DISEÑO E IMPLEMENTACION DE UN PROTOTIPO DE LA ENTIDAD ELEMENTO DE CONTROL DE LA ARQUITECTURA FORCES

Autor:

SUSAN COSTANZA MARTINEZ CORDERO

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA ELECTRÓNICA
BOGOTÁ - COLOMBIA.
NOVIEMBRE DE 2012

DISEÑO E IMPLEMENTACION DE UN PROTOTIPO DE LA ENTIDAD ELEMENTO DE CONTROL DE LA ARQUITECTURA FORCES

Autor:
SUSAN COSTANZA MARTINEZ CORDERO
Ingeniero Electrónico

Proyecto de Grado presentado como requisito para optar el titulo de:
MAGISTER EN INGENIERIA ELECTRONICA

Director
LUIS CARLOS TRUJILLO ARBOLEDA, MsC
Ingeniero Electrónico

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA ELECTRÓNICA
BOGOTÁ - COLOMBIA.
NOVIEMBRE DE 2012

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERÍA

MAESTRÍA EN INGENIERÍA ELECTRÓNICA

RECTOR MAGNÍFICO: R.P. JOAQUIN EMILIO SANCHEZ GARCÍA S.J

DECANO ACADÉMICO: Ing. LUIS DAVID PRIETO, Ph.D

DECANO DEL MEDIO UNIVERSITARIO: R.P SERGIO BERNAL RESTREPO S.J

DIRECTOR DE MAESTRÍA: Ing. CESAR LEONARDO NIÑO, Ph.D

NOTA DE ADVERTENCIA

"La Universidad no se hace responsable de los conceptos emitidos por algunos de sus alumnos en los proyectos de grado. Solo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ello el anhelo de buscar la verdad y la justicia."

Artículo 23 de la Resolución No. 13, del 6 de julio de 1946, por la cual se reglamenta lo concerniente a Tesis y Exámenes de Grado en la Pontificia Universidad Javeriana.

Nota de aceptación

_____

_____

_____

_____

Presidente del Jurado

_____

Jurado

_____

Jurado

Bogotá, 19 de Noviembre de 2012

AGRADECIMIENTOS

A Dios, mi esposo y mi hijo.

TABLA DE CONTENIDO

TABLA DE FIGURAS

1. INTRODUCCION

Con el incremento en la demanda de tráfico, según informe realizado por Cisco [15], y con la prestación de nuevos servicios de voz, datos y video entre otros, las redes de datos, tanto privadas como Internet, requieren equipos que soporten dicha demanda, que sean fiables en la transmisión de información, y que presenten una arquitectura escalable y gran capacidad de procesamiento. Actualmente los equipos que prestan estos servicios son propietarios, tienen una arquitectura cerrada, un costo elevado, y no son escalables; por lo tanto si la red requiere un nuevo servicio y el equipo no lo soporta, es necesario cambiar el dispositivo por uno actualizado, llevando a que cada vez que se realice un mejoramiento en la red, se necesite un cambio de dispositivos, incrementando costos y creando una dependencia con los fabricantes.

Para efectuar una mejora en los dispositivos y en los servicios que prestan en las redes de comunicaciones, los organismos de estandarización se han interesado en la creación de interfaces y elementos de red con una arquitectura abierta, programable, escalable y a un bajo costo. La arquitectura ForCES, creada y estandarizada por el grupo de trabajo de redes IETF ForCES [14], permite desarrollar un elemento de red basándose en la separación física de los elementos del plano de control y plano de datos, especificando un protocolo que permite la comunicación entre dichos elementos.

La arquitectura ForCES ha sido de gran aceptación por varios grupos de investigación a nivel académico como, en la Universidad de Zhejiang Gongshang, el Instituto de Redes e Ingeniería de Comunicaciones en China, Universidad de Ben Gurion en Israel, Universidad de Stanford entre otras, ya que los componentes de un elemento de red, elementos de control (CE) en el plano de control y elementos de reenvío (FE) en el plano de datos, se pueden estudiar de una manera individual, mejorando sus funciones y servicios, para posteriormente integrarlos y formar un solo dispositivo flexible y escalable.

Existen varios proyectos en los que se han realizado trabajos de investigación relacionados con esta arquitectura y en los que se encuentran los siguientes: [4] “ForCES protocol as a solution for interaction of control and forwarding planes in distributed routers”, en el cual se realiza un análisis del protocolo de comunicación ForCES, usado entre los elementos de control y reenvío en un enrutador distribuido, con las especificaciones del grupo de trabajo IETF ForCES. Con base en este análisis, proponen un modelo para el procedimiento de creación de túneles MPLS LSP (Multiprotocol Label Switching-Label Switched Path) aplicable al enrutador distribuido en el ambiente ForCES. En [2] “Analysis and Implementation of an Open Programmable Router Based on Forwarding and Control Element Separation”, se ilustra, el diseño e implementación de un enrutador basado en ForCES. En este artículo, inicialmente se explica la arquitectura ForCES, luego, describe un modelo de software en capas, que ilustra muy bien las características de ForCES, además crea un modelo basado en el diseño e implementación del elemento de control (CE) y Elemento de reenvío (FE), el cual es denominado ForTER, adicionalmente consideran aspectos de seguridad y presentan un algoritmo para prevenir ataques DoS (Denegación de Servicio). Por

último, los experimentos de ForTER se ilustran ejecutando los protocolos de enrutamiento, gestión de redes y prevención contra ataque de DoS, entre otros.

En [22], Inicialmente realiza un análisis en el diseño modular descentralizado, que mejoraría la escalabilidad, flexibilidad y confiabilidad de los routers futuros, luego muestra el diseño e implementación de un router distribuido, basado en la separación física de los diferentes módulos funcionales de los planos de operación: control y datos (arquitectura ForCES) y la comunicación entre ellos por un protocolo diseñado con el apoyo de los mensajes NetLink.

Los artículos anteriores han sido de gran importancia y guia, pero el trabajo de investigación que aporto en gran parte a este proyecto fue el realizado por los investigadores Xiaochun Wu, Ligang Dong, Weiming Wang, Bin Zhuge, Ming Gao, Fenggen Jia, Rong Jin, Jin Yu, del Institute of Network and Communication Engineering Zhejiang Gongshang University, Hangzhou, China. Este grupo diseño e implemento un router con arquitectura y protocolo de interconexión ForCES, su último trabajo de investigación sobre el tema fue el diseño de un sistema de alta disponibilidad, el cual permite configurar un elemento de control como primario y otro como respaldo, para no perder interoperabilidad y asegurar la comunicación en llegado caso de una falla. De esta gran investigación hacen parte [1],[5],[18],[19],[20],[21],[23].

En conclusión, los resultados experimentales demuestran la factibilidad del diseño de un router o elemento de red modular y la viabilidad de la arquitectura ForCES.

Este campo de investigación, de nuevas arquitecturas de elementos de red, es uno de los temas de trabajo del grupo de investigación Sistemas de Telecomunicaciones de la Pontificia Universidad Javeriana, SISCOM. El presente proyecto se centra en el desarrollo e implementación de uno de los componentes que hacen parte de la arquitectura ForCES, en particular en el elemento de control.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Diseñar e implementar un prototipo de elemento de control para un router con base en la arquitectura ForCES.

2.2. OBJETIVOS ESPECIFICOS

1. Diseñar e implementar un esquema que permita al Elemento de Control conocer y gestionar los Elementos de Forwarding en un Elemento de Red.

2. Crear e implementar un módulo que integre las funcionalidades básicas del elemento de control como son enrutamiento y gestión.

3. Diseñar e implementar un esquema para la operación con múltiples Elementos de Control.

4. Diseñar e implementar un entorno de pruebas que permita validar la funcionalidad del Elemento de Control desarrollado.

3. MARCO TEORICO

3.1 ARQUITECTURA DEL ENRUTADOR

Según el RFC 1812 [7], un enrutador, es un dispositivo electrónico, para interconexión de redes que opera en el nivel de red del modelo de referencia OSI. Tiene la capacidad para distribuir o encaminar por medio de protocolos de enrutamiento cada paquete de información que recibe y decidir la mejor ruta de envío al destino. Estas decisiones lógicas las toma dependiendo de las características de la red como densidad de tráfico, velocidad del enlace, y ancho de banda, entre otras.

Un enrutador está encargado de desempeñar dos tareas fundamentales:

1. Enrutamiento: El proceso de enrutamiento busca una ruta entre todas las posibles en una red para que un paquete llegue a su destino, esto lo hace mediante la información que se intercambia entre enrutadores vecinos, usando diferentes protocolos de enrutamiento. Las mejores rutas son almacenadas en una estructura de información denominada Tabla de Enrutamiento.
2. Reenvío de paquetes (forwarding): El proceso de reenvío de paquetes, es aquel que transporta un paquete desde una interfaz de entrada de un enrutador, a la interfaz de salida apropiada, del mismo enrutador, basado en la información contenida en una estructura de información denominada tabla de Reenvío.

Según [7], la arquitectura básica de un enrutador está compuesta de dos partes principales, como se observa en la figura 1:

3.1.1. PLANO DE CONTROL

Es el plano donde el enrutador decide cual es la interface de salida apropiada para la transmisión de los paquetes a determinados destinos.

El Plano de Control construye la tabla de enrutamiento con el conocimiento de sus interfaces locales y del intercambio de información de los protocolos de enrutamientos con otros enrutadores. La tabla de enrutamiento almacena las mejores rutas a determinados destinos de la red, las métricas de enrutamiento asociadas con esas rutas y el camino al próximo enrutador. En este plano también se realizan las funciones de gestión de red.

3.1.2. PLANO DE REENVIO

El plano de reenvío se conoce también como plano de datos. Entre las decisiones más importantes del reenvío está decidir qué hacer cuando se produce congestión. En este plano también se construye la tabla de reenvío, que es consultada por el enrutador para determinar la interface de salida donde necesita reenviarse un paquete entrante, entonces cada entrada en la tabla de reenvío

mapea un prefijo IP a una interface de salida. Dependiendo de la implementación, las entradas deben contener información adicional tal como direcciones MAC para el próximo salto y estadísticas acerca del número de paquetes reenviados a través de una interface.

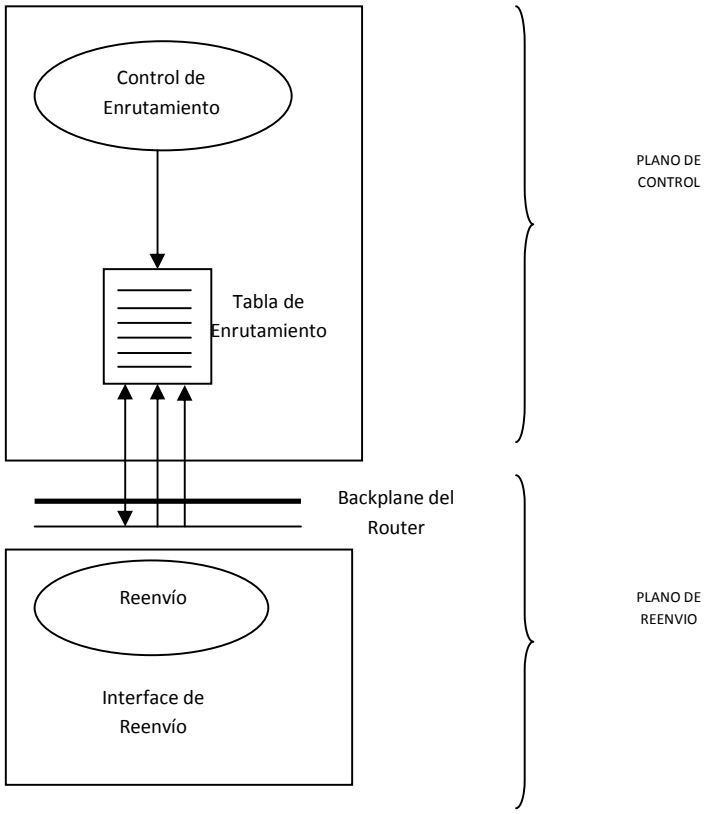


Figura 1. Elementos de un enrutador (Elaborado por, Susan Martínez, basado en el RFC 1812)

3.2 FORCES (FORWARDING AND CONTROL ELEMENT SEPARATION: SEPARACIÓN DE LOS ELEMENTOS DE REENVÍO Y CONTROL)

Un elemento de red (NE, Network Element), sea un enrutador o un switch, está compuesto básicamente por numerosas entidades lógicas separadas que interactúan entre sí para proporcionar una funcionalidad determinada como por ejemplo, enrutamiento; estas entidades se conocen como elementos de control (CE, Control Element) , que hacen parte del Plano de Control y elementos de reenvío (FE, Forwarding Element) en el plano de Datos, los cuales se interconectan y comunican entre sí por medio de un protocolo de comunicación; sin embargo a entidades externas, aparecen como un elemento de red integrado. ForCES proporciona un conjunto estándar de mecanismos para la conexión de estos componentes ofreciendo mayor escalabilidad y permitiendo a los planos de control y reenvío evolucionar de forma independiente, promoviendo así una rápida innovación [8].

3.2.1. ARQUITECTURA FORCES.

La arquitectura ForCES está compuesta por uno o más elementos de control, CE, uno o más elementos de reenvío, FE y un protocolo de comunicación que permite la interacción entre estos elementos, como se observa en la figura 2. El CE es responsable de operaciones como la señalización e implementación de los protocolos de administración y enrutamiento. Basado en la información adquirida a través del procesamiento de información de las funciones básicas, los CE determinan el comportamiento de los paquetes transmitidos por los FE, utilizando el protocolo de interconexión. Por ejemplo, el CE puede controlar a un FE por medio de la manipulación de las tablas de reenvío y el estado de sus interfaces. Los FE operan en el plano de reenvío y son los responsables del procesamiento y manejo de los paquetes. Algunas funciones que desarrollan los FE incluyen el reenvió, firewall, NAT (Traducción de Direcciones de Internet), encapsulación, desencapsulación, y encriptación, entre otras.

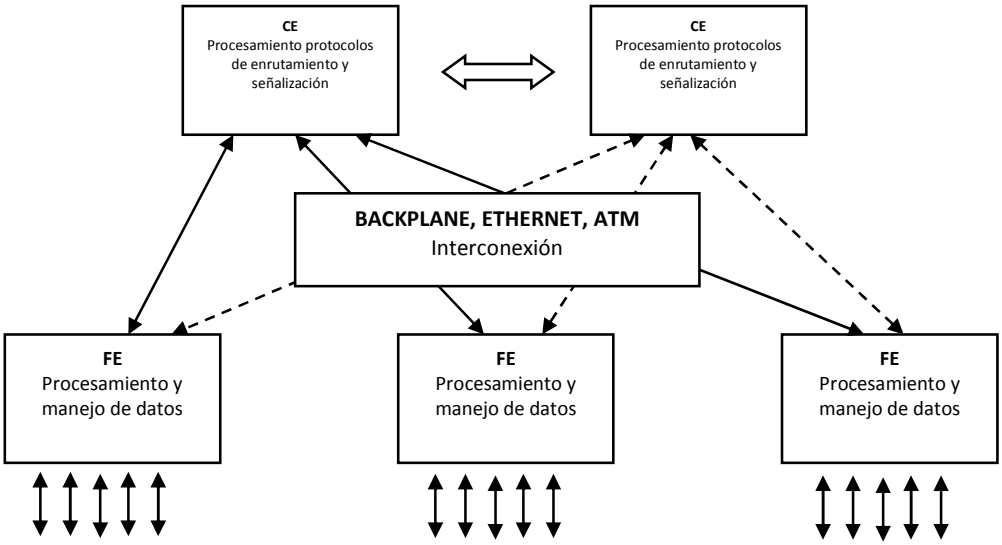


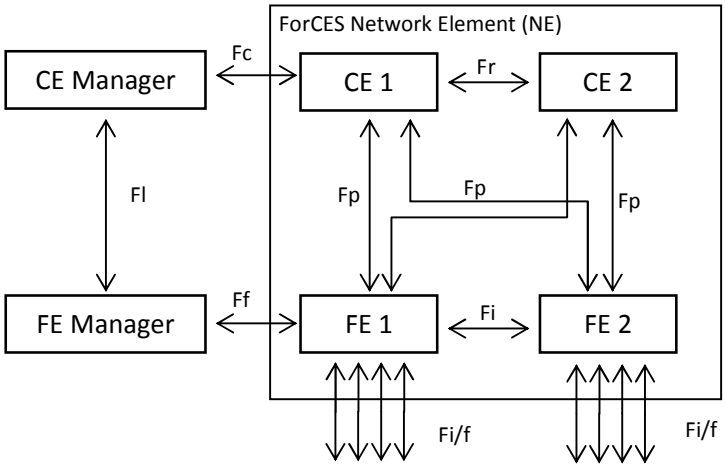
Figura 2. Arquitectura ForCES. (Elaborado por Susan Martínez, basado en el RFC 3654))

3.2.2. PROTOCOLO FORCES

El protocolo ForCES definido por el RFC 3654, es aquel que permite la comunicación entre los elementos de control (CE) y los elementos de reenvío (FE) en un elemento de red (NE). Según el RFC, este protocolo no estandariza la comunicación entre CE-CE, FE-FE o entre gestores FE y CE.

La figura 3, muestra un NE, el cual está compuesto de dos CE y dos FE. Los FE's y los CE's, requieren una configuración mínima como parte del proceso de pre-configuración, el cual es realizado por un Administrador de FE (FE Manager) y un Administrador del CE (CE Manager). La comunicación que se realiza por medio de estos componentes está por fuera del alcance de la

arquitectura y requerimientos del protocolo ForCES, el cual únicamente involucra la comunicación entre CE's y FE's.



Fp: CE-FE interface
Fi: FE-FE interface
Fr: CE-CE interface
Fi/f: interface externa FE
Fc: Interface entre un administrador CE y un CE
Ff: Interface entre un administrador FE y un FE
Fl: Interface entre un administrador CE y un administrador FE

Figura 3. Diagrama de la Arquitectura ForCES (Elaborado por Pedro Luis Gonzalez, basado en el RFC 3654)

3.2.3. BLOQUES LOGICOS FUNCIONALES (LFBS)

Según el draft-ietf-forces-lib-05, un Bloque Lógico Funcional es un bloque que reside en un elemento de reenvío (FE) y es controlado por el elemento de control (CE) operado sobre el protocolo ForCES. Estos bloques se categorizan por las clases LFBs, una Instancia LFB, representa una clase existente y cada clase está representada por un ID. La comunicación entre diferentes LFBs se realiza por medio de metadatos, lo cuales se defienden en el RFC modelo de un FE (5812)como un paquete redirect que comunica el estado de un paquete desde un LFB a otro LFB, el metadato se envía entre FEs y CEs pero no se envía a través de la red.

Una clase LFB, está conformada por uno o varios componentes, los cuales son parámetros operacionales que deben ser visibles a los CE's, estos pueden ser las banderas, argumentos de parámetros únicos o complejos y tablas que el CE puede leer y/o escribir por medio del protocolo ForCES.

La representación de cómo las instancias de los LFBs están lógicamente interconectadas y puestas a lo largo de los datapath en un FE se conoce como topología LFB.

En la figura 4, se puede observar que los LFBs tienen entradas, salidas y componentes que pueden ser requeridos y manipulados por el CE por medio del punto de referencia (Fp) definido en el RFC 3746 y el punto de terminación del protocolo ForCES.
Internamente en el FE se encuentran los LFBs, los cuales se interconectan con las entradas y salidas de cada uno de ellos, P, indica el paquete de datos y M indica el metadato asociado a un paquete. El punto Fp entre el CE y el FE establece una comunicación bidireccional, la comunicación desde el CE al FE es para configuración, control y entrada de paquetes, mientras que la comunicación desde el FE al CE es para redireccionar paquetes al plano de control, reporte de información de monitoreo, conteo de información y reporte de errores. El resultado de la interacción por el CE es la manipulación de los componentes de las instancias de los LFBs.

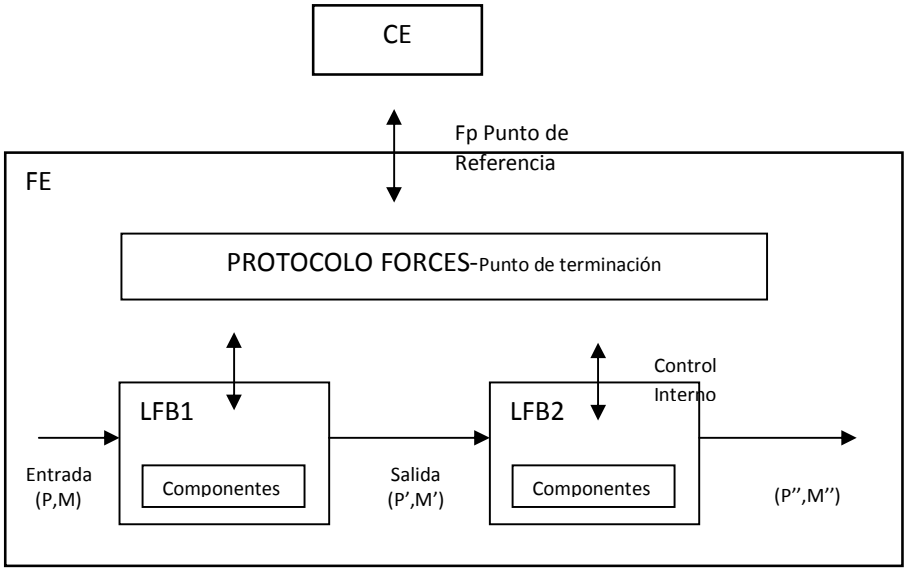


Figura 4. Diagrama de un LFB (Tomada del RFC 5812, página 16)

3.2.3.1. ALCANCE DE LA LIBRERÍA DE LOS LFBs

En el draft-ietf-forces-lib-05, intenta que las clases LFBs sean designadas para proveer las funciones de un router típico, según el RFC 1812, las funciones que debe proveer son:

1. Interface para redes de paquetes e implementación de funciones requeridas por la red como:
   a. Encapsulación y desencapsulación de datagramas IP con conexión a la red (por ejemplo: encabezados Ethernet y checksum).
   b. Envío y recepción de datagramas IP con un máximo de tamaño soportado por la red, este tamaño es la unidad máxima de transmisión o MTU.

c. Traducción de direcciones de destino IP dentro de un nivel de red apropiado para establecer comunicación (ejemplo, dirección de hardware Ethernet) si es necesario.
d. Respuesta para el control de flujo a la red e indicación de error, en su caso.

2. Conformación de protocolos específicos de internet, incluyendo el protocolo de internet (IPv4/IPv6), protocolo de mensajes de internet (ICMP) y otros si es necesario.
3. Recibir y reenviar datagramas de internet.
   a. Reconocer las condiciones de error y generar mensajes de información ICMP de error.
   b. Eliminar datagramas los cuales tienen un tiempo de vida TTL que ha llegado a 0
   c. Fragmentación de datagramas cuando sea necesario para ajustar en la MTU de la próxima red.
4. Escoger el próximo salto de destino para cada datagrama IP, basado sobre la información de la tabla de enrutamiento.
5. Usualmente soporta un protocolo de Gateway interior (IGP), para transportar/distribuir el enrutamiento y algoritmos accesibles con otros router en el mismo sistema autónomo.
   a. En adición, algunos routers necesitan soportar un protocolo de Gateway exterior para intercambio de la información topológica con otros sistemas autónomos y tener enrutamiento estático.
6. Proveer un sistema de gestión de red y un sistema de soporte, incluyendo carga, debugging, reporte de estado entre otros.

Un router típico que emplea ForCES lo constituye un CE corriendo y controlando un IGP y/o EGP e implementando FEs usando bloques lógicos funcionales (LFBs) conforme al RFC 5812.
Los paquetes en un router IP se reciben y se transmiten sobre un medio físico conocido como "Puerto", los diferentes puertos físicos tienen diferentes formas para encapsular tramas salientes y desencapsular tramas entrantes. Los paquetes IP que vienen de los puertos de los LFBs son procesados por un LFB Validador antes de reenviarse al próximo LFB, después del proceso de validación el paquete es enviado a un LFB donde la decisión de reenvío IP se realiza. En el LFB de reenvío, un LFB de acople de prefijo, es usado para buscar la información de destino de un paquete y seleccionar el próximo salto indicado para enviar el paquete.

3.2.3.2. DESCRIPCION DE LAS CLASES DE LOS LFBs

Una clase LFB es una plantilla que representa un aspecto lógicamente separable del procesamiento del FE.

3.2.3.3. LFB PARA PROCESAMIENTO ETHERNET

El protocolo con mayor despliegue en los niveles físico y de enlace de datos es el Ethernet, por tal motivo se convierte en un requisito básico para un router ser capaz de procesar varios paquetes de datos Ethernet, es bueno aclarar que existen varias versiones de protocolo Ethernet como: Ethernet V2, 802.3 RAW, IEEE 802.3/802.2, IEEE 802.3/802.2 SNAP, también existen varias técnicas LAN basadas en Ethernet como: VLANs, MACinMAC, entre otros. En el procesamiento Ethernet que realizan los LFBs se intenta hacer frente a todas estas variaciones de la tecnología Ethernet.

También hay varios tipos de medio físicos Ethernet como el cobre o la fibra óptica, pero por definición los LFBs se centran en el cobre como medio físico Ethernet. En la figura 5, muestra las diferentes clases de la clase LFB Ethernet como son: Clase EtherPHYCop, EtherMACIn, EtherClassifier, EtherEncapsulator y EtherMACOut.

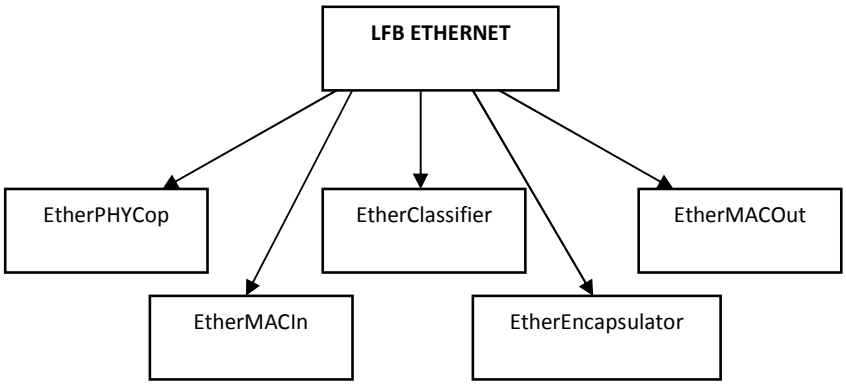


Figura 5. Clase LFB Ethernet (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

3.2.3.3.1. EtherPHYCop

Este LFB resume la interface física Ethernet con medio de transmisión por cobre. Este bloque tiene una entrada para paquetes Ethernet y una salida que por lo general se conecta a un LFB conocido como EtherMACIn. El metadato asociado a esta salida es el PHYPortID, que indica el ID del puerto físico por donde ingreso el paquete.

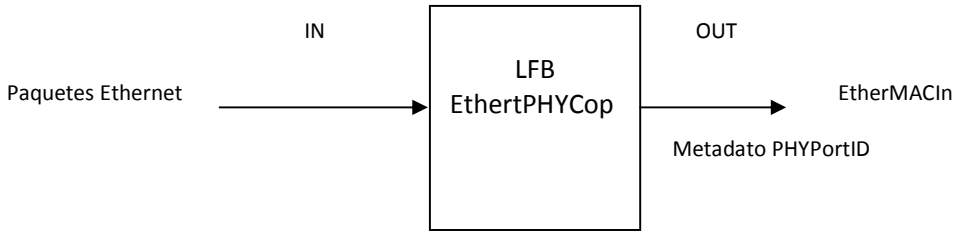


Figura 6. Bloque funcional EtherPHYCop (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Manejo de datos

Este LFB es la interface para el medio físico Ethernet y maneja las tramas Ethernet que entran y salen del FE, por ejemplo, las tramas Ethernet son recibidas desde un puerto físico y pasan a los LFBs como EtherMACIn por medio de una salida simple conocida como EtherPHYOut. Un metadato "PHYPortID" indica por cual puerto físico la trama ingresa, los paquetes Ethernet son recibido por este LFB desde los LFB superiores como el EtherMACOut por medio de una entrada simple conocida como "EtherPHYIn" y luego se envían al exterior, como se observa en la figura 6.

Componentes

A este LFB es controlado por los siguientes componentes:

- AdminStatus. Está definido por el CE para gestionar administrativamente el estado del LFB. El CE puede administrativamente habilitar, deshabilitar y reiniciar el LFB, cambiado el valor del componente, el valor por defecto es "down".
- OperStatus. Captura el estado operacional del puerto físico. También es definido por el CE.
- PHYPortID. Es una identificación única para el puerto físico, está definido como solo de lectura por el CE. Este valor es dado por el FE y este componente produce un metadato "PHYPortID" a la salida del LFB y lo asocia a cada paquete Ethernet que recibe.
- AdminLinkSpeed. Permite al CE configurar la velocidad del enlace por el puerto. El valor por defecto es el modo de auto-negociación.
- OperLinkSpeed. Permite al CE consultar la velocidad del enlace.
- AdminDuplexMode: Permite al CE configurar el modo dúplex para el puerto. Configuración por defecto es auto-negociación.
- OperDuplexMode: Permite al CE consultar el modo de operación actual (dúplex).
- CarrierStatus: Captura el estado de la portadora y especifica si el puerto esta enlazado a un conector operacional.

Capacidades

La información sobre capacidades para este LFB, incluye la velocidad del enlace soportada por el FE (SupportedLinkSpeed) así como el soporte en modo dúplex (SupportedDuplexMode).

Eventos

Este LFB está definido para ser capaz de generar varios eventos en los cuales el CE puede estar interesado. Estos eventos son:

- PHYPortStatusChanged. Cambio del estado del puerto físico, este evento notificara que el estado del puerto físico ha cambiado y reportara el nuevo estado.
- LinkSpeedChanged. Captura el cambio en la velocidad de operación del enlace. Este evento notificara al CE que la velocidad de operación ha sido cambiada y dará un reporte de la nueva velocidad operacional negociada.
- DuplexModeChange. Captura el cambio en el modo dúplex. Este evento notificara al CE que el modo dúplex ha sido cambiado y dará un reporte con el nuevo modo negociado.

3.2.3.3.2. EtherMACIn

Este LFB describe las funciones de procesamiento Ethernet como chequeo local de direcciones MAC, decide si los paquetes pueden ser puenteados y provee un control de flujo.

Está compuesto por una entrada simple, por donde ingresan paquetes Ethernet desde el LFB EtherPHYCop y al cual se le asocia un metadato con el ID del puerto físico por donde ingreso el

paquete desde el exterior. Dos salidas, la primera conocida como NormalPathOut, la cual está asociado un metadato PHYPortID y la segunda L2BridgingPathOut. La figura 7, muestra el diagrama este LFB.

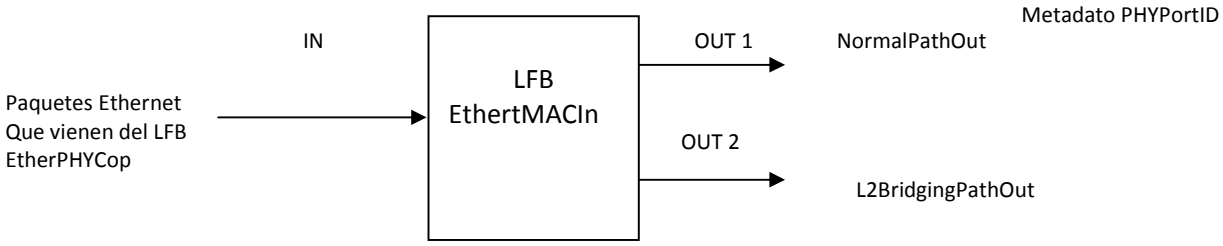


Figura 7. LFB EtherMACIn (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Manejo de datos

El LFB espera recibir todos los paquetes tipo Ethernet por medio de una entrada simple conocida como EtherMACIn, la cual usualmente sale desde algún LFB de nivel físico como EtherPHYCop y un metadato indicando el ID del puerto físico de donde viene el paquete.

Este LFB está definido con dos salidas simples separadas, todos los paquetes que salen están en formato Ethernet. La primera salida es conocida como "NormalPathOut", usualmente esta salida es una entrada para otro LFB como por ejemplo un LFB Clasificador y luego enviarla a un proceso de reenvío de nivel L3 con un metadato asociado de PHYPortID indicando el puerto físico de donde viene el paquete.

La segunda salida es conocida como "L2BridgingPathOut" que encuentra el requerimiento de las funciones de puenteo de nivel L2. Si el FE soporta puenteo de nivel L2, el CE puede ser capaz de habilitarlo o deshabilitarlo por medio del componente "L2BridgingPathEnable" en el FE.

Estos LFBs pueden trabajar en modo promiscuo, permitiendo que todos los paquetes pasen a través del LFB sin ser bloqueados, o de lo contrario pueden realizar un chequeo local basado en el direccionamiento MAC, y aquellos paquetes que no cumplan con los requerimientos serían bloqueados, también pueden realizar un control de flujo a nivel Ethernet, lo cual es implementado en conjunto con los LFBs EtherMACIn e EtherMACOut.

Componentes

- AdminStatus. Está definido por el CE para gestionar administrativamente el estado del LFB. El CE puede habilitar, deshabilitar y reiniciar el LFB cambiando el valor de este componente. Por defecto se encuentra en "Down".
- LocalMACAddress. Especifica la dirección MAC local basándose en el chequeo local que se realiza. Es un arreglo de direcciones MAC y de acceso lectura/escritura.
- L2BridgingPathEnable. Captura si el LFB trabaja como puente L2, si el FE no soporta puenteo de nivel L2 hay una bandera que indica "falso". El valor por defecto es "Falso".

- PromiscuosMode. Especifica si el LFB trabaja en modo promiscuo. EL valor por defecto es "Falso"
- TxFlowControl. Define si el LFB esta desempeñando un control de flujo sobre los paquetes enviados, el valor por defecto es "Falso".
- RxFlowControl. Define si el LFB está desempeñando un control de flujo sobre los paquetes entrantes, el valor por defecto es "Falso".
- MACInStats. Define un conjunto de estadísticas para este LF, incluyendo el número de paquetes recibidos y el número de paquetes bloqueados.

Capacidades.
Este LFB no tiene capacidades

Eventos
Este LFB no tiene eventos especificados.

3.2.3.3.3. EtherClassifier

Este LFB realiza el proceso de desencapsulación de paquetes Ethernet y los clasifica según el encabezado. La figura 8, muestra la representación grafica de este LFB.

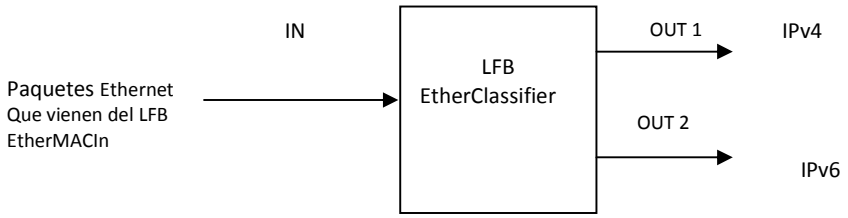


Figura 8. LFB EtherClassifier (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Manejo de datos

El LFB describe el proceso de desencapsulación de paquetes Ethernet y los clasifica dentro de varios niveles de red de acuerdo a la información incluida y los encabezados Ethernet. Este LFB espera recibir todo tipo de paquetes Ethernet incluyendo paquetes VLAN por medio de una entrada simple conocida como "EtherPktsIn", la cual es usualmente la salida de un LFB superior como EtherMACIn. Esta entrada es capaz de multiplexar la información para permitir la conexión de múltiples LFBs.

Usualmente todos los paquetes Ethernet esperados están asociados a un metadato PHYPortID indicando el puerto físico de donde viene el paquete.

Las salidas se definen como un grupo llamado "ClassifyOut" ya que pueden presentarse varios tipos de protocolos de paquetes a la salida. Por ahora las salidas se clasificaran en IPv4 e IPv6.

Componentes

- EtherDispatchTable. Es un arreglo de componentes que se define para el acople de cada paquete Ethernet a un grupo de salida acordado por el ID del puerto lógico asignado por la VLANInputTable para el paquete y el encabezado Ethernet para el tipo Ethernet. Cada fila del arreglo es una estructura que contiene un ID de puerto lógico, un EtherType y un índice de salida. El CE configura la tabla de envio, el LFB puede estar esperando para clasificar varios protocolos de nivel de red y las salidas en diferentes puertos de salida. Se espera que la clasificación de los paquetes sea de acuerdo a los protocolos como IPv4, IPv6, MPLS, ARP y ND.
- VLAMInputTable. Es un arreglo de componentes para clasificar paquetes VLAN Ethernet, cada fila del arreglo es una estructura que contiene un ID del puerto de entrada, un ID de VLAN y un ID de puerto lógico. El ID del puerto Lógico y el ID del puerto físico son configurados por el CE.
- EtherClassifyStats. Es un arreglo de componentes que definen un conjunto de estadísticas para este LFB, se mide el número de paquetes por el EtherType. Cada fila del arreglo es una estructura que contiene un EtherType y un número de paquete.

Capacidades
Este LFB no tiene lista de capacidades

Eventos
Este LFB no tiene lista de eventos.

3.2.3.3.4. EtherEncap

Este LFB resume el proceso para reemplazar o unir apropiadamente los encabezados Ethernet al paquete. La figura 9, muestra la representación grafica de este LFB.

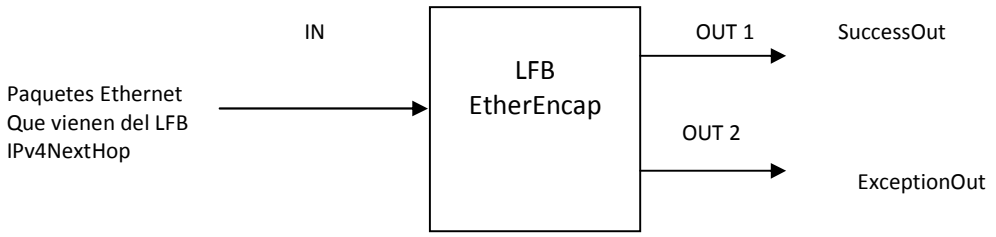


Figura 9. Bloque Funcional EtherEncap (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Manejo de datos

Este LFB realiza el proceso de encapsular paquetes IP a paquetes Ethernet de acuerdo a la información de nivel L2.

Espera recibir paquetes IP incluyendo IPv4 e Ipv6 por medio de una entrada conocida como "EncapIn" la cual se conecta a un LFB IPv4NextHop o un IPv6NextHop, BasicMetadataDispatch o algún LFB que requiera un paquete de salida para la encapsulación Ethernet.

Este LFB opera desde un LFB superior con un metadato asociado "MediaEncapInfoIndex" el cual es usado como un indicador para buscar en la tabla de encapsulación.
Este LFB tiene dos puertos de salida, la primera salida se conoce como "SuccessOut", tras una búsqueda satisfactoria de las direcciones MAC de origen y destino y el puerto lógico (L2PortID) se encuentran en la tabla apropiada.
La segunda salida se conoce como "ExceptionOut" son los datos de salida donde la búsqueda en la tabla falla, se asocia con un metadato "ExceptionID", actualmente se incluye un tipo de excepción que incluye el siguiente dato:

- Valor MediaEncapInfoIndex que no es asignado en la tabla de encapsulación.

Componentes

- EncapTable. Es un arreglo del componentes, cada fila del arreglo es una estructura que contiene la dirección de destino MAC y la dirección de origen MAC, el ID de la VLAN con valor por defecto de 0 y la salida lógica L2portID.

Capacidades
No tiene lista de capacidades

Eventos
No tiene lista de eventos

3.2.3.3.5. EtherMACOut

Resume un puerto Ethernet para nivel de enlace de datos MAC. Este LFB describe el proceso de salida de un paquete. La figura 10, muestra la representación grafica de este LFB.

