Redes de computadoras 2015
Obligatorio 3

Grupo 48
Alonzo Fulchi, Emiliano CI: 4.460.443-0
Damiano Izzi, Rodrigo CI: 4.260.487-0
Flores Saavedra, Martin CI: 4.656.307-0

Tutor: Eduardo Grampin

1. Objetivos del obligatorio

Comprender los conceptos básicos de forwarding, routing y switching, a través de un hipotético caso de aplicación. El problema en cuestión es proporcionar un servicio web de Alta Disponibilidad. El servicio se implementa mediante dos servidores front-end que actuarán como balanceadores de carga de múltiples servidores back-end (o granja de servidores). Además, cada conjunto de servidores (front-end y su correspondiente conjunto de back-ends) se instalarán en datacenters geográficamente separados, y cada datacenter contará con su propia conexión a internet mediante ISPs diferentes

2. Implementación del servicio

2.1. Balanceo de carga utilizando DNS

Para lograr el balanceo de carga utilizando DNS se deben configurar los Name Servers Autoritativos de forma que cuando se quiera averiguar la dirección IP correspondiente a un nombre de dominio, el DNS nos responda con diferentes registros de tipo A dependiendo, en nuestro caso del cliente que realice la consulta. En el laboratorio de ejemplo de netkit utilizan lo que se conocen como vistas, donde cada vista contiene diferente información. El acceso a estas vistas depende de quien haya realizado la consulta. A continuación mostramos un ejemplo.

```
view "US" {
   match-clients { 10.0.0.0/24; };
   zone "." {
         type hint;
         file "/etc/bind/db.root";
   };
   zone "web.com" {
         type master;
         file
"/etc/bind/db.web.com-us";
   };
   rrset-order { order random; };
};
```

Estas líneas son parte del archivo de configuración **named.conf** que se encuentra en el directorio **/etc/bind**. Suponiendo que el cliente quiere saber la direccion IP de **www.web.com**, en la linea **match-clients {...};** es donde se hace el filtrado de los

clientes que realizan la consulta dns, de forma que si la dirección IP del cliente matchea con el prefijo 10.0.0.0/24 la respuesta de la consulta se obtiene del archivo **db.web.com-us**, en este archivo se simula la base de datos del servidor dns, es decir, contiene los registros de tipo A para **www.web.com**. Para realizar el balanceo dns, lo que hace es configurar diferentes vistas, de forma que según la dirección IP del cliente, obtiene la respuesta a la consulta de diferentes archivos. En el ejemplo tiene dos vistas, una para los clientes de USA (10.0.0.0/24) y otra para los clientes de Europa (20.0.0.0/24).

En nuestro problema, nos solicitan implementar el servicio dns en R1 (servidor primario) y en R2 (servidor secundario), por esto entendemos que se solicita que R1 cumpla el rol de master y R2 el rol de slave, de forma tal que el servidor master envíe los registros al servidor slave de manera de simplificar el mantenimiento de los registros DNS sin duplicar la información. Esta configuración involucró inconvenientes con permisos y vistas, imposibilitando compartir los registros desde el master al servidor slave, a pesar de haber configurado los permisos para la transferencia mediante las opciones *allow-transfer* y *allow-notify*
La solución que implementamos fue quitando las vistas, y utilizando una única zona por servidor dns. Para manejar la preferencia en base al cliente que realiza la consulta utilizamos *sort-list*, donde se establece el orden en que se van a devolver los registros en base a la ip que realiza la consulta. Evitando la configuración de las vistas, la configuración del servidor slave no dio problemas de permisos (una vez fueron seteados correctamente).
La forma en que fue implementada la solución utilizando sort-list se muestra en el siguiente recuadro.

```
sortlist {
   { match-ip; {primera-preferencia; segunda-preferencia;}; };
};
```

Esta configuración es parte de las opciones que se configuran en el archivo **`/etc/bind/named.conf`** de los routers que implementan el servicio dns.

