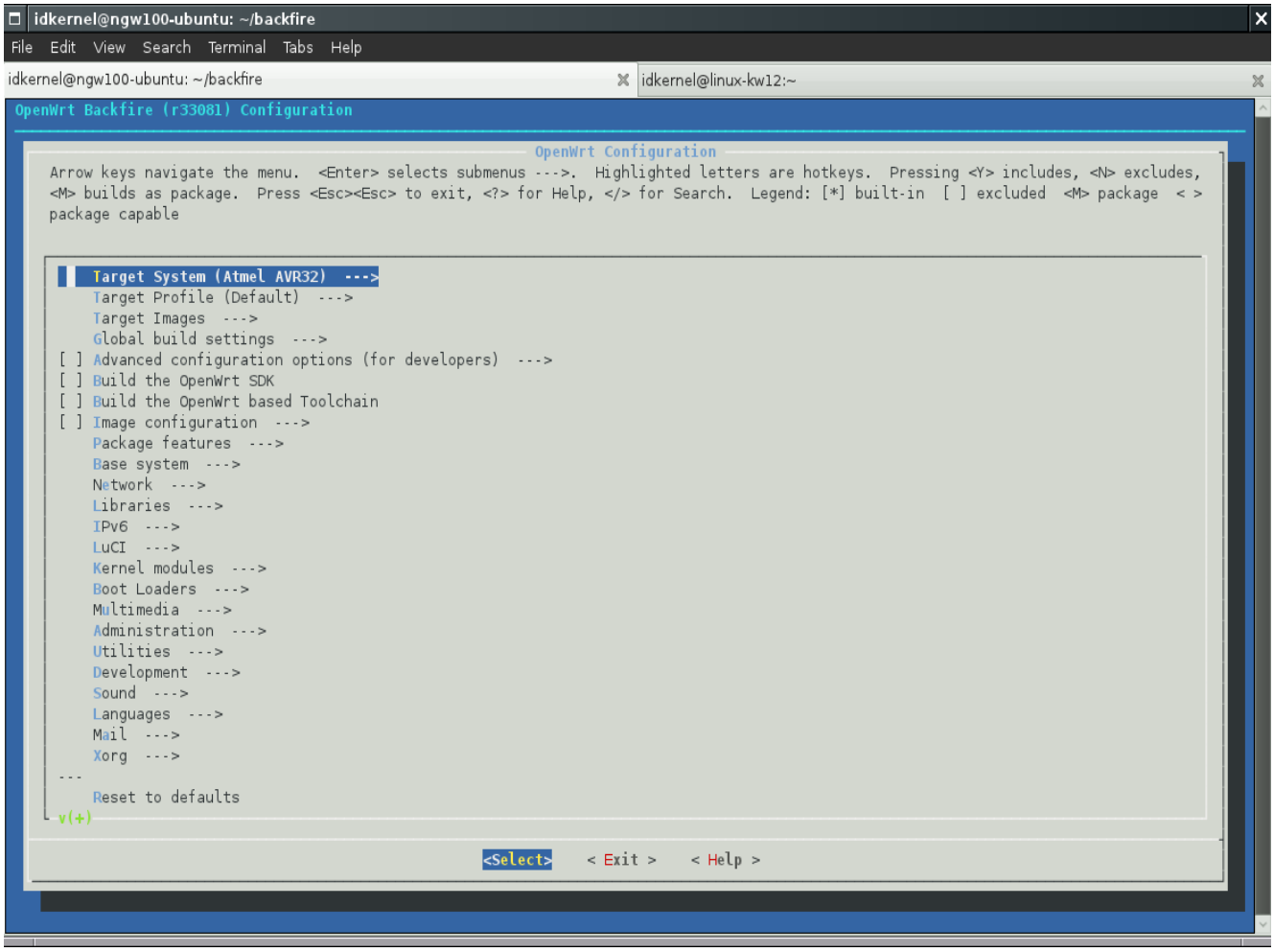


Fig 6. Ejecución de programa de configuración del buildroot OpenWrt backFire

La figura 7 muestra la tabla con los programas que son necesarios para poder realizar una compilación cruzada del kernel para una arquitectura AVR32. Muchos de estos programas no seran indispensables para generar el kernel, pero se recomienda realizar la instalalción de todos ellos, para evitar futuros problemas a la hora de ejecutar el compilador cruzado.

Aunque aquí se muestra esta tabla, es posible que el sistema operativo residente necesite instalar algún paquete extra, a los requeridos inicialmente. Esto se debe, a que las condiciones iniciales pueden variar.

Prerequisite	Debian	Suse	Red Hat	OS X (via MacPorts)	Fedora	NetBSD
asciidoc	asciidoc	asciidoc	asciidoc	asciidoc	asciidoc	?
bash	bash	bash	?	bash	?	bash
binutils	binutils	binutils	binutils	binutils	binutils	?
bzip2	bzip2	bzip2	bzip2	bzip2	bzip2	?
fastjar	fastjar	fastjar	libgcj	fastjar	libgcj	?
flex	flex	flex	?	flex	flex	?
git	git-core	git-core	?	?	?	?
g++	g++	gcc-c++	gcc-c++	?	gcc-c++	?
gcc	gcc	gcc	gcc	?	gcc	?
getopt	util-linux	util-linux	?	getopt	?	getopt
GNU awk	gawk	gawk	gawk	gawk	gawk	?
gtk2.0-dev	libgtk2.0-dev	?	gtk2-devel	gtk2	gtk2-devel	?
intltool-update	intltool	intltool	intltool	intltool	intltool	?
jikes	—	jikes	?	jikes	—	?
libz, libz-dev	zlib1g-dev	zlib-devel	zlib-devel	zlib	zlib-devel	?
make	make	make	?	gmake	make	gmake
ncurses	libncurses5-dev	ncurses-devel	ncurses-devel	ncurses	ncurses-devel	?
openssl/ssl.h	libssl-dev	libopenssl-devel	openssl-devel	openssl	openssl-devel	?
patch	patch	patch	?	patchutils	patch	?
perl-ExtUtils-MakeMaker	perl-modules	perl-ExtUtils-MakeMaker	perl-ExtUtils-MakeMaker	p5-extutils-makemaker	perl-ExtUtils-MakeMaker	?
python2.6-dev	python2.6-dev	python-devel	?	python26	?	?
rsync	rsync	rsync	?	rsync	rsync	?
ruby	ruby	ruby	?	ruby	ruby	?
sdcc	sdcc	sdcc	?	sdcc	sdcc	?
unzip	unzip	unzip	?	unzip	unzip	?
wget	wget	wget	wget	wget	wget	?
working-sdcc	—	?	?	?	—	?
xgettext	gettext	?	?	gettext	gettext	?
xsltproc	xsltproc	libxslt	?	libxslt	libxslt	?
zlib, zlib-static	zlib1g-dev	zlib-devel	?	?	?	?

Package	Prerequisite	Debian	Suse	Red Hat	OS X	Fedora	NetBSD
intltool	[Perl] XML::Parser	libxml-parser-perl	?	?	?	?	?

Fig. 7 Tabla de prerequisitos para la compilación cruzada

openSuSE 11.1

```
# zypper install binutils bzip2 gawk gcc gcc-c++ gettext make ncurses-devel patch unzip wget zlib-devel flex git-core
```

Ubuntu:

```
$ sudo apt-get install build-essential subversion libncurses5-dev zlib1g-dev gawk flex
```

Ubuntu 9.10, I needed also these (30-03-2011):

```
$ sudo apt-get install gcc-multilib bison autoconf screen gcc g++ binutils patch bzip2 flex make gettext unzip libc6 git-core
```

Ubuntu 11.10:

```
$ sudo apt-get install build-essential subversion git-core libncurses5-dev zlib1g-dev gawk flex quilt
```

Ubuntu 12.04LTS:

```
$ sudo apt-get install build-essential subversion git-core libncurses5-dev zlib1g-dev gawk flex quilt libssl-dev xsltproc libxml-parser-perl
```

Ubuntu 64bit:

```
$ sudo apt-get install build-essential subversion libncurses5-dev zlib1g-dev gawk gcc-multilib flex git-core gettext
```

Fig. 9 Comandos de instalación de los paquetes necesarios.

Al igual que ocurrio en este trabajo de proyecto terminal, es posble que el buildroot, no funcione correctamente en algunas distribuciones de Linux, ya que en los inicios de este proyecto, se habia construido un entorno de desarrollo en la version 11,4 de Linux OpenSuse, pero a lo largo de la compilación se presentaron problemas de compatibilidad y algunos de los programas que aparecen en la tabla de los requerimientos no puedieron ser instalados o ya se encuentran descontinuados de sus repositorios o depositos correspondientes, lo que impidio finalmente que se pudiera realizar una sola compilación exitosa, se recomienda utilizar de entrada la versión del Linux más reciente hasta ese momento y ademas tratar de utilizar distribuciones derivadas de Debian, e incluso el mismo Debian.

Finalmente despues de intentar en reiteradas ocaciones con la distribución OpenSuse, se opto por utilizar Ubuntu 12.04, en la cual se puede facilmente instalar los programas requeridos para nuestra compilación, la tabla de la Figura. 9 contiene los comandos de ejecucion del buildroot en cada distribución.

Para la compilación cruzada es recomendable que se tenga un buen espacio de reserva, ya que a pesar de que se contaba con un disco duro de 10 Gb con la version de OpenSuse, al cabo de algunos dias este quedo sobre saturado de información, por lo que tambien se opto por ensancharlo y pasar de un tamaño de 10Gb a los 35Gb con los que finalmente quedo.

SCRIPTS E INTERFACE WEB

Una de las partes fundametales de este proyecto es la interface que se mostrará al usuario, esta sera la forma de comunicación e interacción entre la persona y el dispositivo, se eligio mostar la información a traves de una pagina web ya que resulta ser la forma más natural, y en la que no tendra que instalar absolutamente nada en su computadora, debdo a que la mayoria de los sistemas operativos de computadoras personales cuentan con al menos un explorador de internet, único programa necesario para acceder al router NGW100.

La interface web mostrada en la figura 10, esta construida en base al lenguaje HTML, PHP, JavaScript y posee además modificadores de tipo CSS, corren en un mini servidor web llamado uhttpd que le proporciona funciones con interpretación de programas de CGI y de PHP[6].