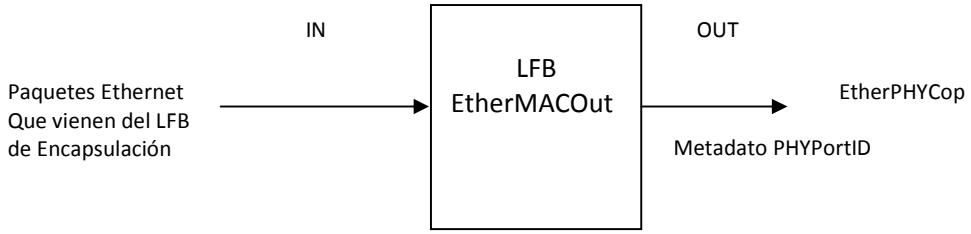


Figura 10. Bloque Funcional EtherMACOut (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Manejo de datos

Este LFB espera recibir todo tipo de paquete Ethernet por medio de una entrada simple conocida como "EtherPktsIn", la cual es usualmente salida de un LFB de Encapsulación Ethernet con un metadato asociado indicando el ID del puerto físico. Tiene una salida simple, todo paquete que sale es Ethernet y también tiene asociado un metadato con el ID del puerto físico, esta salida se enlaza por lo general con un LFB físico EtherPHYCop.
Otra de sus funciones es el control de flujo, que es implementado por el LFB EtherMACIn y EtherMACOut cooperativamente.

Componentes

- AdminStatus. Está definido por el CE para administrar el estado del LFB. El CE puede habilitar, deshabilitar y reiniciar el LFB cambiando el valor por defecto que es "Down"
- MTU. Define la unidad máxima de transmisión
- TxFlowControl. Define si el LFB está desempeñando el control de flujo de paquetes enviados. Valor por defecto "falso"
- RxFlowControl. Define si el LFB está desempeñando el control de flujo de paquetes recibidos. Valor por defecto "falso".
- MACOutStats. Define un conjunto de estadísticas para este LFB, incluyendo el número de paquetes transmitidos y bloqueado.

Capacidades
No tiene lista de capacidades

Eventos
No tiene lista de eventos

3.2.3.4. LFBs de Validación de paquetes IP

Resume el proceso de validación de paquetes IP, específicamente para validación del protocolo IPv4 e IPv6. En la figura 11, se observa la clase principal de este LFB y las clases que la componen.

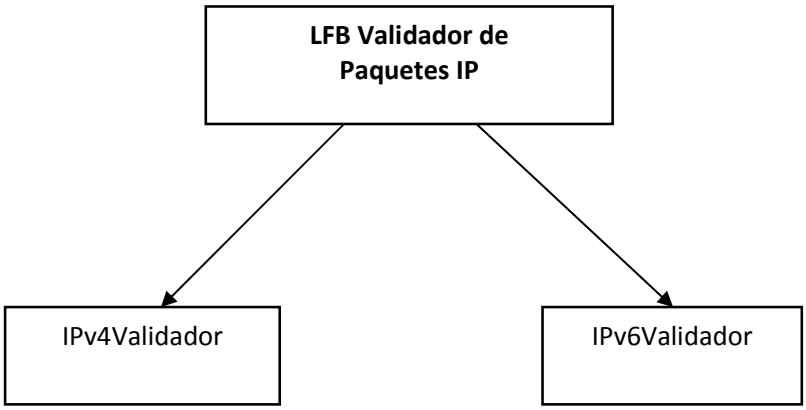


Figura 11. Clase Validador de Paquetes IP (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

3.2.3.4.1. IPv4Validador

Valida paquetes IP versión 4. La figura 12, muestra la representación grafica de este LFB.

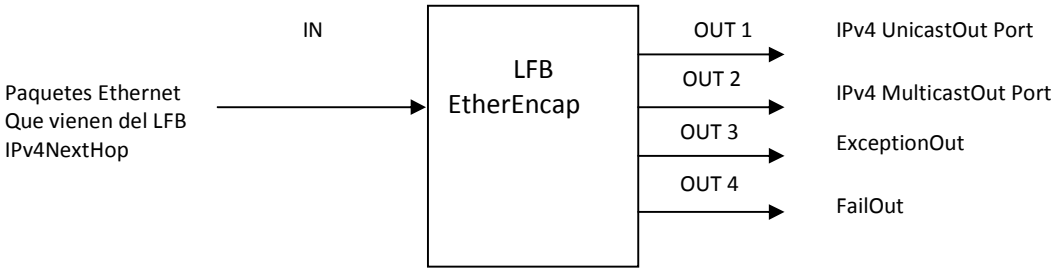


Figura 12. Bloque Funcional EtherEncap (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Manejo de datos

Cada paquete IPv4 sale al puerto correspondiente de acuerdo a la validación del resultado, si un paquete es Unicast, Multicast, ha ocurrido una excepción o la validación falla. Este LFB siempre espera como entrada paquetes los cuales han sido indicados como paquetes IPv4 por un LFB anterior como EtherClassifier. No hay un metadato específico esperado por la entrada del LFB.

Tiene cuatro puertos de salida, definidos para la validación de resultados.

Todos los paquetes Unicast IPv4 validados serían salidas en el puerto "IPv4UnicastOut", todos los paquetes Multicast serán salidas de "IPv4MulticastOut", no hay metadatos requeridos para producir estas salidas. Un puerto conocido como "ExceptionOut" es definido para salida de paquetes los

cuales han sido validados como paquetes de excepción. Un metadato con ID de exception es producido para indicar que ha sucedido una excepción. Se define excepción a:

a. Un paquete con dirección de destino igual a 255.255.255.255
b. Un paquete con TTL expirado
c. Paquetes con longitud de encabezado con más de cinco palabras.
d. Encabezado IP incluyendo la opción de Alerta en el router.

El puerto "FailOut", se define para todos los paquetes en los cuales ha fallado el proceso de validación. Razón de fallo:

a. El tamaño del paquete reportado es menos que 20 bytes.
b. Paquete con versión que no es IPv4
c. Paquete con longitud de encabezado menor a 5
d. Paquete con longitud total menor a 20
e. Paquete con checksum invalido
f. Paquete con dirección de origen igual a 255.255.255.255
g. Paquete con dirección de origen igual a 0
h. Paquete con dirección de origen de forma {127, <any>}
i. Paquete con dirección de origen clase E.

Componentes

- IPv4ValidatorStatisticsType. El cual define un conjunto de estadísticas para procesos de validación, incluyendo el número de paquetes con encabezado errado, número de paquetes con longitud errada, número de paquetes con TTL errado y número de paquetes con checksum no valido.

Capacidades

No tiene lista de capacidades

Eventos

No tiene lista de eventos

3.2.3.4.2. IPv6 Validador

Este LFB desempeña validación de paquetes IPv6 de acuerdo al RFC 2460. No se define porque no se incluye en este trabajo.

3.2.3.5. LFBs IP Forwarding

Este LFB resume el proceso de reenvío IP. Este documento se centra en el reenvío IP Unicast. Multicast se define para trabajos futuros.

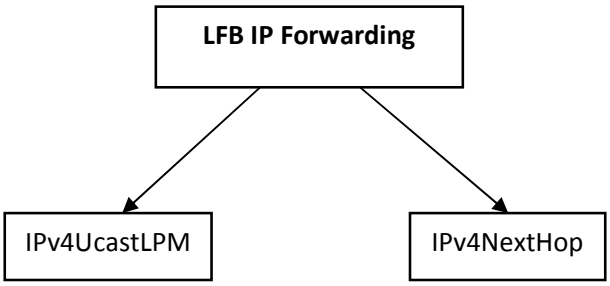


Figura 13. Clase LFB Forwarding (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Un típico trabajo de reenvío IP unicast es usualmente realizado para buscar alguna información en la tabla de forwarding para encontrar la información del próximo salto y basándose en esta información reenviar el paquete a un puerto de salida especifico.
Usualmente este proceso toma dos pasos para hacerlo, primero buscar en la tabla de forwarding por medio de la regla del prefijo más largo (Longest Prefix Matching- LPM), para encontrar el próximo salto, y segundo usar el próximo salto indicado en la tabla de información para encontrar suficiente información para enviar los paquetes a los puertos de salida.
Sin embargo, existen otros modelos como uno que puede tener una información de reenvío base que tiene unida la información del próximo salto con la información de reenvío. Este proyecto se ha de basar en la regla de los dos pasos.

Hay dos clases de LFB IP Forwarding, Unicast LPM y el Next Hop, como se observa en la figura 13.

3.2.3.5.1. IPv4UcastLPM

Resume el proceso del prefijo acoplado más largo (Longest Prefix Match – LPM). La figura 14, muestra la representación grafica de este LFB.

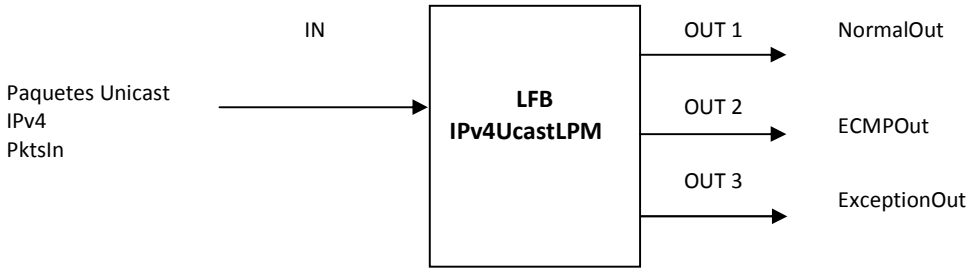


Figura 14. Bloque Funcional LFB IPv4UcastLPM (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Manejo de datos

Se espera como entrada paquetes Unicast IPv4 desde una entrada conocida como "PktsIn". Entonces el LFB usa la dirección IPv4 de destino de cada paquete como índice para buscar en la tabla de prefijo de IPv4 y generar un selector de salto como resultado, este resultado se asocia al paquete con un metadato para salir a otro LFB.

Hay tres puertos de salida que se definen de acuerdo a los resultados de LPM.
La primera salida se conoce como "NormalOut" por la cual salen paquetes IpV4 unicast que han pasado por la búsqueda LPM y conseguido un selector como resultado de esa búsqueda.
La segunda salida se conoce "ECMPOut" que está definida para proveer soporte a los usuarios que deseen implementar ECMP, una bandera ECMP está definida en la tabla LPM para habilitar el LFB que lo soporte.
La salida final es conocida como "ExceptionOut" y es definida para permitir la excepción de paquetes, los casos de excepción incluyen:

a. Los paquetes que no pueden encontrar alguna ruta en la tabla de prefijos.

Los LFB vecinos de este LFB son usualmente IPv4Validador (Upstream) y los vecinos downstream son usualmente IPv4NextHop.

Componentes

- IPv4PrefixTable. Es un arreglo de componentes. Cada fila del arreglo contiene una dirección IPv4, una longitud de prefijo, un selector de salto, una bandera ECMP y una bandera de ruta por defecto. El LFB usa la dirección IPv4 de destino de cada paquete entrante como un indicador para buscar en la tabla y conseguir un selector de salto como resultado. La bandera ECMP indica al LFB el soporte de ECMP.
- IPv4UcastLPMStats. Es una estructura la cual recoge información estadística, incluyendo el número total de entrada de paquetes recibidos y paquetes IPv4 reenviados por este LFB y el número de datagramas IP descartados debido a no encontrar rutas validas.

Capacidades
No tiene lista de capacidades

Eventos
No tiene lista de eventos

3.2.3.5.2. IPv4NextHop

Resume el proceso de selección del próximo salto. La figura 15, muestra la representación grafica de este LFB.

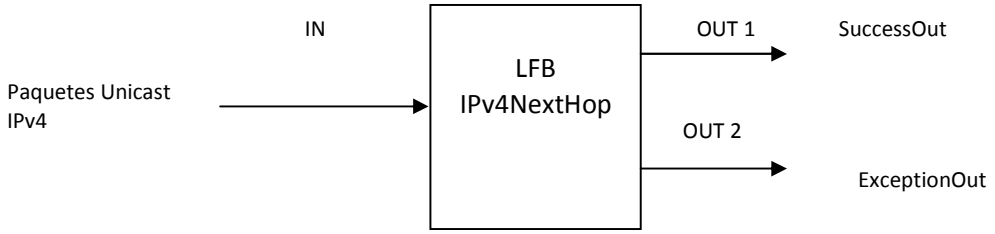


Figura 15. Bloque Funcional IPv4NextHop (Elaborado por Susan Martínez, basado en el draft-ietf-forces-lfb-lib-05)

Manejo de datos

Resume el proceso de aplicación de la información del próximo salto para paquetes IP, este LFB recibe paquetes IPv4 con un ID del próximo salto asociado y usa el ID para buscar en la tabla del próximo salto para encontrar un puerto de salida apropiado.

El LFB espera recibir paquetes unicast IPv4 por medio de una entrada conocida como "PcktsIn" con un metadato HopSelector el cual es usado como un índice para buscar en la tabla del próximo salto.

El procesamiento de datos envuelve el decremento TTL y la recalculación del checksum.

Se define dos puertos de salida, la primera salida es conocida como "SuccessOut", donde el proceso satisfactorio de paquetes de datos es enviado. El paquete de salida es enviada a otro LFB junto con L3PortID y un metadato MediaEncapInfoIndex.

La segunda salida es para el procesamiento de paquetes que fallan, a este paquete de salida se asocia un metadato ExceptionID para indicar que causa la excepción. Los tipos de excepción incluyen:

a. Selector de salto invalido
b. El MTU de salida en menor que el tamaño del paquete
c. El paquete ICMP necesita ser generado.

Componentes

- IPv4NextHopTable. Es un arreglo, cada fila del arreglo es una estructura que contiene:

a. L3PortID, el cual es el ID del puerto lógico que pasa sobre las instancias de los vecinos del LFB. El ID indica que el puerto del vecino es definido por el nivel L3.
b. La unidad máxima de transmisión para el puerto de salida
c. NextHopIPAddr, IP del próximo salto.
d. MediaEncapInfoIndex. Este índice es usado para buscar en la tabla adicional de bajada.
e. LFBOutputSelectIndex. Indica la selección del puerto del LFB de bajada.

Capacidades

No tiene lista de capacidades

Eventos

No tiene lista de eventos

3.2.4. TOPOLOGIAS DE PROTOCOLOS IMPLEMENTADOS EN LOS BLOQUES LOGICOS FUNCIONALES DEL ELEMENTO DE REENVIO EN LA ARQUITECTURA FORCES

Cuando se va a enviar un mensaje de un origen a un destino, esa información pasa por la capa de Aplicación (nivel 7 del modelo OSI), luego por la capa de Presentación (nivel 6), Capa de Sesión (nivel 5), Capa de Transporte (nivel 4), Capa de Red (nivel 3), Capa de Enlace de datos (nivel 2) y finalmente por la Capa física (nivel 1). En cada una de estas capas, se va adicionando al mensaje una serie de encabezados, los cuales son información redundante que ayudan a identificarlo, por ejemplo, de donde viene, hacia donde va, el puerto de entrada/salida, direcciones lógicas de origen destino (direcciones IP), direcciones físicas de origen, destino (direcciones MAC), sistemas de detección de errores, control de flujo, entre otros. De modo que el mensaje alcance el destino de una forma confiable y se pueda recuperar, este proceso se muestra en la figura 16.

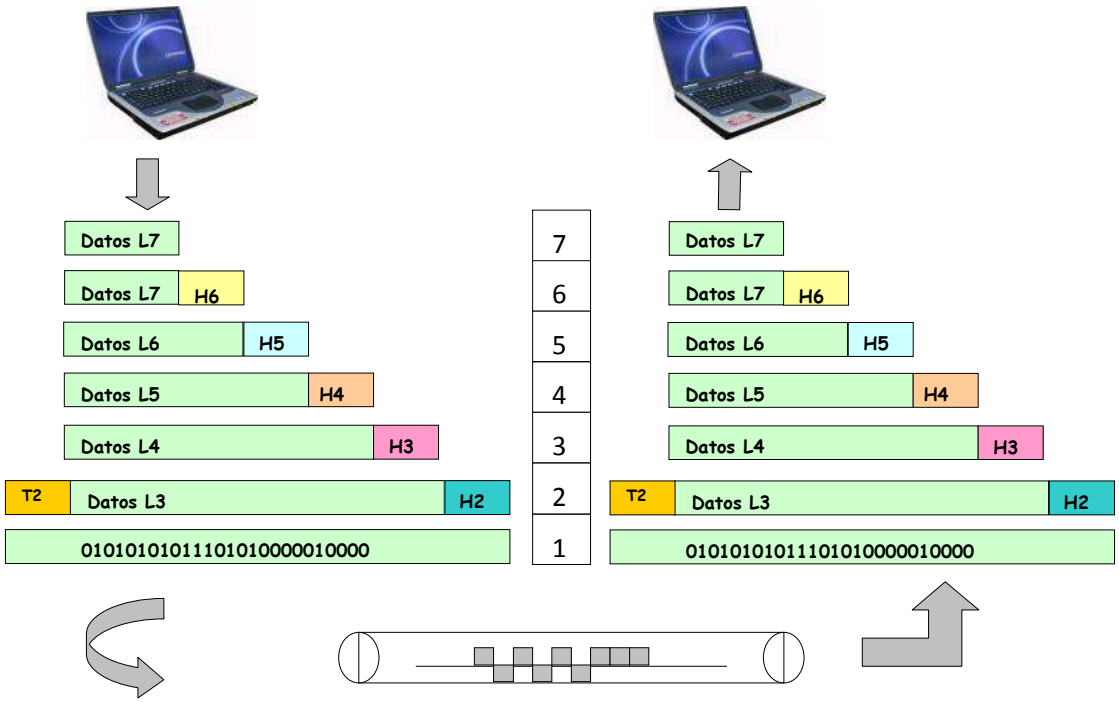


Figura 16. Procesamiento de la información según el modelo OSI (Tomado del libro Redes de Datos, Forouzan)

En cada uno de los diferentes niveles del modelo OSI, operan ciertos protocolos, los cuales trabajan juntos para llevar a cabo la transmisión de datos de un origen a un destino.

En este proyecto se implementaron los protocolos: ARP (Protocolo de Resolución de Direcciones), que hace parte del nivel 2 del modelo OSI, protocolo de enrutamiento RIP Versión 2, que hace parte

del nivel 3 y el protocolo de Gestión de Red SNMP (Protocolo Simple de Administración de Red) perteneciente al nivel 7. En la arquitectura ForCES, el elemento de control (CE), es el encargado de construir una topología de cada protocolo por medio de los diferentes bloques lógicos funcionales (LFBs), luego esta topología por medio del protocolo ForCES es enviada al elemento de reenvío (FE) que la implemente.

3.2.4.1. PROTOCOLO DE RESOLUCION DE DIRECCIONES – ARP

El protocolo ARP (Protocolo de Resolución de Direcciones), traduce direcciones lógicas (direcciones IP) a direcciones físicas (direcciones MAC), de acuerdo a la información que se almacena en una tabla llamada ARP que se encuentra en la caché del host o enrutador.

ARP tiene dos tipos de mensajes, solicitud (request) y respuesta (reply), que como sus nombres lo indican, permiten solicitar una dirección física en caso que no se tenga en la tabla ARP del equipo que desea enviar un paquete a un destino y que el destino de respuesta a dicha solicitud.

Cuando se realiza una solicitud ARP, el mensaje ARP viaja sobre una trama Ethernet, como se puede observar en la figura 17.

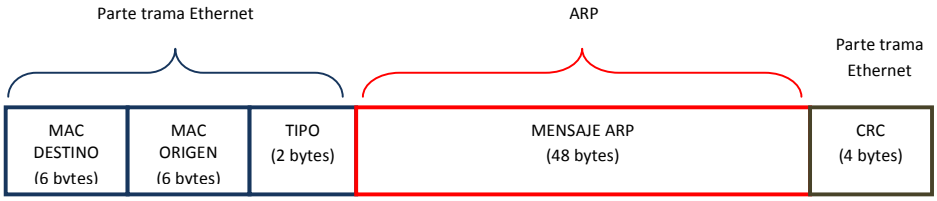


Figura 17. Formato trama ARP sobre Ethernet (Elaborado por Susan Martínez)

En la figura 18, se observa cómo está conformado un mensaje ARP.

TIPO DE HARWARE (2 bytes)		TIPO DE PROTOCOLO (2 bytes)
LONG. DIR.FISICA (1 bytes)	LONG. DIR.LOGICA (1 bytes)	OPERACION (2 bytes)
DIRECCION FISICA ORIGEN (6 bytes-Octetos del 0 al 3)		
DIRECCION LOGICA ORIGEN (4 bytes)		
DIRECCION FISICA DESTINO (6 bytes)		
DIRECCION FISICA ORIGEN (4 bytes)		

Figura 18. Formato trama ARP (Elaborado por Susan Martínez)

- **Tipo de Hardware.** Este campo de 2 bytes, especifica el tipo de una dirección de hardware. El valor "1" representa una dirección Ethernet.
- **Tipo de Protocolo.** Este campo de 2 bytes, especifica el tipo de la dirección de protocolo que se asignan, por ejemplo 0x800 representa el valor hexadecimal de una dirección IP.
- **Longitud de dirección física.** Este campo de 1 byte, representa el tamaño o longitud de una dirección de hardware, en este caso es 6 para una dirección Ethernet.
- **Longitud de dirección de lógica.** Este campo de 1 byte, representa el tamaño o longitud de una dirección de protocolo, en este caso es 4 para una dirección del protocolo IPv4.
- **Operación.** Este campo de 2 bytes, especifica el tipo de mensaje ARP. El valor "1" representa una solicitud de ARP y "2" representa una respuesta ARP, "3" representa una solicitud RARP y "4" representa una respuesta RARP.
- **Dirección física origen.** Este campo de 6 bytes, especifica la dirección hardware (MAC) del dispositivo que envía el mensaje.
- **Dirección IP origen.** Este campo de 4 bytes, especifica la dirección lógica (IP) del dispositivo que envía el mensaje.
- **Dirección física destino.** Este campo de 6 bytes, especifica la dirección hardware (MAC) del dispositivo que va dirigido el mensaje.
- **Dirección IP destino.** Este campo de 4 bytes, especifica la dirección lógica (IP) del dispositivo que va dirigido el mensaje.

El protocolo ARP se utiliza en dos casos principales:

CASO 1.

Cuando dos host están en la misma red y se requiere enviar un paquete de un host a otro host. Supongamos que el host A, con dirección IP 192.168.1.2, va a enviar un mensaje al host B, con dirección IP 192.168.1.5, el procesamiento es el siguiente:

1. Antes de salir el mensaje hacia B, el host A encapsula el mensaje, adicionándole dirección IP de origen (que es la del host A), dirección IP de destino (que es la del host B), dirección MAC de origen (que es la dirección hexadecimal de 48 bits que identifica exclusivamente el dispositivo de red del host A) y dirección MAC de destino (que es la del dispositivo de red del host B).

2. Si el host A no encuentra en su tabla ARP una dirección MAC que corresponda a la dirección IP de destino 192.168.1.5, entonces crea un mensaje ARP de solicitud, el cual es un mensaje de difusión (en el campo de dirección MAC de destino, asigna una dirección FF:FF:FF:FF:FF:FF), que va a llegar a todos los host conectados a esa red.

3. Al llegar el mensaje de solicitud de ARP a todos los host de la red, estos revisan el mensaje y comparan la dirección IP de destino que tiene el mensaje con la dirección IP que tiene asignada cada uno, si no corresponde, entonces el mensaje se descarta y al que le corresponde la dirección, este host, crea un paquete de respuesta ARP donde en el campo de dirección MAC de origen asigna su MAC, en el campo de MAC de destino asigna la dirección MAC con que llego el mensaje de solicitud y lo envía de nuevo al host A, de esta

forma el host A al recibir el mensaje de respuesta ARP, actualiza su tabla ARP asociando la dirección IP del host B 192.168.1.5 con esa MAC.

4. Con la información completa, el host A ya puede encapsular el paquete a formato Ethernet y enviarlo al host de destino, en este caso B.

Este proceso se puede observar en la figura 19.

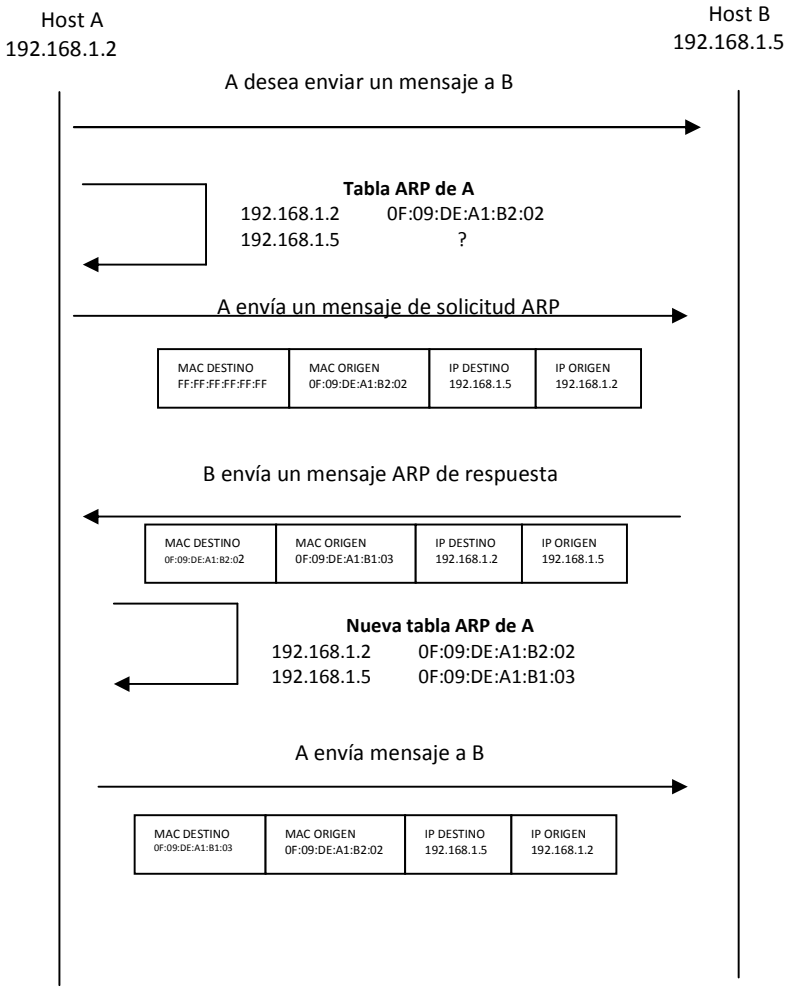


Figura 19. Proceso de ARP entre host de la misma red (Elaborado por Susan Martínez)

CASO 2.

Cuando dos host están en diferente red y se requiere enviar un paquete de un host a otro host a través de un enrutador. ARP también se utiliza para enviar datagramas IP a enrutadores locales de destinos que no se encuentran en la misma red, en este caso ARP resuelve la MAC de la interfaz de un enrutador en la red local.

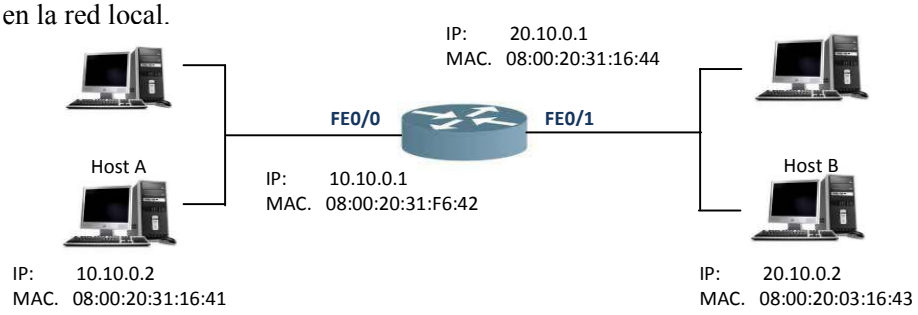


Figura 20. Proceso de ARP entre redes locales diferentes (Elaborado por Susan Martínez)

1. Supongamos que el host A con dirección IP 10.10.0.2, va a enviar un mensaje al host B con dirección IP 20.10.0.2. La dirección IP de la interfaz del enrutador que conecta al host A es 10.10.0.1 y la del enrutador que conecta al host B es 20.10.0.1.

2. Antes de enviar el mensaje al host B, en la tabla de enrutamiento del host A se verifica que la dirección IP de reenvío que se va a utilizar para llegar a B es la 10.10.0.1 y luego se verifica en la tabla ARP del host A, que esta dirección IP tenga asociada una dirección física para realizar el encapsulamiento IP.

3. Si no se encuentra una dirección MAC asociada a la IP de destino, el host A difunde un mensaje de solicitud de ARP a todos los host de la red, incluyendo la interfaz del enrutador.

4. El enrutador determina que la dirección IP especificada en la solicitud ARP es su dirección IP, por tal motivo reenvía un mensaje ARP de respuesta con la dirección IP de origen como la dirección IP de destino de su mensaje e incluye su dirección MAC y dirección IP como direcciones de origen.

5. El host A recibe el mensaje de respuesta ARP y actualiza su tabla ARP.

6. Como todavía se desconoce la dirección MAC de 20.10.0.2, el enrutador realiza nuevamente una solicitud ARP.

7. El host 20.10.0.2 determina que la dirección IP especificada en la solicitud ARP es su dirección IP, por tal motivo reenvía un mensaje ARP de respuesta con la dirección IP de origen como la dirección IP de destino de su mensaje e incluye su dirección MAC y dirección IP como direcciones de origen.

8. El host 20.10.0.1 recibe el mensaje de respuesta ARP y actualiza su tabla ARP.

9. Finalmente se envía el paquete IP desde 10.10.0.2 al 20.10.0.2

Este proceso se puede observar de la siguiente manera:

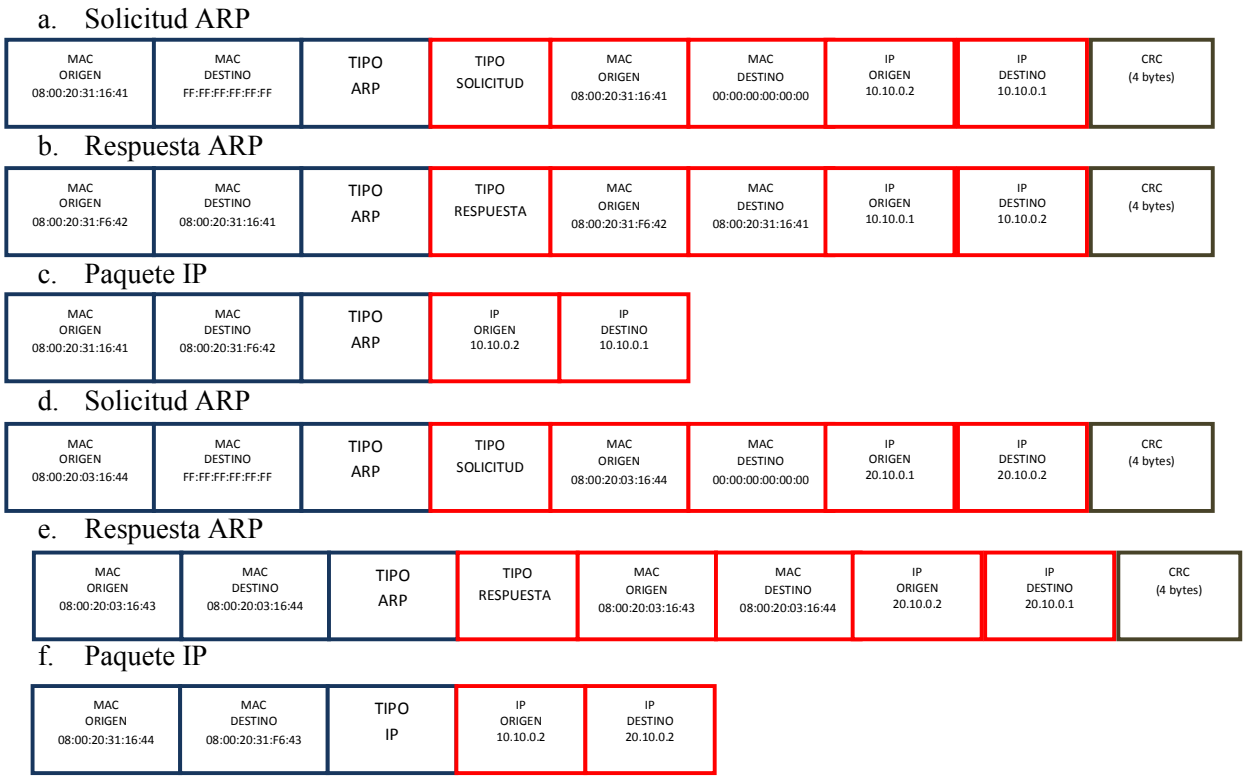


Figura 21- Procesamiento de tramas en ARP entre redes locales diferentes (Elaborado por Susan Martínez)

3.2.4.1.1. Procesamiento de ARP por medio de los Bloques Lógicos Funcionaes en ForCES

El procesamiento de un mensaje de solicitud ARP o una respuesta ARP desde el FE al CE en la arquitectura ForCES, se observa en la figura 22 y sería el siguiente:

Las tramas Ethernet ingresan al FE por medio del LFB EtherPHYCop, en el cual se encuentran las interfaces Ethernet físicas por la entrada llamada "EtherPHYIn", cada puerto físico tiene asociado un metadato que contiene el ID que identifica a cada puerto.

La trama sale del LFB "EtherPHYCop" por la salida "EtherPHYOut", al LFB EtherMACIn, la trama que ingresa a este LFB, viene con un ID, el cual indica el puerto por donde ingreso la información. El LFB EtherMACIn, realiza un chequeo de MAC locales que se encuentran en un arreglo de direcciones MAC dentro del componente "LocalMACAddress", luego de realizar ese chequeo y verificar que la dirección se es conocida, la envía a la salida "NormalPathOut", donde usualmente esta salida es la entrada del LFB EtherClassifier, si la dirección MAC no es conocida va a la salida "L2BridgingPathOut" para realizar un procesamiento a nivel L3.

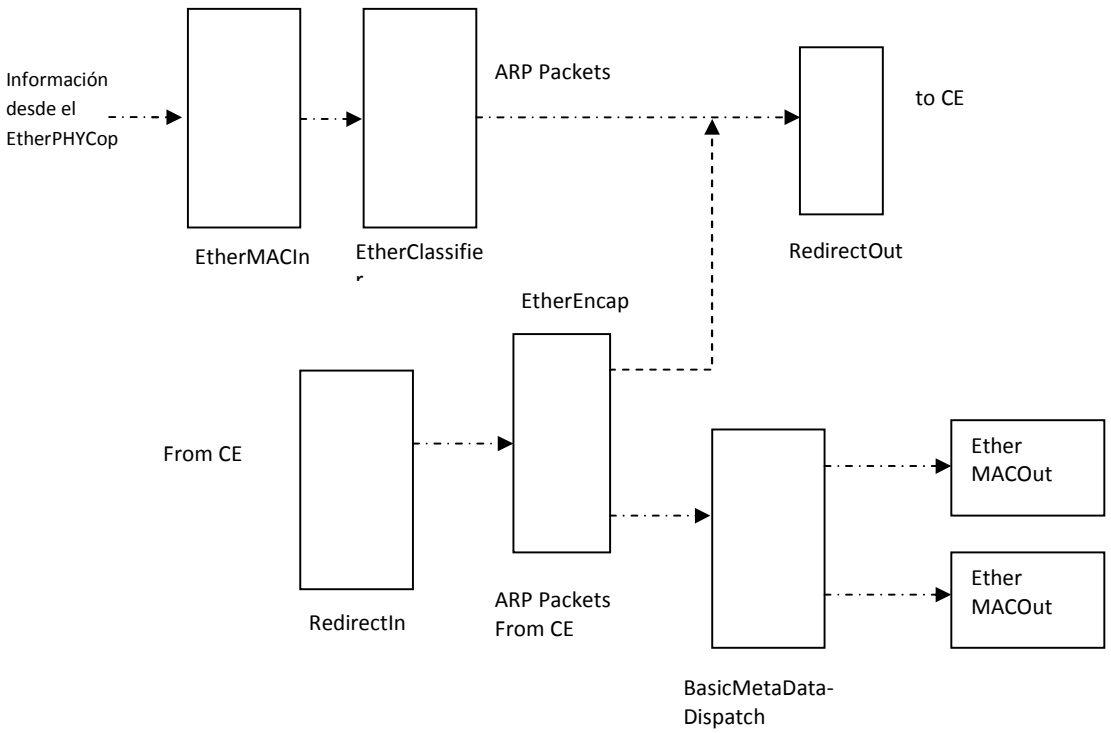


Figura 22. Topología del funcionamiento del protocolo ARP en ForCES (Tomado del draft-ietf-forces-lfb-lib-05)

Como al realizar el chequeo de MAC locales, en el LFB EtherMACIn, el mensaje es enviado al LFB EtherClassifier, el cual es encargado de desencapsular el paquete y clasificarlo (si es IPV4, IPV6, si es ARP, RARP, MPLS, entre otros). En este caso se espera un paquete ARP, que puede ser de respuesta ARP o una solicitud de ARP que viene de otro host o interfaz local.

Después de clasificar el paquete como una solicitud de ARP o respuesta ARP pasa al LFB RedirectOut, que indica que los mensajes o paquetes vienen del FE al CE para que el CE los procese.

El procesamiento desde el CE al FE en caso que el CE envíe una solicitud ARP o una respuesta ARP sería la siguiente:

Cuando un mensaje o paquete viene del CE, pasa por el LFB RedirectIn, que es la conexión desde el CE hacia los otros bloques funcionales del FE.

Como todo mensaje o paquete se debe encapsular antes de salir del dispositivo al exterior, este proceso lo realiza el LFB EtherEncap, el cual adiciona el encabezado Ethernet (MAC destino, MAC Origen) a un paquete IP ya sea IPv4 o IPv6, en nuestro caso son IPv4.

El LFB EtherEncap tiene dos salidas, una llamada “SuccessOut”, donde después de buscar en la tabla de encapsulación Ethernet la dirección MAC de origen y dirección MAC de destino correspondientes a las direcciones IP de origen y destino que vienen del paquete desde el CE, pasarían al LFB IPv4NextHop para su procesamiento, en llegado caso que no encuentre en su tabla la dirección MAC de destino asociada a la dirección IP de destino en la tabla de encapsulación, el mensaje o paquete pasa a la salida “ExceptionOut” y luego al LFB redirectOut y indicándole al CE que el paquete no se pudo encapsular a Ethernet. El CE entonces envía un mensaje de solicitud ARP, el cual tiene la siguiente trama como se observa en la figura 23:

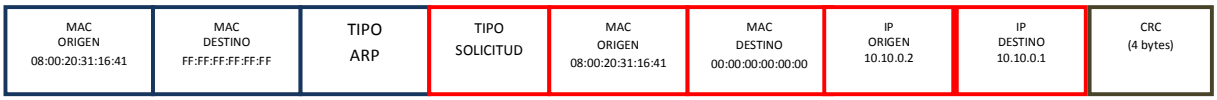


Figura 23. Formato trama solicitud ARP (Elaborado Susan Martínez)

Este mensaje de solicitud ARP pasa al LFB EtherEncap, el cual se encarga de encapsular el mensaje asignándole las direcciones MAC de origen y MAC de destino (FF:FF:FF:FF:FF:FF), que es una dirección de difusión, para que todos los dispositivos de la red reciban el mensaje. Como es un mensaje de difusión ARP según la programación del LFB, este mensaje ingresa al LFB MetadataDispatch, el cual lo recibe junto con un valor de metadato asociado. El LFB MEtadataDispatch, tiene un componente llamado MetadataDispatchTable, el cual es un arreglo que contiene una serie de metadatos (ID) asociados a los puertos de salida de este LFB, en este caso el metadato ID indica que el mensaje debe salir a todos los LFB EtherMACOut conectados a él.