2.2. Balanceo de carga con Web Switches

El balanceo de carga web se implementó utilizando la herramienta iptables. Iptables es una herramienta de administración de filtrado de paquetes ipv4/ipv6, este actúa sobre tablas ip mediante reglas en el kernel de Linux.
Para lograr el balanceo de carga se aplicaron las siguientes reglas:

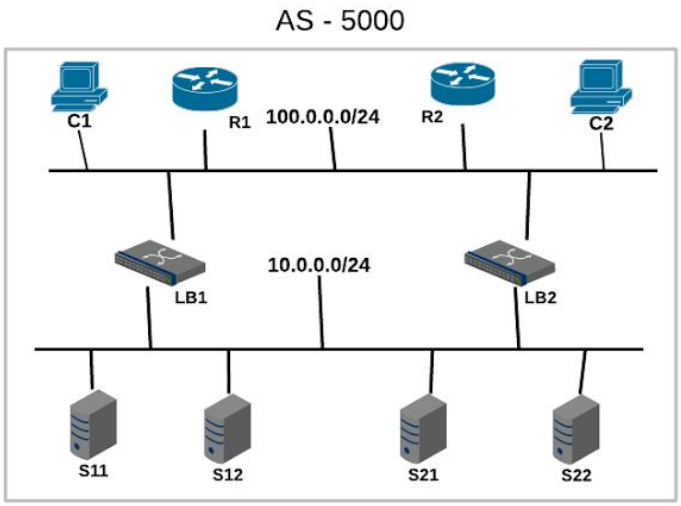
```
iptables --table nat --append PREROUTING --destination 100.0.0.5 --match
statistic --mode nth --every 2 --jump DNAT --to-destination 10.0.0.2
iptables --table nat --append PREROUTING --destination 100.0.0.5 --jump
DNAT --to-destination 10.0.0.1
```

Ambas reglas redireccionan con la IP del parámetro --destination a la IP del parámetro --to-destination.
La primera regla se aplica cada dos paquetes (indicado por los parámetros -mode nth --every 2), de esa forma se logra el balanceo round robin.

Además, se agrega una regla para que los servidores web vean el tráfico como si viniera desde el balanceador de carga. Sin esta regla los servidores web deberían responder a través de uno de los balanceadores de carga. La regla es la siguiente:

```
iptables --table nat --append POSTROUTING --source 100.0.0.0/24
--destination 10.0.0.0/24 --jump MASQUERADE
```

2.3. Configuración del AS5000



En esta sección explicamos las configuraciones que fueron necesarias para replicar el bloque AS-5000. Primero mencionamos los diferentes comandos que fueron necesarios para configurar cada nodo, asi como los archivos de configuración que se crearon.

Para la configuración IP utilizamos el siguiente comando:
`ifconfig ethX <dirIP> netmask <mask> broadcast <bcast> up`
Con este comando asignamos a la interfaz de red **`ethX`** la dirección IP **`dirIP`** junto con su dirección de broadcast **`bcast`** y su máscara de red **`mask`**.

Para configurar las direcciones MAC utilizamos el siguiente comando:
`ifconfig ethX hw ether <dirMAC>`
Con este comando asignamos a la interfaz de red **`ethX`** la dirección MAC **`dirMAC`**.
Estos comandos se encuentran en los archivos **`.startup`** de cada nodo del bloque.
En particular, en los **`.startup`** de los nodos LB1 y LB2 se incluyen los comandos iptables para configurar el balanceo de carga utilizando web switches que mencionamos antes.
En los **`.startup`** de los nodos que implementan el servicio dns fue necesario también incluir el comando **`/etc/init.d/bind start`**, que se encarga de levantar el servidor dns.
Por último, en los **`.startup`** de los nodos que implementan los servidores web fue necesario incluir el comando **`/etc/init.d/apache2 start`**, el cual inicia la ejecución del servidor web en el nodo.