Fig. 10 Página inicial de la interface web del router NGW100

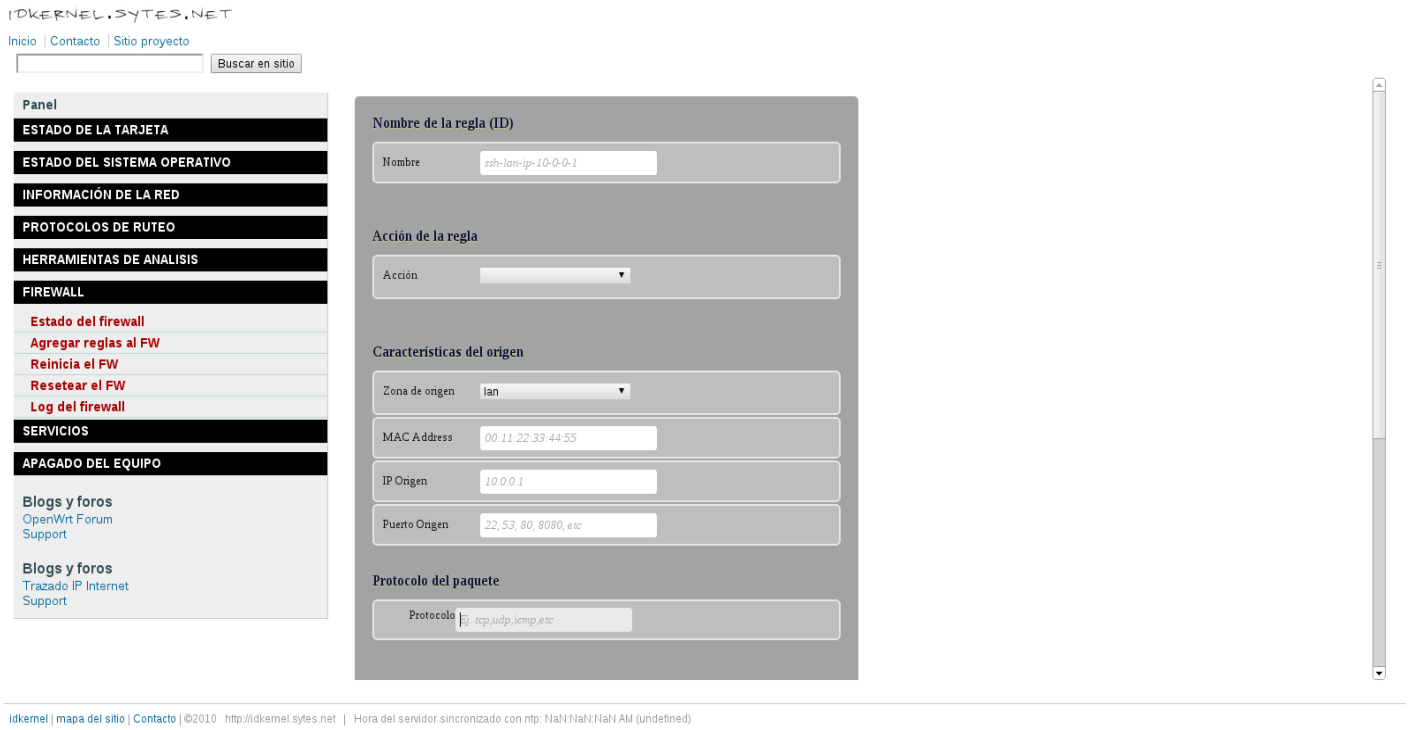


Fig. 11 Interface mostranto la función para agregar reglas al FW

Como plataforma de desarrollo para este proyecto se eligio el lenguage PHP, este nos permite de manera rapida y eficiente, ejecutar los comandos necesarios para llevar a cabo las funciones vitales, tales como la carga del sistema y la admninistración del mismo. Algunos otros scripts fueron creados para dar soporte y facil acceso a otros comandos o para dar formato a las salida de los mismos la figura 11 muestra la entrada de datos desde la interfaz de la pagina web.

Ademas de los scripts creados en PHP, se generaron scripts para el sistema operativo, estos son interpretados directamente desde el shell del linux, se encuentran precompilados en el sistema, por lo que su modificación de ser necesaria requeria de una nueva compilación y generación de kernel.

Esta es la relación de los scripts que fueron generados a lo largo de este proyecto, algunos de ellos ya se encuentra descontinuados, aunque sigan estando presentes en las ultimas versiones de la imagen del kernel.

No.	Nombre del Script	Funcion	Ubiacion	Script de Sistema	Estado
1	config-dhcp-dinamico.sh	Llama al DHCP esta funcion solo es para la interface WAN	/bin	No	Activo
2	config-firewall.sh	Restaura configuracion del FW de la ultima sesion	/bin	Si	Activo
3	config-install-package.sh	Instalacion de paqueteria al arranque del sistema	/bin	No	Activo
4	config-opkg-default.sh	Configura el actualizador del sistema operativo	/bin	No	Inactivo
5	config-overlay-mount.sh	Gestiona la memoria de SWAP y la SD	/bin	Si	Activo
6	config-sys-daemons.sh	Activa los demonios de monitoreo de la red	/bin	No	Activo
7	config-sys-diagnos.sh	Instala los programas necesarios para activar el modo de diagnostico	/bin	No	Activo
8	config-sys-router-buk.sh	Configuracion de l demonio zebra, controla OSPF y RIP	/bin	No	Inactivo
9	config-sys-router.sh	Configura en su totalidad el modo de enrutamiento	/bin	No	Activo
10	fw-backup.sh	Realiza una copia de seguridad de la configuracion actual del FW	/bin	No	Activo
11	fw-clean.sh	Libera todas las reglas de la pared de fuego, permite todas las comuniciones	/bin	No	Activo
12	fw-reload.sh	Reinicia el FW	/bin	No	Activo
13	fw-restore.sh	Script para restaurar automaticamente la pared de fuego	/bin	Si	Inactivo
14	fw-restore.sh.save	Script para restaurar automaticamente la pared de fuego	/bin	Si	Inactivo
15	Fw-stop-all.sh	Script para liberar las reglas del FW, bloquea todas las comunicaciones	/overlay	No	Aun no activo
16	fw-stop.sh	Script para liberar las reglas del FW, detiene la ejecucion del FW	/bin	No	Activo
17	ip-eth0.sh	Script para aislar datos, para la interface web	/bin	No	Activo
18	ip_eth1.sh	Script para aislar datos, para la interface web	/overlay	No	Aun no activo
19	cgi-arpSys.php	Ejecuta el comando arp, para la pagina web	/overlay	No	Activo
20	cgi-cpuinfo.php	Ejecucion del comando cpusage, para la pagina web	/overlay	No	Activo
21	cgi-fwClean.php	Script asistente de la pagina web, ejecuta el script fw-clean.sh	/overlay	No	Activo
22	cgi-fwReload.php	Script asistente de la pagina web, ejecuta el script fw-reload.sh	/overlay	No	Activo
23	cgi-logSys.php	Ejecuta el comando dmesg, para la pagina web	/overlay	No	Activo
24	cgi-rebootSys.php	Ejecuta el comando reboot, para la pagina web	/overlay	No	Activo
25	cgi-shutdownSys.php	Ejecuta el comando poweroff, para la pagina web	/overlay	No	Activo
26	info.php	Muestra la configuracion totaal de PHP	/overlay	No	Inactivo
27	inicio.php	Script de prueba	/overlay	No	Inactivo
28	php-telnet.zip	Script no finalizado, inicia sesion telnet desde la pagina web	/overlay	No	Aun no activo
29	shell-exec.php	Script de prueba	/overlay	No	Inactivo
30	telnet.php	Script no finalizado, inicia sesion telnet desde la pagina web	/overlay	No	Aun no activo

Todos los scripts que componen el sistema, se encuantran disponibles en el CD que será entregado junto con esta documentación. Se anexan tambien los códigos de la pagina web.

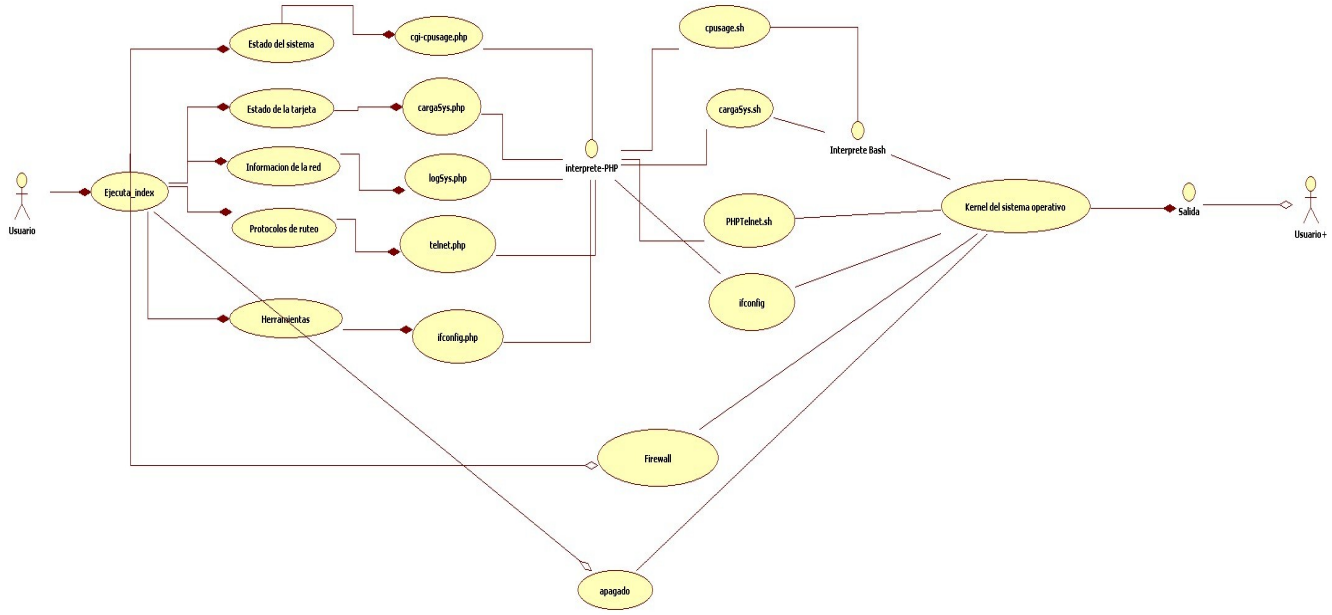


Fig. 13 Esquema del flujo de datos, para las peticiones desde la pagina del router

El esquema de la figura 13 muestra a grandes rasgos el camino que siguie el flujo de datos antes de responder a una peticion lanzada desde el explorador de interne a la hora de acceder a la pagina web. Queda de manifiesto que de no tener soporte de PHP no se podria enitir una peticion hacia el interprete de comanos del sistema operativo.