El mensaje de solicitud ARP llega a todos los LFB EtherMACOut, este LFB espera recibir paquetes tipo Ethernet, junto con un metadato ID que indica el puerto físico de salida del paquete al exterior, todos los LFB EtherMACOut, revisan la dirección MAC asociada al puerto físico directamente conectado a cada uno y si la dirección IP coincide con la del destino del mensaje ARP, envía un mensaje de respuesta ARP con la dirección MAC de origen la de su puerto y con la de destino quien lo solicito.

El mensaje de respuesta ARP tiene la siguiente forma:

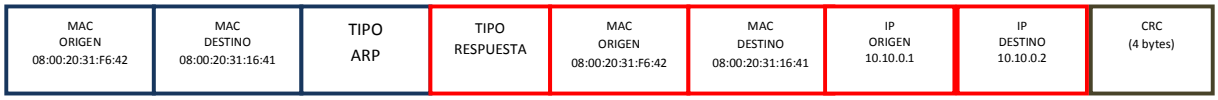


Figura 24. Formato trama respuesta ARP (Elaborado Susan Martínez)

Cuando el mensaje de respuesta ARP llega a quien lo solicito, este dispositivo actualiza su tabla ARP y encapsula el mensaje que deseaba enviar inicialmente.

3.2.4.2. PROTOCOLO DE ENRUTAMIENTO RIP VERSION 2

Un protocolo de enrutamiento es el conjunto de reglas empleadas por un dispositivo de red llamado router para comunicarse con otros dispositivos de red de su misma familia, con el fin de compartir información de rutas a un destino que se almacenan en una tabla llamada tabla de enrutamiento. El protocolo se encarga de crear y actualizar dichas tablas en los routers.

Existen protocolos de enrutamiento estático y dinámico. En este proyecto se empleo el protocolo de enrutamiento dinámico RIPv2 (Protocolo de Enrutamiento de Pasarela Interior, versión 2), que utiliza un algoritmo de enrutamiento de vector distancia y su métrica es el número de saltos.

Inicio del proceso de RIP

El router genera paquetes de solicitudes RIP para ser enviado a todos los puertos. El paquete saldrá satisfactoriamente por un puerto si el puerto:
- Es funcional (si el puerto existe y el protocolo esta up).
- Si está configurado RIP.

Versiones de RIP

El router ofrece diferentes paquetes con RIP dependiendo de la versión que corre de RIP.
- Si está corriendo RIPv1, puede:
  a. Enviar y recibir paquetes RIPv1
  b. Enviar broadcast
- Si está corriendo RIPv2, puede:
  a. Enviar y recibir paquetes RIPv2
  b. Enviar broadcast

Actualizaciones en RIP

Hay dos tipos de actualizaciones por RIP: periódica y activa

- En la actualización periódica, el router envía actualizaciones periódicas cada 30 segundos. La actualización contiene toda la información en la tabla de enrutamiento.
- En la actualización activa, el router envía actualizaciones solamente cuando ha cambiado de estado una interfaz (up o down).

Procesamiento de paquetes RIP entrantes

Cuando un router recibe paquetes RIP:

- Se descarta el paquete si (hay):
  a. El puerto de entrada no tiene una dirección IP válida o no está habilitado RIP
  b. La dirección IP de origen no proviene de una red conectada directamente.
  c. El paquete llegó desde el propio router.
  d. La versión RIP del paquete no coincide con la versión RIP del router.

- Si el paquete es un paquete de petición:
  a. Comprueba el puerto para ver si es una interfaz pasiva.
    - Si es así, descartar el paquete.

- Si no es una interfaz pasiva, procesa el paquete:
  -- Crea un paquete de respuesta de RIP, que contiene información sobre una ruta o la tabla de enrutamiento completa (dependiendo de la petición).
  -- Envía la respuesta RIP fuera el mismo puerto.

- Si el paquete es un paquete de respuesta, el proceso es:

a. Mirar a través de cada porción de ruta RIP del paquete (la porción de identificador de dirección de la familia, o AFI, para la métrica). Un paquete RIP puede contener hasta 25 porciones de rutas RIP.

b. Ignore cualquier parte donde (hay):

--- La métrica es mayor que infinito.

--- EL AFI no es de la familia IP

--- Es un broadcast, Clase D o dirección clase E.

c. Ajustar el próximo salto a la dirección del puerto entrante.

d. En el caso de nuevas rutas, ignorar la parte de la ruta si la métrica es ahora de 16.

f. Para las rutas existentes, la métrica se establece en 16.

g. Si el paquete contiene información sobre una red que no existe en la base de datos RIP, se añade a la base de datos.

h. Si una red cuenta ya con una entrada en la base de datos RIP, se actualizará con la información más reciente.

h. Envío de nuevas rutas y actualizada sobre la próxima actualización disparada

3.2.4.2.1. Procesamiento del protocolo de enrutamiento por medio de los Bloques Lógicos Funcionales en ForCES

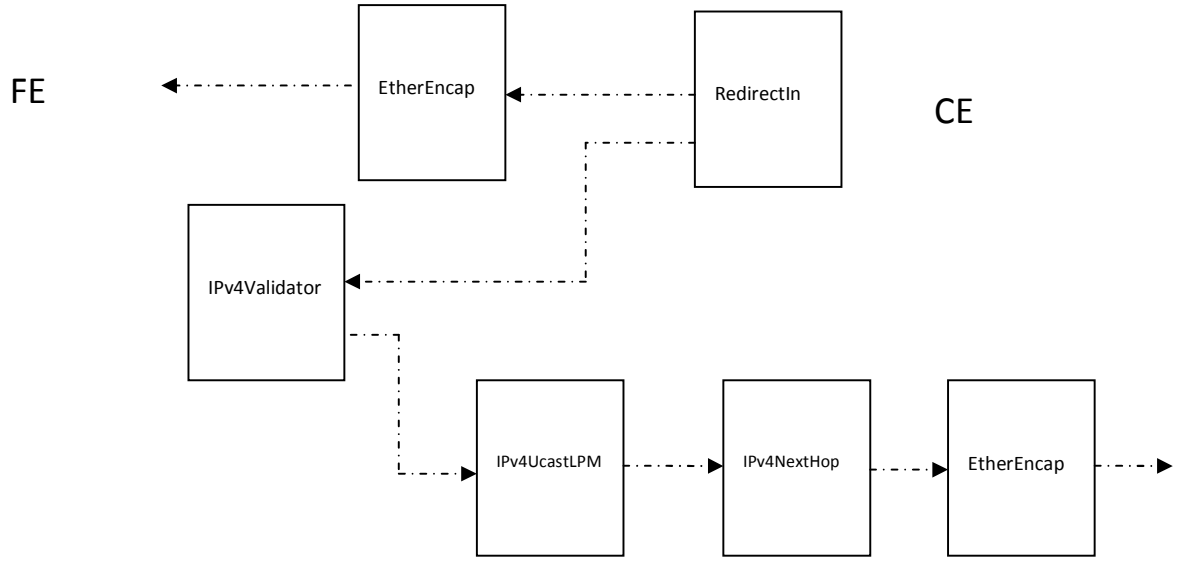


Figura 25. Proceso de Enrutamiento por medio de los LFBs en ForCES (Elaborado por Susan Martínez)

Desde el CE se puede:

a. Consultar la tabla de enrutamiento
b. Configurar el protocolo de enrutamiento
c. Enviar un paquete a un destino fuera de la red.

El proceso para enviar un paquete IPv4 por medio del protocolo ForCES desde el CE hacia a un host que se encuentra en una red externa, se realiza como se muestra en la figura 25.

Cuando un paquete sale desde el CE y se va a enviar a un destino fuera del NE, este paquete pasa primero por el LFB RedirectIn, este bloque como se ha mencionado antes, es el encargado de procesar los paquetes entre el CE y el FE. Luego el paquete llega al LFB de nombre IPV4Validator, este bloque valida los paquetes IPv4 y determina si es un paquete unicast o multicast, por lo general los paquetes son unicast, si es así, este paquete pasa al LFB IPv4 UcastLPM, que es el encargado de buscar en la tabla el prefijo más largo (LPM, Longest Prefix Match). Este bloque tiene un componente llamado IPv4PrefixTable, en el cual se encuentra la información de direcciones IPv4, un selector de salto y una bandera de ruta por defecto.

El LFB IPv4 UcastLPM, usa la dirección de destino del paquete IPv4 como un índice para buscar en la tabla de prefijos LPM y generar un selector de próximo salto como resultado. Por ejemplo, si se tienen las direcciones
192.168.20.16/28
192.16.0.0/16

Con el LPM se escoge la dirección de mayor prefijo siendo en este caso el /28

Luego que se selecciona el prefijo más largo, el paquete pasa al bloque LFB IPv4NextHop, en el cual se realiza el proceso de selección del próximo salto IPv4. A la salida del LFB IPv4UcastLPM, al paquete IPv4 se le asocia un ID indicando el puerto por donde debe salir el paquete ya sea FastEthernet, Ethernet o serial. Este LFB tiene un componente llamado IPv4NextHopTable, que es un arreglo y cada fila del arreglo es una estructura que contiene:

- L3PortID, el cual es el ID del puerto lógico de salida.
- MTU, Unidad Máxima de Transmisión para el puerto de salida
- NextHopIPAddr, dirección IPv4 del próximo salto.

Ya conociéndose el próximo salto, el paquete llega al LFB EtherEncap, que es el encargado de encapsular el paquete en una trama Ethernet y enviarla a un LFB EtherMACOut y finalmente al LFB EtherPHYCop.

3.2.4.3. PROTOCOLO SIMPLE DE GESTION DE RED - SNMP

SNMP, Protocolo Simple de Administración de Red (Simple Network Management Protocol) es un estándar de administración de redes basado en el conjunto de protocolos TCP/IP, que permiten la consulta a los diferentes elementos que constituyen la red.

Existen tres versiones de SNMP, SNMP v1, SNMP v2, y SNMP v3, permitiendo a los administradores de red:

- Supervisar la operación de la red.
- Configurar equipos.
- Encontrar y resolver fallos.
- Analizar prestaciones de los equipos.
- Acceder a la información de productos de diferentes fabricantes de una misma manera, desarrollando una herramienta común de monitoreo.

Una red que trabaje con el protocolo SNMP se basa en cuatro componentes:

1. Estructura de Administración de la Información (SMI), es el lenguaje de definición de datos, que especifica los tipos de datos, un modelo de objetos y reglas para escribir y comprobar la información de administración.
2. Administración de la Base de Información (MIB): un objeto de red, es conocido como objetos MIB. La información de administración se representa como un conjunto de objetos que conforman un almacenamiento de información virtual, conocido como Base de Información de Administración. Un objeto MIB, puede ser un contador (por ejemplo el número de datagramas IP que han sido eliminados en el router debido a los errores en la cabecera del datagrama IP o bien el número de errores de detección de la portadora en una tarjeta de interfaz Ethernet).
3. Protocolo Simple de Administración de Redes (SNMP)
4. Capacidad de seguridad y administración de objetos.

Procesamiento de un comando de usuario en un gestor SNMP

Cuando un administrador de SNMP procesa un comando:

- Si se trata de un comando Get-Request:
  a. El administrador de SNMP, crea un paquete de solicitud SNMP GET y la envía al agente de destino.
- Si se trata de un comando GET-BULK-Request:
  a. El administrador SNMP chequea la versión SNMP seleccionada y sólo envía una solicitud SNMP GET-BULK, si la versión de SNMP es 2 o superior.
- Si se trata de un comando SET-Request:
  a. El administrador SNMP crea un paquete de solicitud SNMP-SET y lo envía al agente de destino.

Procesamiento de paquetes con SNMP
Cuando un administrador de SNMP recibe un paquete:

- Se comprueba si el paquete tiene un encabezado SNMP correcto. Si es correcto, se pasa a la siguiente etapa. De lo contrario, se descarta el paquete.
- Verifica si el tiempo de espera desde la última solicitud enviada a este destino ha expirado. Si no ha expirado, se va a la siguiente etapa. De lo contrario, se descarta el paquete.
- Comprueba si el encabezado SNMP contiene una correcta PDU SNMP. Si es correcta, se pasa a la siguiente etapa. De lo contrario, se descarta el paquete.
- Comprueba si el PDU SNMP es de tipo SNMP GET-Response. Si es correcta, se pasa a la siguiente etapa. De lo contrario, se descarta el paquete.
- Se comprueba el estado de error de la PDU:

  a. Si hay un error, la señal del MIB buscara para mostrar la cadena de errores. O la cadena de error se puede visualizar a través de la línea de comandos.
  b. Si no hay ningún error:
     - Procesa las asignaciones de variables PDU en el SNMP PDU.
     - Señala el navegador MIB para mostrar el resultado. O el resultado se puede visualizar a través de la línea de comandos.

3.2.4.3.1. Procesamiento del protocolo Simple de Gestión de Red (SNMP) por medio de los Bloques Lógicos Funcionales en ForCES

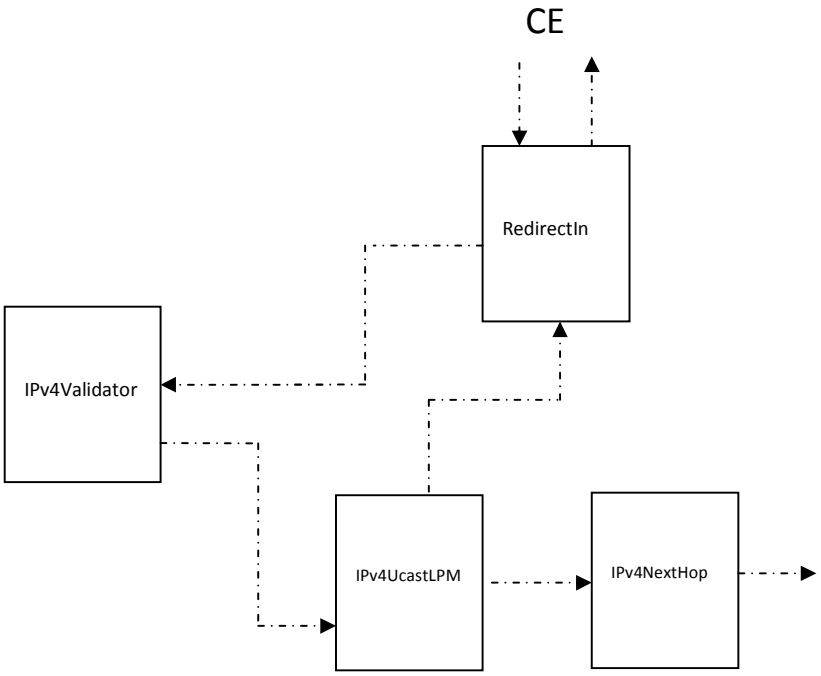


Figura 26. Proceso de Gestión de red por medio de los LFBs en ForCES (Elaborado por Susan Martínez)

Cuando el CE envía una solicitud para visualizar una interfaz en cuanto a paquetes y bytes enviados y recibidos a un destino fuera del NE, este paquete pasa primero por el LFB RedirectIn, este bloque como se ha mencionado antes, es el encargado de procesar los paquetes entre el CE y el FE. Luego el paquete llega al LFB IPV4Validator, este bloque valida los paquetes IPv4 y determina si es un paquete unicast o multicast, por lo general los paquetes son unicast, si es así, este paquete pasa al LFB IPv4 UcastLPM, que es el encargado de buscar en la tabla el prefijo más largo (LPM, Longest Prefix Match). El LFB IPv4 UcastLPM, usa la dirección de destino del paquete IPv4 como un índice para buscar en la tabla de prefijos LPM y generar un selector de próximo salto como resultado, como se explico anteriormente.

3.2.5. PROCESAMIENTO DE IPV4 FORWARDING POR MEDIO DE LOS LFB DEL PROTOCOLO FORCES

Como se ha descrito anteriormente, cuando ingresan los datos por medio del un puerto físico a un NE, en este caso un enrutador, llega primero al LFB EtherPHYCop, que es un bloque funcional que asocia el puerto físico con un ID que lo identifica, estos datos pasan al LFB EtherMACIn, el cual asocia ese puerto a una dirección MAC, los datos pasan al LFB EtherClassifier, el cual se encarga de desencapsular la trama y clasificarla, dependiendo si es IPv4, IPv6, ARP, entre otros, si es un paquete IPv4 pasa la LFB IPv4Validator, donde el paquete es clasificado como unicast o multicast, si es unicast pasa al LFB IPv4UcastLPM, donde se busca en la tabla de prefijos el más largo, como se explico en el numeral anterior, cuando se encuentra el selector de salto, pasa al LFB IPv4NextHop donde se asocia ese selector a la dirección IP de próximo salto y al puerto físico de salida, después de conocer el puerto de salida, el paquete pasa al LFB EtherEncap, el cual se encarga de encapsular el paquete a una trama Ethernet, si el paquete es para difusión por ejemplo una solicitud de ARP, pasa al LFB BasicMetadataDispatch, el cual se encarga de enviarlo a todos LFBs EtherMACOut conectados a él, en el EtherMACOut, se verifica la MAC que tiene la trama Ethernet y la asocia a un puerto físico de salida. Este proceso se puede observa en la figura 27.

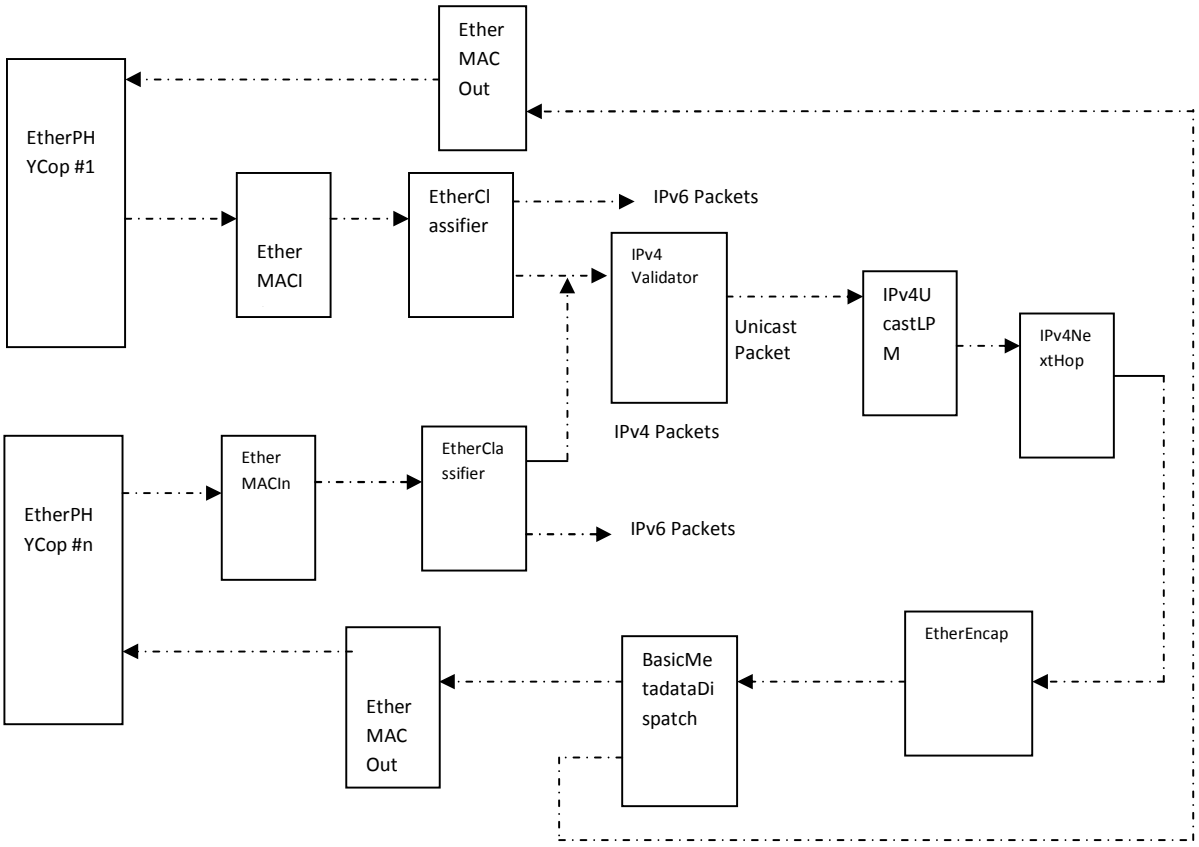


Figura 27. Proceso de Gestión de Forwarding por medio de los LFBs en ForCES (Tomado del ForCES LFB Library draft-ietf-forces-lfb-lib-00)

4. ESTADO DEL ARTE

El IETF creó el grupo de trabajo llamado ForCES [14], ya que con la aparición de nuevos procesadores de red embebidos, que permiten el acceso por la ruta rápida en el plano de reenvió en los elementos de red tales como router, junto con la aparición de nueva generación de señalización, enrutamiento y aplicaciones en el plano de control, ha creado la necesidad de estandarizar mecanismos para permitir que estos componentes se combinen en todas sus funciones. ForCES tiene como objetivo definir un framework y los mecanismos asociados para la estandarización en el intercambio de información entre la separación física del plano de control, incluidas las entidades tales como los protocolos de enrutamiento, control de admisión, y la señalización, y el plano de reenvío, en donde las actividades por paquete tales como reenvío de paquetes, encolamiento, y la edición de encabezados. Al definir un conjunto de mecanismos estándar para la separación de los planos de control y reenvío, ForCES permitirá la rápida innovación en paralelo, manteniendo la interoperabilidad de los planos.

Los productos de este grupo de trabajo son y serán:

a. Un conjunto de requerimientos para los planos de control y reenvío, mecanismos lógicamente separados de un elemento de red IP (NE)

b. Un estamento de aplicabilidad para el modelo y el protocolo ForCES.

c. RFC informativos necesarios para documentar el enfoque actual del modelo funcional y objetos controlados por ellos.

d. Una arquitectura de framework que define las entidades que comprende un elemento de red ForCES y la identificación de las interacciones entre ellos.

e. Una definición formal de los objetos controlados en el modelo funcional de un elemento de reenvío. Esta incluye el reenvío IP, IntServ y DiffServ QoS.

Los RFC definidos por el grupo ForCES del IETF. [7], [8], [9], [10], [11], [12], [13], [23], algunos estandarizados, otros draft, los cuales se han ido mejorando con nuevas versiones, fueron de gran apoyo en el desarrollo del proyecto, ya que indicaban como estaba conformado cada elemento, su función y arquitectura.

En cuanto a proyectos de investigación, en realidad son pocas las universidades y entidades que han trabajado esta arquitectura. La universidad Zhejiang Gongshang de China, tiene un grupo consolidado que se ha enfocado en realizar diseño, implementación y pruebas en routers distribuidos ForCES, el último proyecto elaborado se basa en HA (alta Disponibilidad) en los elementos de control (principal y backup). Entre sus proyectos se tienen [5], [18], [19], [20], [21], [22], [23], entre otros.

5. DISEÑO DEL ELEMENTO DE CONTROL

Este proyecto de grado propone el diseño e implementación de un Elemento de Control basado en la arquitectura ForCES y se comunica con un Elemento de Forwarding por medio del protocolo ForCES. En la figura 28, se puede observar el diagrama de prototipo.

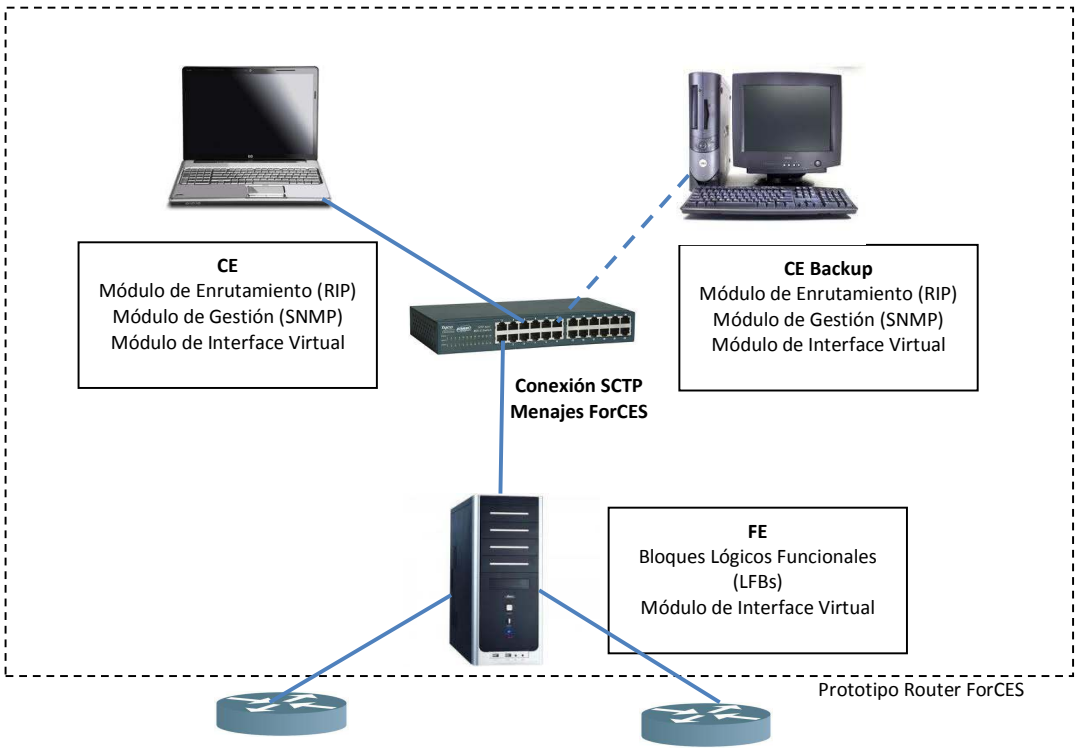


Figura 28. Implementación física del Elemento de Control (Elaborado por Susan Martínez)

El Elemento de Control está diseñado bajo la normatividad y estructura dada por los documentos que establece el grupo de trabajo en Ingeniería en Internet EITF para ForCES, tanto arquitectura como el manejo del protocolo y además sobre la estructura de los bloques lógicos funcionales (LFBs).

El prototipo de CE está compuesto de seis módulos principales: Módulo ForCES, Módulo IP, Módulo de Interfaz Virtual, Módulo de Enrutamiento, Módulo de Gestión y Módulo de Alta Disponibilidad.

El módulo ForCES, es el encargado de establecer la conexión entre el CE-FE y de permitir al Elemento de Control (CE) la manipulación funcional, gestión y configuración de los Elementos de Reenvío (FE) por medio de los LFB y además permite la interacción y gestión de los elementos de

control (CE), cuando falle un CE en el elemento de red, por medio de mensajes Heartbeat se da alarma para que el FE se re-asocie a otro CE configurado como respaldo. El módulo ForCES fue desarrollado por el Ingeniero Pedro Luis González como proyecto para optar el titulo de Msc en Ingeniería Electrónica de la Pontificia Universidad Javeriana.

En el módulo IP, gestiona el direccionamiento de las interfaces físicas del CE, gestionar, se refiere a configurar, consultar y habilitar/deshabilitar las interfaces que hacen parte del FE.

El módulo de Enrutamiento, permite configurar el protocolo de enrutamiento RIP versión 2 y consultar las tablas de enrutamiento.

El módulo de Gestión, por medio del Net-SNMP es el encargado de visualizar algunos parámetros como son: bytes enviados, bytes recibidos, paquetes enviados y paquetes recibidos por cada una de las interfaces, permitiendo monitorear la red y el router.

El módulo de Interfaz Virtual, MIVS (Módulo de Interfaz Virtual Servidor) el cual es una aplicación que permite reflejar las interfaces que se encuentran en el FE como si estuvieran en el CE y de esta forma poder configurar desde el CE el protocolo de enrutamiento RIP versión 2 y luego transportar esa información por medio de túneles virtuales hacia las interfaces físicas en el FE y luego hacia el exterior.

En el FE se encuentran los LFBs, el módulo MIVC (Módulo de Interfaz Virtual Cliente), permite realizar una réplica de las interfaces físicas que tiene el FE en interfaces lógicas para reflejarlas hacia el MIVS que se encuentra en el CE, de modo que el CE asume estas interfaces remotas como suyas, se puede observar en la figura 29.

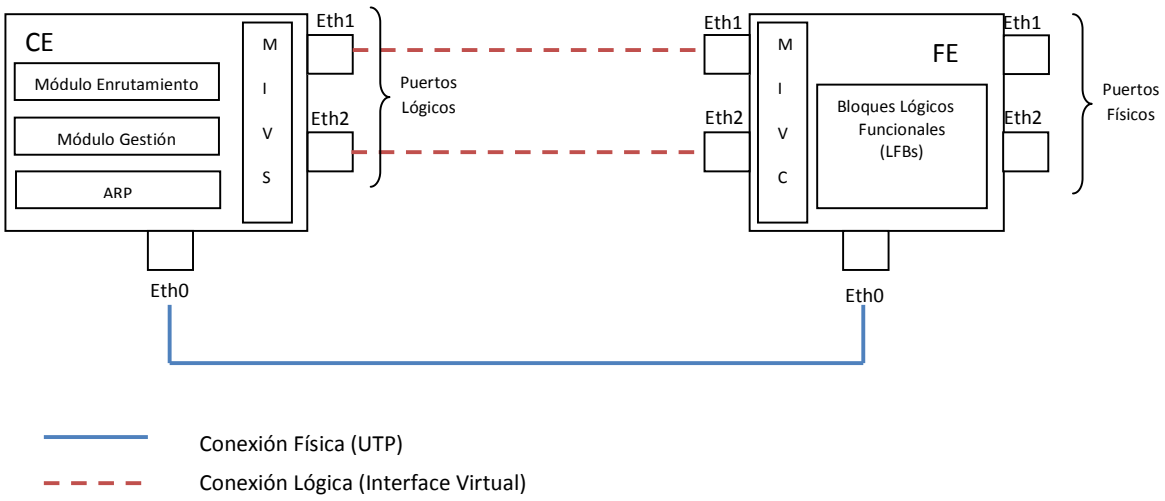


Figura 29. Interconexión lógica del prototipo (Elaborado por Susan Martínez)

El módulo de Alta Disponibilidad es el encargado de mantener la operatividad en el NE, en caso de que falle el CE principal, el FE entra al estado de pre-asociación e inicia la búsqueda de otro CE (backup) para asociarse nuevamente y continuar operativo.

5.1. INGENIERIA DE SOFTWARE DEL ELEMENTO DE CONTROL

Para la construcción de los diagramas de casos de uso de este proyecto, se empleo el método CRUD (Create, Read, Update y Delete; Crear, Obtener, Actualizar y Borrar).

5.2. MÓDULOS DEL ELEMENTO DE CONTROL

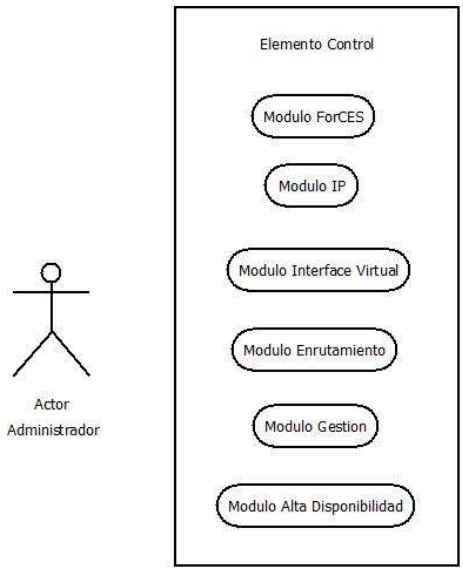


Figura 30. Módulos del Elemento de Control (Elaborado por Susan Martínez)

Como se explico anteriormente el Elemento de Control está compuesto por seis módulos que permiten su funcionamiento, el módulo ForCES, módulo IP, Módulo Interfaz Virtual, Módulo de Enrutamiento, Módulo de Gestión y Módulo de Alta Disponibilidad, como se observa en la figura 30.

5.2.1. CASO DE USO MÓDULO IP

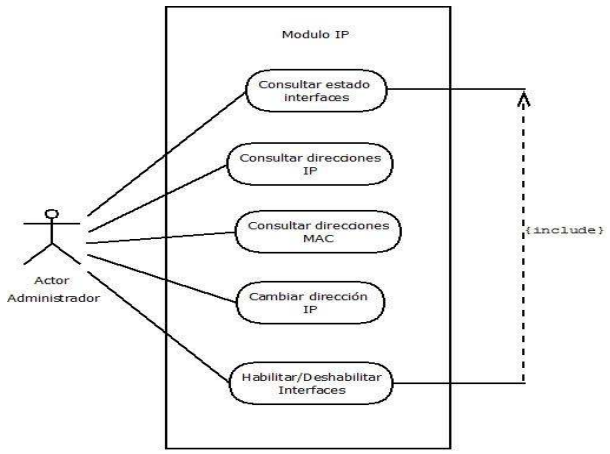


Figura 31. Parámetros del Módulo IP (Elaborado por Susan Martínez)

Actores:

- Administrador. Administrador que configura y controla el módulo del protocolo de Internet.

Casos de Uso:

a. Núcleo del Negocio:

- Consultar estado de Interfaces: El administrador consulta el estado de las interfaces (si están habilitadas, deshabilitadas o configuradas).
- Consultar direcciones IP: El administrador consulta si las interfaces físicas y lógicas tienen configuradas las direcciones IP y cuales direcciones son.
- Consultar direcciones MAC: El administrador consulta las direcciones MAC de las interfaces físicas.

b. CRUD (Create, Read, Update, Delete/Crear, Obtener, Actualizar, Borrar)

- Cambiar dirección IP: El administrador cambia o elimina la dirección IP de una interfaz.
- Habilitar/Deshabilitar Interfaces. El administrador habilita o deshabilita las interfaces dependiendo del estado de las interfaces.

c. Reportes

- No aplica.

Especificación del Caso de Uso "Consultar estado de Interfaces"

CASO DE USO	Consultar estado Interfaces		
Descripción	El administrador consulta el estado de las interfaces (si están habilitadas, deshabilitadas o configuradas).		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El sistema debe haber arrancado de manera exitosa		
Poscondiciones	Mostrar en pantalla o interfaz de usuario las interfaces y su estado		
FLUJO NORMAL DE EVENTOS			

1. El administrador ingresa al modo privilegiado del elemento de control. 2. El administrador consulta las interfaces que tiene el elemento de control. 3. El sistema muestra en pantalla las interfaces que tiene y el estado de cada una (habilitada, deshabilitada, configurada) 4. El administrador visualiza el estado de las interfaces.
FLUJOS ALTERNOS
o El sistema no muestra interfaces. Si en el paso 3. el sistema no se visualiza ninguna interface puede ser porque el comando de consulta esta errado y el sistema muestra un mensaje de error y termina el caso de uso.
EXCEPCIONES

REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

Especificación del Caso de Uso "Consultar direcciones IP"

CASO DE USO	Consultar direcciones IP		
Descripción	El administrador consulta si las interfaces físicas y lógicas tienen configuradas las direcciones IP y cuales direcciones son.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El sistema debe haber arrancado de manera exitosa		
Poscondiciones	Mostrar en pantalla o interfaz grafica de usuario las direcciones IP		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa al modo privilegiado del elemento de control. 2. El administrador consulta las direcciones IP de las interfaces que tiene el elemento de control. 3. El sistema muestra en pantalla las direcciones IP de las interfaces. 4. El administrador visualiza las direcciones IP de las interfaces.
FLUJOS ALTERNOS

o El sistema no muestra direcciones IP. Si en el paso 3. el sistema no se visualiza ninguna dirección IP de las interfaces puede ser porque el comando de consulta esta errado y el sistema muestra un mensaje de error y termina el caso de uso.
EXCEPCIONES

REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

Especificación del Caso de Uso "Consultar direcciones MAC"

CASO DE USO	Consultar direcciones MAC		
Descripción	El administrador consulta las direcciones MAC de las interfaces físicas.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El sistema debe haber arrancado de manera exitosa		
Poscondiciones	Mostrar en pantalla o interfaz grafica de usuario la tabla con direcciones MAC		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa con un usuario que tenga perfil para generar comandos para revisión de ARP. 2. Consultar que las interfaces físicas estén habilitadas o administrativamente arriba. 3. Se digita el comando para visualizar la tabla con direcciones MAC 4.El administrador visualiza las direcciones MAC de las interfaces.
FLUJOS ALTERNOS
o Las interfaces no se encuentran administrativamente arriba o habilitadas Entonces paso 3. Se habilitan las interfaces. 4. Se digita el comando para visualizar la tabla de direcciones MAC. 5. El administrador visualiza las direcciones MAC de las interfaces.
EXCEPCIONES

REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

Especificación del Caso de Uso "Cambiar direcciones IP"

CASO DE USO	Cambiar direcciones IP		
Descripción	El administrador cambia o elimina la dirección IP de una interfaz		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El sistema debe haber arrancado de manera exitosa		
Poscondiciones	Mostrar en pantalla o interface grafica de usuario la nueva dirección IP		

FLUJO NORMAL DE EVENTOS

1. El administrador ingresa con un usuario que tenga perfil para generar comandos para cambio de direcciones.
2. Consultar que las interfaces físicas estén habilitadas o administrativamente arriba.
3. El administrador verifica si la interfaz tiene una dirección IP configurada.
4. El administrador digita el comando para cambiar la dirección IP.
5. El sistema muestra la dirección IP de la interfaz.

FLUJOS ALTERNOS

o Si no hay una dirección IP configurada en la interfaz
Entonces paso 4. El administrador digita el comando para asignar una dirección IP a la interfaz.
5. El sistema muestra la dirección IP de la interfaz.

EXCEPCIONES

REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

Especificación del Caso de Uso "Habilitar/Deshabilitar Interfaces"

CASO DE USO	Habilitar/Deshabilitar Interfaces		
Descripción	El administrador habilita o deshabilita las interfaces dependiendo del estado de las interfaces.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012

Actores	Administrador
Precondiciones	El sistema debe haber arrancado de manera exitosa
Poscondiciones	N/A
FLUJO NORMAL DE EVENTOS	
Habilitar: 1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2. El administrador digita el comando para visualizar el estado de las interfaces. 3. El sistema muestra las opciones de habilitar o deshabilitar interfaces. 4. El administrador selecciona la opción de habilitar. 5. El sistema habilita la interfaz.	
FLUJOS ALTERNOS	
Deshabilitar En el paso 4, del flujo normal, el administrador selecciona la opción de deshabilitar. 5. El sistema deshabilita la interfaz.	
EXCEPCIONES	
REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

5.2.2. CASO DE USO MÓDULO ENRUTAMIENTO

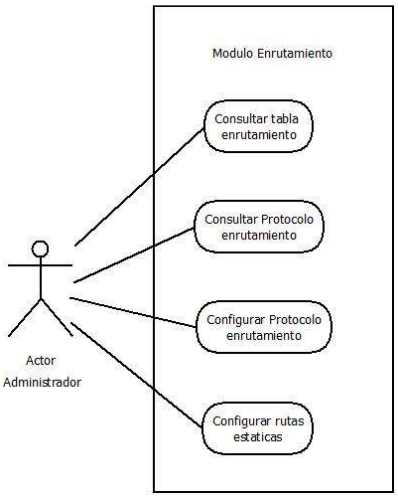


Figura 32. Caso de uso del Módulo de Enrutamiento (Elaborado por Susan Martínez)

Actores:

- Administrador. Administrador que configura y controla el módulo del enrutamiento.

Casos de Uso:

a. Núcleo del Negocio:

- Consultar tabla de enrutamiento: El administrador consulta la tabla de enrutamiento (mejores rutas o rutas optimas).
- Consultar protocolo de enrutamiento: El administrador consulta el protocolo de enrutamiento configurado.

b. CRUD

- Configurar protocolo de enrutamiento: El administrador configura o cambia el protocolo de enrutamiento.
- Configurar rutas estáticas. El administrador configura las rutas estáticas.

c. Reportes

- No aplica.

