**Informe Obligatorio 3**

**Redes de Computadoras 2012**

**INCO – Facultad de Ingeniería – UdelaR**

**Noviembre 2012**

Grupo 59

Álvaro Acuña – CI: 3826062-8

Gabriel Centurión – CI: 2793486-8

Germán Mamberto – CI: 3187102-8

Fernando Mangino – CI: 3621009-1

Tabla de contenido

[Objetivo 3](#_Toc339206233)

[Laboratorio de Routing y Forwarding (rip) 3](#_Toc339206234)

[Laboratorio de Routing y Forwarding (ospf) 3](#_Toc339206235)

[Laboratorio de Bridging (two-switches) 3](#_Toc339206236)

Objetivo

Laboratorio de Routing y Forwarding (rip)

Rip es un algoritmo de ruteo del tipo vector distancia

a) Instalación del laboratorio

Descargamos NetKit y los archivos correspondientes a VMWare

Levantamos en NetKit un servidor ssh para la transmisión de archivos desde el sistema operativo real

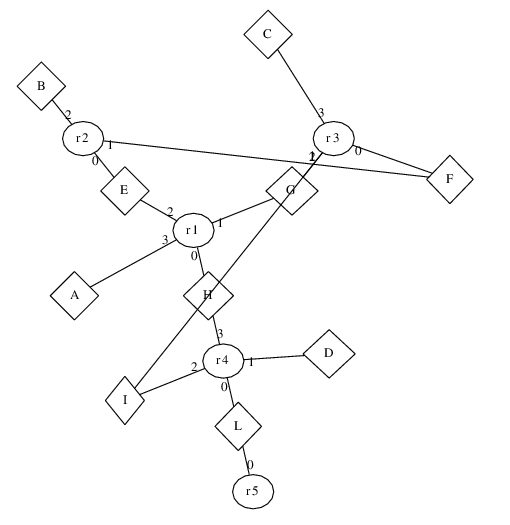
Copiamos netkit-lab\_rip.tar.gz y lo descomprimimos

Ejecutamos ltest para verificar su correctitud.

Nos pareció interesante la representación grafica que nos brinda linfo. Para ello hubo que instalar graphvis siguiendo los siguientes pasos:

* sudo apt-get update    // actualiza las dir de descargas
* sudo apt-get install graphviz  // instala la libreria graphviz
* luego parados en el directorio del lab en cuestion
  + linfo -m nombreGrafica.ps // genera un fichero postcript
  + gs nombreGrafica.ps

Obtenemos la topologia generada con GraphViz.



b)Captura de paquetes en las Interfaces de r1.

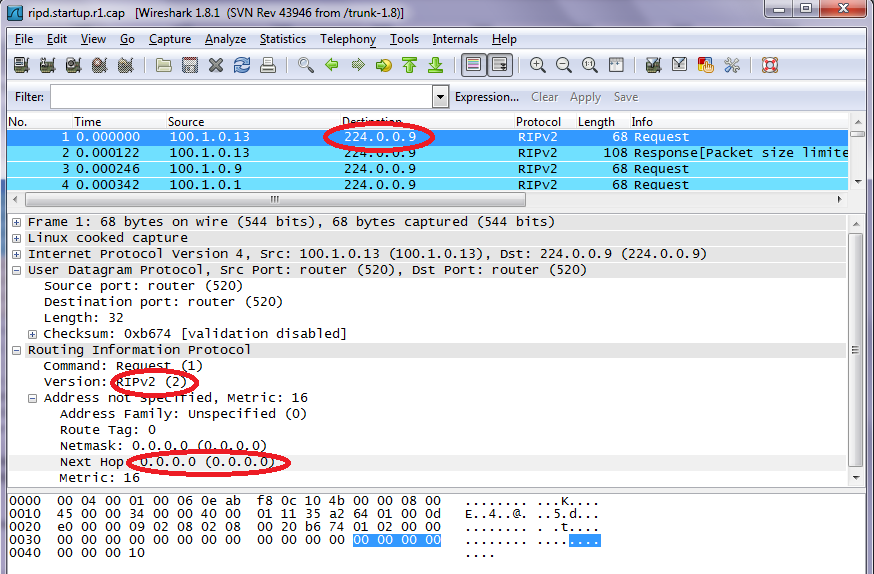
Con *” tcpdump -i any -w /hosthome/ripd.startup.r1.cap &* “ en la terminal correspondiente a r1 iniciamos la captura de paquetes antes de levanter el demonio zebra, liberando la terminal con & .