TRABAJO PENDIENTE

Durante el desarrollo de este proyecto terminal, quedaron pendientes algunos puntos importantes que si bien ya no afectan al funcionamiento del equipo, si convendria revisar, sobre todo por que el trabajo sobre este enrutador se volveria un poco mas facil.

El principal problema que queda pendiente es la generación de una imagen del kernel que pueda gestionar correctamente el proceso de pivot_root para poder ocupar la memoria SD como un disco duro donde se pueda escribir y no se pierda la configuración cada ves que reinicia. Se estudio la posibilidad de generar la imagen y realizar la revision de la misma, pero por cuestiones de tiempo, ya no era factible seguir antendiendo esa necesidad, puesto que el sistema ya puede cargar directo de la memoria secundaria SD a la memoria principal y realiza sus funciones correctamente, este problema queda como el principal de los pendientes de este proyecto terminal.

Otro problema que no se logro resolver en los ultimos dias de desarrollo, es la funcionalidad de enrutamiento OSPF, queda claro que este esta activo dentro del kernel, y que durante las pruebas con la red virtual funciono, pero a partir de ese punto no se ha podido poner en marcha este enrutamiento, por lo que queda pendiente la revision profunda de la configuración, tanto del demonio zebra, como de los archivos de configuración de OSPF.

Algunos programas presentan problemas de programacion, tal es el caso del codigo del programa tcpdump, este programa en un principio fue compilado e incluido en el repositorio de aplicaciones, cuando este programa fue instalado, se realizaron pruebas con el y se vio que tenia problemas cuando ambas interfaces de red estaba en UP, esta condicion de las tarjetas le genera una segmentation fault, la solucion, provicional, es dar de baja temporalmente una de las interfaces, realizar las lecturas que se requieran con el tcpdump e irlas guardando en un archivo, esto evita que este programa falle en tiemo de ejecucion.

PRUEBAS Y RESULTADOS

PRUEBAS DE ENRUTAMIENTO CON LA RED VIRTUAL RIPv.2

Se realizaron dos tipos de pruebas al dispositivo implementado, en materia de enrutamiento, la primera de ellas se realizó con el simulador de redes GNS3, este potente simulador permite conectar varios routers de manera virtual, bajo diferentes protocolos de enrutamiento, ademas permite intercomunicar a las redes virtuales con las redes físicas, tal es el caso de estas pruebas.

Como en este proyecto solo se implementon dos protocolos de enrutamiento, el RIPv.2 y el OSPFv.2, se diseñaron dos redes virtuales con multiples subredes. Una para realizar la prueba de protocolo de enrutamiento RIP y la otra para realiza la revisión de protocolo de enrutamiento OSPFv.2.

El procedimiento fue casi el mismo en el diseño y construcción de las redes virtuales. Primeramente se diseño y construyo la red RIP, duramente este proceso se realizaron pruebas sobre las subredes virtuales, se reviso que estuvieran bien configuradas las interfaces de cada uno de los enrutadores, se reviso si realizaban correctamente el proceso de anuncio de las tablas a sus vecinos via el protocolo RIP, una ves concluida la revisión y vericado que todos los routers conocen las tablas de enrutamiento de sus vecinos, y previa revisión a traves de comando ping a cada una de los segmento desde cada uno de los enrutadores que comeponen esta red virtual, se procedio a realiza la interconexión entre la red virtual compuesta de 4 routers de marca Cisco Mod. 3660 telco a el router NGW100 que se desarrollo en este proyecto, esta interconexión se realiza a traves del Modem-Router que provee la compañia Telmex para dar su servicio de internet, tal como lo muestra la siguiente imagen.

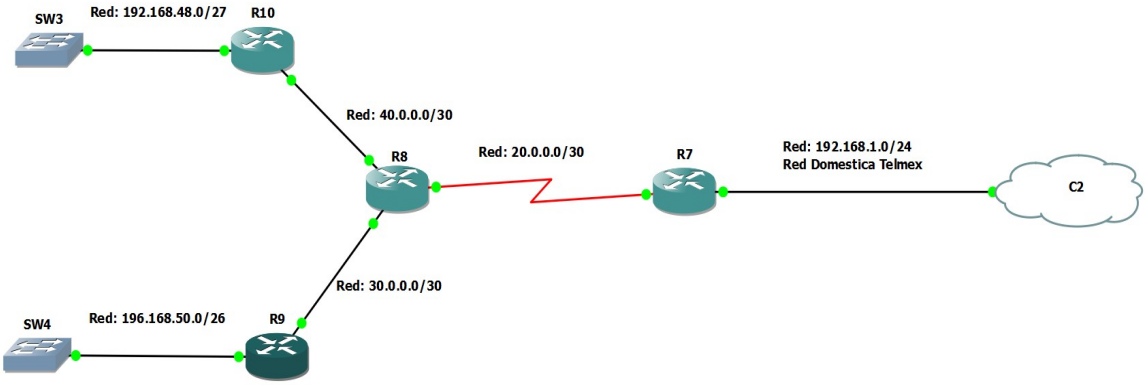
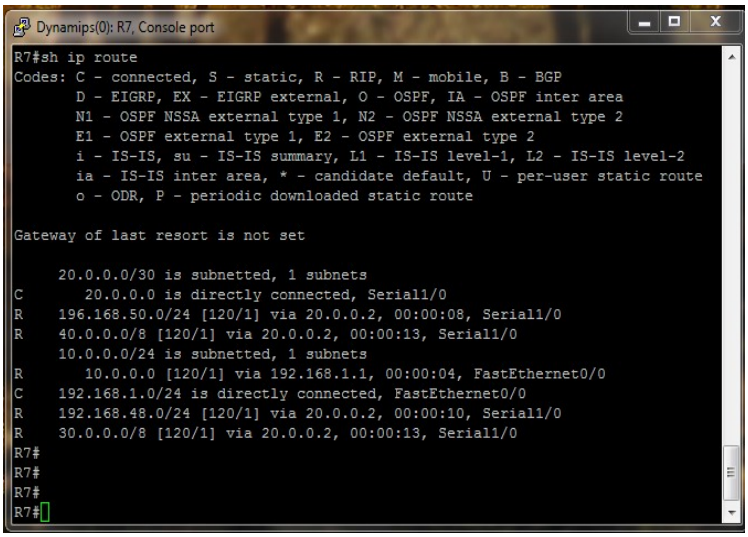
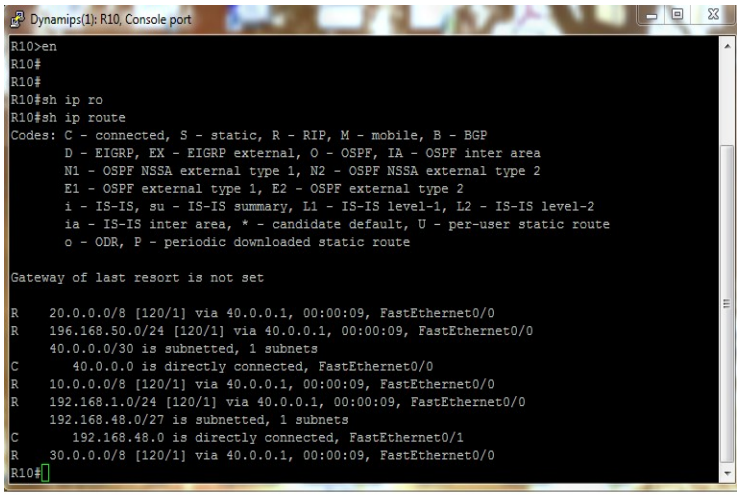
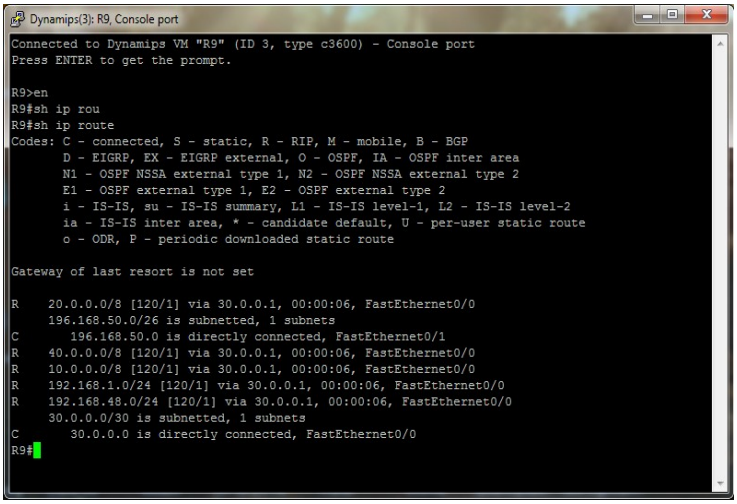


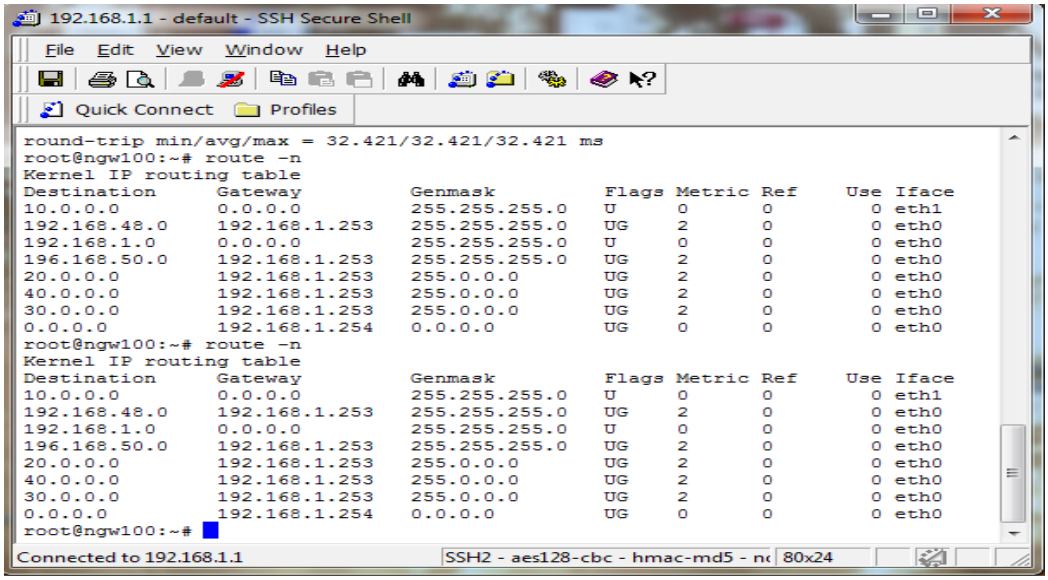
Fig. 12 Pruebas realizadas con la red virtual de enrutamiento dinamico RIPv.2, simulador de redes GNS3.