Especificación del Caso de Uso "Consultar tabla de enrutamiento"

CASO DE USO	Consultar tabla de enrutamiento		
Descripción	El administrador consulta la tabla de enrutamiento (mejores rutas o rutas optimas).		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El sistema debe haber arrancado de manera exitosa		
Poscondiciones	El sistema muestra en pantalla o interfaz grafica de usuario la tabla de enrutamiento		
FLUJO NORMAL DE EVENTOS			

1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2.El administrador digita el comando para visualizar la tabla de enrutamiento 3.El sistema muestra la tabla de enrutamiento.
FLUJOS ALTERNOS
Si no se visualiza la tabla de enrutamiento En el paso 2, del flujo normal, el administrador digita el comando para visualizar la tabla de enrutamiento. 3. El sistema no muestra la tabla de enrutamiento. 4. El administrador configura el protocolo de enrutamiento. 5. El administrador digita el comando para visualizar la tabla de enrutamiento. 6. El sistema muestra la tabla de enrutamiento
EXCEPCIONES

REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

Especificación del Caso de Uso "Consultar protocolo de enrutamiento"

CASO DE USO	Consultar protocolo de enrutamiento		
Descripción	El administrador consulta el protocolo de enrutamiento configurado.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El sistema debe haber arrancado de manera exitosa		
Poscondiciones	El sistema muestra en pantalla o interfaz grafica de usuario el protocolo de enrutamiento		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2. El administrador digita el comando para visualizar el protocolo de enrutamiento configurado 3.El sistema muestra el protocolo de enrutamiento.

FLUJOS ALTERNOS
Si el protocolo de enrutamiento no está configurado En el paso 2, del flujo normal, el administrador digita el comando para visualizar el protocolo de enrutamiento. 3. El sistema no muestra protocolo de enrutamiento configurado. 4. El administrador configura el protocolo de enrutamiento. 5. El administrador digita el comando para visualizar el protocolo de enrutamiento configurado. 6. El sistema muestra rl protocolo de enrutamiento
EXCEPCIONES

REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

Especificación del Caso de Uso "Configurar protocolo de enrutamiento"

CASO DE USO	Consultar protocolo de enrutamiento		
Descripción	El administrador consulta el protocolo de enrutamiento configurado.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El sistema debe haber arrancado de manera exitosa		
Poscondiciones	El sistema muestra en pantalla o interfaz grafica de usuario el protocolo de enrutamiento		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2.El administrador digita el comando para visualizar el protocolo de enrutamiento configurado 3.El sistema muestra el protocolo de enrutamiento.
FLUJOS ALTERNOS

Si el protocolo de enrutamiento no está configurado En el paso 2, del flujo normal, el administrador digita el comando para visualizar el protocolo de enrutamiento. 3. El sistema no muestra protocolo de enrutamiento configurado. 4. El administrador configura el protocolo de enrutamiento. 5. El administrador digita el comando para visualizar el protocolo de enrutamiento configurado. 6. El sistema muestra rl protocolo de enrutamiento
EXCEPCIONES

REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

Especificación del Caso de Uso "Configurar rutas estáticas"

CASO DE USO	Configurar rutas estáticas		
Descripción	El administrador configura las rutas estáticas.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El sistema debe haber arrancado de manera exitosa		
Poscondiciones	N/A		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2. El administrador digita el comando para configurar enrutamiento estático 3.El sistema muestra las rutas de enrutamiento estáticas.
FLUJOS ALTERNOS
EXCEPCIONES

REFERENCIAS	Formato router convencional
ANOTACIONES	N/A

5.2.3. CASO DE USO MÓDULO DE GESTION

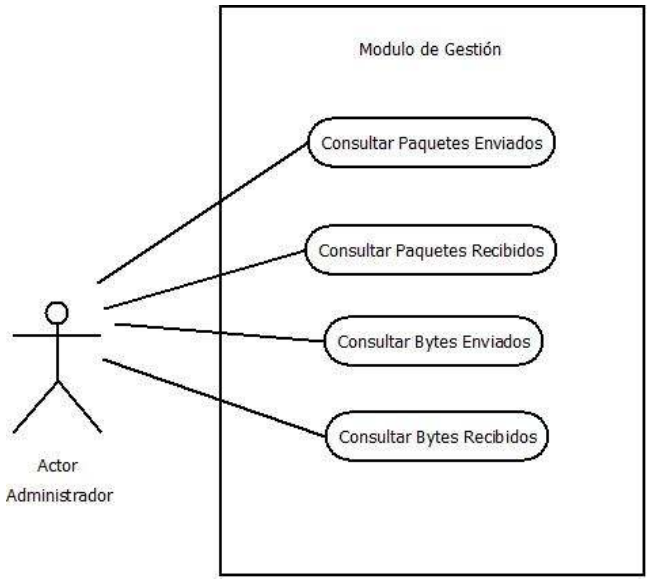


Figura 33. Diagrama Caso de Uso Módulo de Gestión (Elaborado por Susan Martínez)

Actores:

- Administrador. Administrador que configura y controla el módulo del gestión.

Casos de Uso:

a. Núcleo del Negocio:

- Consultar paquetes enviados: El administrador consulta los paquetes enviados por las diferentes interfaces.

- Consultar paquetes recibidos: El administrador consulta los paquetes recibidos por las diferentes interfaces.

- Consultar bytes enviados: El administrador consulta los bytes enviados por las diferentes interfaces.

- Consultar bytes recibidos: El administrador consulta los bytes recibidos por las diferentes interfaces.

Especificación del Caso de Uso "Consultar Paquetes Enviados"

CASO DE USO	Consultar paquetes enviados
Descripción	El administrador solicita la consulta de cuantos paquetes se enviaron en cada una de las

	interfaces.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES El módulo de Interfaz Virtual debe estar activo		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			
1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2. El administrador digita el comando para consulta de paquetes enviados. 3. El sistema visualiza el número de paquetes enviados por cada interface del elemento de forwarding (FE).			
FLUJOS ALTERNOS			
EXCEPCIONES			
REFERENCIAS	Comandos Net-SNMPv2		
ANOTACIONES	N/A		

Especificación del Caso de Uso "Consultar Paquetes Recibidos"

CASO DE USO	Consultar paquetes recibidos		
Descripción	El administrador solicita la consulta de cuantos paquetes se recibieron en cada una de las interfaces.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES El módulo de Interfaz Virtual debe estar activo		
Poscondiciones	N/A		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2. El administrador digita el comando para consulta de paquetes recibidos. 3. El sistema visualiza el número de paquetes recibidos por cada interfaz del elemento de forwarding (FE).
FLUJOS ALTERNOS
EXCEPCIONES

REFERENCIAS	Comandos Net-SNMPv2
ANOTACIONES	N/A

Especificación del Caso de Uso "Consultar Bytes Enviados"

CASO DE USO	Consultar bytes enviados		
Descripción	El administrador solicita la consulta de número de bytes se enviaron en cada una de las interfaces.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES El módulo de Interfaz Virtual debe estar activo		
Poscondiciones	N/A		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2. El administrador digita el comando para consulta del número de bytes enviados. 3. El sistema visualiza el número de bytes enviados por cada interfaz del elemento de forwarding (FE).

FLUJOS ALTERNOS	
EXCEPCIONES	
REFERENCIAS	Comandos Net-SNMPv2
ANOTACIONES	N/A

Especificación del Caso de Uso "Consultar Bytes Recibidos"

CASO DE USO	Consultar bytes recibidos		
Descripción	El administrador solicita la consulta de cuantos bytes se recibieron en cada una de las interfaces.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES El módulo de Interfaz Virtual debe estar activo		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			
1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2. El administrador digita el comando para consulta del número de bytes recibidos. 3. El sistema visualiza el número de bytes recibidos por cada interface del elemento de forwarding (FE).			
FLUJOS ALTERNOS			
EXCEPCIONES			
REFERENCIAS	Comandos Net-SNMPv2		
ANOTACIONES	N/A		

5.2.4. CASO DE USO MÓDULO DE INTERFAZ VIRTUAL

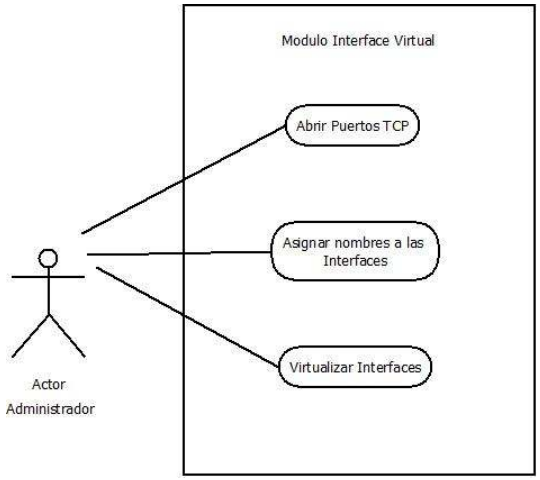


Figura 34. Diagrama Caso de Uso Módulo Interfaz Virtual (Elaborado por Susan Martínez)

Actores:

- Administrador. Administrador que configura y controla el módulo interfaz virtual.

Casos de Uso:

a. Núcleo del Negocio:

- Abrir puertos TCP: El administrador configura puertos TCP asignar uno a cada interfaz que se va a virtualizar.

- Asignar nombres a las interfaces: El administrador asigna un nombre a cada interfaz real que se va a virtualizar.

- Virtualizar Interfaces: El administrador virtualiza las interfaces reales, teniendo nombre a cada interfaz y puerto asignado.

Especificación del Caso de Uso "Abrir Puertos TCP"

CASO DE USO	Abrir puertos TCP		
Descripción	El administrador configura puertos TCP para establecer la conexión virtual de las interfaces físicas del FE.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES		

Poscondiciones	N/A
FLUJO NORMAL DE EVENTOS	
1. El administrador ingresa con un usuario que tenga perfil para configurar los puertos TCP en modo escucha. 2. El administrador elige los puertos TCP (0-65535) por los que va a escuchar cada interfaz física. 3. El sistema visualiza las interfaces físicas en estado de espera.	
FLUJOS ALTERNOS	
EXCEPCIONES	
REFERENCIAS	Comandos Etherpuppet
ANOTACIONES	N/A

Especificación del Caso de Uso "Asignar nombre a las Interfaces "

CASO DE USO	Asignar nombre a las Interfaces		
Descripción	El administrador asigna nombre a las interfaces físicas que se van a virtualizar, por medio de este nombre el CE identificara las interfaces del FE como si fueran propias.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			
1. El administrador ingresa con un usuario que tenga perfil para configurar los puertos TCP en modo escucha. 2. El administrador toma los puertos TCP que elige escuchar cada interfaz física y le asigna un nombre .. 3. Relaciona la dirección lógica que cada interfaz física tiene configurada junto con el puerto TCP elegido y el nombre asignado.			
FLUJOS ALTERNOS			

EXCEPCIONES	
REFERENCIAS	Comandos Etherpuppet
ANOTACIONES	N/A

Especificación del Caso de Uso "Virtualizar Interfaces"

CASO DE USO	Virtualizar interfaces		
Descripción	El administrador habilita la virtualización de las interfaces, por lo tanto las interfaces físicas del FE, el CE las puede controlar y gestionar como si fueran sus propias interfaces.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			
1. El administrador ingresa con un usuario que tenga perfil para configurar los puertos TCP en modo escucha. 2. El administrador habilita las interfaces virtuales.			
FLUJOS ALTERNOS			
EXCEPCIONES			
REFERENCIAS	Comandos Etherpuppet		
ANOTACIONES	N/A		

5.2.5. CASO DE USO MÓDULO FORCES

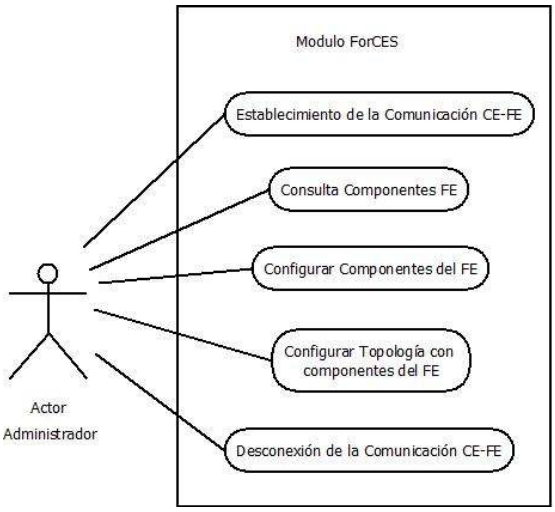


Figura 35. Diagrama Caso de Uso Módulo ForCES (Elaborado por Susan Martínez)

Casos de Uso:

a. Núcleo del Negocio:

- Establecimiento de la comunicación CE-FE: El administrador realiza la conexión entre CE y FE por medio de mensajes ForCES.
- Consultar componentes FE: El administrador consulta las instancias de los diferentes componentes del FE.
- Desconexión de la comunicación CE-FE: El administración realiza la desconexión entre CE-FE.

b. CRUD

- Configurar componentes del FE: El administrador configura o cambia los valores de las diferentes componentes del FE.
- Configurar topologías con componentes del FE: El administrador configura las topologías de SNMP, Enrutamiento, Forwarding y ARP con los LFBs que se encuentran en el FE.

c. Reportes

No aplica

Especificación del Caso de Uso "Establecimiento de la comunicación CE-FE"

CASO DE USO	Establecimiento de la comunicación CE-FE

Descripción	El administrador establece la pre-asociación y asociación por medio de la interfaz ForCES.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	Tener instalada la interfaz del protocolo ForCES		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			
1. El administrador ingresa con un usuario que tenga perfil para administrar el sistema. 2. El administrador realiza la pre-asociación y asociación del protocolo. 3. El sistema visualiza el establecimiento de la conexión ForCES.			
FLUJOS ALTERNOS			
EXCEPCIONES			
REFERENCIAS	Comandos ForCES		
ANOTACIONES	N/A		

Especificación del Caso de Uso "Consulta de los Componentes FE"

CASO DE USO	Consulta de los componentes FE		
Descripción	El administrador solicita la consulta si las interfaces están en estado activo o en estado inactivo en el elemento de control.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES El módulo de Interfaz Virtual debe estar activo		
Poscondiciones	N/A		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa con un usuario que tenga perfil para generar comandos en modo privilegiado. 2. El administrador digita el comando para consultar el estado de las interfaces. 3. El sistema visualiza si el estado de cada interfaz del elemento de forwarding (FE) está activo o inactivo.
FLUJOS ALTERNOS
EXCEPCIONES

REFERENCIAS	Comandos Net-SNMPv2
ANOTACIONES	N/A

CASO DE USO	Consultar estado de las interfaces		
Descripción	El administrador desde la aplicación que hace parte del CE, solicita la consulta de los componentes que hacen parte del FE que son los bloques lógicos funcionales.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES		
Poscondiciones	N/A		

FLUJO NORMAL DE EVENTOS
1. El administrador ingresa con un usuario que tenga perfil para gestionar el sistema. 2. El administrador digita el comando para consultar los componentes del FE, es decir los LFBs. 3. El sistema visualiza en el CE los componentes del FE.
FLUJOS ALTERNOS
EXCEPCIONES

REFERENCIAS	Comandos ForCES
ANOTACIONES	N/A

Especificación del Caso de Uso "Configurar Componentes del FE"

CASO DE USO	Configurar componentes del FE		
Descripción	El administrador desde la aplicación que hace parte del CE, configura las diferentes instancias de los componentes que hacen parte del FE que son los bloques lógicos funcionales.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			
1. El administrador ingresa con un usuario que tenga perfil para gestionar el sistema. 2. El administrador digita el comando para configurar las instancias de los componentes del FE, es decir los LFBs. 3. El sistema visualiza la configuración de las instancias en el CE de los diferentes componentes del FE.			
FLUJOS ALTERNOS			
EXCEPCIONES			
REFERENCIAS	Comandos ForCES		
ANOTACIONES	N/A		

Especificación del Caso de Uso "Configurar Topologías con Componentes del FE"

CASO DE USO	Configurar topologías con componentes del FE
Descripción	El administrador desde la aplicación que hace parte del CE, configura las diferentes instancias de los componentes que hacen parte del FE que son los bloques lógicos funcionales y configura topologías para el funcionamiento de ciertos protocolos como son ARP, RIP y SNMP
Autor	Susan Martínez Cordero

Fecha Creación	30/10/2011	**Fecha última modificación**	30/10/2011
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			
1. El administrador ingresa con un usuario que tenga perfil para gestionar el sistema. 2. El administrador digita el comando para configurar los componentes del FE, es decir los LFBs. 3. El sistema visualiza las topologías que configuro el CE con los diferentes componentes del FE			
FLUJOS ALTERNOS			
EXCEPCIONES			
REFERENCIAS	Comandos ForCES		
ANOTACIONES	N/A		

Especificación del Caso de Uso "Desconexión de la comunicación CE-FE"

CASO DE USO	Desconexión de la comunicación CE-FE		
Descripción	El CE o FE solicita la desconexión establecida.		
Autor	Susan Martínez Cordero		
Fecha Creación	30/10/2011	**Fecha última modificación**	16/11/2012
Actores	CE o FE		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			

1. El administrador ingresa con un usuario que tenga perfil para gestionar el sistema. 2. El CE o FE solicita desconexión. 3. El sistema se desasocia, conexión inactiva.
FLUJOS ALTERNOS
EXCEPCIONES

REFERENCIAS	Comandos ForCES
ANOTACIONES	N/A

5.2.6. CASO DE USO MÓDULO ALTA DISPONIBILIDAD

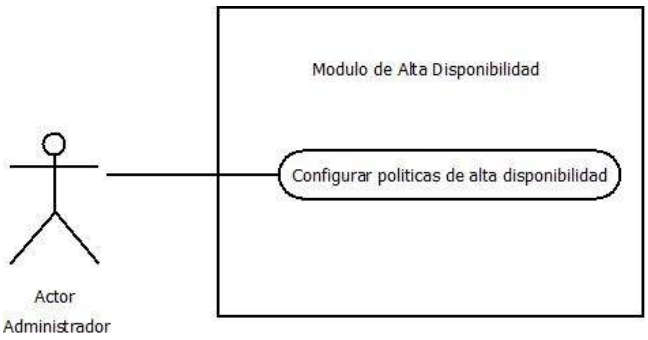


Figura 36. Diagrama Caso de Uso Módulo Alta Disponibilidad-HA (Elaborado por Susan Martínez)

Casos de Uso:

d. Núcleo del Negocio:

- Configurar políticas de Alta Disponibilidad: El administrador configura las diferentes políticas de alta disponibilidad, de modo que el elemento de red en lo posible quede operativo.

Especificación del Caso de Uso "Configurar políticas de Alta Disponibilidad"

CASO DE USO	Configurar políticas de Alta Disponibilidad		
Descripción	El administrador desde la aplicación que hace parte del CE, configura las diferentes componentes del FE que hacen parte de las políticas de alta disponibilidad del elemento de red.		
Autor	Susan Martínez Cordero		
Fecha Creación	1/11/2012	**Fecha última modificación**	16/11/2012
Actores	Administrador		
Precondiciones	El FE debe estar asociado a un CE y exitosa la conexión por ForCES		
Poscondiciones	N/A		
FLUJO NORMAL DE EVENTOS			
1.El administrador ingresa con un usuario que tenga perfil para gestionar el sistema. 2.El administrador configura desde el CE los componentes del FE Protocol LFB, que hacen parte de la Alta Disponibilidad en el FE. 3.El administrador configura el temporizador de búsqueda de CEs de Backup, tiempo en el cual el FE busca un CE de respaldo en caso de que el CE principal se desconecte. 4.El administrador configura el temporizador que indica al FE que no encontró CE de backups y el Elemento de Red deja de funcionar. 5. El sistema visualiza la configuración de los componentes del FE Protocol LFB en el CE.			
FLUJOS ALTERNOS			
EXCEPCIONES			
REFERENCIAS	Comandos ForCES		
ANOTACIONES	N/A		

5.3. DIAGRAMA DE SECUENCIA

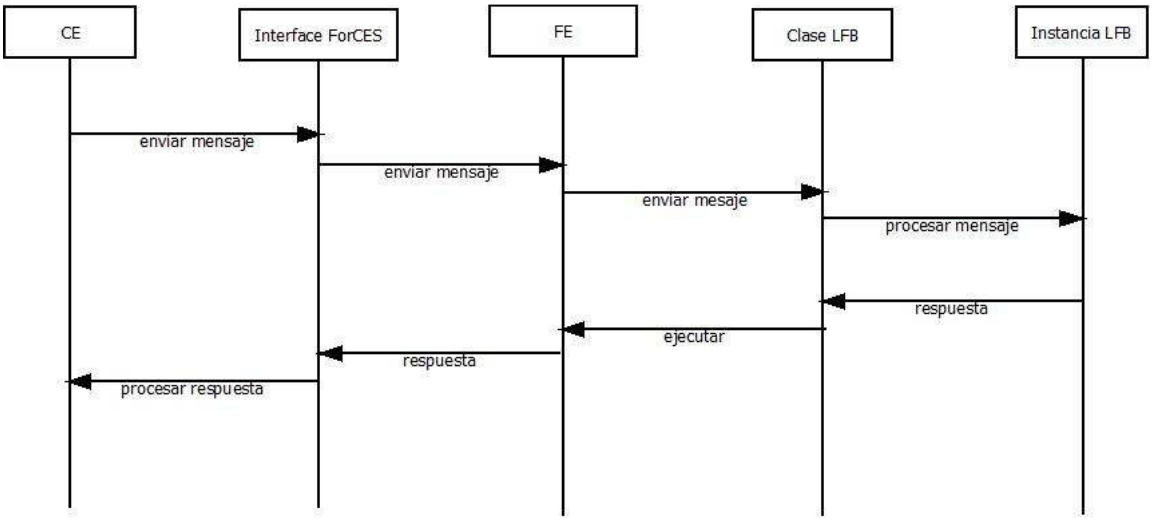


Figura 37. Diagrama de Secuencia (Elaborado por Susan Martínez)

Este diagrama de secuencia indica la interacción de los objetos de cada módulo que hacen parte de la aplicación. En la figura 37, se puede observar el envío general de mensajes desde el CE hacia el FE y su mensaje de respuesta.

5.4. DIAGRAMA DE CLASES

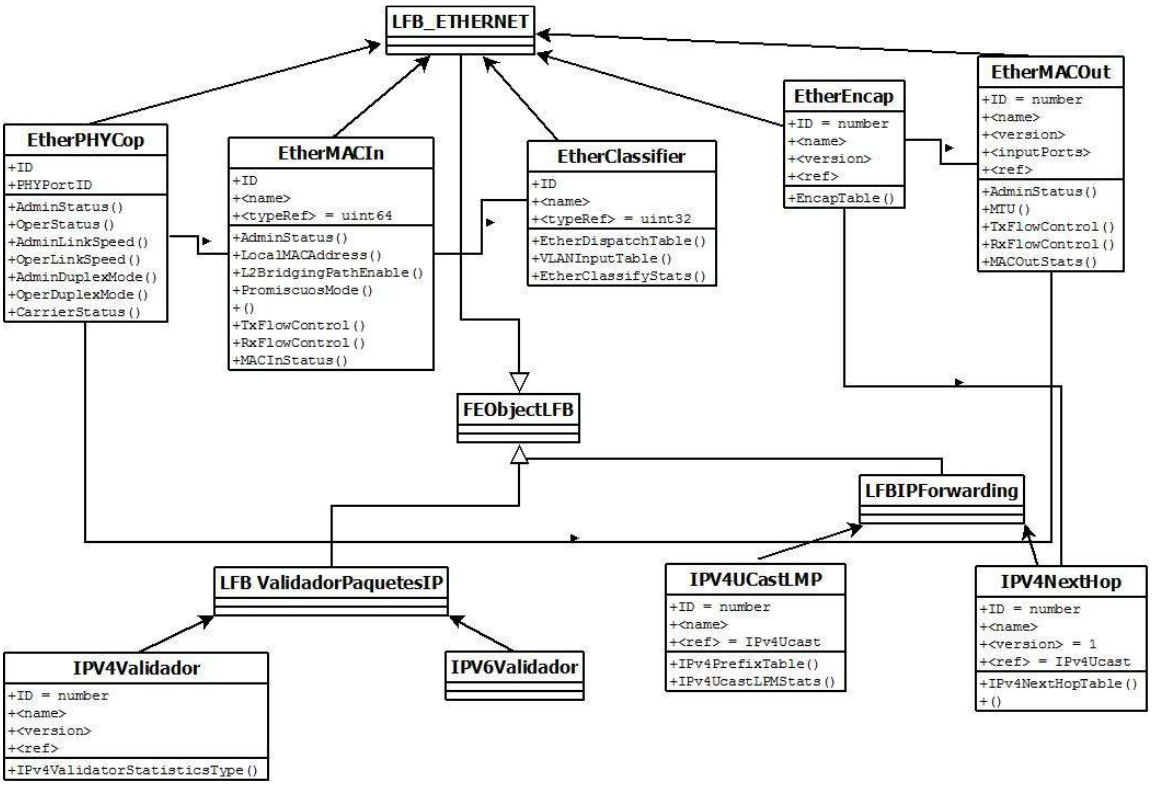


Figura 38. Diagrama de Clases de los LFBs

Este diagrama de clases, como se observa en la figura 38, describe cada uno de los grupos de objetos que hacen parte de los diferentes LFBs que hacen parte del FE, los cuales tienen ciertos atributos y características específicas.

6. DESARROLLO DEL ELEMENTO DE CONTROL

6.1. ELEMENTO DE CONTROL

El plano de control, es la parte lógica de un elemento de red (NE). Internamente el plano de control está formado por elementos de control llamados CEs, basados en software, los cuales controlan y gestionan la operatividad de un enrutador.

Este desarrollo esta implementado en el sistema operativo Linux con la distribución Ubuntu 10, la interfaz Grafica y la interoperabilidad entre cada módulo: Módulo ForCES, Módulo de Interfaz Virtual (Etherpuppet), Módulo IP, Módulo de Enrutamiento (Quagga), Módulo de Gestión (Net-SNMP) y Módulo de Alta Disponibilidad (HA), se implementaron en Eclipse, plataforma de desarrollo basada en Java.

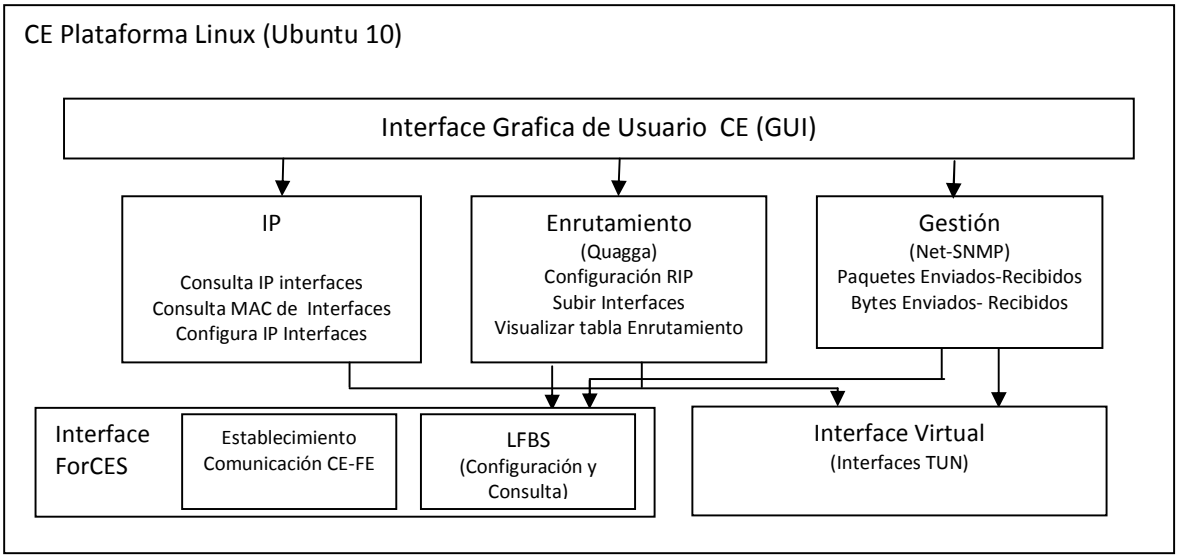


Figura 39. Estructura del Elemento de Control (Elaborado por Susan Martínez)

En la figura 39, se observa el diagrama de bloques del Elemento de Control diseñado y los diferentes módulos que hacen parte de él.

A continuación se explica cómo fue desarrollado cada módulo del prototipo.

6.1.1. MÓDULO FORCES

El módulo ForCES es el encargado de permitir la interconexión y comunicación entre el CE y FE, por medio del protocolo ForCES. Inicialmente realiza la Fase de Pre-Asociación, en la cual hay un descubrimiento de los componentes y capacidades de cada elemento, en cada FE existen las entidades conocidas como Bloques Lógicos Funcionales o LFBs, explicadas anteriormente. Cada Clase LFB tiene una serie de componentes y cada componente tiene capacidades, esa información es la que el CE aprende del FE. Una vez se realiza ese descubrimiento entra la Fase de Pos-Asociación en la cual se establece la comunicación entre el CE y FE por medio del protocolo ForCES, esto se puede ver detalladamente en [17].

Ya establecida la comunicación, el CE puede consultar al FE o FEs asociados que componentes tiene y las capacidades de cada uno. La consulta es realizada por medio del mensaje Query, donde el FE responde al CE con el mensaje Query Response.

También el CE está en la capacidad de configurar algunos componentes de los LFBs del FE y establecer topologías con la unión de varios LFBs para que cumplan una función específica, por ejemplo: enrutamiento, ARP, SNMP y Forwarding, por medio del mensaje de configuración ForCES, Config.

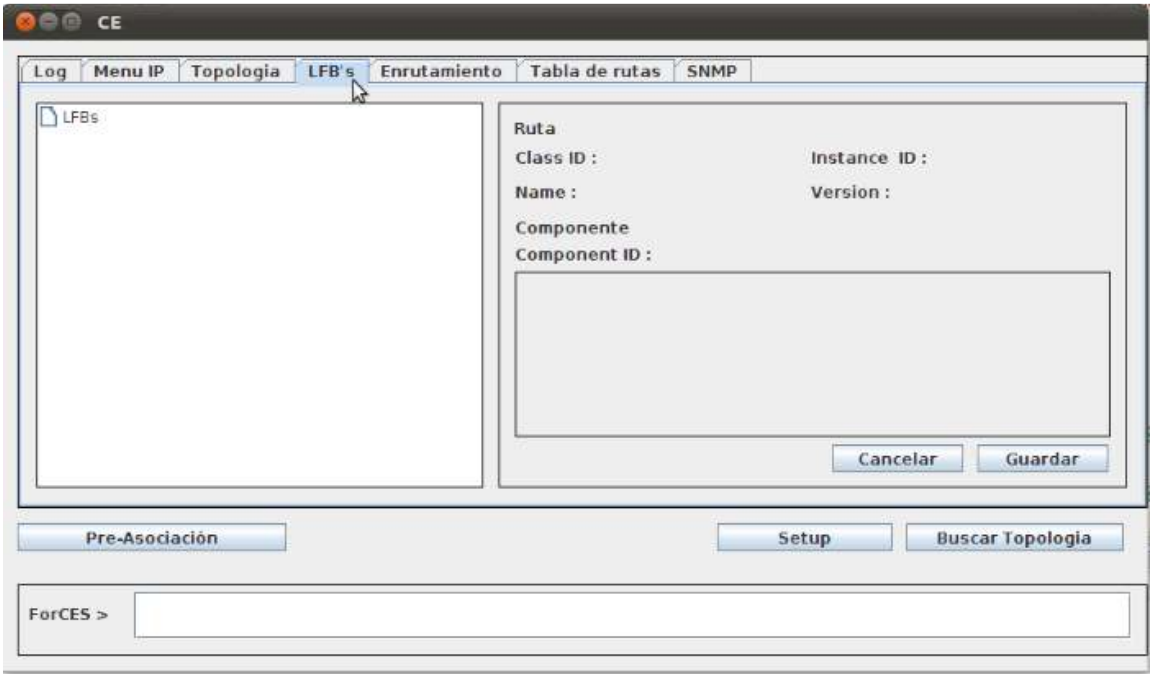


Figura 40. Interfaz Grafica (GUI) del Módulo ForCES en el CE

En la figura 40, se observa la interfaz grafica del Módulo ForCES en el CE. Al realizar la asociación el CE consulta los diferentes parámetros de clases LFBs que componen el FE, entre los cuales están ClassID, InstanceID y ComponentID.

Es bueno aclarar que este módulo utiliza la interfaz del Protocolo ForCES desarrollada por el Ingeniero Pedro Luis González y descrita en [17] y al cual se le realizo una modificación en cuanto a topologías, ya que anteriormente no era posible elegir una topología y cargarla, había que digitar el comando **config – file "nombre_archivo.txt",** el cual traía un archivo de texto con la configuración de los LFBs y se cargaba en el árbol del FE. Con la modificación hecha, se puede cargar una topología, como por ejemplo ARP, Enrutamiento, SNMP y Forwarding , que también están en un archivo de texto, guardados en el **paquete files** que se encuentran en el **workspace/src/files**, los mismos archivos de texto se deben encontrar en **workspace/bin/files** , los cuales contienen la interconexión entre LFBs y la ruta para llenar el árbol en el FE. Para listar los archivos de texto en la interfaz grafica GUI CE módulo de topología, se llama al archivo topology.propertier. Los archivos se listan como lo muestra la figura 41, para cargarlos se oprime el **botón cargar,** que no es más que comandos ***config*** que llevan la información a cada una de las estructuras creadas dentro del componente ***LFBTopology,*** que son los valores que corresponden para seleccionar los LFBs que se van a emplear, esta selección se realizar por medio de la ruta –ClassId 1 –InstanceId 1 –path 1.1.1.1 –value X, en el FE es donde se construye la topología, como se observa en la figura 42.

En la figura 41, se observa que el número de líneas de la instrucción es de seis para interconectar dos LFBs, tres para origen y tres para destino. Las instrucciones llevan información de los ID de los LFBs de origen y destino, la interconexión y el puerto. El número de líneas depende de que tan extensa sea la topología. Lo mejor es guardar cada configuración en un archivo de texto (.txt) y luego llamarla desde la interfaz grafica para mayor facilidad.

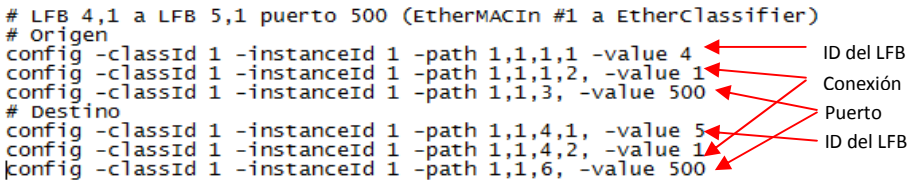


Figura 41. Ruta topología al FE

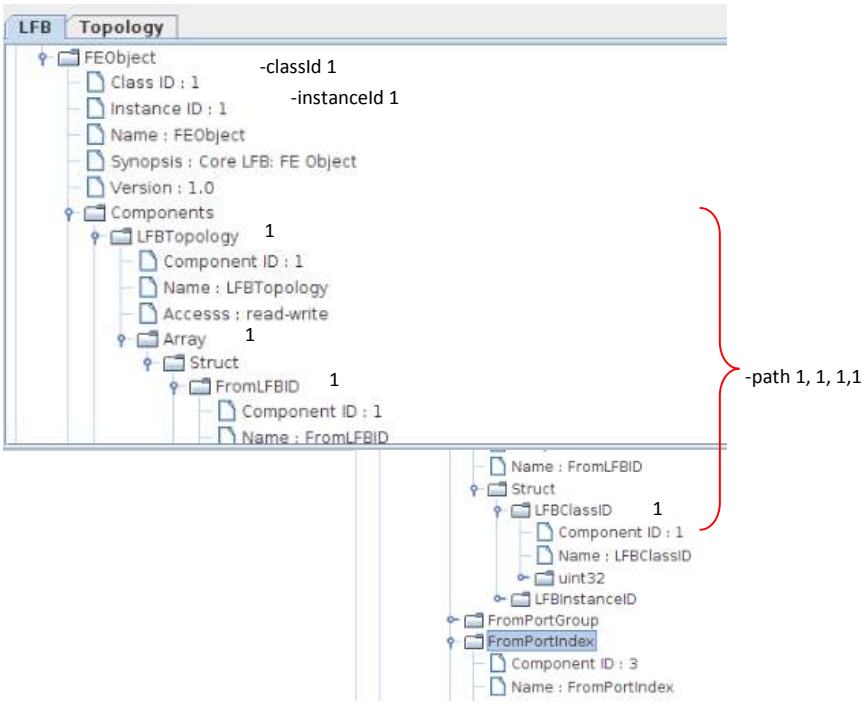


Figura 42. Ruta en el LFBTopology del FE

Para enviar el archivo de texto con los mensajes **Config** que contienen las diferentes topologías a configurar desde el CE hacia el FE, primero se selecciona la topología y luego se oprime el botón cargar, como se observa en la figura 43.

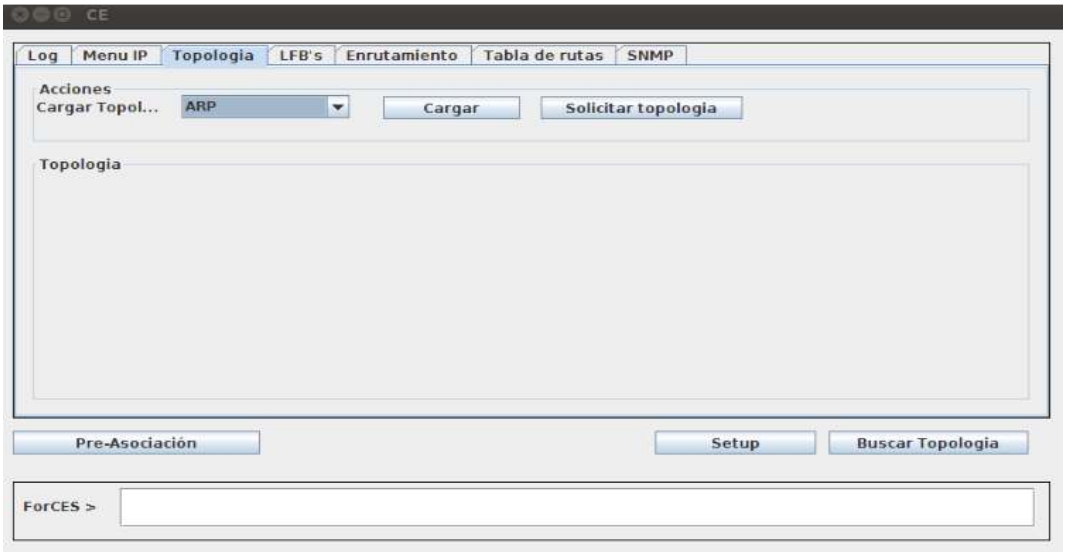


Figura 43. Interfaz Topología LFBs

Para solicitar la topología desde el CE se oprime el botón ***Solicitar topología,*** que internamente ejecuta una serie de comandos **Query (query –classId 1 –instanceId 1 –componentId X),** solicitando toda la información que se encuentra en el ***LFBTopology del FE,*** como se observa en la figura 44.