Analisis de los parámetros de tcpdump

* -i indica que interfaces queremos capturar al decirle “any” las toma todas
* -w nombreArchivo , nos permite indicarle el archivo de salida de la captura.

Luego levantamos el demonio de zebra en cada terminal con *“/etc/init.d/zebra start”*

c) Análisis del archivo ripd.startup.r1.



1. ¿Cuál es el contenido de los mensajes *Request*?

El contenido de los mensajes Request de RipV2 es

* Comando  
  Campo de 1 byte con valor 0x01 indicando que es un Request [http://www.networksorcery.com/enp/protocol/rip.htm#Command]
* Versión

Campo de 1 byte con valor 0x02 indicando que es RipV2

* Datos de la tabla de entradas de RipV2 (si corresponde)[ http://www.networksorcery.com/enp/protocol/rip.htm#RIPv2%20entry%20table]
  + Address family. 16 bits.
  + Route tag. 16 bits.

Este es un atributo asignado a una ruta que debe ser preservado. El uso previsto de esta etiqueta es proporcionar un método para separar rutas de las redes en el dominio de enrutamiento RIP, de rutas que pueden haber sido importados de un EGP (protocolo exterior a un sistema autónomo) o Otro IGP (protocolo interior a un sistema autónomo).

* + IPv4 address. 32 bits.
  + Subnet mask. 32 bits.

Esta es la máscara de subred en la que se puede aplicar a la dirección IPv4 para resolver la parte de red de la dirección. Si el campo se pone a 0, no se especifica la máscara de subred.

* + Next hop. 32 bits.

Indica la ip donde los paquetes para esta entrada deberían de ser transferidos.

* + Metric. 32 bits.

Contiene un valor del 1 to 15 que indica la actual métrica para el destino. El valor 16 indica que el destino no es alcanzable.

1. ¿Cuál es la dirección de destino de los mensajes *Request*? ¿Por qué?

La dirección de destino de los mensajes de Request es la 224.0.0.9.

Es una dirección Multicast particular del protocolo Rip versión 2.[ http://en.wikipedia.org/wiki/Multicast\_address]

Se utiliza para enviar información de routing a todos los routers RIP2 en un segmento de red.

Utilizando esta dirección los routers Rip puede coexistir con los routers que no son Rip puesto que estarían asociados a otra Multicast.

3. ¿Cuál es el *Next Hop* especificado en las entradas de un comando *Response*? ¿Por

qué?

El Next Hop especificado en las entradas de un comando Response es 0.0.0.0.

Esto es porque está indicando que el ruteo lo efectúa el router en cuestión y no a través de la subred lógica indicada.

4. Especifique cómo visualizar la información de mensajería de RIP utilizando la utilidad

debug del demonio ripd. Implemente el debug en **r1** y capture los mensajes en el

archivo /hosthome/ripd.log.r1.cap.

Para visualizar la información de mensajería de RIP utilizando la utilidad debug tenemos dos maneras de hacerlo. La primera es modificando el archivo ripd.conf agregando las líneas debug rip events , debug rip packet y debug rip zebra y reiniciando el demonio para que lso cambios surjan efecto.

La segunda es conectándose por telnet contra el demonio ripd. Esto es posible porque ripd es VTY (Virtual TeletYpe interface). Utilizando los comandos de debug de ripd [http://pf.itd.nrl.navy.mil/ospf-manet/quagga.pdf pag 31] [http://www.ige.unicamp.br/manuais/zebra/zebra.info.RIP\_Debug\_Commands.html]

levantamos los modos debug necesarios (events, packet, zebra) y cambiando la configuración [http://pf.itd.nrl.navy.mil/ospf-manet/quagga.pdf pag 24] indicamos en que archivo realizar la captura.