Ya conetado el dispositivo NGW100 a la red cuyo segmento es 192.168.1.0/24, se activa el demonio de enrutamiento llamado quagga, se libera temporalmente la pared de fuego que protege al NGW100 y se comienza a ver la transmisión de los paquetes y la modificación de las tablas de enrutamiento, tanto en los routers Cisco 3660 como en el router NGW100, esto se demuestra en la siguientes imagenes .


Dynamips(0): R7, Console port
R7#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
20.0.0.0/30 is subnetted, 1 subnets
C 20.0.0.0 is directly connected, Serial1/0
R 196.168.50.0/24 [120/1] via 20.0.0.2, 00:00:08, Serial1/0
R 40.0.0.0/8 [120/1] via 20.0.0.2, 00:00:13, Serial1/0
10.0.0.0/24 is subnetted, 1 subnets
R 10.0.0.0 [120/1] via 192.168.1.1, 00:00:04, FastEthernet0/0
C 192.168.1.0/24 is directly connected, FastEthernet0/0
R 192.168.48.0/24 [120/1] via 20.0.0.2, 00:00:10, Serial1/0
R 30.0.0.0/8 [120/1] via 20.0.0.2, 00:00:13, Serial1/0
R7#
R7#
R7#
R7#


Dynamips(1): R10, Console port
R10>en
R10#
R10#
R10#sh ip ro
R10#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
R 20.0.0.0/8 [120/1] via 40.0.0.1, 00:00:09, FastEthernet0/0
R 196.168.50.0/24 [120/1] via 40.0.0.1, 00:00:09, FastEthernet0/0
40.0.0.0/30 is subnetted, 1 subnets
C 40.0.0.0 is directly connected, FastEthernet0/0
R 10.0.0.0/8 [120/1] via 40.0.0.1, 00:00:09, FastEthernet0/0
R 192.168.1.0/24 [120/1] via 40.0.0.1, 00:00:09, FastEthernet0/0
192.168.48.0/27 is subnetted, 1 subnets
C 192.168.48.0 is directly connected, FastEthernet0/1
R 30.0.0.0/8 [120/1] via 40.0.0.1, 00:00:09, FastEthernet0/0
R10#


Dynamips(3): R9, Console port
Connected to Dynamips VM "R9" (ID 3, type c3600) - Console port
Press ENTER to get the prompt.
R9>en
R9#sh ip rou
R9#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
R 20.0.0.0/8 [120/1] via 30.0.0.1, 00:00:06, FastEthernet0/0
196.168.50.0/26 is subnetted, 1 subnets
C 196.168.50.0 is directly connected, FastEthernet0/1
R 40.0.0.0/8 [120/1] via 30.0.0.1, 00:00:06, FastEthernet0/0
R 10.0.0.0/8 [120/1] via 30.0.0.1, 00:00:06, FastEthernet0/0
R 192.168.1.0/24 [120/1] via 30.0.0.1, 00:00:06, FastEthernet0/0
R 192.168.48.0/24 [120/1] via 30.0.0.1, 00:00:06, FastEthernet0/0
30.0.0.0/30 is subnetted, 1 subnets
C 30.0.0.0 is directly connected, FastEthernet0/0
R9#


192.168.1.1 - default - SSH Secure Shell
File Edit View Window Help
Quick Connect Profiles
round-trip min/avg/max = 32.421/32.421/32.421 ms
root@ngw100:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
192.168.48.0 192.168.1.253 255.255.255.0 UG 2 0 0 eth0
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
196.168.50.0 192.168.1.253 255.255.255.0 UG 2 0 0 eth0
20.0.0.0 192.168.1.253 255.0.0.0 UG 2 0 0 eth0
40.0.0.0 192.168.1.253 255.0.0.0 UG 2 0 0 eth0
30.0.0.0 192.168.1.253 255.0.0.0 UG 2 0 0 eth0
0.0.0.0 192.168.1.254 0.0.0.0 UG 0 0 0 eth0
root@ngw100:~# route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
10.0.0.0 0.0.0.0 255.255.255.0 U 0 0 0 eth1
192.168.48.0 192.168.1.253 255.255.255.0 UG 2 0 0 eth0
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
196.168.50.0 192.168.1.253 255.255.255.0 UG 2 0 0 eth0
20.0.0.0 192.168.1.253 255.0.0.0 UG 2 0 0 eth0
40.0.0.0 192.168.1.253 255.0.0.0 UG 2 0 0 eth0
30.0.0.0 192.168.1.253 255.0.0.0 UG 2 0 0 eth0
0.0.0.0 192.168.1.254 0.0.0.0 UG 0 0 0 eth0
root@ngw100:~#
Connected to 192.168.1.1
SSH2 - aes128-cbc - hmac-md5 - n
80x24

Las imagenes anteriores demuestran claramente la propagación de las tablas de enrutamiento entre cada uno de los elementos que conforman la red, includo el router que fue implementado.

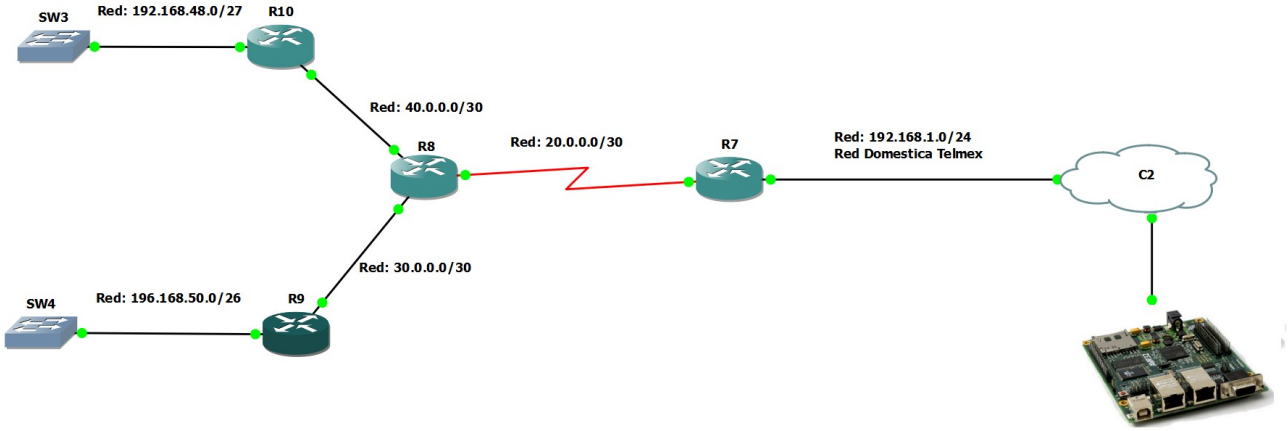


Fig. 18 Red virtual de prueba para la revision del protocolo de enrutamiento RIP.

PRUEBAS DE ENRUTAMIENTO CON LA RED VIRTUAL OSPFv.2

Para las pruebas con el protocolo de enrutamiento OSPFv.2 utilizamos la siguiente red virtual que consta de 5 routers Cisco 3520, todos los routers fueron previamente configurados y revisados, al igual que en la prueba anterior. Cabe destacar que esta red a diferencia de la RIPv.2 posee dos areas propias del protocolo OSPF.

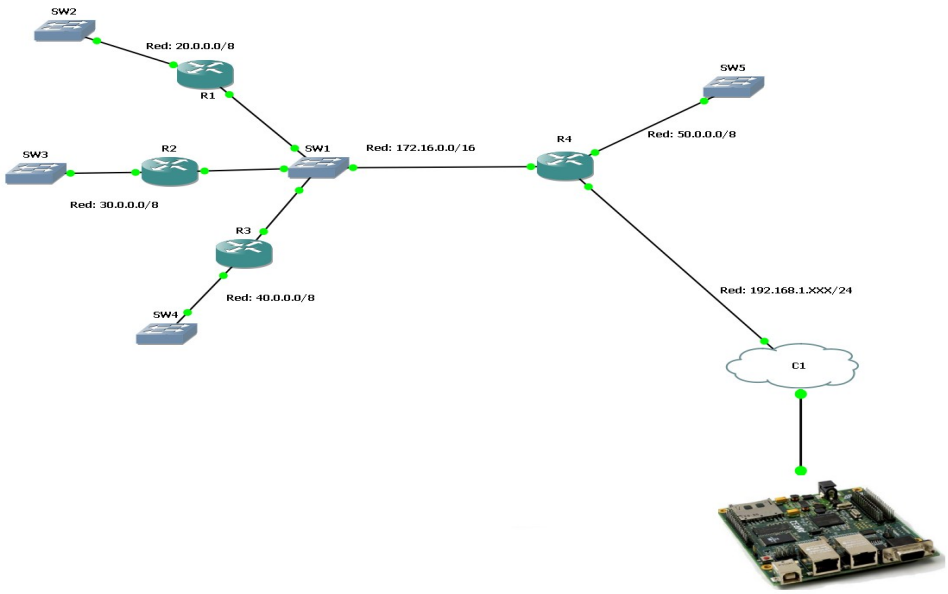


Fig. 19 Red virtual de prueba para la revision del protocolo de enrutamiento OSPF.

Durante las pruebas con la red que se muestra en la figura 19, no se vio en ningun momento el anuncio de adyacencia de vecinos o la actualización de las tablas de enrutamiento en ninguno de los dispositivos de la red, por lo que se puede decir que este protocolo aunque se encuentra presente en el router desarrollado en este proyecto terminal, no está funcionando de manera correcta, se procedera a continuación con las pruebas físicas, con el fin de determinar si se trata de un problema con la simulación o efectivamente el protocolo de enrutamiento OSPF de la tarjeta NGW100 no está funcionando correctamente.

PRUEBAS DE ENRUTAMIENTO CON LA RED FÍSICA, SE UTILIZA UN ENRUTADOR CISCO SERIE 1700

Despues de realizar las pruebas con las redes virtuales, se procedio a realizar pruebas con equipos físicos, para observar el comportamiento del router que se implemento en la tarjeta NGW100, en un medio con multiples comptadoras. Se realiza esta prueba para saber si posee las capacidades minimas en un ambiente controlado y ver que tantas tranasacciones soporta el equipo y que cantidad de computadoras en una red puede atenter simultameamente.