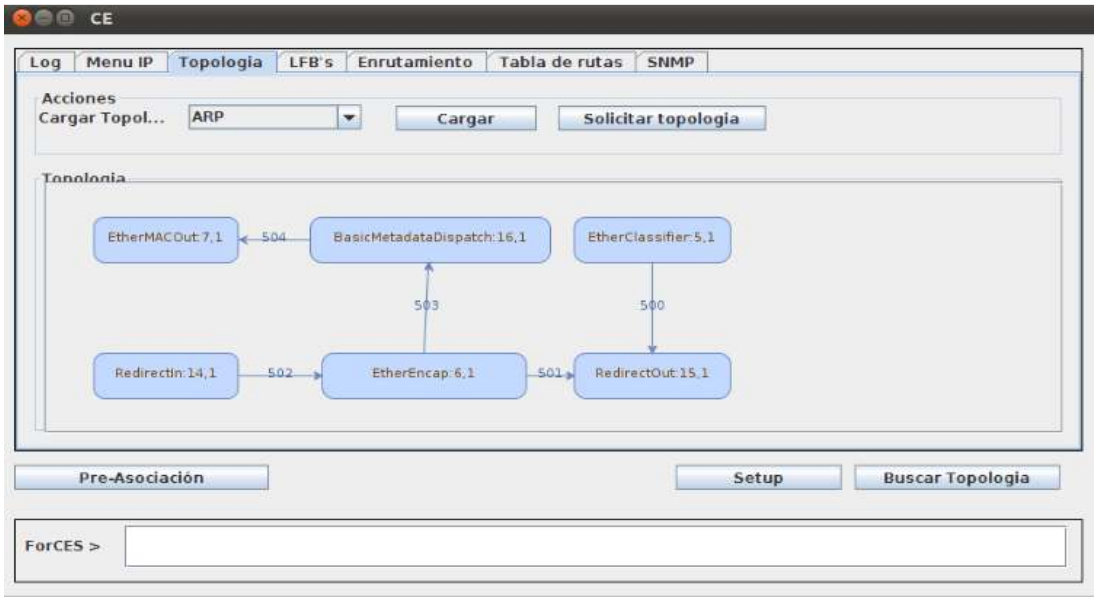


Figura 44. Visualización Topología desde el CE

```
# LFB 4,1 a LFB 5,1 puerto 500 (EtherMACIn #1 a EtherClassifier)
# Origen
config -classId 1 -instanceId 1 -path 1,1,1,1 -value 4
config -classId 1 -instanceId 1 -path 1,1,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,1,3, -value 500
# Destino
config -classId 1 -instanceId 1 -path 1,1,4,1, -value 5
config -classId 1 -instanceId 1 -path 1,1,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,1,6, -value 500

# LFB 5,1 a LFB 15,1 puerto 501 (EtherClassifier #1 a RedirectOut)
# Origen
config -classId 1 -instanceId 1 -path 1,2,1,1 -value 5
config -classId 1 -instanceId 1 -path 1,2,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,2,3, -value 501
# Destino
config -classId 1 -instanceId 1 -path 1,2,4,1, -value 15
config -classId 1 -instanceId 1 -path 1,2,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,2,6, -value 501

# LFB 14,1 a LFB 6,1 puerto 502 (RedirectIn #1 a EtherEncap)
# Origen
config -classId 1 -instanceId 1 -path 1,3,1,1 -value 14
config -classId 1 -instanceId 1 -path 1,3,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,3,3, -value 502
# Destino
config -classId 1 -instanceId 1 -path 1,3,4,1, -value 6
config -classId 1 -instanceId 1 -path 1,3,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,3,6, -value 502

# LFB 6,1 a LFB 16,1 puerto 503 (EtherEncap #1 a BasicMetaDataDispatch)
# Origen
config -classId 1 -instanceId 1 -path 1,4,1,1 -value 6
config -classId 1 -instanceId 1 -path 1,4,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,4,3, -value 503
# Destino
config -classId 1 -instanceId 1 -path 1,4,4,1, -value 16
config -classId 1 -instanceId 1 -path 1,4,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,4,6, -value 503

# LFB 16,1 a LFB 7,1 puerto 504 (BasicMetaDataDispatch #1 a EtherMACOut)
# Origen
config -classId 1 -instanceId 1 -path 1,5,1,1 -value 16
config -classId 1 -instanceId 1 -path 1,5,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,5,3, -value 504
# Destino
config -classId 1 -instanceId 1 -path 1,5,4,1, -value 7
config -classId 1 -instanceId 1 -path 1,5,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,5,6, -value 504

# LFB 6,1 a LFB 15,1 puerto 505 (EtherEncap #1 a RedirectOut)
# Origen
config -classId 1 -instanceId 1 -path 1,6,1,1 -value 6
config -classId 1 -instanceId 1 -path 1,6,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,6,3, -value 505
# Destino
config -classId 1 -instanceId 1 -path 1,6,4,1, -value 15
config -classId 1 -instanceId 1 -path 1,6,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,6,6, -value 505
```

Figura 45. Archivo de texto para la configuración de la topología ARP

```
# LFB 14,1 a LFB 8,1 puerto 500 (RedirectIn #1 a IPv4Validator)
# Origen
config -classId 1 -instanceId 1 -path 1,1,1,1 -value 14
config -classId 1 -instanceId 1 -path 1,1,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,1,3, -value 400
# Destino
config -classId 1 -instanceId 1 -path 1,1,4,1, -value 8
config -classId 1 -instanceId 1 -path 1,1,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,1,6, -value 400

# LFB 8,1 a LFB 10,1 puerto 401
# Origen
config -classId 1 -instanceId 1 -path 1,2,1,1 -value 8
config -classId 1 -instanceId 1 -path 1,2,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,2,3, -value 401
# Destino
config -classId 1 -instanceId 1 -path 1,2,4,1, -value 10
config -classId 1 -instanceId 1 -path 1,2,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,2,6, -value 401

# LFB 10,1 a LFB 12,1 puerto 402
# Origen
config -classId 1 -instanceId 1 -path 1,3,1,1 -value 10
config -classId 1 -instanceId 1 -path 1,3,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,3,3, -value 402
# Destino
config -classId 1 -instanceId 1 -path 1,3,4,1, -value 12
config -classId 1 -instanceId 1 -path 1,3,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,3,6, -value 402

# LFB 12,1 a LFB 6,1 puerto 403
# Origen
config -classId 1 -instanceId 1 -path 1,4,1,1 -value 12
config -classId 1 -instanceId 1 -path 1,4,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,4,3, -value 403
# Destino
config -classId 1 -instanceId 1 -path 1,4,4,1, -value 6
config -classId 1 -instanceId 1 -path 1,4,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,4,6, -value 403

# LFB 14,1 a LFB 6,1 puerto 404
# Origen
config -classId 1 -instanceId 1 -path 1,5,1,1 -value 14
config -classId 1 -instanceId 1 -path 1,5,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,5,3, -value 404
# Destino
config -classId 1 -instanceId 1 -path 1,5,4,1, -value 6
config -classId 1 -instanceId 1 -path 1,5,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,5,6, -value 404
```

Figura 46. Archivo de texto para la configuración de la topología Enrutamiento

```
# LFB 14,1 a LFB 8,1 puerto 200 (RedirectIn #1 a IPv4Validator)
# Origen
config -classId 1 -instanceId 1 -path 1,1,1,1 -value 14
config -classId 1 -instanceId 1 -path 1,1,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,1,3, -value 200
# Destino
config -classId 1 -instanceId 1 -path 1,1,4,1, -value 8
config -classId 1 -instanceId 1 -path 1,1,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,1,6, -value 200

# LFB 8,1 a LFB 10,1 puerto 201
# Origen
config -classId 1 -instanceId 1 -path 1,2,1,1 -value 8
config -classId 1 -instanceId 1 -path 1,2,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,2,3, -value 201
# Destino
config -classId 1 -instanceId 1 -path 1,2,4,1, -value 10
config -classId 1 -instanceId 1 -path 1,2,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,2,6, -value 201

# LFB 10,1 a LFB 12,1 puerto 202
# Origen
config -classId 1 -instanceId 1 -path 1,3,1,1 -value 10
config -classId 1 -instanceId 1 -path 1,3,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,3,3, -value 202
# Destino
config -classId 1 -instanceId 1 -path 1,3,4,1, -value 12
config -classId 1 -instanceId 1 -path 1,3,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,3,6, -value 202

# LFB 10,1 a LFB 15,1 puerto 203
# Origen
config -classId 1 -instanceId 1 -path 1,4,1,1 -value 10
config -classId 1 -instanceId 1 -path 1,4,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,4,3, -value 203
# Destino
config -classId 1 -instanceId 1 -path 1,4,4,1, -value 15
config -classId 1 -instanceId 1 -path 1,4,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,4,6, -value 203
```

Figura 47. Archivo de texto para la configuración de la topología SNMP

```
# LFB 3,1 a LFB 4,1 puerto 600 (Ether PHY co #1 a Ether MACIn)
# Origen
config -classId 1 -instanceId 1 -path 1,1,1,1 -value 3
config -classId 1 -instanceId 1 -path 1,1,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,1,3, -value 600
# Destino
config -classId 1 -instanceId 1 -path 1,1,4,1, -value 4
config -classId 1 -instanceId 1 -path 1,1,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,1,6, -value 600

# LFB 4,1 a LFB 5,1 puerto 601
# Origen
config -classId 1 -instanceId 1 -path 1,2,1,1 -value 4
config -classId 1 -instanceId 1 -path 1,2,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,2,3, -value 601
# Destino
config -classId 1 -instanceId 1 -path 1,2,4,1, -value 5
config -classId 1 -instanceId 1 -path 1,2,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,2,6, -value 601

# LFB 5,1 a LFB 8,1 puerto 602
# Origen
config -classId 1 -instanceId 1 -path 1,3,1,1 -value 5
config -classId 1 -instanceId 1 -path 1,3,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,3,3, -value 602
# Destino
config -classId 1 -instanceId 1 -path 1,3,4,1, -value 8
config -classId 1 -instanceId 1 -path 1,3,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,3,6, -value 602

# LFB 8,1 a LFB 10,1 puerto 603
# Origen
config -classId 1 -instanceId 1 -path 1,4,1,1 -value 8
config -classId 1 -instanceId 1 -path 1,4,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,4,3, -value 603
# Destino
config -classId 1 -instanceId 1 -path 1,4,4,1, -value 10
config -classId 1 -instanceId 1 -path 1,4,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,4,6, -value 603

# LFB 10,1 a LFB 12,1 puerto 604
# Origen
config -classId 1 -instanceId 1 -path 1,5,1,1 -value 10
config -classId 1 -instanceId 1 -path 1,5,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,5,3, -value 604
# Destino
config -classId 1 -instanceId 1 -path 1,5,4,1, -value 12
config -classId 1 -instanceId 1 -path 1,5,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,5,6, -value 604

# LFB 12,1 a LFB 6,1 puerto 605
# Origen
config -classId 1 -instanceId 1 -path 1,6,1,1 -value 12
config -classId 1 -instanceId 1 -path 1,6,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,6,3, -value 605
# Destino
config -classId 1 -instanceId 1 -path 1,6,4,1, -value 6
config -classId 1 -instanceId 1 -path 1,6,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,6,6, -value 605

# LFB 6,1 a LFB 16,1 puerto 606
# Origen
config -classId 1 -instanceId 1 -path 1,7,1,1 -value 6
config -classId 1 -instanceId 1 -path 1,7,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,7,3, -value 606
# Destino
config -classId 1 -instanceId 1 -path 1,7,4,1, -value 16
config -classId 1 -instanceId 1 -path 1,7,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,7,6, -value 606

# LFB 16,1 a LFB 7,1 puerto 607
# Origen
config -classId 1 -instanceId 1 -path 1,8,1,1 -value 16
config -classId 1 -instanceId 1 -path 1,8,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,8,3, -value 607
# Destino
config -classId 1 -instanceId 1 -path 1,8,4,1, -value 7
config -classId 1 -instanceId 1 -path 1,8,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,8,6, -value 607

# LFB 7,1 a LFB 3,1 puerto 608
# Origen
config -classId 1 -instanceId 1 -path 1,9,1,1 -value 7
config -classId 1 -instanceId 1 -path 1,9,1,2, -value 1
config -classId 1 -instanceId 1 -path 1,9,3, -value 608
# Destino
config -classId 1 -instanceId 1 -path 1,9,4,1, -value 3
config -classId 1 -instanceId 1 -path 1,9,4,2, -value 1
config -classId 1 -instanceId 1 -path 1,9,6, -value 608
```

Figura 48. Archivo de texto para la configuración de la topología Forwarding

6.1.2. MÓDULO IP

Realiza una comunicación entre el CE y FE, la cual permite conocer las interfaces físicas, direcciones lógicas y direcciones físicas que tiene el FE y consultadas por el CE, a su vez, el CE está en la capacidad de tener gestión sobre esas interfaces como si fueran locales, aunque realmente son remotas.

Para establecer la comunicación en este módulo, se crea un protocolo llamado ***Background*** que emplea el mensaje con el mismo nombre, también se implementa una interfaz en java que se llama ***serializable,*** la cual emplea métodos propios de java para que la clase pueda ser transportable y sea posible enviar los mensajes por la interfaz de red y los datos llegue intactos.

```
            break;
        case BackGroundTypes.ACTIVE_VIRTUAL_INET_RESPONSE:
            if (message.getState() == BackGroundTypes.RESPONSE_OK)
                {
                    connectVirtuals();
                    JOptionPane.showMessageDialog(this,"Interfaces activas con exito");
                    jbActiveVirtuals.setEnabled(false);
                    jbDesactiveVirtuals.setEnabled(true);
                }
            else
                JOptionPane.showMessageDialog(this,
                        "No se han podido activar la interfaces virtuales");
            break;
        case BackGroundTypes.DESACTIVE_VIRTUAL_INET_RESPONSE:
                    DefaultTableModel modelt = (DefaultTableModel) jtInterfaces.getModel();
                    for(int i=modelt.getRowCount()-1;i>=0;i--)
                            {
                                modelt.setValueAt(false,i,7);
                            }
                    jbActiveVirtuals.setEnabled(true);
                    jbDesactiveVirtuals.setEnabled(false);
                    JOptionPane.showMessageDialog(this,"Interfaces desactivadas con exito");
```

Figura 49. Mensaje Background

El mensaje está compuesto por:

- ***Texto:*** Que tiene información de chequeo
- ***Tipo:*** Define el tipo de mensaje
- ***HashMap:*** Contiene toda la información, es una colección clave/valor, lo que significa que es un arreglo con dos valores, en este caso nombre/valor, donde nombre son las interfaces (Eth0, Eth1, Eth2) y el valor es toda la información de cada interfaz.
- ***Estado:*** Que indica si se completa toda la operación o no.

El *HashMap* es el que extre la información de las interfaces desde el kernel de Linux, para este proceso se crea una clase llamada ***interface,*** que es una clase tipo estructura, la cual captura todos los datos de cada interfaz como el nombre, MTU, dirección de red, dirección de hardware, dirección de broadcast y mascara de red.

Para cada parte del mensaje se crea una clase. En el caso del Tipo, se crea la clase BackgroundTypes, la cual indica:

- 10: Activación de la interfaz virtual
- 11: Respuesta al mensaje de activación interface virtual
- 12: Desactivar la interfaz virtual
- 13: Respuesta al a desactivación de la interfaz virtual.

La interfaz grafica del módulo IP, se observa en la figura 52. En este módulo se realizan dos operaciones:

- El CE consulta las interfaces físicas del FE, información de direcciones IP si están configuradas, mascara de red, dirección física y nombre de la interfaz.

- El CE configura las direcciones IP de las interfaces.

Las dos columnas finales de la tabla, puerto virtual y virtualizar hacen parte del módulo de virtualización que se explicara en el siguiente numeral, se encuentra en el menú IP porque está muy relacionado con cada interfaz del FE capturada en este módulo.

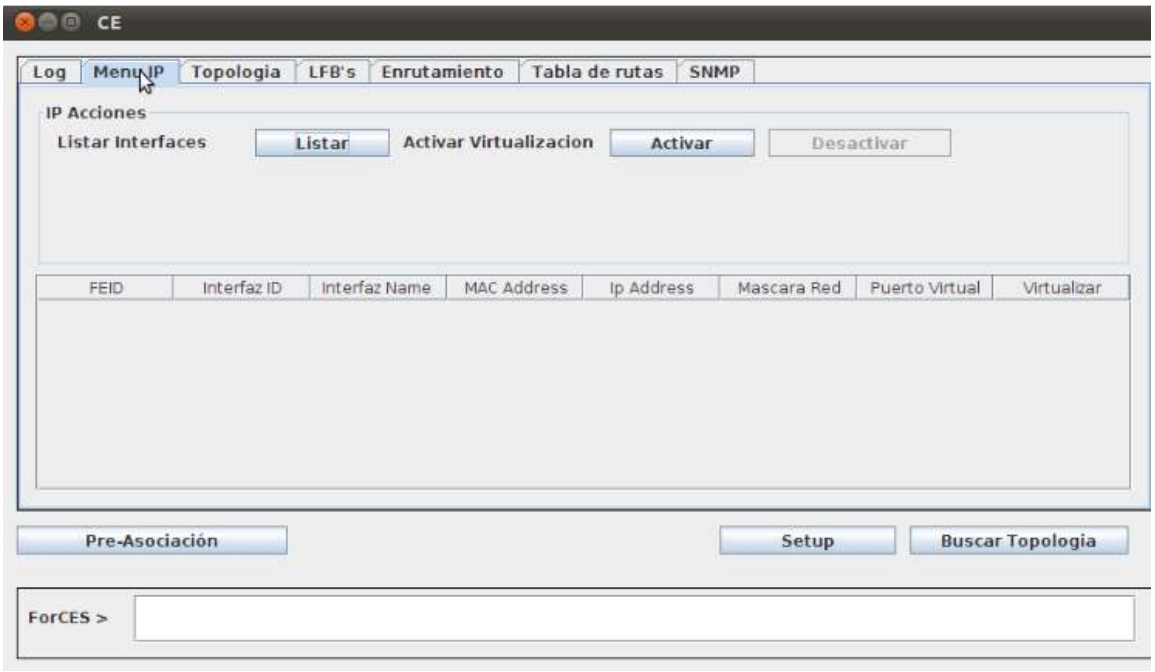


Figura 50. Interfaz Grafica del Módulo IP

Al oprimir el botón Listar, se visualizan las interfaces que hacen parte del FE, como se observa en la figura 53, de esta forma el CE configura con una dirección IP y una máscara de subred cada una de ellas (en este caso las interfaces Eth0 y Eth1), la interfaz Eth2 no se configura ya que por esta interfaz esta la conexión ForCES entre el CE y FE. Posteriormente configuradas se realiza la virtualización de las mismas.

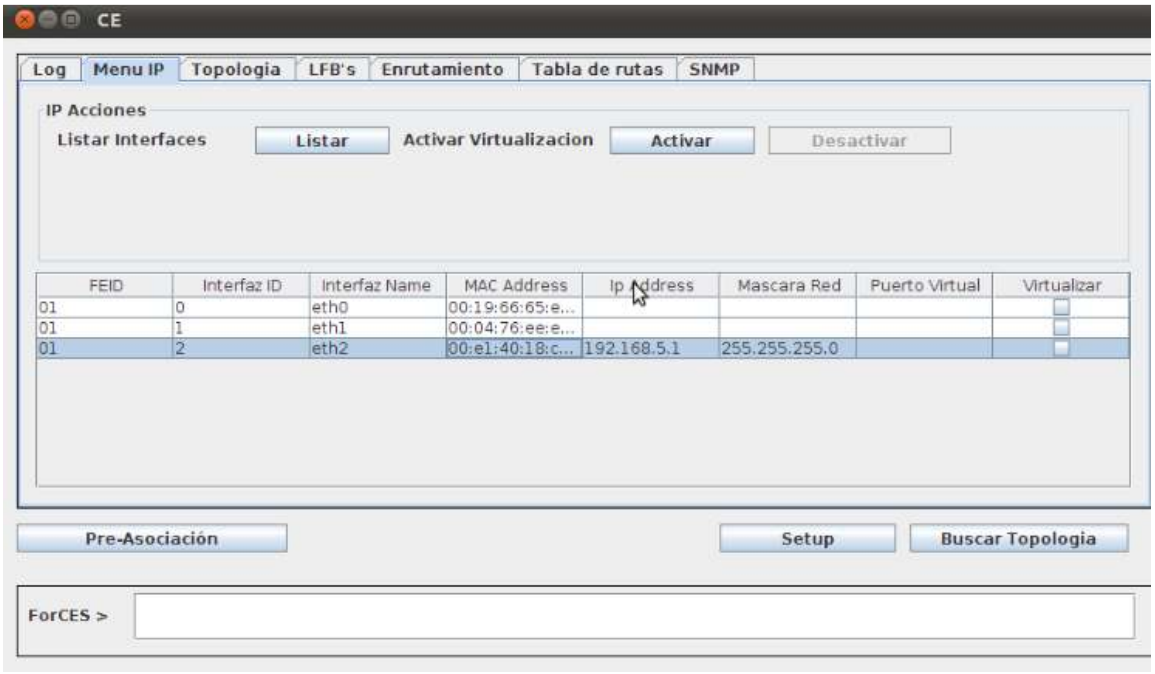


Figura 51. Listar Interfaces Físicas

6.1.3. MÓDULO INTERFAZ VIRTUAL (MIV)

El módulo de Interfaz Virtual emplea la aplicación para Linux llamada Etherpuppet, que crea interfaces virtuales (TUN/TAP) desde una maquina a otra por medio de la interfaz Ethernet a través de puertos TCP. Esta aplicación permite que todo lo que se vea por las interfaces reales se vea también en las interfaces virtuales y lo que se envié por las interfaces virtuales, sea enviado por las interfaces reales, permitiendo reflejar las interfaces que se encuentran en el FE como si fueran interfaces locales del CE y de esta forma, el CE puede configurar las direcciones IP de las interfaces, habilitarlas/deshabilitarlas y configurar también el protocolo de enrutamiento RIP y luego transportar esa información por medio de túneles virtuales hacia las interfaces físicas en el FE y luego hacia el exterior.

El código fuente del aplicativo Etherpuppet se puede descargar de la página ***www.secdev.org/projects/etherpuppet/***, el archivo se llama ***etherpuppet.c v0.3***.

El archivo se compila en el CE y en el FE, el CE queda como cliente y el FE queda como servidor. Para compilar se digita la siguiente instrucción,

root@susanpc:/home/susan# gcc etherpuppet.c –o etherpuppet

Después de compilado el archivo, en el FE, que es el servidor de la interfaz TUN/TAP o interfaz virtual, se digita la instrucción:

root@susanpc:/home/susan# ./etherpuppet/ -s 4444 –i Eth0
root@susanpc:/home/susan# ./etherpuppet/ -s 4445 –i Eth0

donde:

-s: Escuchar sobre un puerto TCP, en este caso se escucha sobre el puerto 4444
-i: Interfaz a virtualizar, como el FE tiene tres interfaces físicas (Eth0, Eth1 y Eth2), la interfaz Eth2 es la interface conectada al CE para la comunicación ForCES, no se usaría para virtualizar. Solo quedan la Eth0 y la Eth1.

Las instrucciones para el CE para configurar las interfaces virtuales son:

root@susanpc:/home/susan# ./etherpuppet/ -m <IP>:4444 –I Tun3
root@susanpc:/home/susan# ./etherpuppet/ -m <IP>:4445 –I Tun4

donde:

-m: Modo maestro
-c <IP>:<Port> : Conectar <IP>:<Port>
-I <ifname>: Elige el nombre de la interfaz virtual, en este caso Tun

En la figura 52, se observa la configuración y la activación de las interfaces virtuales. Cuando el FE recibe el mensaje de activar virtualización, la clase FEConsole.java ejecuta el comando de interfaz virtual servidor.

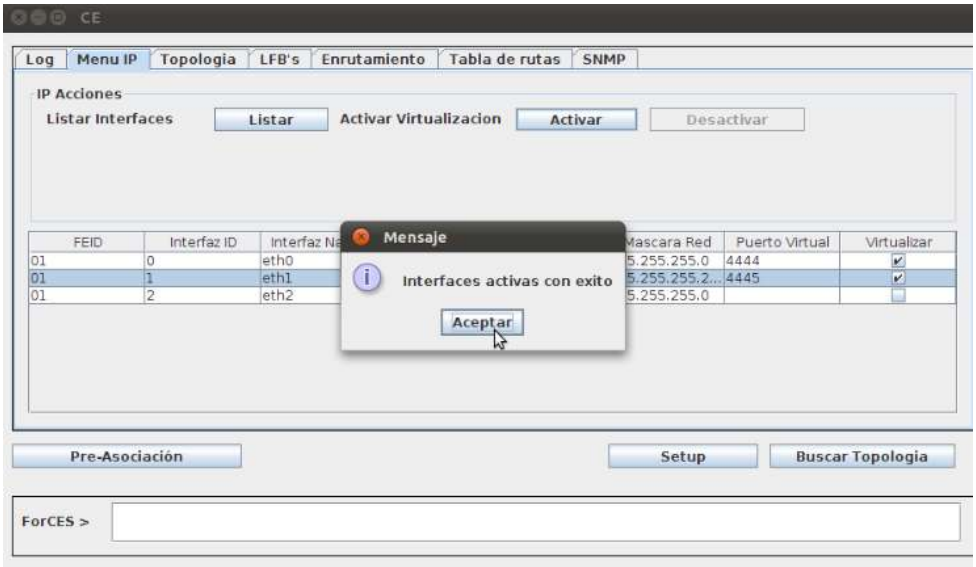


Figura 52. Virtualización de las interfaces

En la figura 53, se observa en el terminal de Linux del CE, las interfaces virtualizadas, en este caso la TUN3 es la interfaz virtual de la interfaz real Eth0 del FE y la TUN4 la interfaz virtual de la interfaz real Eth1 del FE.

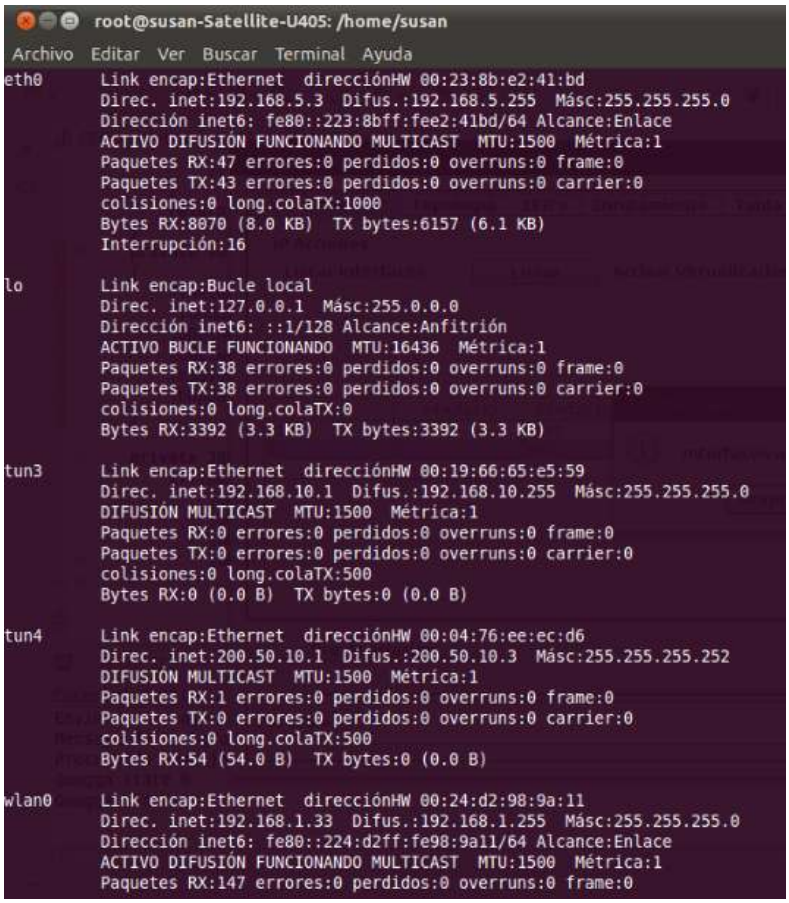


Figura 53. Interfaces virtualizadas vistas desde el CE

6.1.4. MÓDULO DE ENRUTAMIENTO

Este módulo emplea la aplicación ***quagga***, que es un paquete de software de enrutamiento para redes TCP/IP que provee implementaciones de RIPv1, RIPv2, OSPFv2, OSPFv3 y BGP para plataformas Unix, entre ellos Linux. Quagga es una derivación de la aplicación Zebra GNU. El core de quagga es principalmente el demonio Zebra, que actua como una capa de abstracción al kernel de Linux y presenta el API Zserv. Existen clientes Zserv, los cuales implementan un protocolo de enrutamiento y actualizan tablas de enrutamiento para el demonio Zebra. Estos Zserv son:

- Ospfd (Implementa OSFPv2)
- Ripd (Implementa Ripv1 y Ripv2)
- Ospf6d (Implenta OSPFv3/IPv6)

- Ripngd (Implementa Ripng/IPv6)
- Bgpd (Implementa BGPv4)

La configuración de los demonios de Quagga es por medio de un CLI así como se configura cualquier otro enrutador. Tomado página oficial de quagga www.nongnu.org/quagga/.
Para la instalación de Quagga:

- Descargar de la página oficial el archivo quagga-0.99.17.tar.gz.
- **root@susanpc:/home/susan#** apt –get install quagga
- Se ponen los ficheros de configuración de quagga en su sitio con el siguiente comando.
  root@susanpc:/home/susan# cd /usr/share/doc/quagga/examples/
  root@susanpc:/home/susan/examples# cp * /etc/quagga/
- Se renombran los ejemplos de configuración, para usarlos y activar Quagga con la configuración por defecto**.**

  root@susanpc:/home/susan# cd /etc/quagga/
  root@susanpc:/home/susan/quagga# cp zebra.conf.sample zebra.conf
  root@susanpc:/home/susan/quagga# cp ripd.conf.sample ripd.conf
- Se activo Zebra y el protocolo RIP, si se quiere activar otro protocolo, el proceso es similar

  vim daemons

  Y se editan las líneas:

  zebra=yes

  ripd=yes
- Por último se reinicia Quagaa

  root@susanpc:/home/susan# /etc/init.d/quagga restart

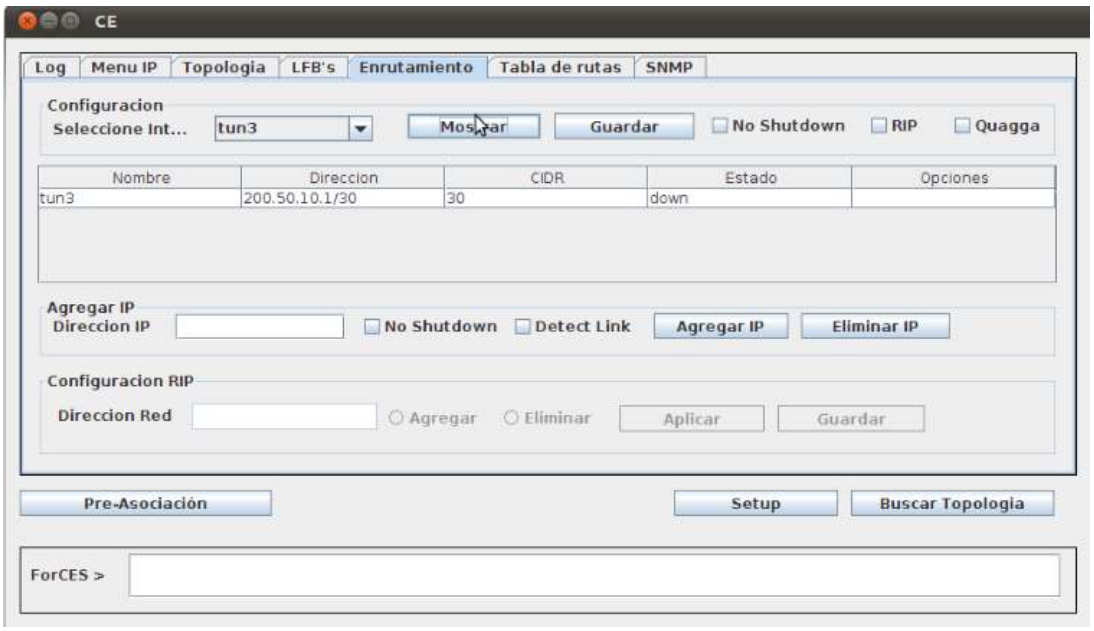


Figura 54. Módulo de Enrutamiento

En la figura 54, se observa la interfaz grafica del Módulo de Enrutamiento, para este prototipo se configura el protocolo RIPv2, aunque como se dijo anteriormente **quagga** soporta OSPFv2,v3, BGP que se podrían configurar en una futura versión de este prototipo.

En la sección de configuración (parte superior izquierda), se pueden visualizar las interfaces TUN (que son las interfaces virtuales). Primero se selecciona la interfaz y luego se oprime el botón Mostrar, el cual en la clase Route.java se ejecuta el comando show interfaz, el cual visualiza en la tabla el nombre de la interfaz, la dirección IP, la máscara de subred y el estado (up/down). Los componentes de la clase Route.java se observan en la figura 55.

```
package util;

import java.util.HashMap;

public class Route {

    public static int TYPE=1;
    public static int IP_ADDRESS=2;
    public static int CIDR=3;
    public static int INTERFACE=4;

    public static String viaExp = "(?:.*\\[(\\d+)/(\\d+)\\].*via (\\d{0,3}\\.\\d{0,3}\\.\\d{0,3}\\.\\d{0,3}).*$)";
    public static String exp ="(?:([A-Z])>\\* (\\d{0,3}\\.\\d{0,3}\\.\\d{0,3}\\.\\d{0,3})/(\\d{0,2}).*, ([a-zA-Z]+\\d).*$)";

    private char type;
    private String ipaddress;
    private int cidr;
    private String iface;
    private String netmask;
    private String via;
    private String metric;
    private String beforeMetric;
    private HashMap<Character, String> types;


    public String getMetric() {
        return metric;
    }

    public void setMetric(String metric) {
        this.metric = metric;
    }
```

Figura 55. Clase Route.java

Para activar la sección Configuración RIP (parte inferior izquierda), se activa RIP oprimiendo la casilla, en esa sección se configuran las redes en RIPv2. Por consola y en un router convencional se activarían con las siguientes líneas de comandos:

Router(config)#router rip
Router(config)#version 2
Router(config-route)#network <dirección IP de red/CIDR>

En la interfaz grafica el botón RIP, activa RIPv2 (#router rip), el botón Agregar IP, al oprimirse agrega las redes (#network <dirección IP de red/CIDR>), también se encuentra la opción de Eliminar redes, la cual se ejecuta al oprimir el botón Eliminar IP (ejecuta el comando #no network <dirección IP de red/CIDR>).

6.1.4.1. TABLA DE ENRUTAMIENTO

La tabla de enrutamiento es tomada de Quagga, con la función de java ***Runtime rt***, que es una función propia de java (ejecuta un programa del sistema operativo). Toda la información de las rutas que se encuentran en Quagga, las captura la clase Route.java y esa información se muestra en la tabla de rutas de la interfaz grafica del módulo de enrutamiento.

```
private void addNetwork(String network)
{

    Runtime rt = Runtime.getRuntime();
    try {
        Process pr = rt.exec(new String[]{"vtysh", "-d","ripd","-c","configure terminal","-c","router
        System.out.println(network);
        BufferedReader input = new BufferedReader(new InputStreamReader(pr.getInputStream()));
```

Figura 56. Clase Route.java con el comando Runtime rt

La tabla de enrutamiento que se observa en la figura 57, contiene el tipo de conexión K, C y R, en la siguiente columna, se encuentra la descripción de cada tipo de conexión. En la tercera columna esta la dirección de red a la que pertenece cada interfaz, en la columna Métrica, indica en este caso por ser RIPv2, la distancia administrativa/número de saltos (120/1). En la columna Próximo Salto, indican la dirección IP del próximo salto para llegar a las redes que se aprendieron por RIP, y en la columna Interfaz, indica la interfaz de ese próximo salto.

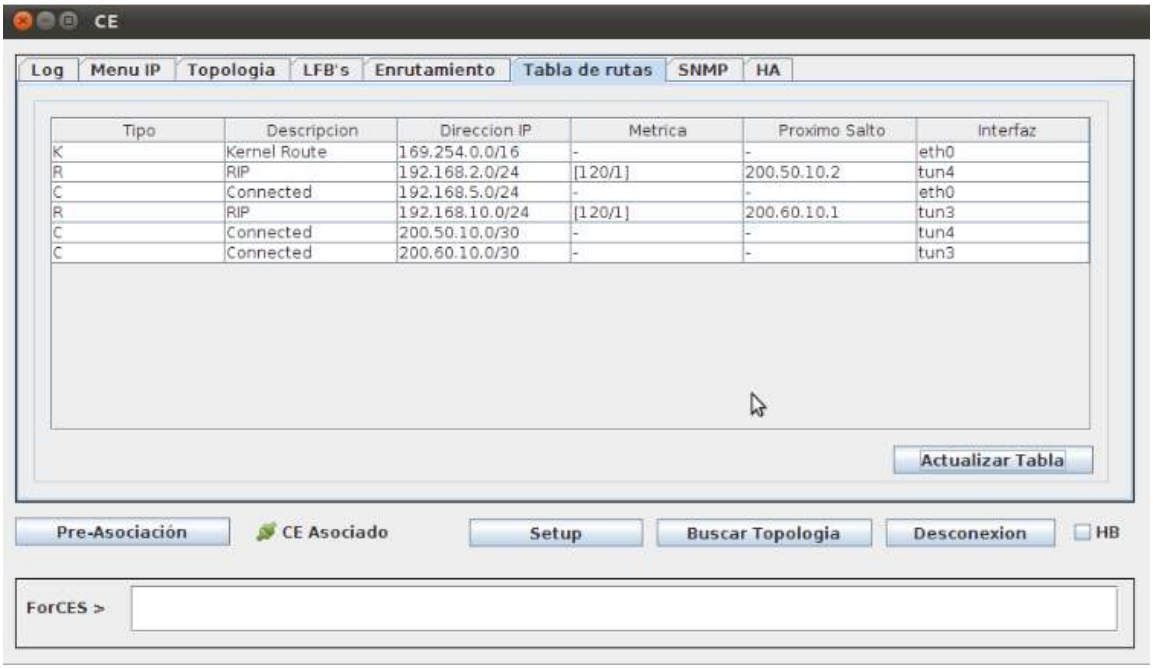


Figura 57. Tabla de Enrutamiento

6.1.5. MÓDULO SNMP

SNMP es el Protocolo Simple de Administración de Red, el cual permite a los administradores de red, administrar los dispositivos que hacen parte de la red, conocer el estado de las interfaces, paquetes y bytes transmitidos y recibidos, entre otros y diagnosticar algún problema en la red.

Elementos básicos de SNMP:

- ***Dispositivos administrados:*** Son elementos de Red (NE), como routers, servidores, concentradores, PC con un agente SNMP, entre otros. Tienen la función de capturar y almacenar la información, la cual es enviada a un sistema de administración de red (NMS) cuando es solicitada.
- ***Agente:*** Es una aplicación de administración de red que se encuentra en un dispositivo administrado. El agente tiene conocimiento local de la información administrada (paquetes transmitidos, paquetes recibidos, bytes transmitidos, byte recibidos, estado de las interfaces, entre otros).
- ***Sistema Administrador de Red (NMS):*** Es una interfaz que supervisa y controla los dispositivos administrados.

6.1.5.1. MIB (Bases de Información de Gestión)

Es un conjunto de información o repositorios de datos donde se almacena información de gestión organizada jerárquicamente en una estructura de árbol y para acceder a las MIBs se requiere un protocolo de gestión de red. También se puede decir que las MIB contienen objetos que describen

parámetros de los dispositivos, cada elemento del árbol se identifica con un OID (Identificador del Objeto) número o texto, por ejemplo .1.3.6.1.2.1.1.1 (número) o iso.org. dod. Internet. mgmt.mib-2. System.sysDescr (texto).