En nuestro primer intento nos encontramos con que debemos crear y dar permisos al archivo donde vamos a realizar la captura. Reintentamos luego de creado el archivo y brindados los permisos.

telnet localhost ripd // levantamos la consola contra ripd en r1

zebra // ingresamos password

enable // habilitamos usuario con privilegios

zebra // ingresamos password

debug rip packet // le indicamos al debug que nos loguee los paquetes rip con info detallada origen, nro de puerto etc.

debug rip zebra // le indicamos al debug que nos loguee la comunicacion entre ripd y zebra

debug rip events // le indicamos al debug que nos loguee los eventos rip , envios recepcion de paquetes timers y cambios en la interface.

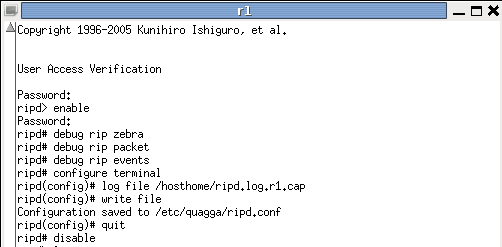
configure terminal // ingresamos a modo configuracion

log file /hosthome/ripd.log.r1.cap // indicamos el Nuevo archive de logueo

quit // cerramos modo configuracion

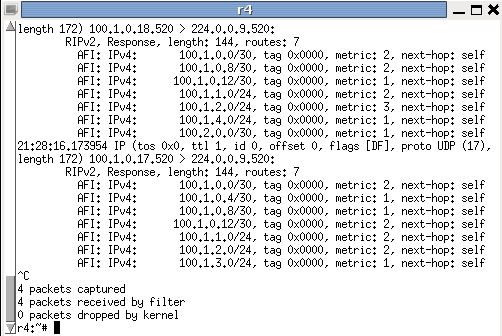
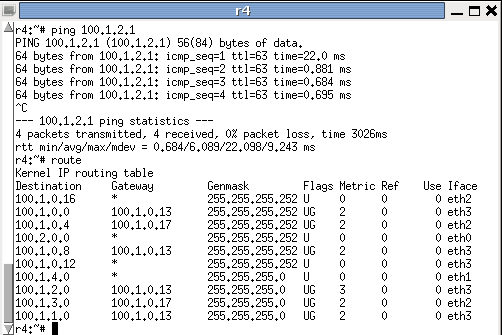
disable // deshabilitamos usuario con privilegios

realizamos una prueba ejecutando el comando write file con la intención de comitear los cambios en el archivo ripd.conf del laboratorio , sin embargo los cambios se guardaron /etc/quagga/ripd.conf , lo que creemos que es una copia de la versión de laboratorio , por lo que los cambios realizados por los comandos anteriores no se verán reflajados en ripd.conf del laboratorio.

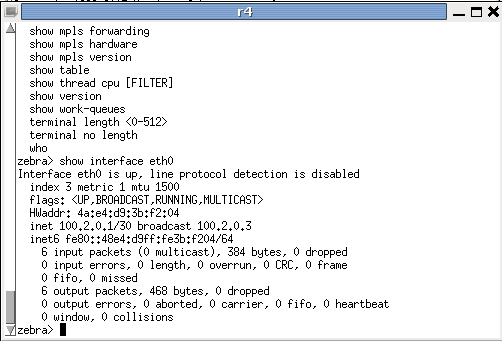
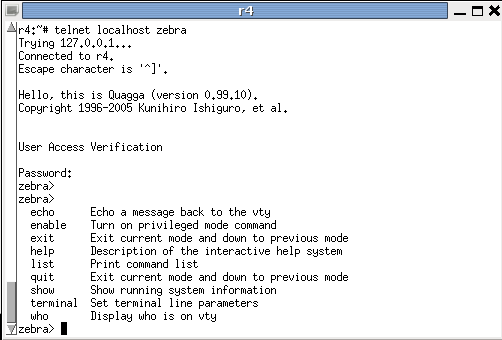


d) Siga adelante con el laboratorio según lo especificado en [6].