Estas pruebas son realizadas con un equipo Cisco 1700, el cual puede realizar enrutamiento dinamico RIP y OSPF, para este caso solo ocuparemos un equipo enrutador, debido a que solo se cuenta con uno.

En conjunto con el asesor del proyecto terminal se realizan las siguientes pruebas, con un equipo router de la marca Cisco, el modelo de este dispositivo es el 1700, este dispositivo posee enrutamiendo dimanico de tipo RIP v.2 y OSPF.

Primeramente se realiza la prueba al enrutamiento dinamico con el protodolo RIP v.2, de lo cual se concluye por lo que observamos durante la comunicación entre los equipo, que son anunaciadas correctamente las tablas de enrutamiento, ademas de realizar pruebas con el programa ping para ver si los dispositivos que estan conecados en diferentes segmentos de red contestaban, y lo hicieron satisfactoriamente.

Por lo que queda concluido que el Router NGW100 tiene la función de realizar enrutamiento dinamico a traves del protocolo de enrutamiento RIP v.2.

Por otro lado, OSPF se implenta en este proyecto como una función extra, debido a que en un principio no se planeto de origen que el router aqui desarrollado tuviera habilitado este protocolo de enrutamiento.

Aun cuando este esta habilitado durante las pruebas realizadas tanto en el ambiente virtual como con el dispositivo fisico de Cisco, este protocolo no respondio satisfactoriamente, es muy probable que se tenga un problema en la configuración del super demonio zebra que es quien controla los protocolos de enrutamiento, tanto RIP como OSPF.

Para esta prueba, el equipo Cisco fue conectado a la tarjeta NGW100 y configurado apropiadamante, pero durante el periodo de pruebas no se vio en ningun momento en anuncio de las adyacencia de vecinos ni las tablas de ruteo, por lo que se descarta por el momento que este protocolo este funcionando por el momento, debido a los retrasos y a lo recortado del tiempo, es muy probable que este protocolo no funcione para cuando finalice este proyecto terminal.

Sin embargo el soporte de este protocolo de enrutamiento queda instalado en la versión del kernel que actualmente corre en la tarjeta, por lo que en determinado momento se puede revisar la configuración para hallar el error.

PRUEBAS DE FUNCIONAMIENTO DEL FIREWALL, BLOQUEO DE PUERTOS

ESPECÍFICOS.

Para esta fase de pruebas se puso a prueba el funcionamiento de la pared de fuego, en ambas interfaces de la NGW100, por defecto se establece, como casi en cualquier dispositivo de seguridad el cierre completo de las comunicaciones y se van abriendo con forme se necesiten.

De entrada, se establecen reglas de comunicación para los puertos de 22 (sshd) y 80 (httpd) desde la red local, autentificando el acceso a traves de la MAC[30] Address y de la IP del equipo que intenta establecer comunicación con estos puertos, para asegurarnos que solo el administrador del equipo pueda acceder a la configuración.

A continuación se muestra un esquema de la forma en la cual funciona el firewall de este equipo.

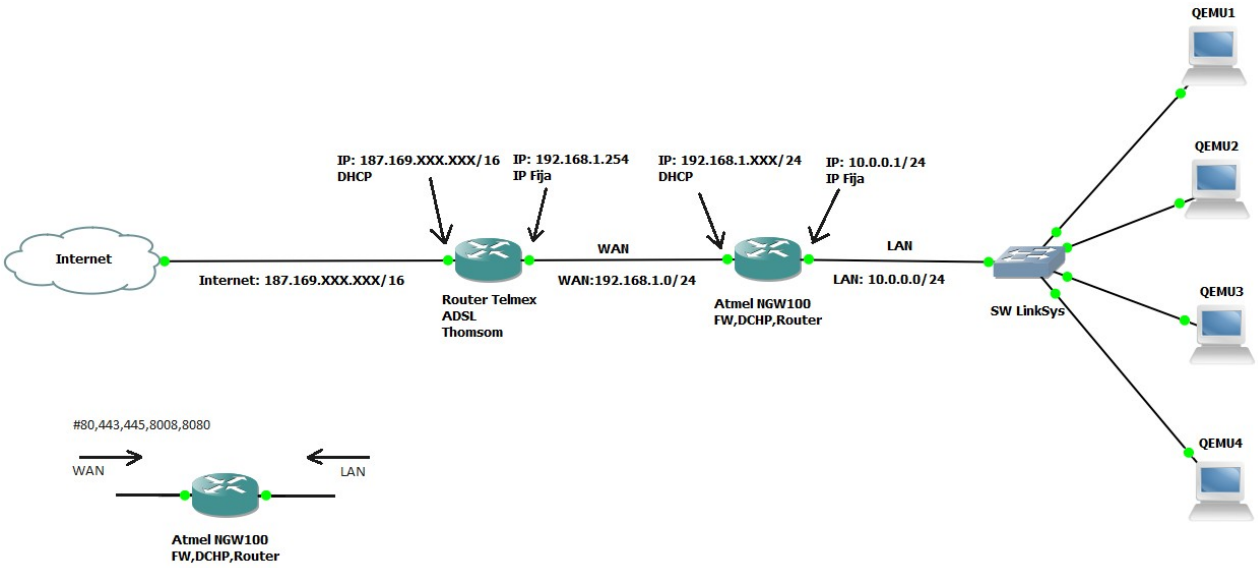


Fig. 20 Esquema general del proyecto y funcionamiento del router y pared de fuego.

Este esquema puede variar de red en red, para realizar las pruebas en el laboratorio se presentara un esquema completamente diferente al mostrado con anterioridad.

Definición de reglas en el firewall

La distribución OpenWrt tiene una forma peculiar de agregar las reglas de la pared de fuego, ya que no hace uso directo del comando ***IPTABLES***, las reglas son agregadas a traves de un archivo que se ubica en la ruta ***/etc/config/firewall*** y este a su ves se encarga de verificar la sintaxis de cada una de las reglas que finalmente seran agregadas como entradas a traves del comando ***IPTABLES***, esto no significa de ninguna manera que este sistema operativo no pueda agregar entradas al firewall de la manera tradicional, a traves de la linea de comandos[31].

El formato que se ocupa para agregar las reglas a traves del archivo ***/etc/config/firewall*** es el

30 En las redes de computadoras, la dirección MAC (siglas en inglés de media access control; en español "control de acceso al medio") es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una tarjeta o dispositivo de red. Se conoce también como dirección física, y es única para cada dispositivo. Está determinada y configurada por el IEEE (los últimos 24 bits) y el fabricante (los primeros 24 bits) utilizando el organizationally unique identifier.

31 Interfaz de Línea de Comandos (CLI), por su acrónimo en inglés de Command Line Interface (CLI), es un método que permite a las personas dar instrucciones a algún programa informático por medio de una línea de texto simple.

siguiente:

```
config rule
        option name         Nombre de la regla                          -- Cadena de id. optativa
        option target       ACCEPT, DROP, REJECT                        -- Dependiendo de lo que se quiera hacer
        option src          lan,wan ,*                                  -- Dependiendo del origen del paquete
        option src_mac      Dirección fisica de la interface de origen
        option src_port     Puerto de origen, o rango de puertos        -- Puerto solo en número de puerto o rango 1:65535
        option proto        tcp,udp,tcpudp,etc                          -- Establece el protocolo especifico de comunicación
        option dest         lan,wan,*                                   -- Dependiendo del destino del paquete
        option dest_port    Puerto destino o rango de puertos           -- Puerto solo en número de puerto o rango 1:65535
        option family       ipv4, ipv6                                  -- Puede ser de la version IPv.4 o IPv.6
        option limit        tiempo/seg, min, hrs                        -- Tiempo de validez de la regla.
```

La salida de la bitácora del sistema muestra que ciertas comunicaciones son detenidas o denegadas y otras permitidas, ademas se ve de donde provienen cada uno de los paquetes, y a donde se dirigen dichos paquetes, a continuación se ejemplifican algunas de las salidas de la bitácora.

MSSFIX(wan):IN=eth1 OUT=eth0 SRC=10.0.0.141 DST=31.13.75.23 LEN=40 TOS=0x00 PREC=0x00 TTL=127 ID=24758 PROTO=TCP SPT=61548 DPT=80 WINDOW=65535 RES=0x00 ACK URGP=0

MSSFIX(wan):IN=eth0 OUT=eth1 SRC=66.220.151.99 DST=10.0.0.141 LEN=245 TOS=0x00 PREC=0x80 TTL=85 ID=29704 DF PROTO=TCP SPT=5222 DPT=53744 WINDOW=21440 RES=0x00 ACK PSH URGP=0

DROP(wan):IN=eth0 OUT= MAC=01:00:5e:00:00:01:58:98:35:34:a9:46:08:00 SRC=192.168.1.254 DST=224.0.0.1 LEN=36 TOS=0x00 PREC=0xC0 TTL=1 ID=63615 DF PROTO=2

MSSFIX(wan):IN=eth1 OUT=eth0 SRC=10.0.0.141 DST=66.220.151.99 LEN=226 TOS=0x00 PREC=0x00 TTL=127 ID=24780 PROTO=TCP SPT=53744 DPT=5222 WINDOW=65535 RES=0x00 ACK PSH URGP=0

MSSFIX(wan):IN=eth0 OUT=eth1 SRC=66.220.151.99 DST=10.0.0.141 LEN=40 TOS=0x00 PREC=0x80 TTL=85 ID=29706 DF PROTO=TCP SPT=5222 DPT=53744 WINDOW=21440 RES=0x00 ACK URGP=0

MSSFIX(wan):IN=eth1 OUT=eth0 SRC=10.0.0.141 DST=199.59.149.232 LEN=48 TOS=0x00 PREC=0x00 TTL=127 ID=24785 PROTO=TCP SPT=55304 DPT=80 WINDOW=65535 RES=0x00 SYN URGP=0

MSSFIX(wan):IN=eth0 OUT=eth1 SRC=199.59.149.232 DST=10.0.0.141 LEN=48 TOS=0x00 PREC=0x00 TTL=46 ID=0 DF PROTO=TCP SPT=80 DPT=55304 WINDOW=14600 RES=0x00 ACK SYN URGP=0

MSSFIX(wan):IN=eth1 OUT=eth0 SRC=10.0.0.141 DST=199.59.149.232 LEN=40 TOS=0x00 PREC=0x00 TTL=127 ID=24786 PROTO=TCP SPT=55304 DPT=80 WINDOW=65535 RES=0x00 ACK URGP=0

MSSFIX(wan):IN=eth1 OUT=eth0 SRC=10.0.0.141 DST=66.220.151.99 LEN=261 TOS=0x00 PREC=0x00 TTL=127 ID=24794 PROTO=TCP SPT=53744 DPT=5222 WINDOW=65535 RES=0x00 ACK PSH URGP=0