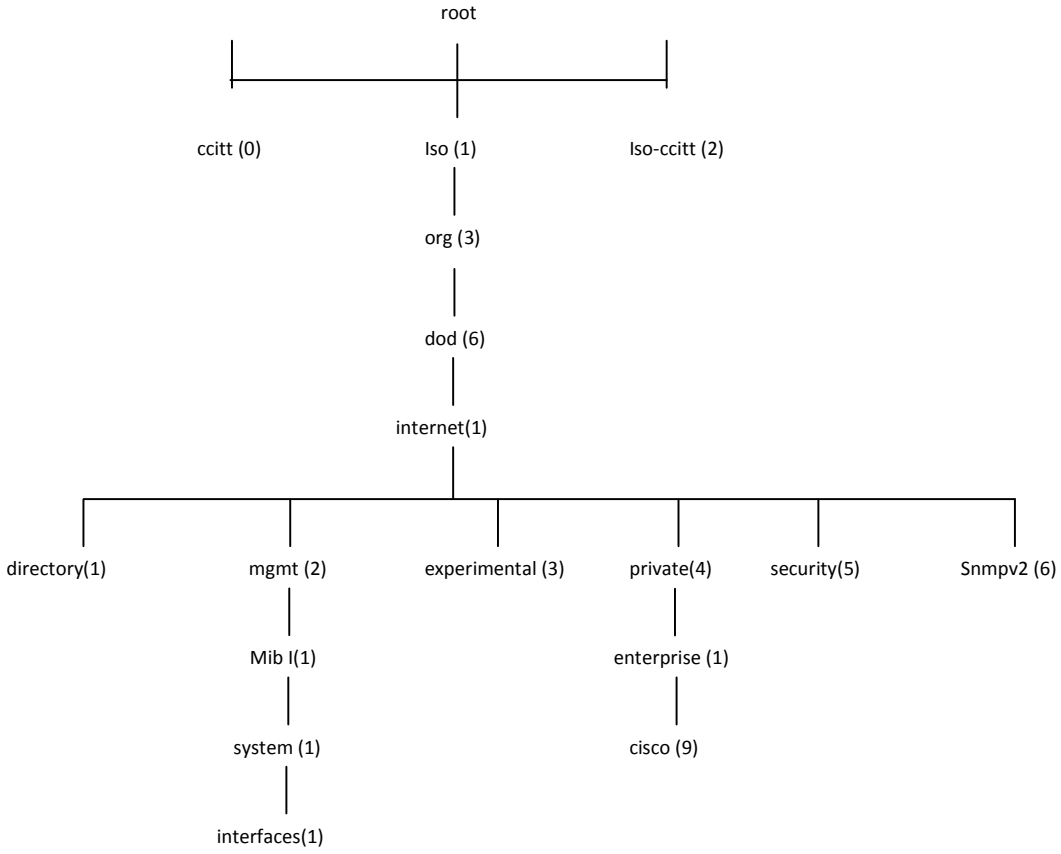


Figura 58. Árbol MIB

Para el desarrollo de este módulo se empleo la aplicación Net-SNMP, que es un conjunto de aplicaciones que se usa para implementar el protocolo SNMP.

Los comandos empleados son:

- ***Snmpget, snmpgetnext, snmpwalk, snmptable:*** Los cuales toman información de los dispositivos administrados.
- ***Snmpset:*** Manipula información sobre configuración de dispositivos.
- ***Snmptranslate:*** Traduce OIDs numéricos y textuales de los objetos de la MIB y visualiza el contenido y estructura de la MIB.

La instalación de Net-SNMP se realiza desde los repositorios de Linux con el siguiente comando:

root@susanpc:/home/susan# apt-get install snmpd

Para que se instale correctamente es necesario estar conectado a Internet.

Después de instalado, hay que habilitarle el acceso a los clientes y habilitarle que escuche en todas las interfaces, como se observa en la figura 59. El comando para habilitar escuchar por el puerto 161 es el siguiente:

root@susanpc:/home/susan# nano /etc/snmp/snmpd.conf

agentAddress udp:161

Permisos para los que quieran ser clientes

Recommunity public localhost <dirección IP>

Y por último se reinicia

root@susanpc:/home/susan# /etc/init.d/snmpd restart

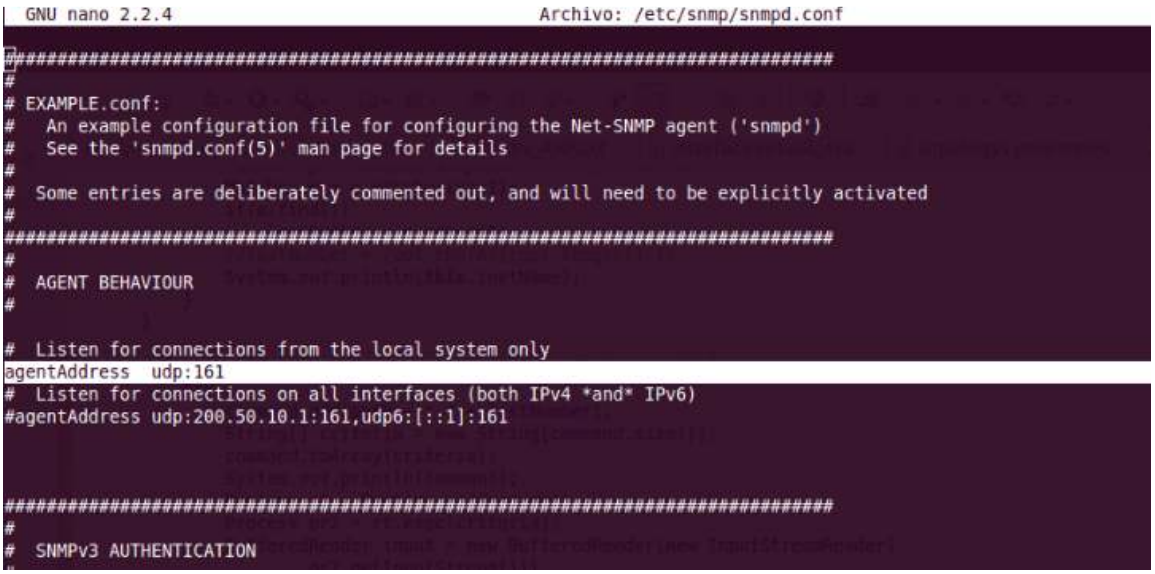


Figura 59. Consola Linux SNMP

Para permitir el acceso a una red para ser gestionada por SNMP, en el archivo de configuración de SNMP en el CE, hay que adicionar la dirección con el siguiente comando:

root@susanpc:/home/susan# nano /etc/snmp/snmpd.conf

Recommunity public <dirección IP>

Y por último se reinicia nuevamente. Las direcciones quedan configuradas como se muestra en la figura 60.

root@susanpc:/home/susan# /etc/init.d/snmpd restart

Figura 60. Permiso SNMP

En la figura 61, se observa la interfaz grafica del módulo SNMP, donde se visualiza una tabla con la información de las interfaces, ruta del árbol MIB, bytes enviados, bytes recibido, paquetes envidos y paquetes recibidos.

Para obtener la información SNMP en la tabla de la interfaz, se emplearon en la clase SNMPInterface.java los comandos:

Snmpwalk –v2c localhost –c public 1.3.6.1.2.1.31.1.1.1.1 (lista las interfaces)
Snmpget –v2c localhost –c public 1.3.6.1.2.1.31.1.1.1.1.1 (lista el nombre de la interfaz 1)
Snmpget –v2c localhost –c public 1.3.6.1.2.1.31.1.1.1.2 (lista el nombre de la interfaz 2)
Snmpget –v2c localhost –c public 1.3.6.1.2.1.31.1.1.6.1 (bytes recibidos por la interfaz 1)
Snmpget –v2c localhost –c public 1.3.6.1.2.1.31.1.1.10.1 (bytes transmitidos por la interfaz 1)
Snmpget –v2c localhost –c public 1.3.6.1.2.1.31.1.1.7.1 (paquetes recibidos por la interfaz 1)
Snmpget –v2c localhost –c public 1.3.6.1.2.1.31.1.1.11.1 (paquetes enviados por la interfaz 1)

-c <nombre de la comunidad>
-v <versión 1, 2c, 3>
-m <lista módulos MIB a incluir>
-t <tiempo de espera>

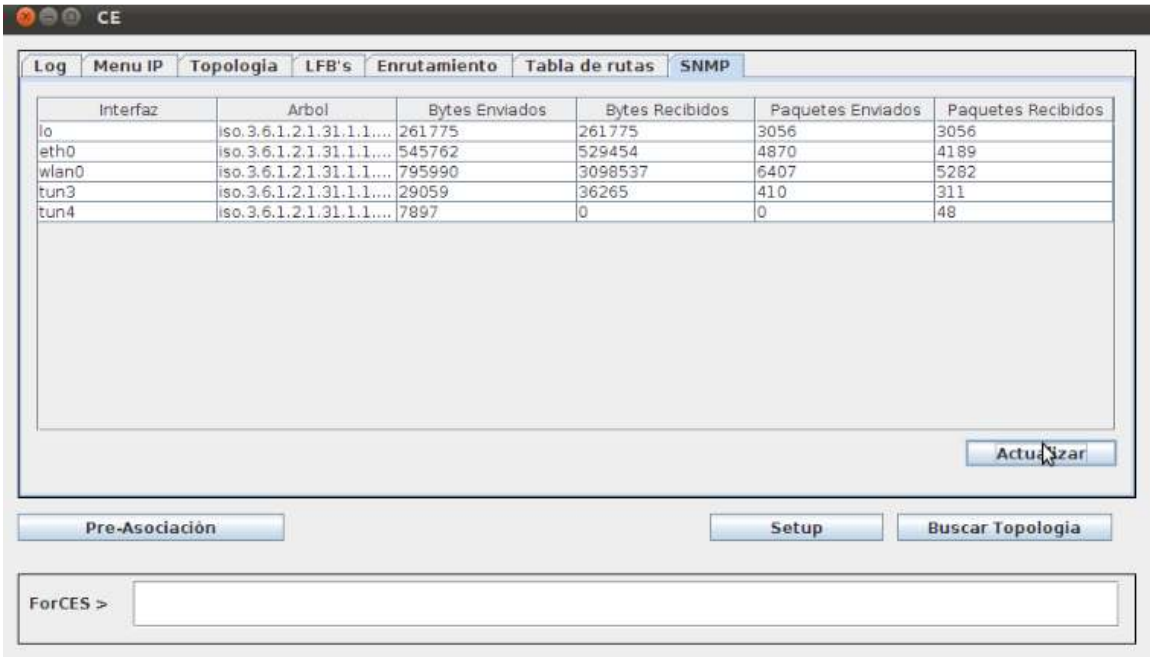
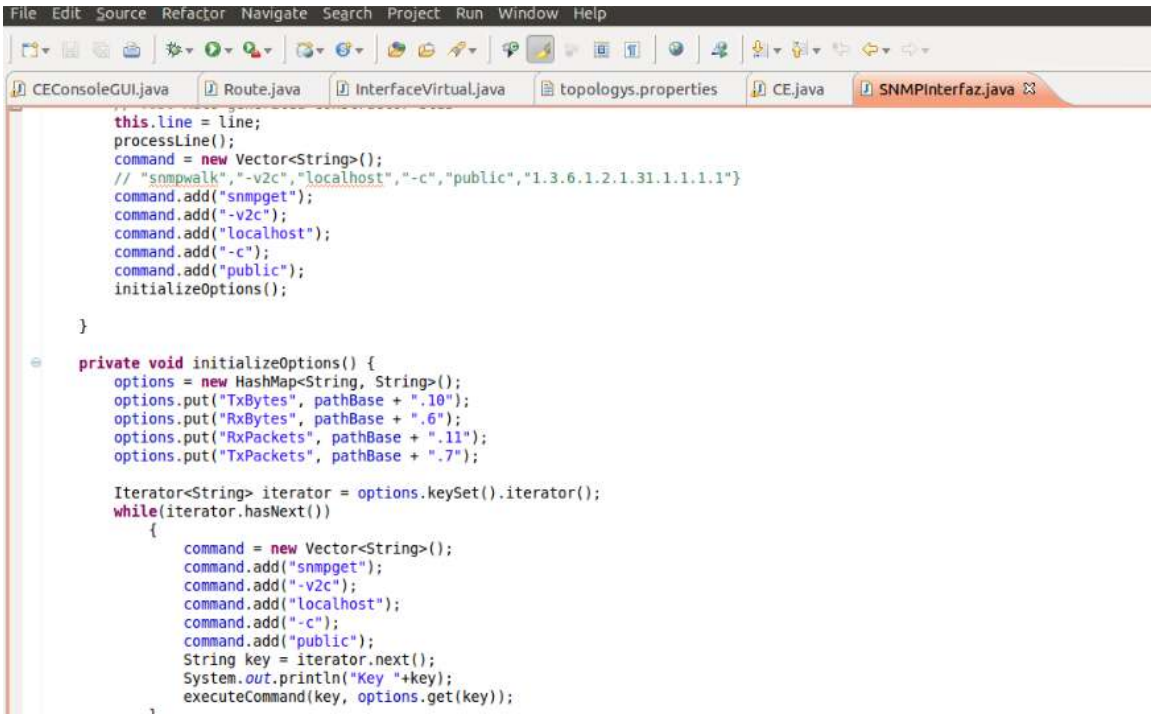


Figura 61. Interfaz Módulo SNMP

La figura 61, muestra los parámetros capturados por la interfaz grafica SNMP, es decir los parámetros de la clase SNMPInterfaz.java




```
        this.line = line;
        processLine();
        command = new Vector<String>();
        // "snmpwalk","-v2c","localhost","-c","public","1.3.6.1.2.1.31.1.1.1.1"}
        command.add("snmpget");
        command.add("-v2c");
        command.add("localhost");
        command.add("-c");
        command.add("public");
        initializeOptions();

    }

    private void initializeOptions() {
        options = new HashMap<String, String>();
        options.put("TxBytes", pathBase + ".10");
        options.put("RxBytes", pathBase + ".6");
        options.put("RxPackets", pathBase + ".11");
        options.put("TxPackets", pathBase + ".7");

        Iterator<String> iterator = options.keySet().iterator();
        while(iterator.hasNext())
            {
                command = new Vector<String>();
                command.add("snmpget");
                command.add("-v2c");
                command.add("localhost");
                command.add("-c");
                command.add("public");
                String key = iterator.next();
                System.out.println("Key "+key);
                executeCommand(key, options.get(key));
```

CEConsoleGUI.java | Route.java | InterfaceVirtual.java | topologys.properties | CE.java | SNMPInterfaz.java

```
package util;

import java.io.BufferedReader;

public class SNMPInterfaz implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    private String pathBase = "1.3.6.1.2.1.31.1.1.1";
    private HashMap<String, String> options;
    private Vector<String> command;
    private final String patterLine = "(?:^(iso\\.[\\d\\.]+)\\s=\\s(?:.+)\\:\\s(.+))";
    private String root;
    private String inetName;
    private String txBytes;
    private String rxBytes;
    private String rxPackets;
    private String txPackets;
    private String line;
    private String lastNumber;

    public SNMPInterfaz(String line) {
        // TODO Auto-generated constructor stub
        this.line = line;
        processLine();
        command = new Vector<String>();
        // "snmpwalk","-v2c","localhost","-c","public","1.3.6.1.2.1.31.1.1.1.1"}
        command.add("snmpget");
        command.add("-v2c");
        command.add("localhost");
```

Figura 62. Componentes y declaraciones de Clase SNMPInterfaz.java

6.1.6. MÓDULO DE ALTA DISPONIBILIDAD (HA)

6.1.6.1. CORE ForCES LFBs

Hay dos LFBs que son usados para controlar la operación del Protocolo ForCES e interactuar con FEs y CEs, FE Protocol LFB y FE Object LFB.

Aunque estos LFBs tienen la misma forma e interfaces de los otros LFBs, son especiales en muchos aspectos, ya que ellos fijan las bien conocidos ID de Clases e Instancias, además su estado no puede ser cambiado por el protocolo, alguna operación para cambiar el estado de tales LFBs (para deshabilitar el LFB) debe resultar en un error.

Por otra parte estos LFBs deben existir antes de que el primer mensaje ForCES pueda ser enviado o recibido. Todos los componentes en estos LFBs deben tener predefinidos valores por defecto.

6.1.6.2. FE Protocol LFB

El FE Protocol LFB es una entidad lógica en cada FE que es usado para controlar el protocolo ForCES. El FE Protocol LFB Class ID es asignado el valor 0x2, el FE Protocol LFB Instance ID se le asigna el valor 0x1, la cual solo debe haber una instancia de este LFB en el FE. Los valores de los componentes en el FE Protocol LFB se predefinen con valores por defecto que son especificados en aquí, si se desean cambiar esos valores, se usa un mensaje **Config** desde el CE. Las capacidades de este LFB son de solo lectura y los componentes son de lectura y escritura.

6.1.6.2.1. Componentes del FE Protocol LFB para Alta Disponibilidad

- **CEHBPolicy**

CE Heartbeat Policy- Esta política, con el parámetro CE Heartbeat Dead Interval (CEHDI), define los parámetros de operación para que el FE chequee si el CE se encuentra operativo. Los valores de la política son:

- "0" (por defecto). Especifica que el CE enviara un mensaje Heartbeat al FE(s) siempre que el CE alcance un intervalo de tiempo dentro del cual no son enviados mensajes PL desde el CE a los FE(s).
- "1"- El CE no generará ningún mensaje Heartbeat, lo que significa que el CE no quiere que el FE los este chequeando.

- **CE Heartbeat Dead Interval (CEHDI)**

Intervalo de tiempo que el FE usa para chequear si el CE esta operativo. Si el FE no recibe algún mensaje desde el CE dentro de este intervalo de tiempo, el FE deduce pérdida de conectividad, el cual implica que el CE dejo de estar operativo o la asociación se perdió. El valor por defecto es de 30 segundos.

- **FEHBPolicy**

El FE Heartbeat Policy. Esta política, con el parámetro FE Heartbeat Interval (FE HI), define parámetros de operación de cómo el FE debe comportarse para que el CE pueda deducir su operatividad.

- "0" (defecto). El FE no debe generar mensajes Heartbeat. En este escenario el CE es responsable de chequear la operatividad del FE enviando los mensajes Heartbeat con el campo de la bandera ACK un "AlwaysACK", el FE debe responder ese mensaje, con un mensaje Heartbeat, el cual en el campo de la bandera ACK va configurado "NoACK".
- "1". Especifica que el FE debe activamente enviar mensajes Heartbeat si alcanza el intervalo de tiempo asignado por el FEHI.

- **FE Heartbeat Interval**

Intervalo de tiempo que el FE debe usar para enviar un mensaje Heartbeat, siempre y cuando no haya otros mensajes enviados desde el FE al CE durante este tiempo. El valor por defecto son 500 milisegundos.

- **CEFailoverPolicy**

Especifica el comportamiento del FE cuando la asociación con el CE se pierde. Cuando se pierde la asociación dependiendo de la configuración, una de las siguientes políticas se activa:

- "0" (Defecto). El FE deja de funcionar inmediatamente y transiciona al modo FEOperDisable.

- "1". Indica que el FE es capaz de reiniciar con Alta Disponibilidad. En tal caso el FE pasa al estado no asociado y el temporizador CEFTI es reiniciado. El FE puede continuar enviando paquetes mientras intenta re-asociación con el CEID primario o un posible CE de backup, si falla la re-asociación con algún CE y el CEFTI expira, entonces el FE pasara al estado de pre-asociación.
- "2". Indica Alta Disponibilidad sin reinicio delicado (graceful restart), si existe algún cambio en la configuración del CE no lo tiene en cuenta, hay que volver a subir los servicios.
- "3" Indica Alta disponibilidad con reinicio delicado. Cuando se realiza la re-asociación con el mismo CE o un nuevo CE, no es necesario subir los servicios ya que estos siguen activos.

- **CE Failover Timeout Interval**

Intervalo de tiempo asociado con el CEFailoverPolicy. Su valor por defecto es de 300 segundos.
Hay que tener en cuenta que es conveniente establecer el valor del intervalo CEFTI mucho mayor que el intervalo CEHDI ya que el efecto de expiración de este parámetro es devastador para el funcionamiento del FE.

- **FERestartPolicy**

Especifica el comportamiento del FE durante un reinicio del FE. El reinicio puede ser por fallo del FE o por razones que han hecho que el FE deje de operar y luego tenga que reiniciar. Los valores definidos son:

- "0" (Defecto). Reinicia el FE a partir de cero. El FE debe partir de la fase previa a la pre-asociación.
- "Otro". No se ha definido hasta el momento.

6.1.6.3. FUNCIONAMIENTO DEL MÓDULO DE ALTA DISPONIBILIDAD

El protocolo ForCES provee mecanismos para redundancia y fallo de CEs, lo que se conoce como Alta disponibilidad. La arquitectura ForCES permite a los FEs tener en cuenta múltiples CEs pero obliga a que solamente un CE sea el controlador maestro. Esto es conocido en la industria como redundancia 1+N. El CE maestro controla los FEs por medio del protocolo ForCES operando en la interfaz Fp. Si el CE maestro falla, el CE de backup asume la operación del NE.

La parametrización de alta disponibilidad en el FE se acciona mediante la configuración del LFB FE Protocol Object. El FE Heartbeat Interval, CE Heartbeat Dead Interval (CEHDI), and CE Heartbeat Policy ayudan en la detección de problemas de conectividad entre un FE y un CE. El CE Failover Policy define la reacción sobre una falla detectada.

6.1.6.3.1. Procesamiento de Alta Disponibilidad

La tabla de los CE's del FE Object Protocol LFB versión 2, contienen todos los CEIDs que el FE puede conectar y asociar como backupCEs.
El orden de los CE IDs en la tabla, define la prioridad en el cual el FE se conectara a los CEs. En la fase de pre-asociación, el primer CE ID (el índice menor en la tabla) en la tabla de los CEs, debe ser

el primer CE ID que el FE tendrá en cuenta para conectarse y asociarse. Si falla la conexión y asociación del FE con ese primer CE ID, intentara la conexión con el segundo CE ID y así sucesivamente, el ciclo regresará al inicio de la lista hasta que haya una conexión y asociación con él.

El FE debe asociarse con al menos un CE. Tras una asociación exitosa, el componente FEPO's CEID, identifica el CE maestro asociado.

Para evitar conflictos el FE debe responder los mensajes del CE maestro solamente, por ejemplo el FE debe ignorar los mensajes que vienen del CE de backup. Sin embargo los eventos asíncronos y los heartbeats son enviados a los CEs asociados. El intervalo Heartbeat, el CEHBPolicy y el FEHB Policy deben ser los mismos para todos los CEs.

La figura 63, muestra el diagrama de bloques que facilita la recuperación de la conexión con alta disponibilidad. Una vez el FE se ha asociado con el CE maestro se mueve a la fase de post-asociación (Estado de Asociación). En este estado, el CE maestro puede actualizar la lista de los CE backups. Se asume que el CE maestro se comunicara con los otros CEs dentro del NE para el propósito de sincronización por medio de la interfaz CE-CE, pero esta parte esta fuera del alcance del proyecto.

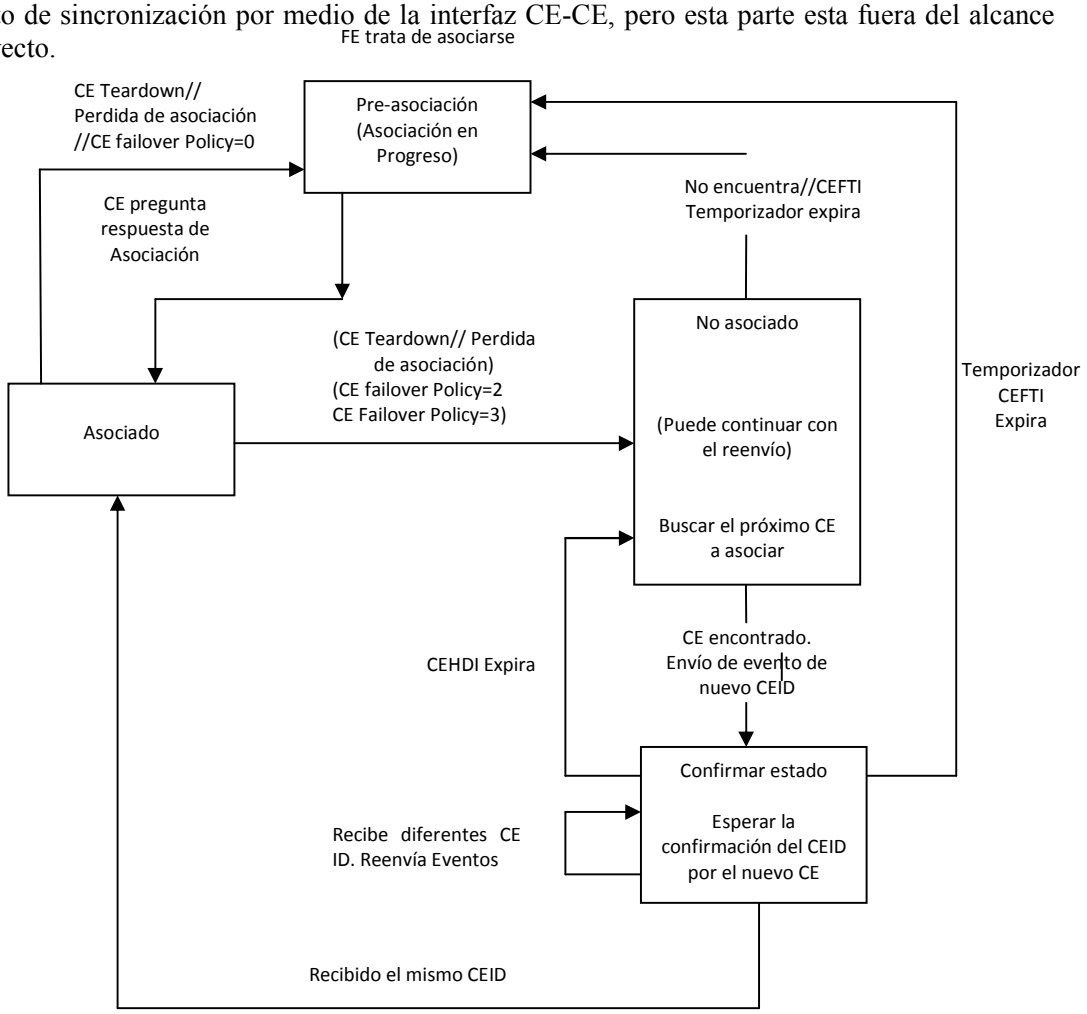


Figura 63. Diagrama de bloques del proceso de Disponibilidad (HA)

Mientras en la fase de post-asociación, si el CE Failover Policy es 2 (Alta Disponibilidad sin Reinicio-without Graceful Restart) o 3 (Alta Disponibilidad con Reinicio- with Graceful Restart), el FE, después de asociarse satisfactoriamente con el CE maestro, debe intentar conectarse y asociarse con todos los CEs que están latentes. En la figura 66, el paso 1 y 2 ilustran el FE asociado con el CE #1 como el maestro y procede al paso 3I al 3N para asociarse con los backup CE's en este caso CE #2. Si el FE falla al conectarse o asociarse con algún CEs, entonces el FE puede marcar como inalcanzable para evitar continuos intentos de conexión. El FE puede reintentar re-asociarse con CEs inalcanzables cuando sea posible.

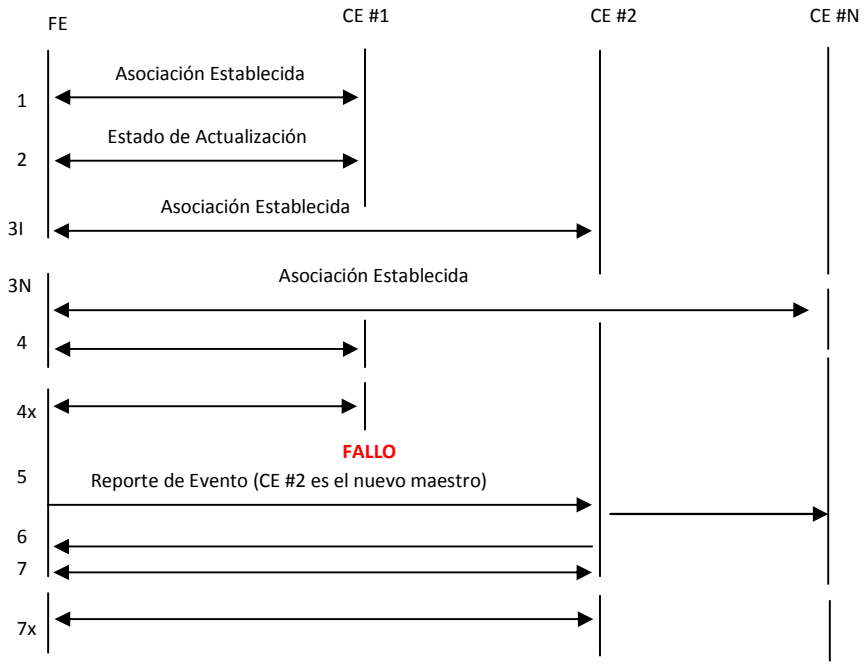


Figura 64. Proceso al fallo del CE maestro

Cuando el CE maestro por alguna razón es considerado inactivo, entonces el FE tratara de encontrar el primer CE asociado de la lista de todos los CE y continua con el recorrido hasta conectarse y asociarse con un CE. Una vez el FE selecciona el CE asociado para ser usado como nuevo maestro, el FE envía una notificación de cambio de evento High Availability Primary CE a todos los CEs asociados, para notificarles que el CE primario se encuentra inactivo y reportar cual CE es el nuevo maestro.

El nuevo CE maestro debe configurar el componente CE ID del FE dentro del tiempo límite definido en el CEHDI Failover Timeout como una confirmación de que el FE ha tomado la decisión correcta.
Si el FE no realiza la confirmación dentro del CEHDI Failover Timeout, seleccionara el próximo CE en la lista y lo anunciara como el nuevo maestro.

Si el temporizador CEFTI expira en cualquiera de los estados de no-asociación o confirmación sin un nuevo maestro CE confirmado, entonces el FE debe revertir a la etapa de pre-asociación.

En la figura 65, se observa la interfaz grafica del módulo de Alta Disponibilidad y los parámetros que se deben configurar.

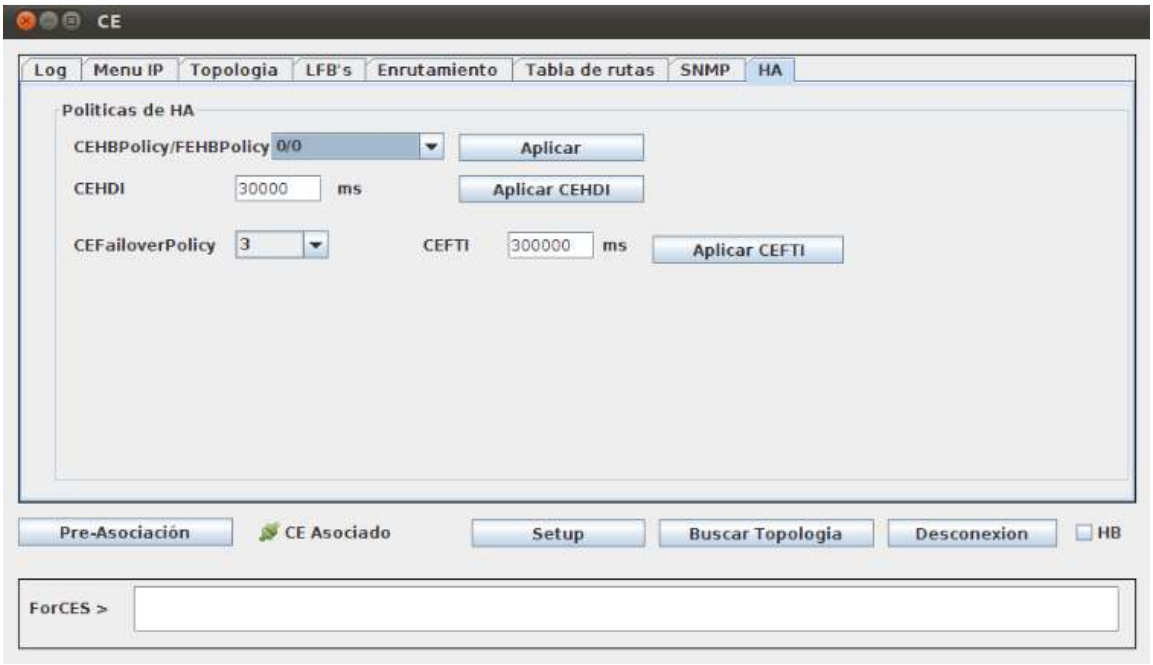


Figura 65. Módulo de Alta Disponibilidad (HA)

6.2. ELEMENTO DE REENVIO (FE)

Para realizar las pruebas de que el prototipo CE diseñado funciona, se desarrolló un FE de prueba, el cual contiene los LFBs, estos se pueden visualizar en una interfaz grafica FE, también tiene tres interfaces físicas Ethernet (Eth0, Eth1 y Eth2), una se emplea para interconexión CE-FE (por medio del protocolo ForCES), y las otras dos se emplean para interconexión con otros NEs, en este caso routers.

Al emplear la arquitectura ForCES, que es una arquitectura modular ya que separa el Plano de Control y el Plano de Reenvío y su elementos (CEs-FEs), se presenta el inconveniente que al requerir consultar y configurar las interfaces por el protocolo de enrutamiento y a su vez el envío y recepción de paquetes o realizar una gestión de las interfaces por el protocolo SNMP del prototipo, estas dos aplicaciones, solo observan las interfaces propias del equipo donde se instalaron, es decir en el CE y para enviar información de enrutamiento y recibir paquetes de enrutamiento o SNMP desde el FE, es necesario que el CE se apropie de las interfaces físicas del FE, por lo tanto se crea un módulo de interfaz virtual que se encarga de este proceso, que como se había mencionado anteriormente permite realizar una réplica de las interfaces físicas que tiene el FE en interfaces lógicas para reflejarlas hacia en el CE.

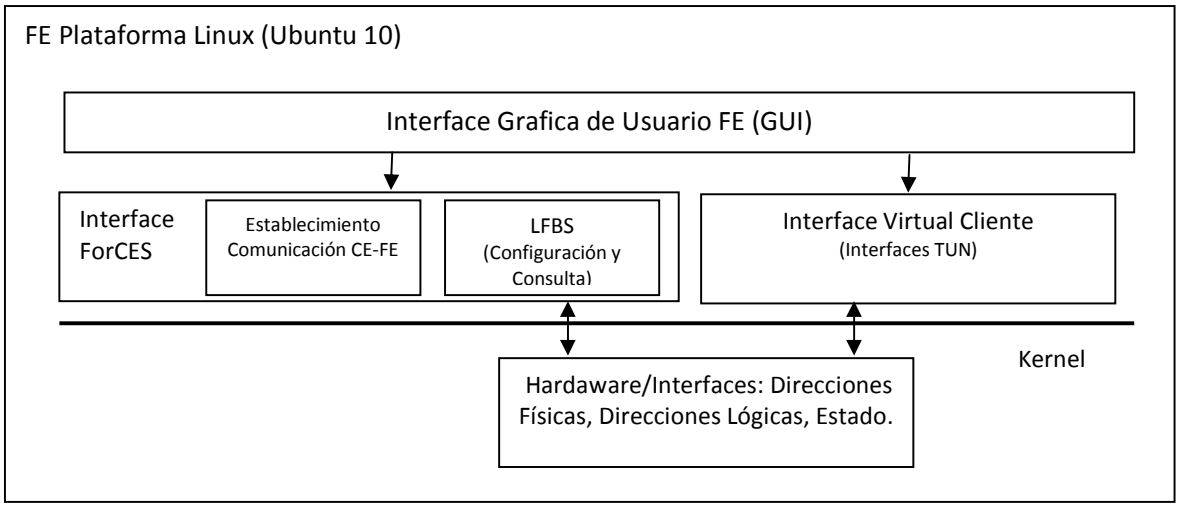


Figura 66. Estructura del Elemento de Reenvío

En la figura 66, se observa el diagrama de bloques de la estructura del FE diseñado, tiene un módulo de Interfaz ForCES que contiene los mensajes ForCES para el establecimiento de la conexión con el CE, un módulo de LFBs, donde se encuentran los XML que contienen las diferentes clases de los Bloques Lógicos Funcionales propios del FE y que el CE puede consultar y configurar cuando hay una asociación entre FE-CE. El módulo de interfaz Virtual Cliente como se explico anteriormente, permite realizar una réplica de las interfaces físicas que tiene el FE en interfaces lógicas para reflejarlas hacia el módulo de Interfaz Virtual que se encuentra en el CE, de modo que el CE cree que estas interfaces remotas son suyas.

Este FE, es una CPU con procesador Pentium IV, 512 de RAM y disco duro de 80 GB. Tiene tres tarjetas de red (Eth0, Eth1, Eth2). Las tarjetas con las interfaces Eth0 y Eth1, son las interfaces que se van a virtualizar y la tarjeta con la interfaz Eth2 es la que se conecta al CE por medio del protocolo ForCES.

En la figura 67, se observa la interfaz grafica del FE, donde se encuentran los botones que realizar la pre-asociación, la escucha del puerto y la asociación entre el FE y CE por medio del protocolo ForCES.

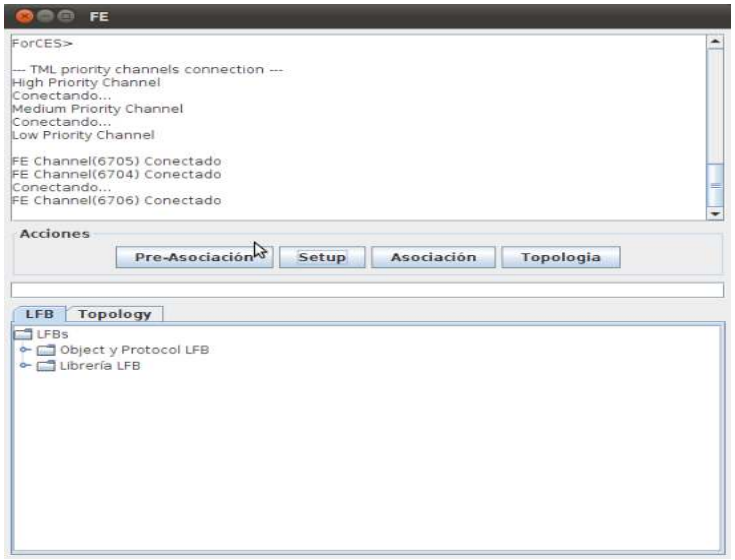


Figura 67. Interface grafica FE

7. PROTOCOLO DE INTEROPERABILIDAD DEL PROTOTIPO

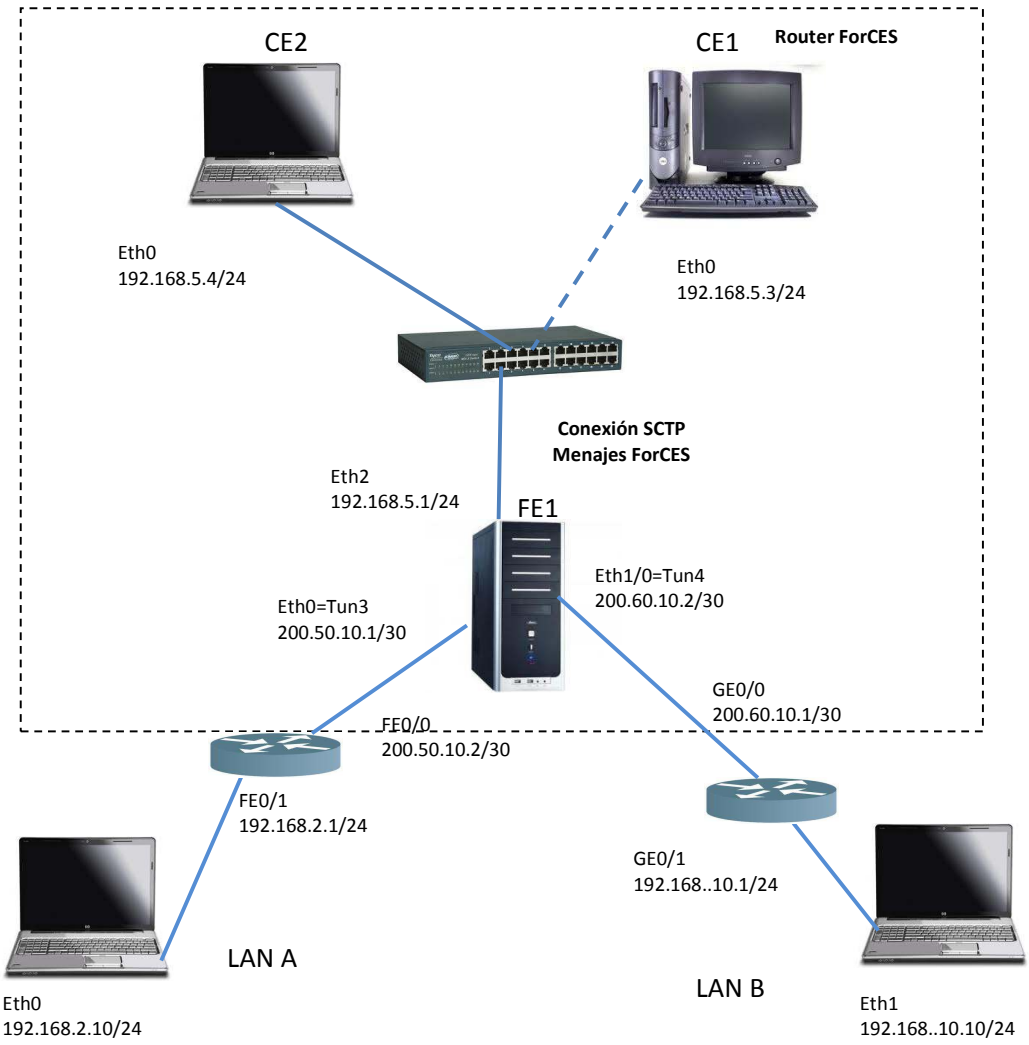


Figura 68. Topología para el protocolo de pruebas

En la figura 68, se muestra la topología que se empleó para realizar el protocolo de pruebas del prototipo. Se utilizaron dos PC (uno de escritorio y un portátil) como CE, uno principal (con dirección 192.168.5.3) y el otro de backup (con dirección 192.168.5.4). Otro PC se configuró como FE, este computador tiene tres tarjetas de red con las interfaces Eth0 y Eth1 (para ser virtualizadas

por el CE) y la interfaz Eth2 (conexión ForCES con el CE). También se configuraron dos routers CISCO que permiten la interconexión entre la LAN A y la LAN B.