En esta parte solo tenemos dos dominios de colisión, **VLAN1 (100.0.0.0/24)** donde están conectados los routers R1 y R2, los clientes C1 y C2 y los balanceadores de carga LB1 y LB2.
Y **VLAN2** (10.0.0.0/24) donde están conectados los servidores S11, S12, S21 y S22 y también los balanceadores de carga LB1 y LB2. (Obs, los LB necesitan tener configuradas dos interfaces de red, una para cada dominio de colisión)
A continuación se muestra una tabla con las direcciones IP asignadas a cada nodo, junto con sus dirección de broadcast y máscara de red correspondiente.

Host	Interfaz	Direccion IP	Direccion MAC	Mascara de red	Broadcast
C1	eth0	100.0.0.1	00::01:04	255.255.255.0	100.0.0.255
C2	eth0	100.0.0.2	00::02:04	255.255.255.0	100.0.0.255
R1	eth0	100.0.0.3	00::00:01	255.255.255.0	100.0.0.255
R2	eth0	100.0.0.4	00::00:03	255.255.255.0	100.0.0.255
LB1	eth0	100.0.0.5	00::00:05	255.255.255.0	100.0.0.255
	eth1	10.0.0.5	00::00:06	255.255.255.0	10.0.0.255
LB2	eth0	100.0.0.6	00::00:07	255.255.255.0	100.0.0.255
	eth1	10.0.0.6	00::00:08	255.255.255.0	10.0.0.255
S11	eth0	10.0.0.1	00::00:11	255.255.255.0	10.0.0.255
S12	eth0	10.0.0.2	00::00:12	255.255.255.0	10.0.0.255
S21	eth0	10.0.0.3	00::00:21	255.255.255.0	10.0.0.255
S22	eth0	10.0.0.4	00::00:22	255.255.255.0	10.0.0.255

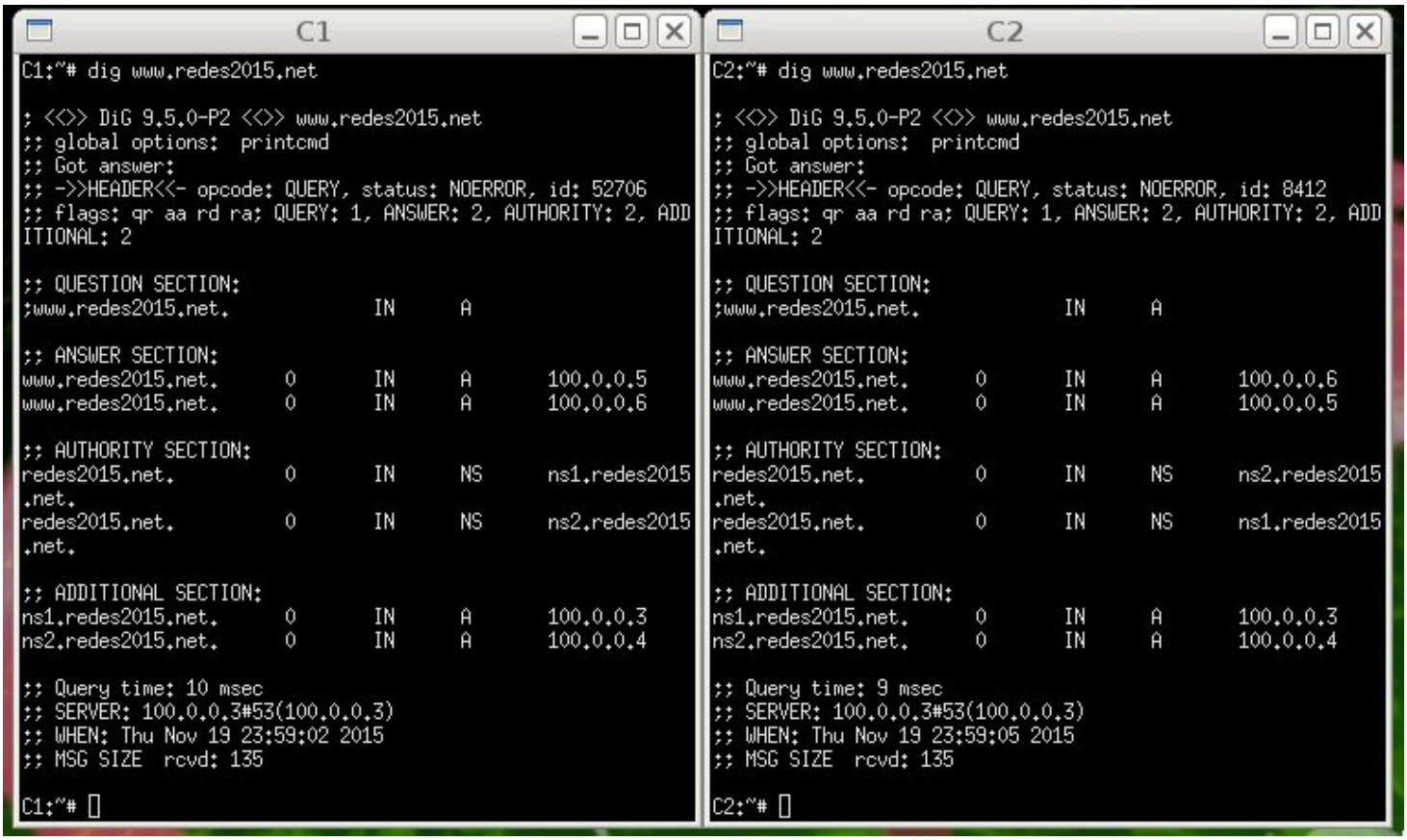
Obs. si bien las direcciones MAC no son necesarias para esta parte, optamos por configurarlas desde el principio por un tema de comodidad.