MSSFIX(wan):IN=eth0 OUT=eth1 SRC=66.220.151.99 DST=10.0.0.141 LEN=40 TOS=0x00 PREC=0x80 TTL=85 ID=29715 DF PROTO=TCP SPT=5222 DPT=53744 WINDOW=21440 RES=0x00 ACK URGP=0

MSSFIX(wan):IN=eth1 OUT=eth0 SRC=10.0.0.141 DST=199.59.149.232 LEN=48 TOS=0x00 PREC=0x00 TTL=127 ID=24796 PROTO=TCP SPT=55572 DPT=80 WINDOW=65535 RES=0x00 SYN URGP=0

DROP(lan):IN=eth1 OUT= MAC=00:04:25:1c:62:2b:3c:74:37:52:81:08:08:00 SRC=10.0.0.141 DST=10.0.0.1 LEN=29 TOS=0x00 PREC=0x00 TTL=128 ID=24801 PROTO=ICMP TYPE=8 CODE=0 ID=16014 SEQ=7

DROP(lan):IN=eth1 OUT= MAC=00:04:25:1c:62:2b:3c:74:37:52:81:08:08:00 SRC=10.0.0.141 DST=10.0.0.1 LEN=29 TOS=0x00 PREC=0x00 TTL=128 ID=35330 PROTO=ICMP TYPE=8 CODE=0 ID=52318 SEQ=4

PRUEBAS DE FUNCIONAMIENTO DEL FIREWALL, ESCANEO DE PUERTOS CON NMAP.

Un escaneo de puertos es una forma de ataque a un segmento de red o ha un dispositivo de ese segmentro, la intensión de dicho ataque es exponer las vulnerabilidades de la victima en cuestion. El programa *nmap,* es una herramienta que escanea cada uno de los puertos de un dispositivo, en busca de puertos abiertos, para explotar posibles vulnerabilidades y asi lograr ingresar de manera ilegar al sistema que es atacado.

Es por ello que se realizaron pruebas de escaneo de puertos hacia la tarjeta del router desarrollado, para observar su comportamiento durante el ataque, se realizaron los siguientes escaneos:

Un escaneo de puertos hacia una PC perteneciente a la red de prueba y un escaneo de puertos desde la PC hacia la tarjeta del proyecto, ambas pruebas quedaron documentadas en el log del sistema, estas pruebas se realizaron en una red casera, pero se extendieron al laboratorio, en donde previamente se habian realizado las pruebas de enrutamiento, para saber como se comportaba la pared de fuego en una red más estresada, donde existe mayor cantidad de tráfico.

PRUEBAS DE FUNCIONAMIENTO DEL FIREWALL, ESCANEO DE PUERTOS DESDE UNA PC DE LA RED LOCAL HACIA EL ROUTER.

```
root@bt:~# nmap -sU 10.0.0.1
Starting Nmap 5.35DC1 ( http://nmap.org ) at 2012-12-31 20:46 UTC
Nmap scan report for ngw100.lan (10.0.0.1)
Host is up (0.0015s latency).
Not shown: 998 closed ports
PORT   STATE         SERVICE
53/udp open          domain
67/udp open|filtered dhcps
MAC Address: 00:04:25:1C:62:2B (Atmel)

root@bt:~# nmap -O 10.0.0.1
Starting Nmap 5.35DC1 ( http://nmap.org ) at 2012-12-31 20:44 UTC
Nmap scan report for ngw100.lan (10.0.0.1)
Host is up (0.0023s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
2601/tcp open  zebra
2604/tcp open  ospfd
MAC Address: 00:04:25:1C:62:2B (Atmel)
Device type: broadband router
Running: Linux 2.6.X
OS details: OpenWrt Kamikaze - Backfre 10.03 (Linux 2.6.19 - 2.6.32)
Network Distance: 1 hop
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.93 seconds

root@bt:~# nmap -sP 10.0.0.0/24
Starting Nmap 5.35DC1 ( http://nmap.org ) at 2012-12-31 20:42 UTC
Nmap scan report for ngw100.lan (10.0.0.1)
Host is up (0.0089s latency).
MAC Address: 00:04:25:1C:62:2B (Atmel)
Nmap scan report for bt.lan (10.0.0.154)
Host is up.
Nmap done: 256 IP addresses (2 hosts up) scanned in 4.26 seconds
root@bt:~#

root@bt:~# nmap -PN 10.0.0.1
Starting Nmap 5.35DC1 ( http://nmap.org ) at 2012-12-31 20:43 UTC
Nmap scan report for ngw100.lan (10.0.0.1)
Host is up (0.018s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
53/tcp   open  domain
80/tcp   open  http
2601/tcp open  zebra
2604/tcp open  ospfd
MAC Address: 00:04:25:1C:62:2B (Atmel)

Nmap done: 1 IP address (1 host up) scanned in 1.82 seconds
root@bt:~# nmap -Pn 10.0.0.1

Starting Nmap 5.35DC1 ( http://nmap.org ) at 2012-12-31 20:44 UTC
Nmap scan report for ngw100.lan (10.0.0.1)
Host is up (0.011s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
```

22/tcp open ssh
53/tcp open domain
80/tcp open http
2601/tcp open zebra
2604/tcp open ospfd
MAC Address: 00:04:25:1C:62:2B (Atmel)

PRUEBAS DE FUNCIONAMIENTO DEL FIREWALL, ESCANEO DE PUERTOS DESDE EL ROUTER HACIA UNA PC EN LA RED LOCAL

root@ngw100:~# nmap -O 10.0.0.154
Starting Nmap 5.51 (http://nmap.org) at 2013-01-02 04:06 CST
Nmap scan report for 10.0.0.154
Host is up (0.0014s latency).
All 1000 scanned ports on 10.0.0.154 are closed
MAC Address: 00:08:54:91:08:B8 (Netronix)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Aerohive HiveAP 320 WAP (HiveOS 3.4) (98%), Aruba 3400 or 6000 wireless LAN controller (ArubaOS 3.3.2) (98%), AT&T 3G MicroCell WAP (98%), AXIS 211A Network Camera (Linux 2.6) (98%), AXIS 211A Network Camera (Linux 2.6.20) (98%), AXIS Network Camera (207, 221, or 232D+) or 241Q Video Server (Linux 2.6.16 - 2.6.17) (98%), Buffalo TeraStation Pro III NAS device (98%), Check Point SBox-200 firewall (98%), Check Point VPN-1 UTM appliance (98%), Chip PC XtremePC thin client (98%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 51.55 seconds

root@ngw100:~# nmap -sU 10.0.0.154
Starting Nmap 5.51 (http://nmap.org) at 2013-01-02 04:09 CST
Nmap scan report for 10.0.0.154
Host is up (0.0016s latency).
Not shown: 999 closed ports
PORT   STATE   SERVICE
68/udp open|filtered dhcpc
MAC Address: 00:08:54:91:08:B8 (Netronix)

PRUEBAS DE FUNCIONAMIENTO DEL FIREWALL, INTENTO DE CONEXÍON HACIA UN PUERTO PROTEGIDO POR REGLAS DEL FIREWALL

Se realiza un intento de comunicación hacia un puerto que esta cerrado por la pared de fuego, por lo que desde una PC de la red local se intenta establecer una conexion hacia un puerto especifico en el firewall, lo cual queda registrado la bitacora del sistema en reiteradas ocaciones, dado que algunas de las reglas que se establecieron son de mayor rigidez que otras, se puedo comprobar que se tiene un mayor nivel de seguridad en las comunicaciones.

Primeramente se toma como punto de partida de la seguridad en las pruebas, la contraseña asignada en el punto de acceso de la red, ya que para el caso de las pruebas realizadas en la red casera se utilizo un punto de acceso inalambrico, como siguiente nivel de seguridad se generan reglas en el firewall que identifican la dirección física (MAC Address) de cada dispositivo que intenta o accesa a la red aun que existe la posibilidad de burlar este nivel de seguridad a traves de programas como macchanger, Technitiumque, etc, que generan una MAC Address ficticia (clon) y modifican las tramas enviadas a traves de la red hacia la pared de fuego.

Finalmente se realiza una asociación de esas reglas a la IP de cada uno de los dispositivos, por lo que se sugiere utilizar para maquinas de la red IP fijas y para maquinas de visitas seria conveniente

se configurara una vlan con restricciones.

Quedaría pendiente un nivel mas de seguridad que no se implementa en este momento, pero que existe la posibildad de hacer en un futuro, con dispositivos externos, un servidor RADIUS, similar a un servidor de tacacs de autentificación. Con este nivel de seguridad se estaria integrando un sistema muy complejo de autentificación para redes de computadoras, que incrementaria en gran medida el nivel de seguridad al interior de una organización

PRUEBAS DE FUNciónAMIENTO DEL FIREWALL, ATAQUE DoS AL FIREWALL.

Finalmente, se realiza un ataque DoS[32], a traves de la herramienta llamada slowloris.pl, que no es más que un script programado en perl[33] para realizar este tipo de ataques en contra de servidores web y servidores de archivos.

Esta herramienta genera varios hilos[34], para poder crear diferentes procesos y sockets, y así simular un multiple ataque hacia un servidor web, en este caso se intento dejar fuera de servicio el servidor web uhttp de la tarjeta de NGW100, pero el resultado de esta prueba demostro que la pared de fuego a traves de la instrucción *option syn_flood **1*** pudo contener el ataque de este tipo.

La unica diferencia perceptible en el momento del ataque fue la extrema lentitud con la que contestaba la pagina web del dispositivo, pero en ningun momento nos indico que no podia atender la petición hecha.

Terminada la prueba con la pared de fuego se procedio a desactivarla para ver la comparación entre la pared de fuego activa y desactivada, en este caso se volvio a atacar al servidor web de la tarjeta NGW100 con la misma instrucción con la que se realizo el ataque anterior:

*root@bt:~# **./slowloris.pl -dns 10.0.0.1 -port 80 -timeout 1 -num 1000 -cache***

Ambos ataques se realizaron desde la ultima versión de backTrack Linux en una PC dentro de la red, y en esta ocación fue evidente que la pagina web no respondio a las peticiones despues de varios segundos de ataque al servidor web, por lo que se dejo correr aun mas el ataque, hasta el punto de llegar a los casi 27000 paquetes enviados.

Estas son las salidas del programa slowloris.pl en donde se puede apreciar la cantidad de paquetes enviados hacia el firewall durante la prueba, la figura 21 muestra el resultado de la prueba cuando el firewall fue deshabilitado.