7.1. ESCENARIO 1. INTERCONEXION POR EL PROTOCOLO FORCES

Para la interconexión del protocolo ForCES, tanto en el CE como en el FE se oprime el botón con el comando Pre-Asociación, que tiene asociado el comando ***preassociation setup***. Al realizar la pre-asociación, el CE lee el CE. Propertier y visualiza los datos de la consola CE, como se muestra en la figura 69.

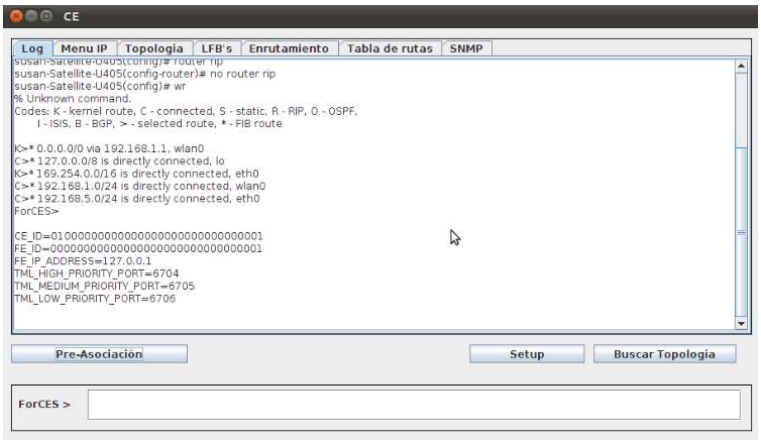


Figura 69. Pre-Asociación

Luego se realiza conexión del canal, lo cual se ejecuta oprimiendo el botón ***setup*** que tiene asociado el comando tmlsetup, ese se ejecuta primero en el CE y luego en el FE. El CE se queda escuchando en los puertos hasta que el FE realice la conexión, como se observa en la figura 70.

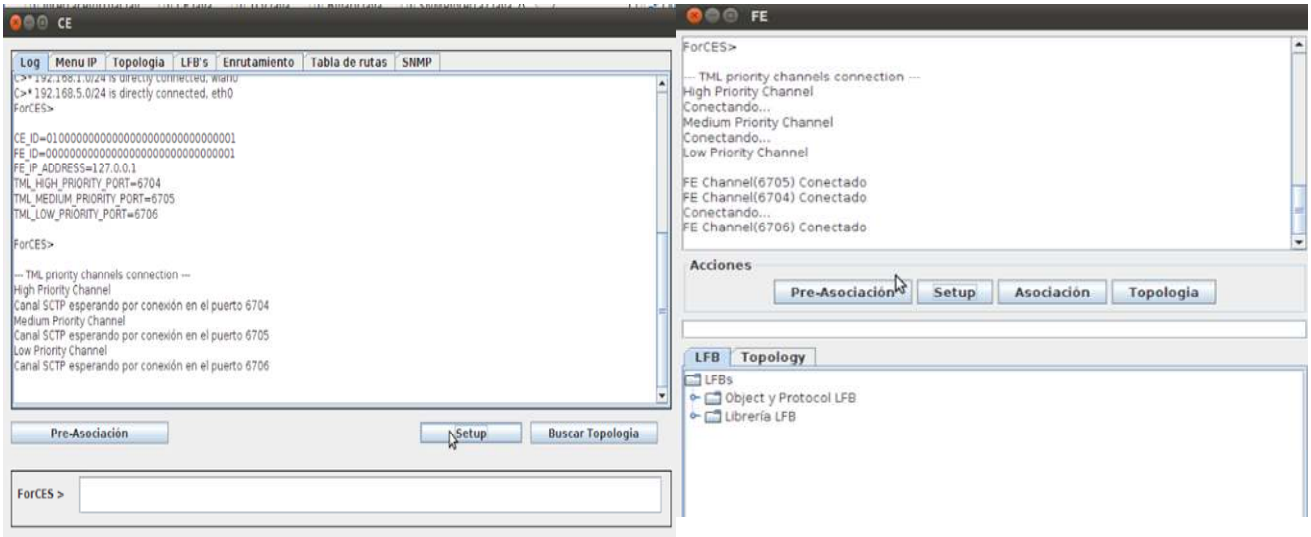


Figura 70. TMLSetup

Finalmente el FE se asocia al CE oprimiendo el botón Asociación, asociado al mensaje association setup. El CE responde con un association Setup Response, como se observa en la figura 71.

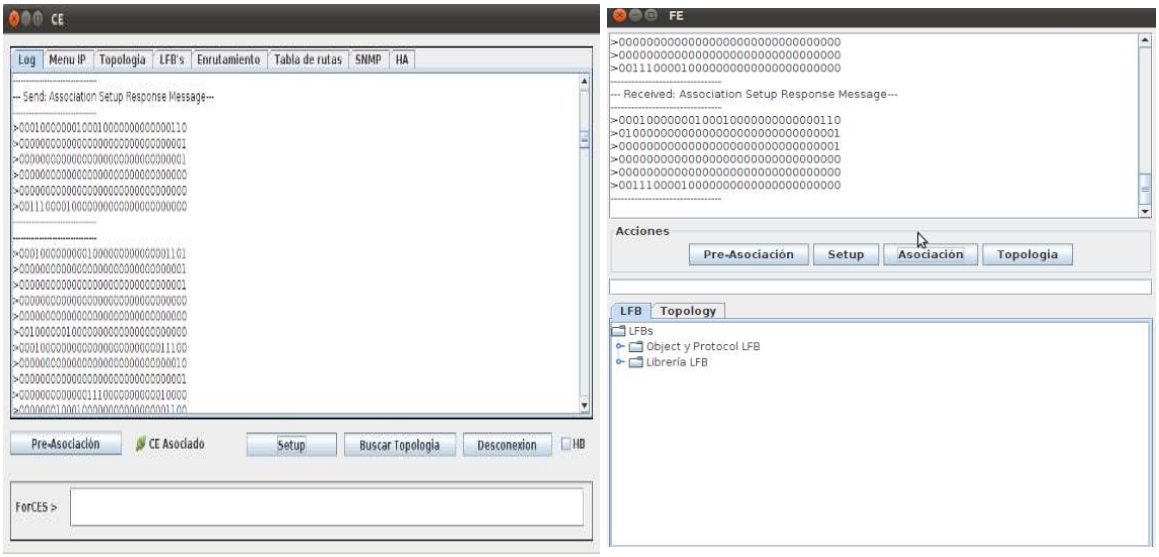


Figura 71. Conexión protocolo ForCES entre el CE-FE

7.2. ESCENARIO 2. CONFIGURACION DE LAS TOPOLOGIAS EN LOS LFBS

La configuración de las topologías se realizar por medio del comando ***config.***

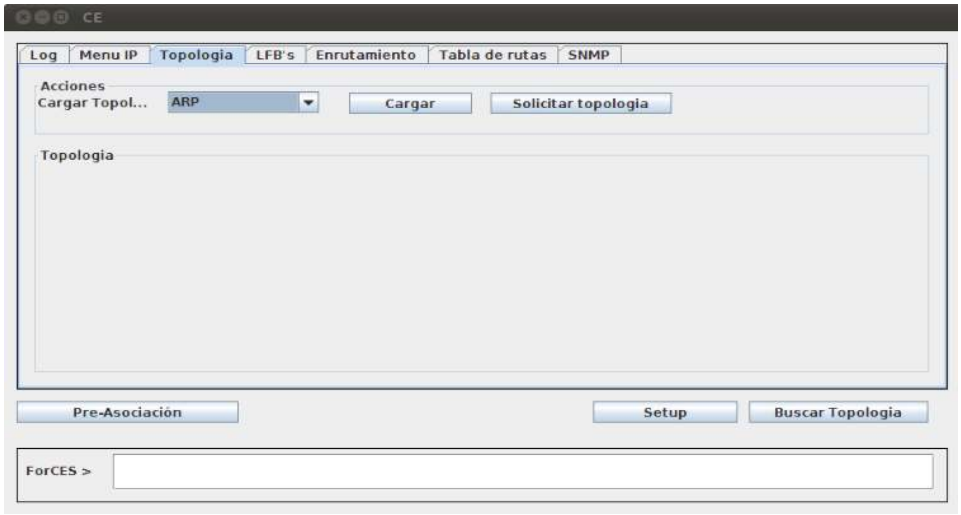


Figura 72. Interface Topología LFBs

Las topologías a configurar, se encuentran en un archivo de texto guardado en el paquete files de los archivos bin y source en el workspace de java. En la figura 72, se observa la interfaz de la topología, para cargar cada topología solo se despliega la pestaña y se elige el protocolo que se

quiere configurar con las clases LFBs en el FE, una vez elegida, se oprime el botón cargar y que enviar comandos config al LFBTopology en el FE donde se configura la topología. Si el CE solicita visualizar la topología configurada en el FE, oprime Solicitar topología, que contiene comandos query.

7.2.1. TOPOLOGIA ARP

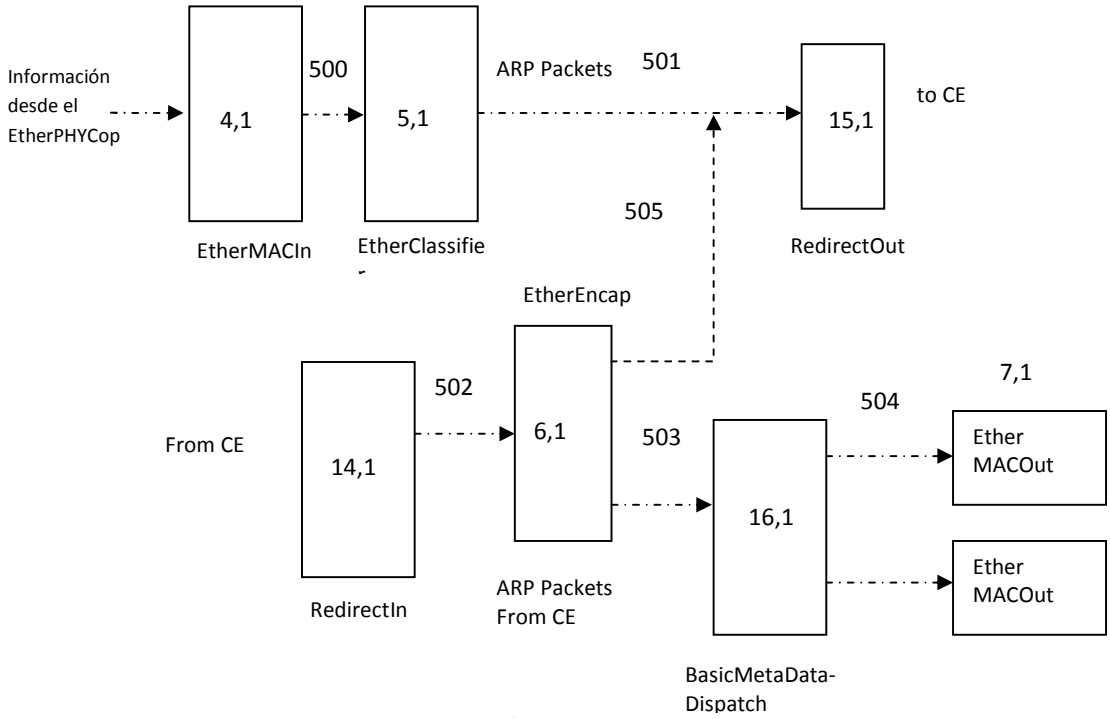


Figura 73. Topología LFBs para el proceso de ARP

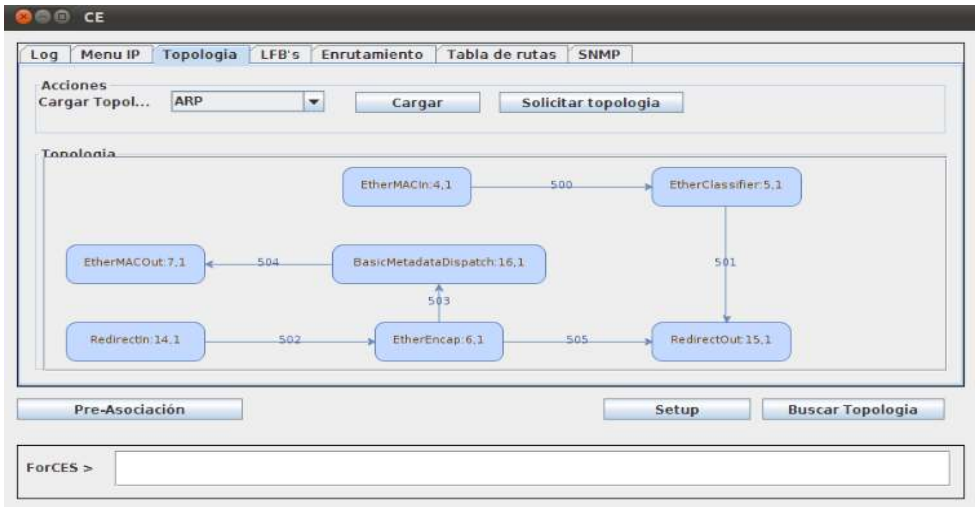


Figura 74. Topología LFBs para el proceso de ARP solicitada en el CE

7.2.2. TOPOLOGIA DE ENRUTAMIENTO

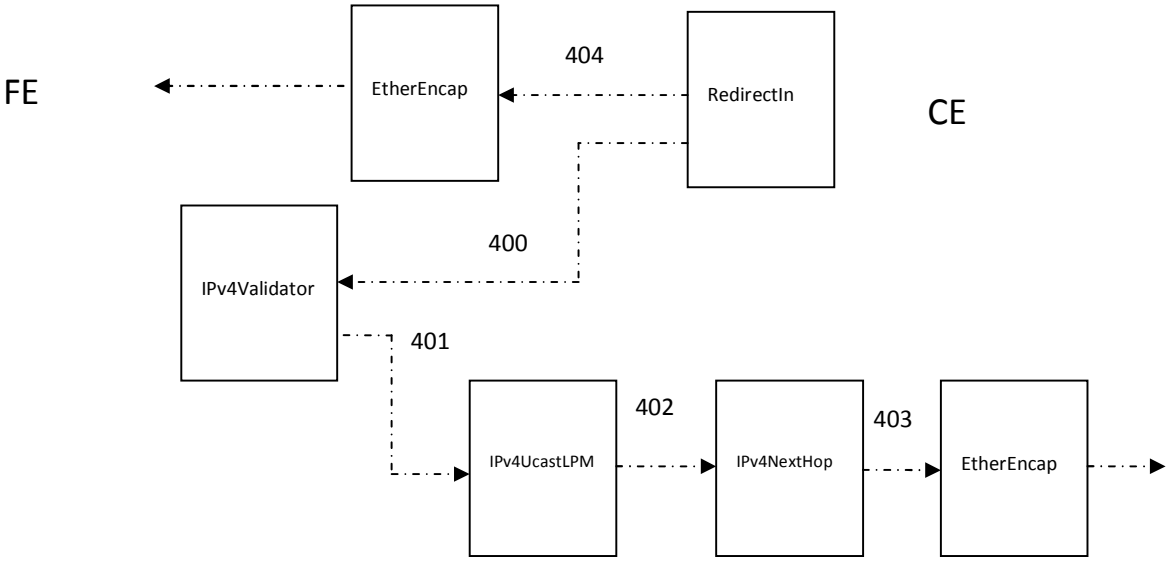


Figura 75. Topología LFBs para el proceso de Enrutamiento

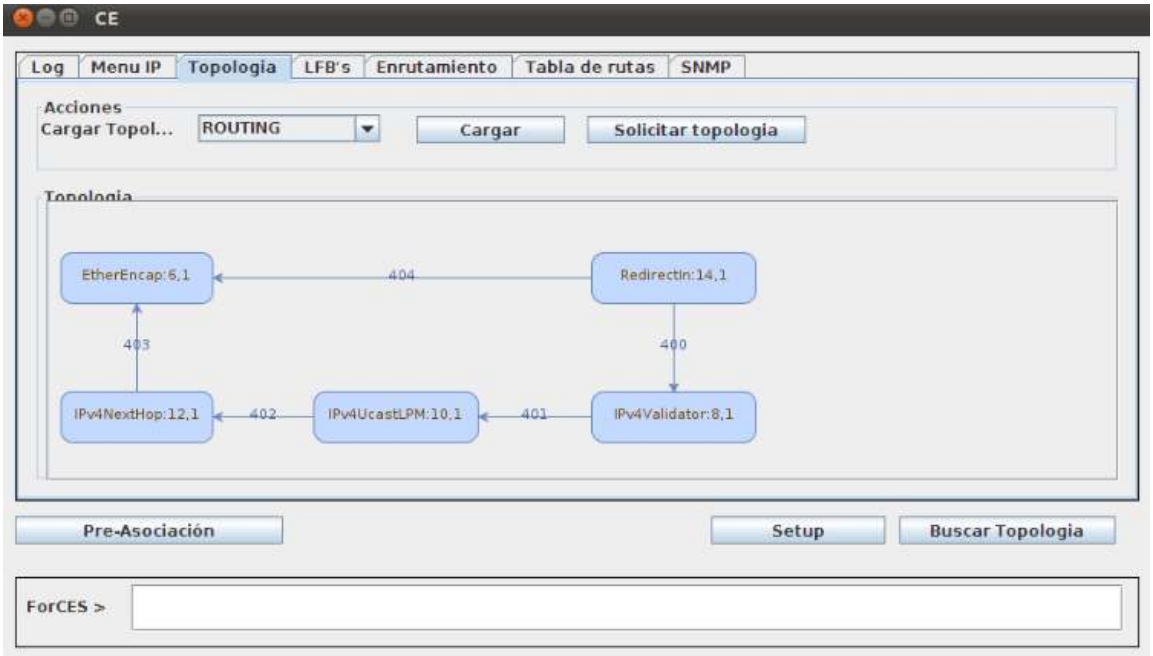


Figura 76. Topología LFBs para el proceso de Enrutamiento solicitada en el CE

7.2.3. TOPOLOGIA DE FORWARDING

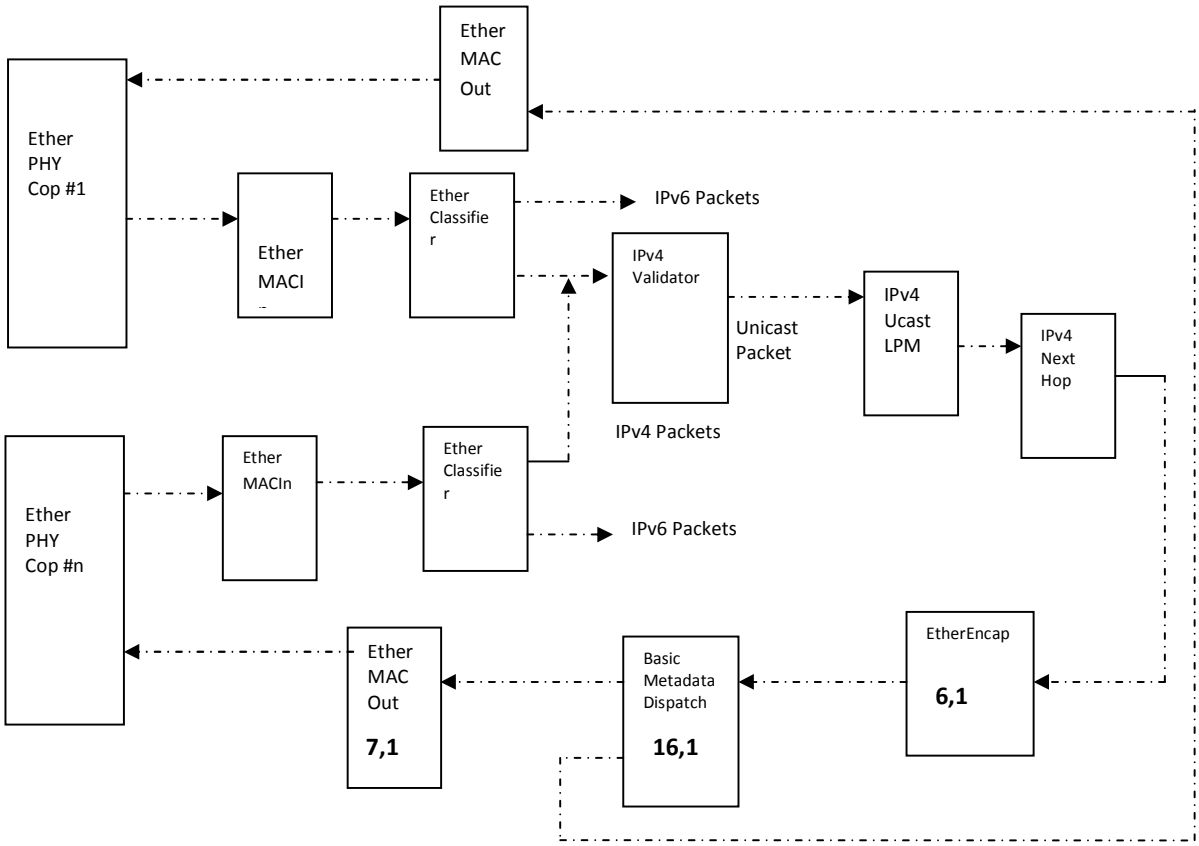


Figura 77. Topología LFBs para el proceso de Forwarding

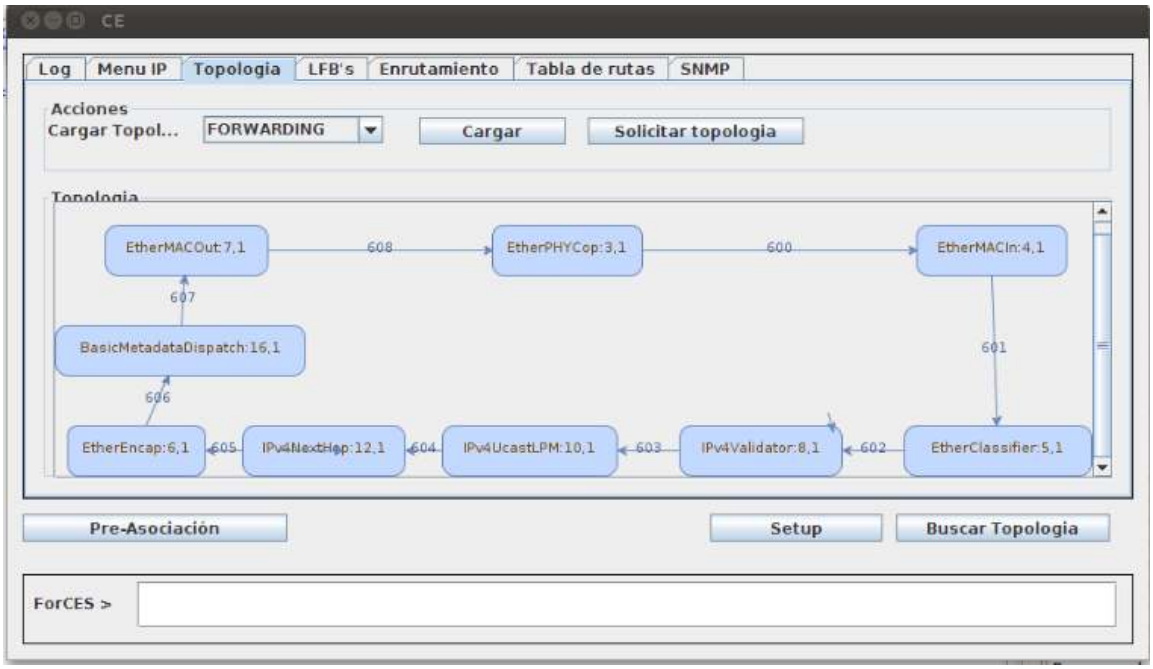


Figura 78. Topología LFBs para el proceso de Forwarding solicitada en el CE

7.2.4. TOPOLOGIA SNMP

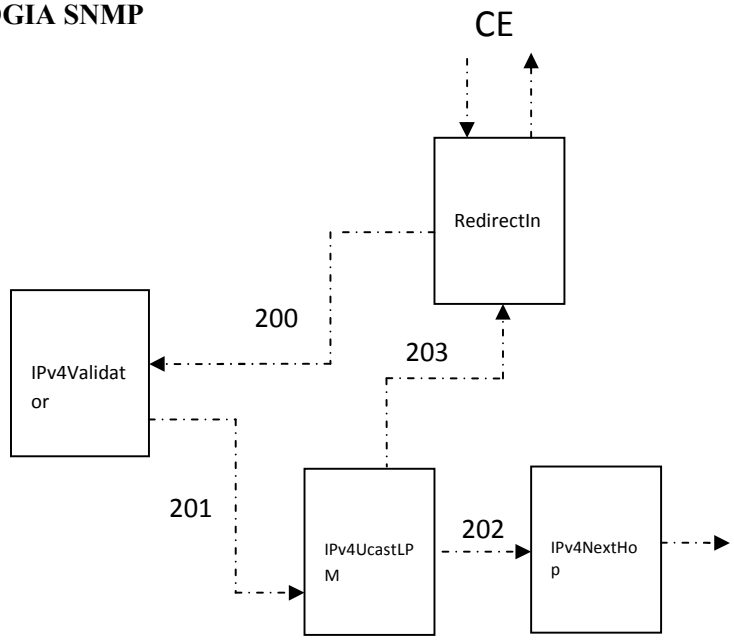


Figura 79. Topología LFBs para el proceso de SNMP

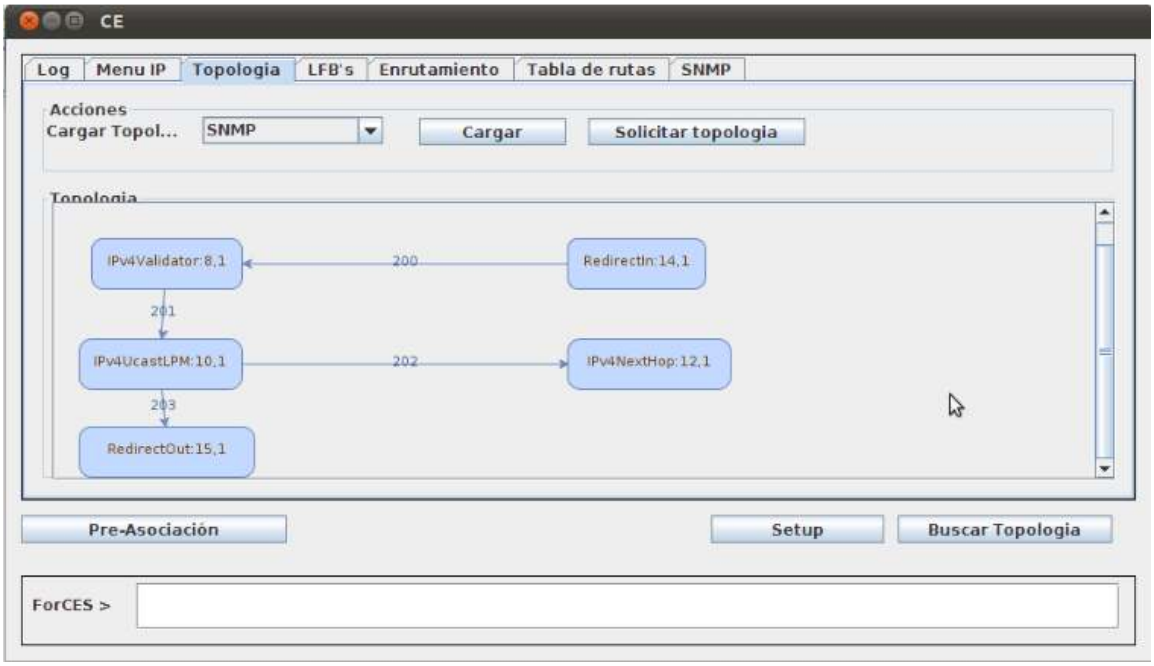


Figura 80. Topología LFBs para el proceso de SNMP solicitada en el CE

7.3. ESCENARIO 3. CONFIGURACION DE LAS INTERFACES VIRTUALES

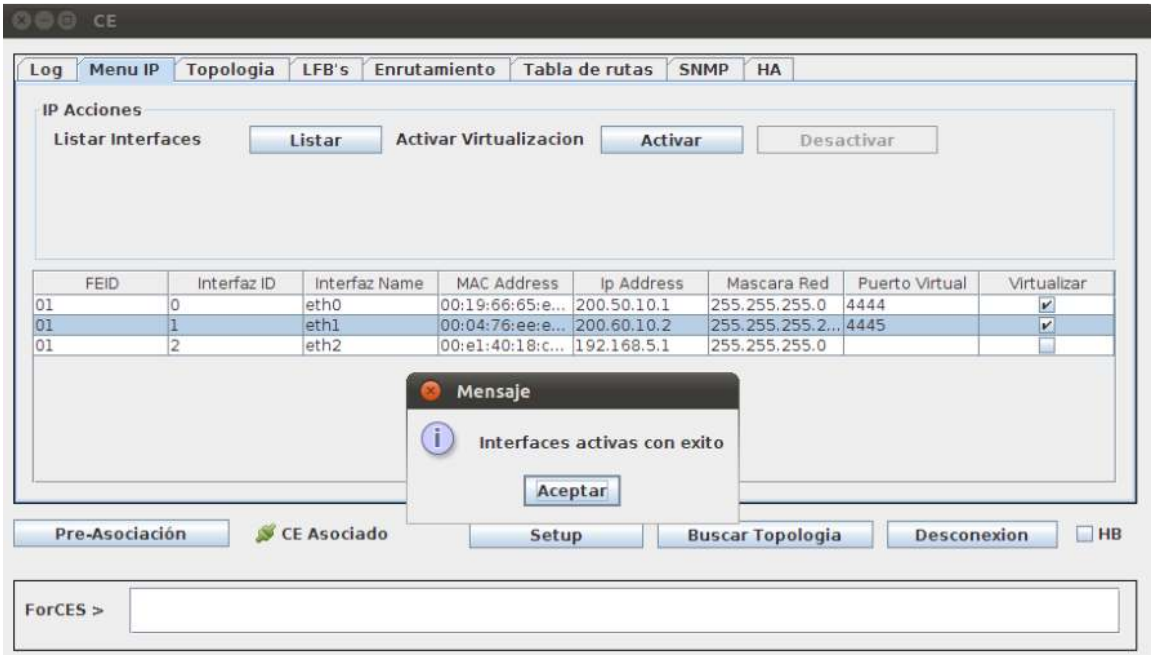


Figura 81. Configuración de las interfaces virtuales

Al oprimir el botón listar, se listan las interfaces que pertenecen al FE, como se observa en la figura 81, Eth0, Eth1 y Eth2. Para virtualizarlas se configura primero la dirección IP en cada una con su respectiva mascara de subred, luego se configura el puerto TCP, en este caso se selecciono 4444 y4445 y por último se oprime activar, para enviar las direcciones a las interfaces del FE. Para el CE la interfaz Eth0 que va por el puerto 4444 es la TUN3 y la Eth1 que va por el puerto 4445 es la TUN4. El figura 82, se observa las direcciones IP que se configuraron desde el CE a las interfaces Eth0 y Eth1 del FE.

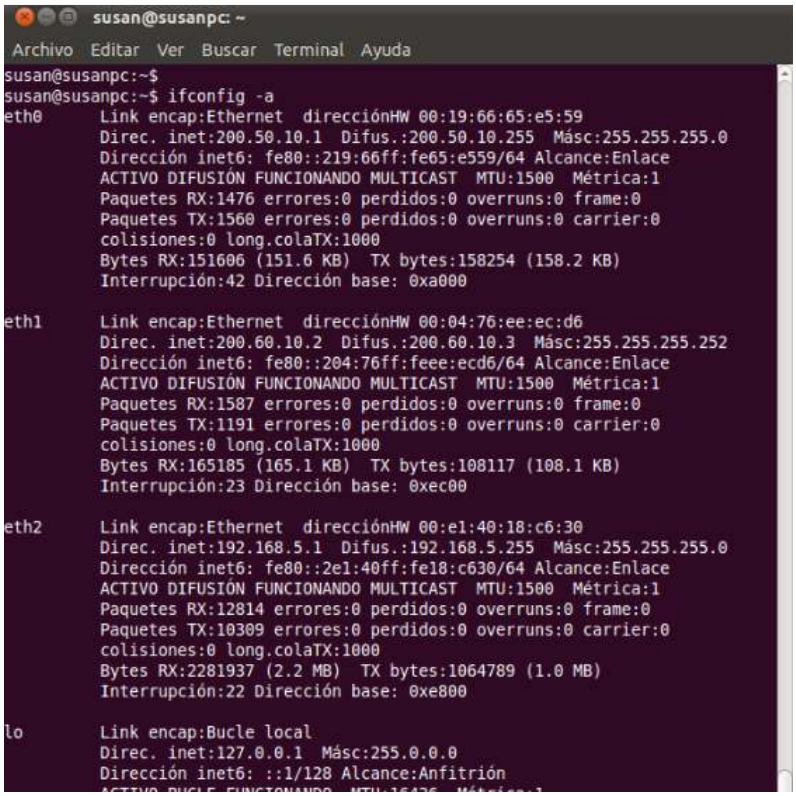


Figura 82. Direcciones IP de las interfaces en el FE

En la figura 83, se observa las interfaces virtuales TUN3 y TUN4 que en realidad con las Eth0 y Eth1 del FE, el CE ve las interfaces TUN como si fueran propias.

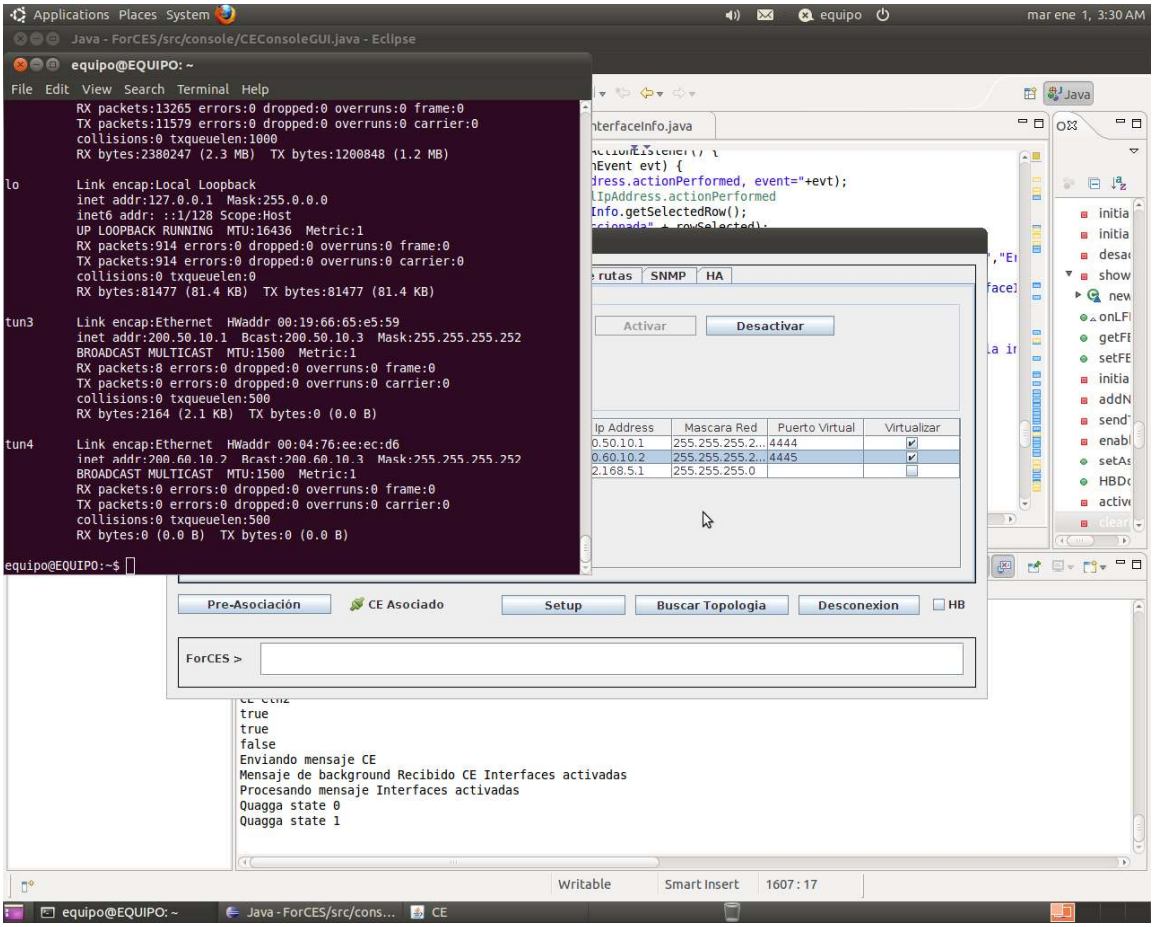


Figura 83. Direcciones IP de las interfaces en el CE

7.4. ESCENARIO 4. CONFIGURACION DEL PROTOCOLO DE ENRUTAMIENTO

Después de configurar las interfaces virtuales, se procede a subir las interfaces virtuales en el módulo de enrutamiento, activando el shutdown. En la figura 84, se observa las interfaces virtuales TUN3 y TUN4 activas.