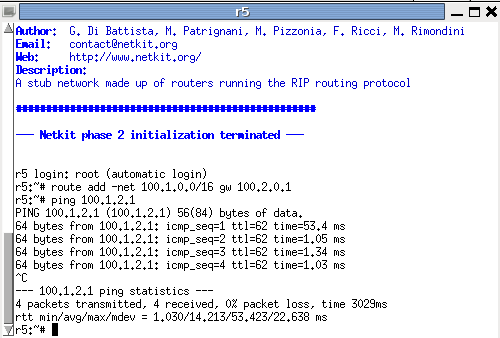
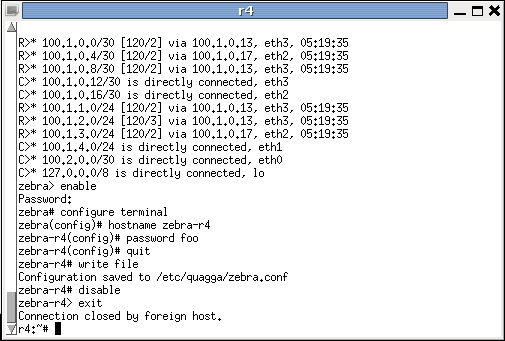
- Verificando conectividad - Sniffing con Tcpdump



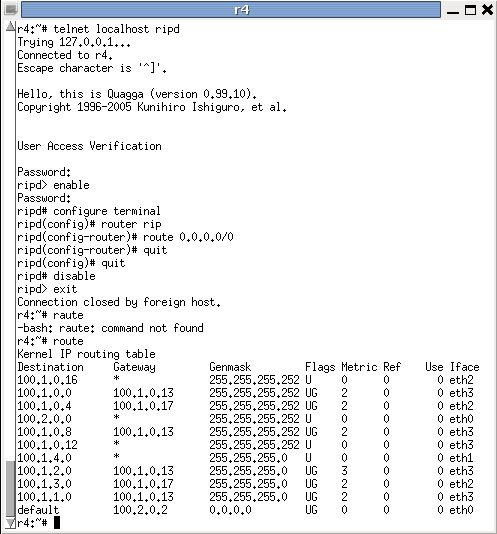
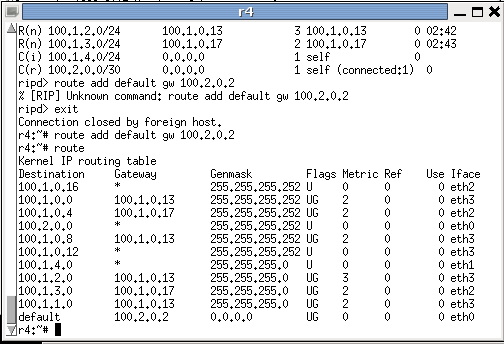
- Conectando con Zebra telnet - Inspección de interface



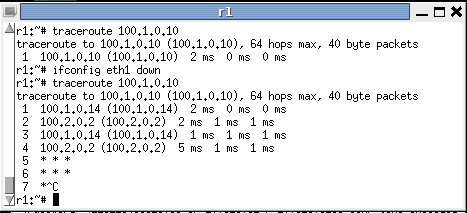
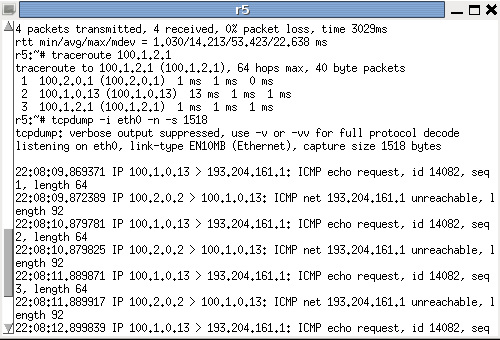
- Configurando Zebra - Agregando ruta estático a R5



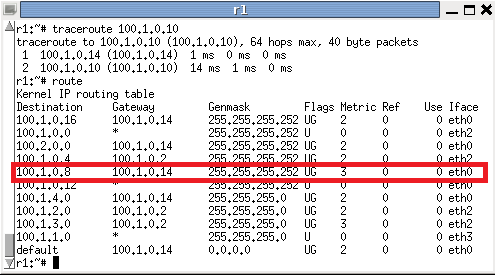
- Conf ruta por defecto en r4 - Propagamos la ruta por defecto



- Recepcion de echo request en r5 - Eliminamos eth1 de r1



- Actualización de tabla ruteo



e) Conéctese al demonio zebra en **r3** con el comando telnet localhost zebra.