Building sockets.
Sending data.
Current stats: Slowloris has now sent 26024 packets successfully.
This thread now sleeping for 1 seconds...

Building sockets.
Sending data.

32 Es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima.

33 Lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado bourne shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

34 Unidad de procesamiento más pequeña que puede ser planificada por un sistema operativo. La creación de un nuevo hilo es una característica que permite a una aplicación realizar varias tareas a la vez (concurrentemente)

Current stats: Slowloris has now sent 26128 packets successfully.
This thread now sleeping for 1 seconds...

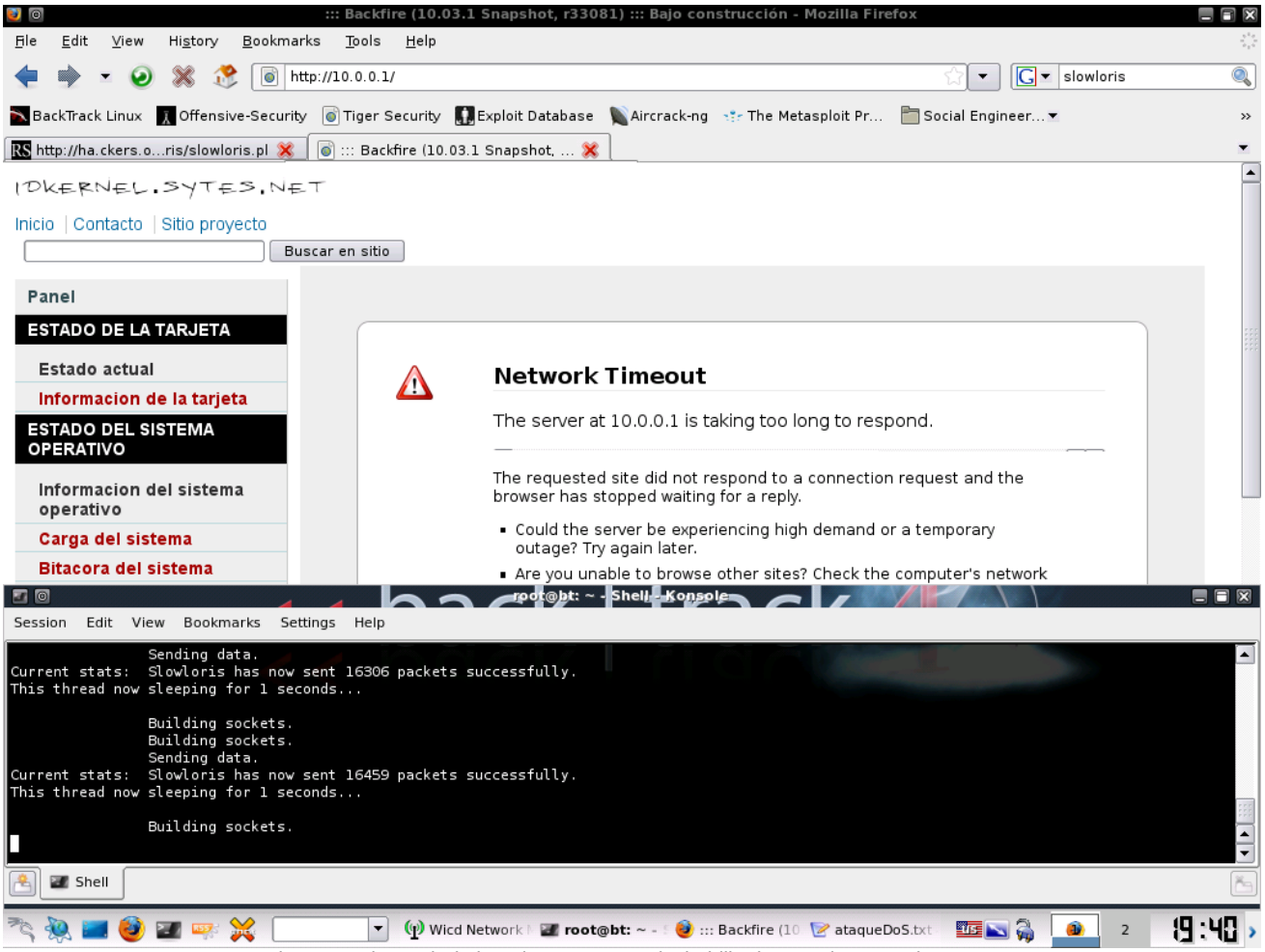


Fig. 21 Pagina web de la tarjeta NGW100 deshabilitada tras el ataque de DoS.

CONCLUSIONES

Al finalizar el presente trabajo, se concluye que:

Se da por teminado el proyecto, dado que se lograron cumplir los objetivos mínimos planteados al principio del presente documento. Después de realizar la pruebas pertienentes se concluye que este dispositivo de enrutamiento funciona apropiadamente con la distribución que le fue instalada, se verifican que los niveles de procesamiento en el cpu, no pasan del 1% sostenido.

Se logra alcanzar una de las metas mas importantes del proyecto, poder administrar el sistema desde un entorno basado en web, los scripts que realizan la funcion de interface con el sistema operativo, se ejecutan correctamente hasta este punto.

Dentro de la parte de enrutamiento, se valida que está funcionando y realiza su cometido, las validaciones del enrutamiento fueron hechas tanto en redes físicas como en la red virtual utilizando el

simulador GNS3, solo quedaría pendiente la revisión del protocolo de enrutamiento que no funcionó.

Debido a lo estrecho del tiempo muchas de las aplicaciones que corren bajo esta arquitectura ya no pudieron ser probadas, dentro de estas aplicaciones, se encuentra, servidores de correo electronico, servidores de VPN, servidored de archivos, telefonia sobre IP, etc. Por tal motivo se deja en la memoria secunadaria SD la compilacion de toda la paqueteria que soporta la distribución.

BIBLIOGRAFÍA

[1] Tanenbaum Andrew S, “Diseño e Implementación” en Sistemas operativos, Prentice Hall, Tercera Edición, 1998, Capítulo 3, pp.115

[2] Raj Kamal, “Architecture, programming and design” en Embedded systems, Tata-McGraw Hill, Segunda edición, 2009, Capitulos 1, 7, 8.

[3] Stallings William, Sistemas operativos, NY:Pearson education, 2005.

[4] Jonathan Corbet, “Linux Device Drivers”, O'Reilly, Tercera Edición, 2004, Capitulo 2, pp. 15-39

[5] A. O. Mulani, Embedded systems, Primera edición, India, 2009, Capitulo 6, pp 6-6

[6] Daniel Bovet, “Understanding the Linux Kernel”, O'Reilly, Tercera Edición, 2005, Capitulo 3, pp, 79-

[7] Atmel. (2010,05,25). Mature NGW100 Network Gateway Kit [Online]. Available: http://www.atmel.com/tools/MATURENGW100NETWORKGATEWAYKIT.aspx

[8] Angstrom Linux. (2009.11,15). The Ångström Distribution, Embedded power [Online]. Available: http://narcissus.angstrom-distribution.org/

[9] Florida State University. (2010,05,22). Ant Colony Optimization used in Network Routers [Online]. Available: http://ww2.cs.fsu.edu/~guidry/AI/AntColonyOptimization.pdf

[10] Craig Hollabaugh, "Hardware, Software, and Interfacing" en Embedded Linux, Pearson edication corporate, Primera edición, IN:Pearson, 2002, Capitulos 2, 4, 6, y 9.

Ken O. Burtch, "A comprehesive guide and reference for linux users and administrators" en Linux shell scripting with bash, Primera edición, USA, 2004, Capitulos 2-6, pp. 13-113.

Gene Sally, "Linux embedded system", Apress, Segunda edición, USA, 2010, Capítulos 7, 11, 14.

Excercise Linux. (2011,08,15). Installing a Beagle OS [Online]. Available: http://www.elinux.org/EBC_Exercise_00_Installing_Angstrom_on_SD

RoboHobby. (2011,10,06). How to use ATMEL NGW100 board for Java robotics [Online]. Available: http://www.robohobby.com/atmel_ngw100_hobby_java_robotics.jsp

Alyda. (2011,08,03). AVR32 Network Gateway 100 development board for getting started with embedded Linux [Online]. Available: http://www.alyda.nl/index.php?option=com_content&task=view&id=49&Itemid=43

Yagarto. (2008,12,15). Atmel NGW100 [Online]. Disponible en: http://www.emb4fun.de/avr/avr32ngw100/index.html

Hallinan Cristopher, "A Practical Real-World Approach" en Embedded Linux Primer, Prentice Hall, Segunda Edición, 2010.

Yaghmoun Karin, Building Embedded Linux Systems, CA:O'Reilly Media, 2008

APENDICE

Proceso de instalación de la imagen del sistema operativo en la memoria SD:

Listamos todos los dispositivos montados en la terminal con el comando:***mi_maquina:~ # fdisk -l***

1. Esperamos que la salida sea similar a la siguiente:

Device	Boot	Start	End	Blocks	id	System
/dev/sde1		249	1983743	991747+	83	Linux

2. Generamos una carpeta temporal en la trayectoria ***/tmp***, para depositar ahí el contenido del sistema operativo descargado de la página angstrom linux, cuyo formato es zip, el comando para llevar a cabo esta operación es:

mi_maquina:~ $ mkdir -p /tmp/carpeta_temporal

3. Montamos la memoria SD en la carpeta temporal, para este ejemplo el comando completo quedaria de la siguiente manera:***mi_maquina:~ # mount /dev/sde1 /tmp/carpeta_temporal***

4. Desempacamos en /tmp/carpeta_temporal el zip resultante de la generación del kernel de la pagina de la distribución Amstrong Linux:

mi_maquina:~ # tar -vxf random-6851924c-image-atngw100.tar -C /tmp/carpeta_temporal/

5. Finalmente sincronizamos la escritura con el comando:***mi_maquina:~ $ sync*** como usuario ***sin privilegios***.

6. Desmontamos la unidad: ***mi_maquina:~ # umount /tmp/carpeta_temporal***

Nota: este proceso debera ser prescedido por un formateo tipo ext2 a la memoria, con una configuración de 128 inodos.

Proceso de formateo a 128 i-Nodos con mkfs.ext2

Mensaje de error a la hora de cargar el sistema operativo desde la memoria SD-1Gb:
Unable to read "/boot/uImage" from mmc 0:1
##Booting image at 00800000 ...
Bad Magic Number

La posible solución es la siguiente:
"Newer versions of ext2 filesystem code use by default 256 byte inodes (and thus, newer versions of Linux). However, the current u-boot code (as of 2008.10) seems to not handle this, and this issue can be solved by creating the ext2 filesystems with 128 byte inodes via the -I 128 option. If you have an old version of linux, it probably already worked because it may not have supported 256 byte inodes."