En cuanto a las configuraciones particulares de cada nodo:
En los clientes agregamos en el directorio **`/etc/`** un archivo **`resolv.conf`** donde especificamos las direcciones IP de los nameservers, es decir las direcciones IP de donde dirigir las consultas dns. en ambos clientes primero se pone la dirección IP de R1 y segundo la dirección IP de R2, de forma que primero dirijan sus consultas dns a R1, y en caso que este no les responda, las dirijan a R2.
En cada servidor web agregamos en el directorio **`/var/www/`** el archivo **`index.html`** el cual implementa una página web que despliega el mensaje "**`Llegaste al servidor Sxx`**" siendo xx=11, 12, 21, 22
Por último, como se menciono antes, en los routers R1 y R2, en el directorio **`/etc/bind/`** se agregan los archivos **`db.redes2015.net`** y **`named.conf`** utilizados para configurar los servidores DNS de forma que R1 sea el servidor primario y R2 el secundario. (**Obs.** en R2 solo fue necesario agregar el archivo **`named.conf`**).

2.4. Pruebas con Clientes

Las pruebas realizadas en esta parte del obligatorio intentan verificar que el balanceo de carga de los LB funciona correctamente, asi como los servidores DNS. Además, en los servidores DNS se debe verificar que cuando el servidor primario (**R1**) se cae, los clientes continúan obteniendo las respuestas del servidor secundario (**R2**).

Primero que nada comprobamos la disponibilidad de los servidores dns ejecutando el comando dig contra la url www.redes2015.net. Ya en esta consulta podemos verificar que para cada cliente se obtienen respuestas diferentes, donde se cumple con la prioridad requerida en la letra, es decir que el cliente C1 prefiera acceder por LB1 y el cliente C2 prefiera acceder por LB2. Se puede apreciar que en la respuesta para el cliente C1 se ofrece primero la dirección IP del LB1 y en la respuesta del cliente C2 se ofrece primero la dirección IP del LB2.