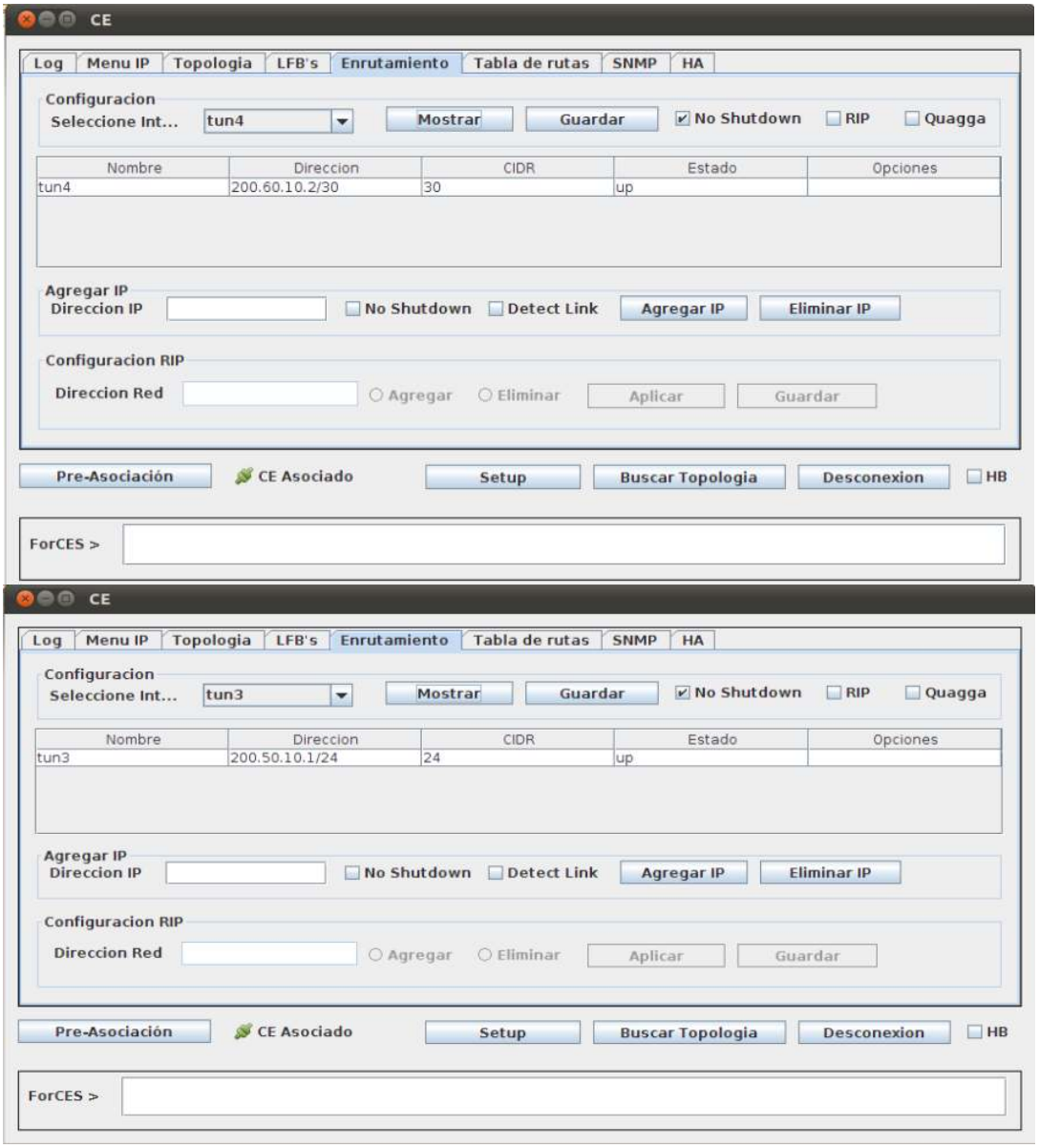


Figura 84. Interfaces TUN3 y TUN4 activas.

Por último se configura el protocolo de enrutamiento RIPv2, activando en la parte superior derecha la casilla RIP, y en la parte inferior izquierda se configura la red o redes directamente conectadas. Se observa en la figura 85.

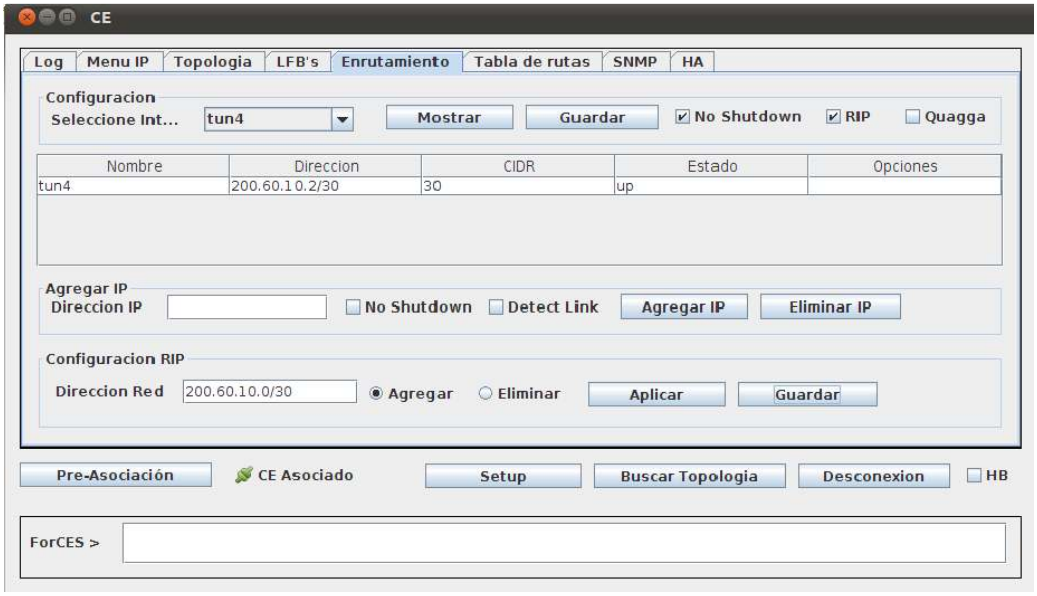


Figura 85. Configuración de RIP

En la figura 86, se observa la tabla de enrutamiento, donde aparecen las redes aprendidas por RIP, las directamente conectadas, la métrica y el próximo salto.

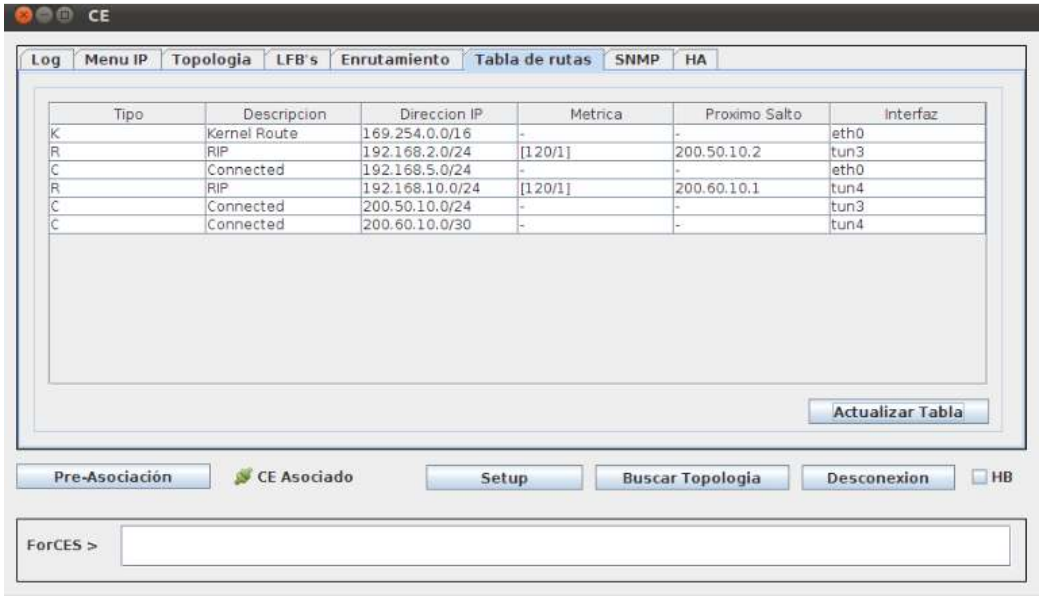


Figura 86. Tabla de Enrutamiento

7.5. ESCENARIO 5. CONFIGURACION DEL PROTOCOLO DE GESTION NET-SNMP

En la figura 87, se observan los parámetros que se visualizan en el módulo SNMP, como son los byte enviados, bytes recibidos, paquetes enviados y paquetes recibidos, en cada una de las interfaces del prototipo.

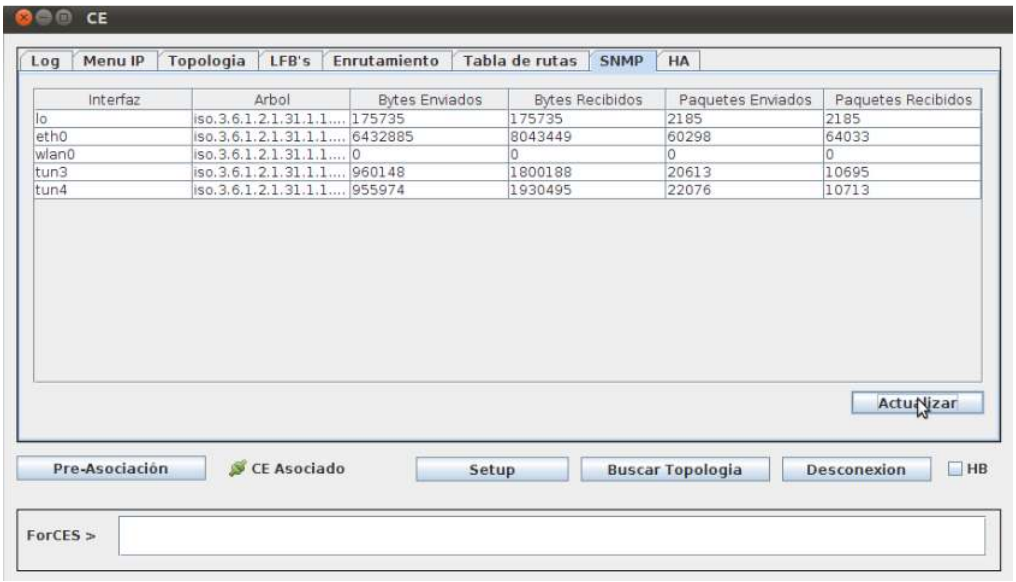


Figura 87. Parámetros del módulo SNMP

Para verificar el funcionamiento del protocolo de gestión Net-SNMP implementado en el prototipo CE, se utilizo el software MIB Browser en host de cada red LAN configurada. Estas pruebas se observan en la figura 88 y figura 89.

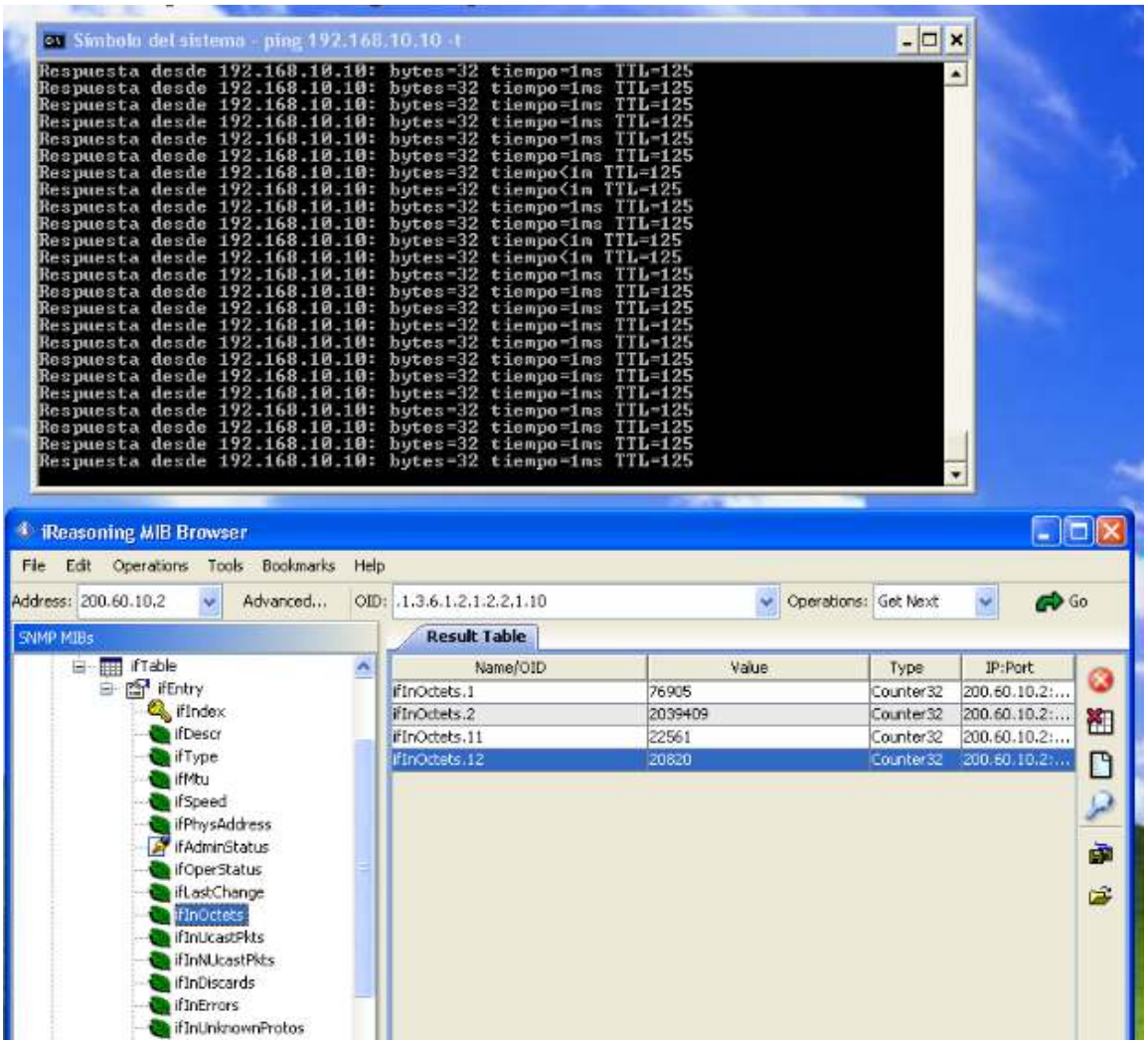


Figura 88. MIB Browser en la red LAN 192.168.2.0/24

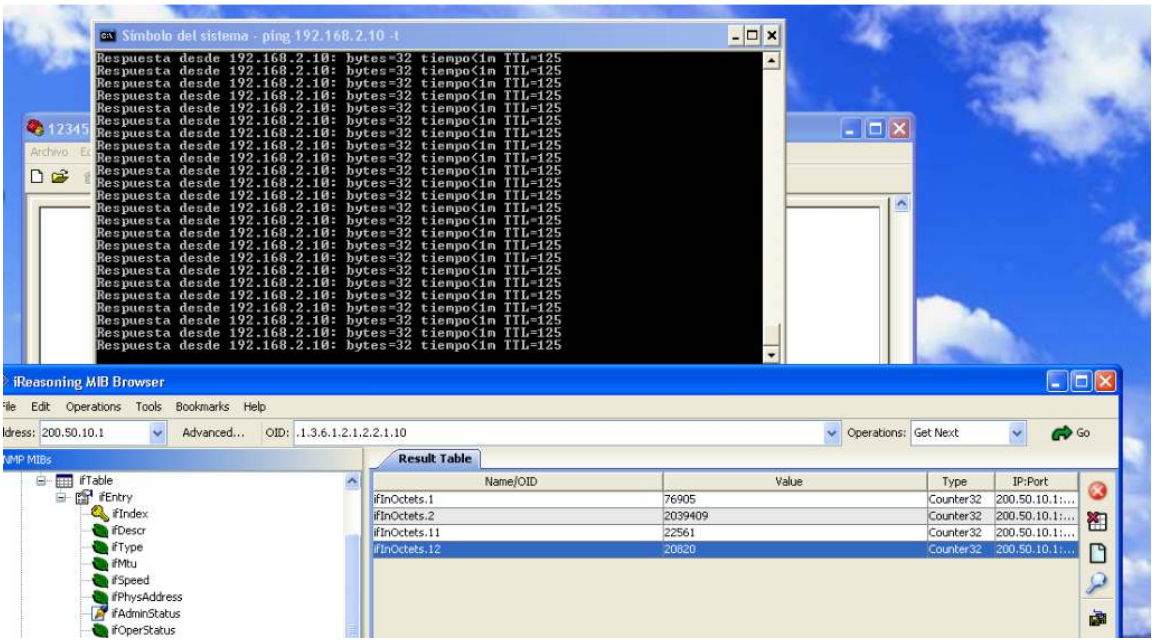


Figura 89. MIB Browser en la red LAN 192.168.10.0/24

7.6. ESCENARIO 6. CONFIGURACION Y PUESTA EN MARCHA DEL SISTEMA DE ALTA DISPONIBILIDAD

En este escenario, se configuran los parámetros de Alta Disponibilidad, estos parámetros deben ser los mismos en todos los CEs que hacen parte del Elemento de Red, en este caso hay solamente dos CEs. El CEHBPolicy y FEHBPolicy se configuran en 0/0, que indica que el CE está enviando mensajes Heartbeat Request hacia el FE y el FE le responde con mensajes Heartbeat Response. Si se cae la conexión entre el CE principal y el FE, el CE sigue enviando mensajes CE hacia el FE, si en 20 segundos el CE no recibe un HB Response, indica hubo desconexión con el FE, sin embargo el componente CEHDI se configuro a 30 segundos, que indica que si se cae la conexión con el CE principal, tiene ese tiempo para buscar en la tabla de CEs un CE de respaldo para re-asociarse.

El componente CEFailoverPolicy se configura en 3, indicando que hay alta disponibilidad con sin reinicio, que significa que los servicios que estaban activos es necesario volverlos a subir, estos servicios son las interfaces virtuales, enrutamiento y SNMP.

Otro componente a tener en cuenta es el tiempo CEFTI, en este caso se configuro en un tiempo de 300000 milisegundos, este parámetro indica que si el FE no logra re-asociarse con un CE, el NE pierde definitivamente la conexión. La configuración de estos parámetros se observa en la figura 90.

En la figura 91, se observa el mensaje HB Request y mensaje HB Response, cuando las políticas de Alta Disponibilidad en el CEHBPolicy/FEHBPolicy se configuran en 0/0.

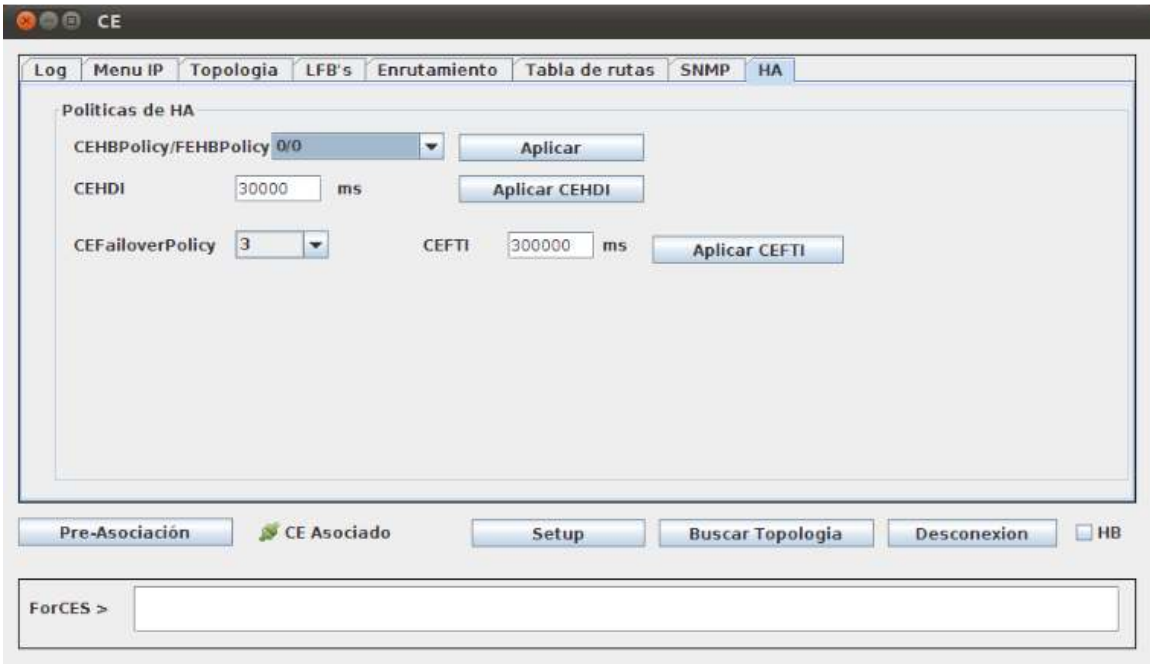


Figura 90. Configuración de los componentes de Alta Disponibilidad.

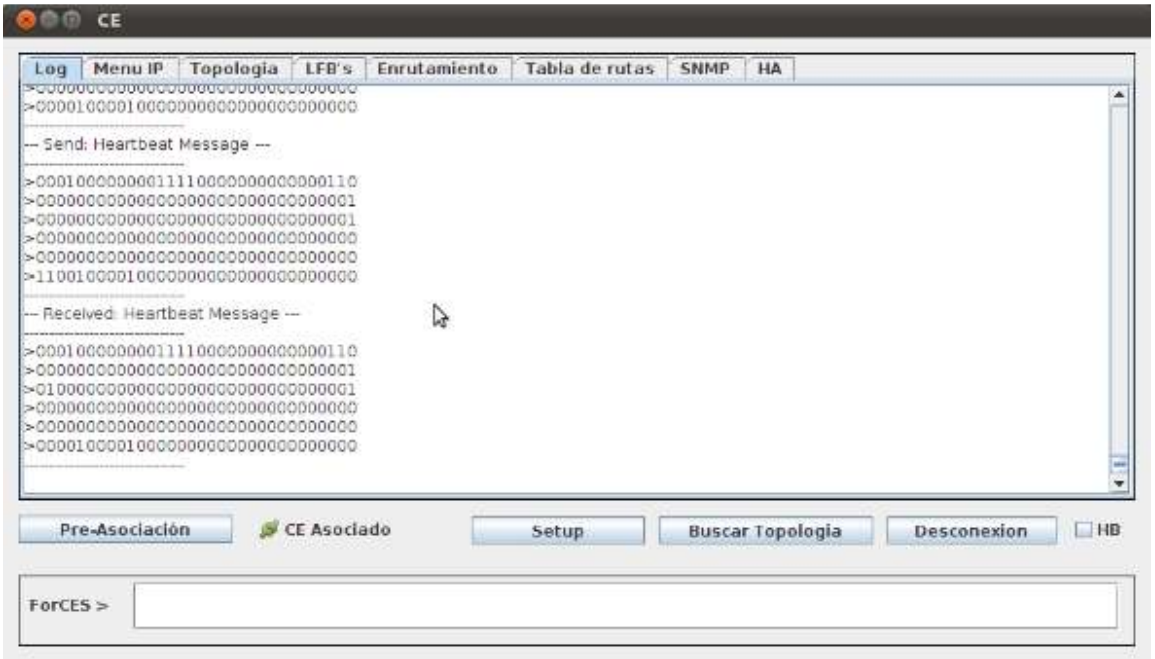


Figura 91. Mensaje Heartbeat enviado y recibido desde el CE-FE y FE-CE respectivamente.

8. CONCLUSIONES

- Durante el desarrollo del proyecto se presentaron ciertas dificultades, una de ellas fue la poca información bibliográfica acerca de implementaciones en la arquitectura y protocolo ForCES, solamente dos grupos de investigación han diseñado routers modulares ForCES, el grupo de investigación de Zhejiang Gongshang University de China, interconectar varios elementos de control y varios elementos de reenvío remotamente (en ciudades diferentes) por medio del protocol ForCES y el grupo de investigación de Royal Institute of Technology en Estocolmo, quienes diseñaron un router modular con arquitectura ForCES pero un protocolo de interconexión entre CE-FE desarrollado por ellos. Los artículos sobre estos dos proyectos dan información superficial del diseño de los dispositivos.

- Los routers convencionales son dispositivos que tienen una arquitectura cerrada, no son escalables ni reconfigurables ya que el elemento de control y el elemento de reenvío hacen parte de una sola estructura y no es posible realizar mejoras por separado en cada uno. Si el dispositivo no soporta ciertos servicios como QoS o VoIP, por ejemplo, y por tanto es necesario hacer un cambio del mismo. El prototipo diseñado, al trabajar con la arquitectura ForCES, que es una arquitectura abierta, hace de este un dispositivo modular, escalable y de bajo costo, ya que permite realizar mejoras en cada elemento que lo compone por separado (elemento de control y elemento de reenvío).

- La implementación de las interfaces virtuales entre los elementos de control y reenvío, permite a la arquitectura ForCES ver estos dos elementos como un solo elemento de red, así se encuentren en sitios remotos cada uno. Para el desarrollo de este módulo, se contó con la asesoría del grupo de investigación de Zhejiang Gongshang University y Royal Institute of Technology.

- El sistema de alta disponibilidad desarrollado, permite al prototipo ser un poco más robusto, ya que si se pierde conexión entre el elemento de control principal y el elemento de reenvío, que hacen parte del elemento de red, entra a funcionar otro elemento de control como respaldo, de esta forma el dispositivo sigue funcionando a menos que la falla se presente en el FE.

9. RECOMENDACIONES

- Actualmente la puesta en marcha de este prototipo se realiza de una forma manual, conexión entre CE-FE por medio del protocolo ForCES, virtualización de las interfaces, configuración del protocolo de enrutamiento y configuración de alta disponibilidad, la idea es que en desarrollos futuros se pueda implementar la operación del sistema en forma automática, de modo que la detección de interfaces por parte del CE y alta disponibilidad se configuren al momento del arranque del dispositivo y solo el administrador configure como un router convencional los servicios de enrutamiento y gestión.

- Otra mejora que se puede realizar, es el desarrollo de algoritmos para optimizar el tiempo de re-asociación en el módulo de alta disponibilidad, puesto que los valores empleados, son valores por defecto que se especifican en [23].

- Al implementarse el Elemento de Reenvío (FE), en un procesador Intel Pentium IV con 512 de RAM, las entidades lógicas que lo componen no son dinámicas, por tanto el Elemento de Control (CE) puede configurarlas y crear topologías pero no puede eliminarlas, debido a que la arquitectura del procesador no es reconfigurable porque ya está establecida con parámetros del fabricante. La idea es desarrollar el FE en una tarjeta NetFPGA lo que permitiría a este elemento ser dinámico.

10. BIBLIOGRAFIA

[1] ForTER – An Open Programmable Router Based on Forwarding and Control Element Separation. Weiming Wang, Ligang Dong, Bin Zhuge. Institute of Networks and Communications Engineering, Zhejiang Gongshang University. Sixth International Conference on Networking (ICN 2007), 22-28 April 2007, Sainte-Luce, Martinique, France.

[2] ForCES: Forwarding and Control Element Separation in IP Networks. Dr. Patrick Droz IBM Zurich Research Lab. IETF 2001

[3] Análisis y Estudio del plano de control de un Open Router basado en el Sistema Operativo Linux y en el Open Source Software Xorp. Olga Jaramillo, Rebeca Estrada, Raffaele Bolla. Escuela Superior Politécnica del Litoral (ESPOL), Facultad de Ingeniería en Electricidad y Computación (FIEC). Febrero 23-2009.Articulos tesis de grado-FIEC

[4] ForCES protocol as a solution for interaction of control and forwarding planes in distributed routers. Ivo Kovačević, dipl.ing. Ericsson, Belgrade, Serbia. ivo.kovacevic@ericsson.com

[5] Design and Implementation of an Open Programmable Router Compliant to IETF ForCES Specifications. Weiming Wang, Ligang Dong, Bin Zhuge, Ming Gao, Fenggen Jia, Rong Jin, Jin Yu, Xiaochun Wu. Institute of Network and Communication Engineering Zhejiang Gongshang University, Hangzhou, China. Networking, April,2007. ICN '07. Sixth International Conference on Communication, Networking & Broadcasting ; Computing & Processing (Hardware/Software).

[6] ForCES Protocol Specification draft-ietf-forces-protocol-02.txt. February, 2005. Network Working Group Internet-Draft

[7] RFC 1812. Requirements for IP Version 4 Routers. June, 1995. Network Working Group F. Baker, Editor. Request for Comments: 1812 Cisco Systems

[8] RFC 3654. Requirements for Separation of IP Control and Forwarding.

[9] RFC 3746. Forwarding and Control Element Separation (ForCES) Framework. Network Working Group. L. Yang, Intel Corp. R. Dantu, Univ. of North Texas. T. Anderson, Intel Corp. R. Gopal, Nokia, April 2004

[10] ForCES LFB Library draft-ietf-forces-lfb-lib-05. Weiming Wang, Evangelos Haleplidis, Kentaro Ogawa, Chuanhuang Li, J. Halpern. July, 2011.

[11] RFC 5810. Forwarding and Control Element Separation (ForCES) Protocol Specification. Internet Engineering Task Force (IETF). A. Doria, Ed. Lulea University of Technology. J. Hadi Salim, Znyx. R. Haas, Ed, IBM. H. Khosravi, Ed, Intel. W. Wang, Ed. L. Dong, Zhejiang Gongshang University. R. Gopal, Nokia. J. Halpern, March 2010.

[12] RFC 5812. Forwarding and Control Element Separation (ForCES) Forwarding Element Model. A. Doria, Ed. Lulea University of Technology. J. Hadi Salim, Znyx. R. Haas, Ed, IBM. H. Khosravi, Ed, Intel. W. Wang, Ed. L. Dong, Zhejiang Gongshang University. R. Gopal, Nokia. J. Halpern, March 2010.

[13] RFC 1850. OSPF Version 2 Management Information Base. Network Working Group. F. Baker, Cisco Systems. R. Coltun, RainbowBridge Communications, November , 1995

[14] IETF ForCES Working Group www.ietf.org/html.charters/forces-charter.html

[15] Cisco Visual Networking Index:Forecast and Methodology, 2009–2014.

[16]www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html

[17] Diseño e Implementación del protocolo ForCES. Pedro Luis Gonzalez Ramirez. Pontificia Universidad Javeriana. Trabajo de grado de Maestría. Julio, 2012

[18] ForCES Intra-NE High Availability. Draft-ietf-forces-ceha-03. W. M. Wang. Zhejiang Gongshang University, E. Haleplidis University of Patras, J. Hadi Salim Mojatatu Networks. February, 2012

[19] Research on High-Availability Based on Architecture of ForCES. Qun Li, Ligang Dong and Ming Gao. College of Information & Electronic Engineering, Zhejiang GongShang. University , Xuezheng Str., Xiasha University Town, Hangzhou, July, 2009 Asia-Pacific Conference on Information Processing.

[20] Research and Design of the High Availability Mechanism in the ForCES Router. Xiaochun Wu, Ligang Dong. Institute of Network and Communication Engineering Zhejiang Gongshang University. Networks, 2009. ICN '09. Eighth International Conference on Communication, Networking & Broadcasting ; Computing & Processing (Hardware/Software).

[21] Inter-FE topology discovery and maintenance technology on ForCES routers. Weiping Yu,Bin Zhuge,Weiming Wang College of Information & Electronic Engineering Zhejiang Gongshang University Hangzhou, P.R.China May, 2010

[22] Design and Implementation of a Distributed Router. Olof Hagsand, Markus Hidell, Peter Sjödin, Royal Institute of Technology. IEEE International Symposium on Signal Processing and Information Technology. 2005

[23] ForCES Intra-NE High Availability draft-ietf-forces-ceha-03. February, 2012.

11. ANEXO A.

```
<!-- XXX -->
<dataTypeDefs>
  <dataTypeDef>
    <name>CEHBPolicyValues</name>
        <synopsis>
           The possible values of CE heartbeat policy
        </synopsis>
      <atomic>
      <baseType>uchar</baseType>
      <specialValues>
        <specialValue value="0">
         <name>CEHBPolicy0</name>
         <synopsis>
            The CE heartbeat policy 0
         </synopsis>
         </specialValue>
        <specialValue value="1">
          <name>CEHBPolicy1</name>
          <synopsis>
             The CE heartbeat policy 1
          </synopsis>
        </specialValue>
       </specialValues>
       </atomic>
  </dataTypeDef>
  <dataTypeDef>
    <name>FEHBPolicyValues</name>
        <synopsis>
           The possible values of FE heartbeat policy
       </synopsis>
      <atomic>
      <baseType>uchar</baseType>
       <specialValues>
        <specialValue value="0">
         <name>FEHBPolicy0</name>
         <synopsis>
            The FE heartbeat policy 0
         </synopsis>
        </specialValue>
        <specialValue value="1">
          <name>FEHBPolicy1</name>
          <synopsis>
             The FE heartbeat policy 1
          </synopsis>
         </specialValue>
```

```
    </specialValues>
    </atomic>
</dataTypeDef>
<dataTypeDef>
<name>FERestartPolicyValues</name>
    <synopsis>
       The possible values of FE restart policy
    </synopsis>
    <atomic>
    <baseType>uchar</baseType>
    <specialValues>
      <specialValue value="0">
       <name>FERestartPolicy0</name>
       <synopsis>
          The FE restart policy 0
       </synopsis>
       </specialValue>
    </specialValues>
    </atomic>
</dataTypeDef>
<dataTypeDef>
<name>CEFailoverPolicyValues</name>
    <synopsis>
       The possible values of CE failover policy
    </synopsis>
    <atomic>
    <baseType>uchar</baseType>
    <specialValues>
     <specialValue value="0">
       <name>CEFailoverPolicy0</name>
       <synopsis>
          The CE failover policy 0
          No High Availability or Graceful Restart.
       </synopsis>
      </specialValue>
      <specialValue value="1">
        <name>CEFailoverPolicy1</name>
        <synopsis>
           Graceful Restart
        </synopsis>
       </specialValue>
       <specialValue value="2">
         <name>CEFailoverPolicy2</name>
         <synopsis>
            High Availability without Graceful Restart
         </synopsis>
        </specialValue>
       <specialValue value="3">
         <name>CEFailoverPolicy3</name>
         <synopsis>
```

```
            High Availability with Graceful Restart
          </synopsis>
        </specialValue>
       </specialValues>
      </atomic>
 </dataTypeDef>
<dataTypeDef>
  <name>FEHACapab</name>
        <synopsis>
          The supported HA features
        </synopsis>
      <atomic>
      <baseType>uchar</baseType>
      <specialValues>
        <specialValue value="0">
         <name>GracefullRestart</name>
         <synopsis>
             The FE supports Graceful Restart
         </synopsis>
         </specialValue>
        <specialValue value="1">
          <name>HA</name>
          <synopsis>
              The FE supports HA
          </synopsis>
        </specialValue>
       </specialValues>
      </atomic>
 </dataTypeDef>
 <dataTypeDef>
 <name>CEStatusType</name>
             <synopis>
                   Status values. Status for each CE.
             </synopis>
             <atomic>
             <baseType>uchar</baseType>
      <specialValues>
        <specialValue value="0">
         <name>Disconnected</name>
         <synopsis>
             No connection attempt with the CE yet.
         </synopsis>
         </specialValue>
        <specialValue value="1">
          <name>Connected</name>
          <synopsis>
              The FE has connected with the CE.
          </synopsis>
        </specialValue>
        <specialValue value="2">
```

```
            <name>Associated</name>
            <synopsis>
               The FE has associated with the CE.
            </synopsis>
            </specialValue>
           <specialValue value="3">
             <name>Lost_Connection</name>
             <synopsis>
                The FE was associated with the CE
                but lost the connection.
             </synopsis>
           </specialValue>
           <specialValue value="4">
             <name>Unreachable</name>
             <synopsis>
                The CE is deemed as unreachable by the FE.
             </synopsis>
           </specialValue>
         </specialValues>
              </atomic>
    </dataTypeDef>
    <dataTypeDef>
              <name>AllCEType</name>
              <synopsis>
                    Table Type for AllCE component.
              </synopsis>
              <struct>
                   <component componentID="1">
                        <name>CEID</name>
                        <synopsis>ID of the CE</synopsis>
                        <typeRef>uint32</typeRef>
                     </component>
                     <component componentID="2">
                          <name>CEStatus</name>
                          <synopsis>Status of the CE</synopsis>
                          <typeRef>CEStatusType</typeRef>
                     </component>
               </struct>
   </dataTypeDef>
</dataTypeDefs>
<LFBClassDefs>
  <LFBClassDef LFBClassID="2">
    <name>FEPO</name>
    <synopsis>
      The FE Protocol Object
    </synopsis>
    <version>2.0</version>
<components>
     <component componentID="1" access="read-only">
       <name>CurrentRunningVersion</name>
```

```
  <synopsis>Currently running ForCES version</synopsis>
  <typeRef>u8</typeRef>
</component>
<component componentID="2" access="read-only">
  <name>FEID</name>
  <synopsis>Unicast FEID</synopsis>
  <typeRef>uint32</typeRef>
</component>
<component componentID="3" access="read-write">
   <name>MulticastFEIDs</name>
   <synopsis>
      the table of all multicast IDs
   </synopsis>
   <array type="variable-size">
    <typeRef>uint32</typeRef>
   </array>
</component>
<component componentID="4" access="read-write">
  <name>CEHBPolicy</name>
  <synopsis>
   The CE Heartbeat Policy
  </synopsis>
  <typeRef>CEHBPolicyValues</typeRef>
</component>
<component componentID="5" access="read-write">
  <name>CEHDI</name>
  <synopsis>
    The CE Heartbeat Dead Interval in millisecs
  </synopsis>
   <typeRef>uint32</typeRef>
 </component>
 <component componentID="6" access="read-write">
   <name>FEHBPolicy</name>
   <synopsis>
     The FE Heartbeat Policy
   </synopsis>
   <typeRef>FEHBPolicyValues</typeRef>
 </component>
 <component componentID="7" access="read-write">
   <name>FEHI</name>
   <synopsis>
     The FE Heartbeat Interval in millisecs
   </synopsis>
   <typeRef>uint32</typeRef>
 </component>
 <component componentID="8" access="read-write">
   <name>CEID</name>
   <synopsis>
      The Primary CE this FE is associated with
   </synopsis>
```

```
    <typeRef>uint32</typeRef>
  </component>
  <component componentID="9" access="read-write">
    <name>AllCEs</name>
    <synopsis>
      The table of all CEs.
    </synopsis>
    <array type="variable-size">
     <typeRef>AllCEType</typeRef>
    </array>
  </component>
  <component componentID="10" access="read-write">
    <name>CEFailoverPolicy</name>
    <synopsis>
     The CE Failover Policy
    </synopsis>
    <typeRef>CEFailoverPolicyValues</typeRef>
  </component>
  <component componentID="11" access="read-write">
    <name>CEFTI</name>
    <synopsis>
     The CE Failover Timeout Interval in millisecs
    </synopsis>
    <typeRef>uint32</typeRef>
  </component>
  <component componentID="12" access="read-write">
    <name>FERestartPolicy</name>
    <synopsis>
      The FE Restart Policy
    </synopsis>
    <typeRef>FERestartPolicyValues</typeRef>
  </component>
  <component componentID="13" access="read-write">
    <name>LastCEID</name>
    <synopsis>
      The Primary CE this FE was last associated with
    </synopsis>
    <typeRef>uint32</typeRef>
  </component>
 </components>
<capabilities>
  <capability componentID="30">
    <name>SupportableVersions</name>
    <synopsis>
      the table of ForCES versions that FE supports
    </synopsis>
    <array type="variable-size">
     <typeRef>u8</typeRef>
    </array>
  </capability>
```

```
      <capability componentID="31">
        <name>HACapabilities</name>
        <synopsis>
          the table of HA capabilities the FE supports
        </synopsis>
        <array type="variable-size">
         <typeRef>FEHACapab</typeRef>
        </array>
      </capability>
    </capabilities>
    <events baseID="61">
      <event eventID="1">
        <name>PrimaryCEDown</name>
        <synopsis>
           The pimary CE has changed
        </synopsis>
        <eventTarget>
           <eventField>LastCEID</eventField>
        </eventTarget>
        <eventChanged/>
        <eventReports>
          <eventReport>
            <eventField>LastCEID</eventField>
          </eventReport>
        </eventReports>
      </event>
      <event eventID="2">
        <name>HAPrimaryCEDown</name>
                <synopsis>The primary CE has changed</synopsis>
                <eventTarget>
                  <eventField>LastCEID</eventField>
                </eventTarget>
                <eventChanged/>
                <eventReports>
                  <eventReport>
                    <eventField>CEID</eventField>
                       <eventField>LastCEID</eventField>
                  </eventReport>
                 </eventReports>
               </event>
    </events>
   </LFBClassDef>
  </LFBClassDefs>
</LFBLibrary>
```