Utilizando el comando sh ip route identifique los *Next Hop* para las redes 100.1.0.12/30 y 100.1.0.0/30. Identifique la distancia administrativa y la métrica en los números entre corchetes [120/2]; comente.

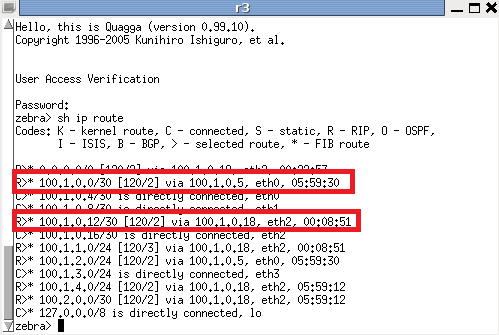
El next hop lo identificamos mirando en “via 100.1.0.18” que indica la dirección del router para el próximo salto

Para 100.1.0.12/30 es 100.1.0.18

Para 100.1.0.0/30 es 100.1.0.5

[120/2] significa la distancia administrativa y la distancia métrica.

En Rip la distancia administrativa por defecto es 120 y el 2 significa que hay 2 hops entre origen y destino.



La distancia administrativa califica la confiabilidad de la información de enrutamiento, se utiliza como criterio de selección cuando el dispositivo tiene multiples rutas hacia el mismo destino. La mejor ruta es la que tenga menor distancia administrativa.[ http://librosnetworking.blogspot.com/2010/03/distancia-administrativa.html].

La distancia métrica es la cantidad de hops para llegar llegar a destino.

f) ¿Por qué es necesario agregar una ruta estática a 100.1.0.0/16 en **r5**?

Es necesario para alcanzar desde r5 a la subred 100.1.0.0/16. O sea siendo r5 un router externo a nuestra red es necesario indicarle el Gateway para acceder a 100.1.0.0/16 a través de r4.En el laboratorio se agregó dicha ruta estatica.

g) ¿Por qué se instala una ruta por defecto en **r4**? ¿Por qué se redistribuye en RIP?

Se instala una ruta por defecto en r4 para que el tráfico que reciba el router que no sea para la red la reenvíe por dicha ruta. Se redistribuye para que los demás routers conozcan la ruta por defecto, cargándola en sus tablas de ruteo.

h) Especifique las modificaciones que debería realizar en la configuración **r5** para que el

comando ping 193.204.161.1 ejecutado en cualquiera de los routers **r1**..**r4** sea

exitoso. Impleméntelo y compruebe. Almacene los archivos del laboratorio modificado en

el directorio IRC-lab\_rip.

Aprovechando la ruta estática a r5 desde r4 y la propagación realizada en el laboratorio restaría por agregar en r5 una alias para la interfaz (eth0:0) y configurarle la ip 193.204.161.1

Utilizaremos el siguiente comandos en la terminal correspondiente a r5.

ifconfig eth0:0 193.204.161.1 // para configurar el alias a la interfaz con la ip 193.204.161.1

Laboratorio de Routing y Forwarding (ospf)

Dado el ejemplo anterior (con las modificaciones especificadas en el apartado h), cambie el

protocolo de *routing* de RIP a OSPF con una única área de *backbone*.

a) ¿Se puede hacer desde en línea (sin re-levantar el laboratorio)? Justifique su respuesta.