Para lo cual seguiremos las siguientes instrucciones:

1. Insertar la tarjeta en el lector de memorias SD.

2. Utilizar el comando ***df*** para ubicar el punto de montaje del dispositivo, esto es similar a:

```
Filesystem  1k-blocks  ...  Mounted on
/dev/hda1   ...        ...  ...
...         ...        ...  ...
/dev/sda1   ...        ...  /media/usbdisk-1
```

3. Se asume que ***/dev/sda1*** esta montada en ***/media/usbdisk-1.***

4. Para formatear la memoria SD y crear un sistema de archivos de tipo ext2 que soporte inodes de 128 bytes es necesaria la siguiente lista de instrucciones como usuario root:

- ***# umount /media/usbdisk-1***
- ***# /sbin/e2fsck /dev/sda1***
- ***# /sbin/mkfs.ext2 -I 128 -L etiqueta_SD /dev/sda1***
- ***# mount /dev/sda1 /media/usbdisk-1***

Proceso de actualización del gestor de arranque das UBoot

Para poder realizar el proceso de actualización del Uboot, es necesario contar con un cableado RS-232 a DB-9, que será la interconexión a la consola, durante el proceso de arranque natural de la tarjeta de desarrollo NGW100, para poder acceder al UBoot, debemos utilizar el programa minicom.
En los primeros segundos se debera presionar la tecla space para acceder al gestor de arranque. Desde este punto le indicaremos al actual gestor que sera leido el archivo de actualización de la memoria SD:
Nota muy importante: Este proceso no debera ser suspendido ya sea por alimentación eléctrica o por detener la ejecución durante la instalación de la nueva versión del UBoot, ya que se corre el riesgo de dañar el gesto de arranque, lo cual provocaria que la tarjeta quedará inservible, por lo que se recomienda, tener al menos un respaldo eléctrico al realiza este procedimiento.

Despues de haber tomado en cuenta los riesgo que implica esta sección del proyecto, se procede a realizar el proceso de actualización, para ello tomamos como referencia la pagina holandesa alyda, que proporciona las instrucciones que fueron llevadas a cabo en la actualización de esta tarjeta y de una segunda tarjeta propiedad del profesor Arturo Zuñiga Lopez.

Para realizar el proceso de actualización del gestor de arraque de la tarjeta NGW100 desde la memoria secuandaria SD, se necesita realizar una conexion por medio de la consola a traves del RS232 a BD-9 conexion que se realiza por un puerto serial de la computadora, sera necesario descargar la imagen de la actualización de la siguiente dirección:
http://www.atmel.no/buildroot/buildroot-u-boot.html

Esta imagen se llama ***flash-upgrade-atngw100-v2008.10.uimg*** es la versión actualizada del UBoot, que corrige el error de lectura de memorias con capacidades mayores a 512Mb, error que con anterioridad se habia mencionado.
El BuildRoot de OpenWrt tiene la posibilidad de generar imagenes de UBoot, el menu principal de dicho BuildRoot muestra que junto con la generación de una imagen del kernel, OpenWrt tiene la posibildad de generar su propia version de UBoot si asi se desea.

Una ves obtenida la imagen de la actualización, sea cual sea el origen de la misma, necesitaremos una memoria de tipo SD de poca capacidad, esto es entre 32Mb y 512Mb, para que el sistema operativo que posee la tarjeta y sobre todo el UBoot la reconozca y pueda acceder a la información.
Seran necesario formatear la SD, de ser posible se recomienda utilizar Ubuntu Linux, su sistema de archivos debera ser el EXT-2.

Despues de terminal proceso de formateado se coloca la imagen llamada ***flash-upgrade-atngw100-v2008.10.uimg*** en la memoria recien formateada, se inserta en la ranura de memoria SD que posee la tarjeta de desarrollo NGW100 y se procede a alimentarla con energia eléctrica.
Nota: Para poder acceder a UBoot, es necesario contar con el cable de consola conectado a la PC y a la tarjeta de desarrollo, y en cuanto se alimente la tarjeta, se debera presionar la tecla SPACE para detener la carga del sistema operativo, que viene precargada, esto permite el acceso al prompt del programa UBoot

Dentro del programa UBoot, una de las primeras cosas que podemos observar es la version que esta instalada en la tarjeta:
U-Boot 1.1.4-at0 (Jan 3 2007 – 10:30:09)
Esta versión fue compilada en el año 2007, por Atmel para integrala a sus tarjetas empotradas NGW100 y ATNGW1000.

El primer comando que se introducira al prompt[35] del UBoot es el siguiente:
Uboot > ***mmcinit***

La función de este comando es la de mostrar las características de la memoria mmc, es casi un hecho que si este comando no logra leer la memoria que esta depositada en el slot SD de la tarjeta NGW100, el proceso de formateo de la memoria fué incorrecto, o presenta una daño el sistema de archivos, será necesario nuevamente revisar el formato de la memoria, para ello tenemos los comando de la PC ***fdisk*** en modo consola o bien ***gparted*** en modo gráfico.
Y en tal caso tambien será conveniente que se revise la integridad de la memoria con el comando ***e2fsck,*** este comando en caso de encontrar alguna irregularidad, la reparará.

Las siguiente, es una salida del comando *mmcinit,* funcionando correctamente.

```
Manufacturer ID:                03
OEM/Application ID:             5344
Product name:                   SD128
Product Revision:               5.8
Product Serial Number:          6496992
Manufacturing Date:             05/10
SD Card detected (RCA 58916)
CSD data: 00260032 1f5983c0 fefa4fff 924040ab
CSD structure version:          1.0
MMC System Spec version:        0
Card command classes:           1f5
Read block length:              512
Supports partial reads
Write block length:             512
Does not support partial writes
Supports group WP:              32
Card capacity:                  125960192 bytes
File format:                    0/0
Write protection:
```

Las partes que han sido seleccionadas con color rojo son las mas importantes caracteristicas de la memoria.

35 Se llama prompt al carácter o conjunto de caracteres que se muestran en una línea de comandos para indicar que está a la espera de órdenes. Éste puede variar dependiendo del intérprete de comandos y suele ser configurable.

El segundo comando a utiliza es el siguiente:
Uboot > ***ext2ls mmc 0:1***

Este comando lista el contenido de la memoria, de la primera partición la salida de este comando es la siguiente:

```
....<DIR>          1024 .
....<DIR>          1024 ..
....<DIR>          12288 lost+found
....<DIR>          1024 .Trash-1000
....   70447 flash-upgrade-atngw100-v2008.10.uimg
```

En esta memoria se puede observar que al final del listado está la imagen de actualización del gestor de arranque, misma que se utilizará en el siguente comando.

El siguiente comando es el siguiente:
Uboot > ***ext2load mmc 0:1 0x10400000 flash-upgrade-atngw100-v2008.10.uimg***

Este comando ordena a la tarjeta NGW100 leer la memoria SD y cargar de la primera partición hacia la dirección de memoria 104000000 de la NGW100, el archivo ***flash-upgrade-atngw100-v2008.10.uimg.***

En este momento se tiene cargada en memoria la actualización del gestor de arranque, la salida de este comando puede variar, pero se presenta de la siguiente manera:

```
.....
70447 bytes read
```

Finalmente se ejecutara el archivo en la localidad de memoria 0x10400000 con el siguiente comando.
Uboot > ***bootm 0x10400000***

Despues de presionar la tecla Enter ya no hay paso hacia atras, se ejecutará la actualización, es en este momento en el que la tarjeta queda más expuesta a fallos y errores, ya que en este momento se retira temporalmente la protección contra escritura y se escribe directamente en ella, es por ello que se da la recomendación que este proceso, asi como todas las actualizaciones de firmware sean hechas con soporte eléctrico de emergencia (No break), ademas de no interrumpir tal procedimiento.

Durante el proceso el programa de instalación hará la siguiente pregunta:

Going to copy 104024 bytes to offset 0x00000000 in flash
Press `y' to continue, or any other key to abort

La actulización no deberá demorar más de 15seg. Una ves realizado el acto la tarjeta activa nuevamente la protección contra escritura de la memoria donde reside el cargador de arraque, si todo ha salido correctamente el programa de instalación pedira que sea reinciada la tarjeta.

Flash upgrade successful. Please press reset or cycle power.

La salida completa de tal proceso es la siguiente:

```
Uboot> bootm 0x10400000
## Booting image at 10400000 ...
Image Name:                  AVR32 Flash Upgrade Utility v0.2
Image Type:                  AVR32 Linux Kernel Image (gzip compressed)
Data Size:                   70383 Bytes = 68.7 kB
Load Address:                10000000
Entry Point:                 90000000
Verifying Checksum ... OK
Uncompressing Kernel Image ... OK
Starting kernel at 90000000 (params at 11fc0040)...
AVR32 Flash upgrade utility version 0.2
HSMC configuration:          0x00030001 0x06030504 0x00080008 0x00001103
Atmel AT49BV642D found at address 0x00000000
cfi: using AMD/Fujitsu command set
cfi: 2 erase regions (total size: 8388608 bytes)
0 8 sectors, 8192 bytes each
1 127 sectors, 65536 bytes each
Going to copy 104024 bytes to offset 0x00000000 in flash
Press `y' to continue, or any other key to abort
Erasing... done
Programming... done
Verifying... done
Flash upgrade successful. Please press reset or cycle power.
```

Una ves reiniciada la tarjeta se puede observar la versión nueva del gestor de arranque que es la siguiente:

U-Boot 2008.10 (Apr 16 2009 – 10:33:49)

He inmediatamente procedera a realizar la carga de sus sistema operativo que posee de fabrica

GLOSARIO

- ***FW***: Abreviatura de firewall o pared de fuego.

- ***Memoria SD***: Secure Digital (SD) es un formato de tarjeta de memoria inventado por Panasonic. Se utiliza en dispositivos portátiles tales como cámaras fotográficas digitales, PDA, teléfonos móviles, computadoras portátiles e incluso videoconsolas (tanto de sobremesa como portátiles), entre muchos otros.

- ***Procesamiento por lotes***: Ejecución de un programa sin el control o supervisión directa del usuario.

- ***Scripts***: Archivo de órdenes o archivo de procesamiento por lotes, es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

- ***GNS3:*** Simulador grafico de redes complejas de computadoras, este simulador tiene la peculiaridad de correr versiones originales de los ios de Cisco, ademas tambien soporta versiones de JunOS.