C1 obtiene como primer respuesta a LB1 **C2 obtiene como primer respuesta a LB2**

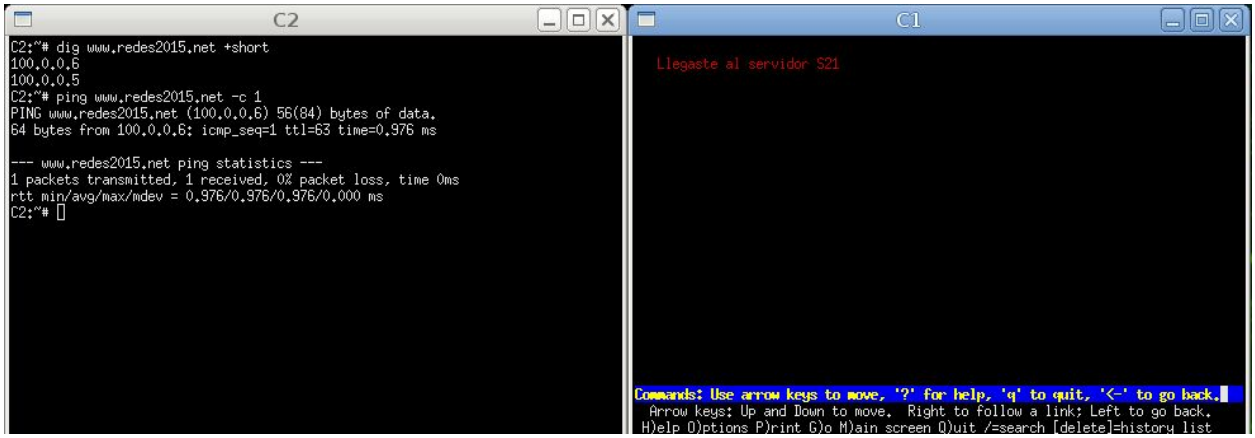
Se comprobó que en caso de bajar el servidor primario las consultas DNS son dirigidas al servidor secundario, para esto se realizó una captura de tráfico en ambos clientes donde se puede apreciar que la dirección IP de origen de las respuestas DNS es la dirección IP del router R2 (100.0.0.4). Ver capturas respR2_DNS_C1.cap y respR2_DNS_C2.cap entregadas en la carpeta anexo aplicando el filtro dns.

También se realizó una prueba del balanceo dns a nivel de aplicación:

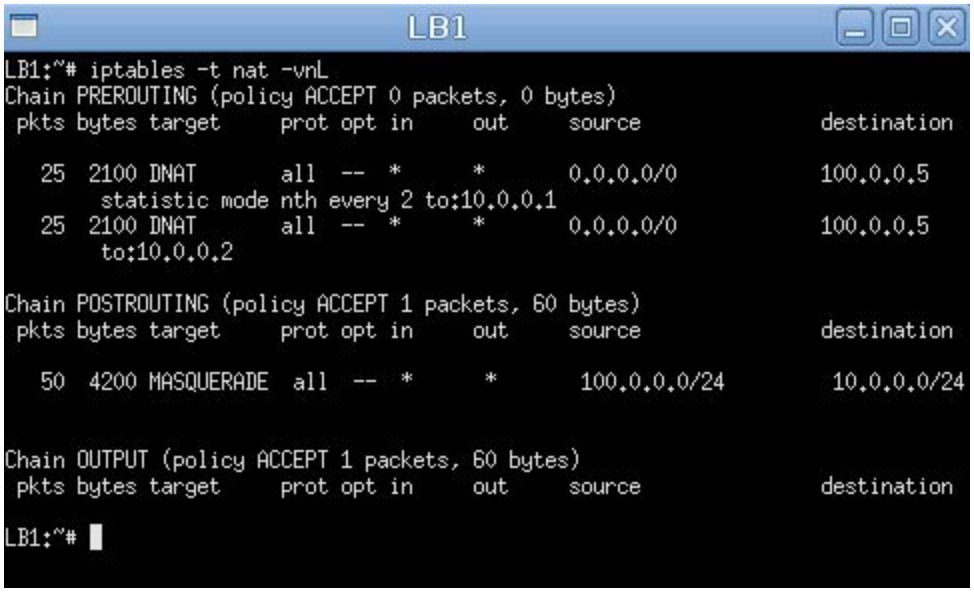
Apagamos la máquina LB1 con el comando *halt*