Almacene los archivos del laboratorio modificado en el directorio IRC-lab\_ospf.

b) Una vez que cambie a OSPF, siga las mismas instrucciones detalladas en [6] para el

laboratorio RIP, pero ahora sabiendo que el protocolo de *routing* utilizado es OSPF.

c) Antes de levantar el demonio zebra inicie una captura de paquetes completos con

Página 2 de 4

Redes de Computadoras Obligatorio 3-2012

tcpdump en TODAS las interfaces de **r1** en el archivo

/hosthome/ospfd.startup.r1.cap; una vez iniciada la captura, proceda a levantar

los demonios zebra en **r1**, **r2**, **r3**, **r4**. Luego de levantados todos los demonios, espere 30

segundos y cierre el archivo de captura.

d) Abra el archivo ospfd.startup.r1.cap con wireshark y consulte en [8] para

responder lo siguiente:

1. Describa las características de los mensajes *LS Request, LS Update y LS*

*Acknowledge.*

2. ¿Cuál es la dirección de destino de los mensajes *Hello*? Por qué?

3. ¿Cuáles son los *Router ID* para **r1**..**r4**?

4. Consulte el archivo de log del ospfd en **r1** y comente sobre el proceso *DR-Election*.

¿Cuál es el rol del *DR* y el *Backup-DR*?

e) Siga adelante con el laboratorio según lo especificado en [6].

f) ¿Cómo se redistribuye la ruta por defecto hacia **r5** en OSPF? Verifique que se propague

efectivamente en **r1**..**r3**.

g) Conéctese al demonio ospf en **r3** con el comando telnet localhost ospfd.

Utilizando el comando sh ip ospf route identifique las rutas externas de OSPF tipo 2

(E2). Comente el resultado.

h) Conéctese al demonio zebra en **r3** con el comando telnet localhost zebra.

Utilizando el comando sh ip route identifique los *Next Hop* para las redes

100.1.0.12/30 y 100.1.0.0/30. Compare con el resultado obtenido para el protocolo RIP y

comente.

Laboratorio de Bridging (two-switches)

La configuración y presentación detallada de este laboratorio se encuentra disponible en la *Wiki*

*de Netkit → Official Labs → Advanced Topics*.

a) Instale el laboratorio de *Bridging* [9] e inicie la ejecución.

b) Vincule el contenido del archivo lab.conf con la topología de red que aparece en la

diapositiva 3 de la presentación [10].

c) Utilizando el comando ifconfig verifique que el plan de numeración (tanto IP como

MAC) es el documentado en la misma diapositiva.

d) Modifique el plan de numeración IP de toda la red de forma que pertenezcan al prefijo

172.31.0.0/24 manteniendo el valor del último byte de cada dirección IP según el plan de

numeración que aparece en la diapositiva 3.

e) Modifique las direcciones MAC de toda la red de forma que se emulen tarjetas de red

fabricadas por la empresa *Speakercraft Inc.*

f) Documente los comandos utilizados para realizar las dos partes anteriores.

g) A partir de aquí, documente y analice los cambios relevantes en el contenido de las tablas

de direcciones MAC de los switches de la topología, antes y después de cada comando

ejecutado.

1. Genere tráfico desde **pc2** hacia **pc3** (mediante el comando ping) pero previamente y

utilizando el comando tcpdump, capture el tráfico en **pc1** y en **pc3** en los archivos

/hosthome/ping.pc1.g1.cap y /hosthome/ping.pc3.g1.cap

respectivamente. Luego de unos 15 segundos de estar generando tráfico, detenga el

ping y las capturas. Analice con wireshark las capturas realizadas y explique el

comportamento observado.

2. ¿Cuál es el tiempo de vida (*lifetime*) por defecto de las entradas en las tablas de

direcciones MAC de ambos switches? Configure su valor en 3 segundos.

3. Luego que las entradas correspondientes a las direcciones MAC de los pcs hayan

desaparecido de las tablas de los switches, vuelva a generar tráfico desde **pc2** hacia

**pc3** (utilizando nuevamente el comando ping) pero ahora forzando que cada

mensaje *ICMP echo* (*Request*) se envíe cada 7 segundos; previamente y utilizando

nuevamente el comando tcpdump, capture el tráfico en **pc1** y en **pc3** en los archivos

/hosthome/ping.pc1.g3.cap y /hosthome/ping.pc3.g3.cap

respectivamente. Luego de unos 30 segundos de estar generando tráfico, detenga el

ping y las capturas. Analice con wireshark las capturas realizadas y explique el

comportamiento observado. Almacene los archivos del laboratorio modificado en el

directorio IRC-lab\_two-switches.