Desde C1, ejecutamos *lynx www.redes2015.net*

El resultado obtenido es que la respuesta llega desde S21 y S22. Cabe destacar que al realizar consultas mediante *ping www.redes2015.net* da como resultado destino inalzanzable, esto se debe a que ping, toma la primer ip devuelta por dns y envía el paquete ICMP, que al estar caído el LB1 da como resultado host inalcanzable. En cambio aplicaciones como lynx y wget en caso de no poder establecer conexión TCP con una ip, intentan con otra de la lista.



Para el balanceo de carga en LB1 y LB2, se comprobó que al recargar la página en el navegador lynx, se desplegaran los mensajes correspondientes, es decir, cuando accedo con el cliente C1, se muestran los mensajes “Llegaste al servidor S11” y “Llegaste al servidor S12” y cuando accedo con el cliente C2 se muestran los mensajes “Llegaste al servidor S11” y “Llegaste al servidor S11”. Para cada cliente los mensajes se deben ir alternando.
También comprobamos en los LB, con el comando **`iptables -t nat -vnL`** la cantidad de paquetes en los que se aplicó cada regla y que fuera consistente, es decir que si se recibieron 20 paquetes en el LB2, 10 fueron redireccionados al servidor S21 y 10 al servidor S22.
A modo de ejemplo a continuación se muestra el resultado de ejecutar el comando mencionado en LB1.



Por último, utilizando el comando traceroute en los clientes, vemos que llegan a los servidores web en dos pasos. Cabe destacar que ambos pasos muestran la misma dirección IP y la razón es porque en los balanceadores de carga LB se modifica la respuesta de los servidores web.

3. Implementación de la conectividad IP

3.1. Numeración IP

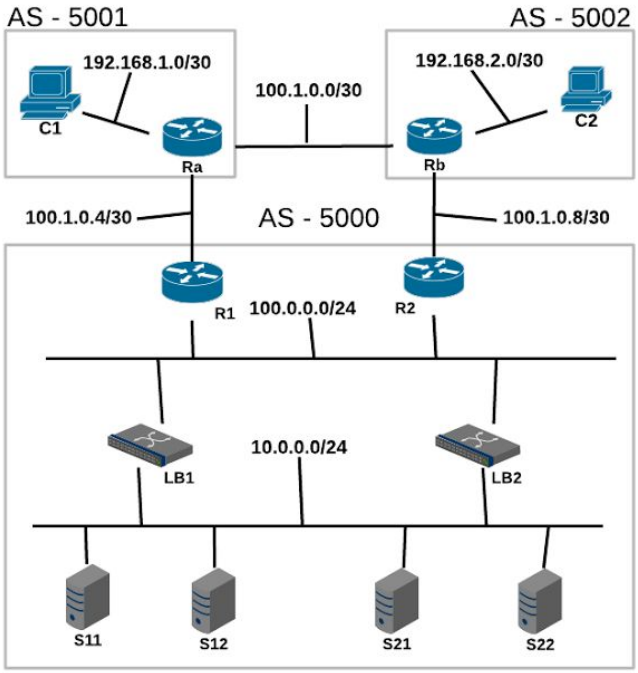
Para esta parte se agregan 5 dominios de colisión, uno para conectar al router **Ra** del AS-5001 con el router **Rb** el AS-5002 con el prefijo **100.1.0.0/30**, uno para conectar al router **R1** del AS-5000 con el router **Ra** del AS-5001 con el prefijo **100.1.0.4/30**, uno para conectar al router **R2** del AS-5000 con el router **Rb** del AS-5002 con el prefijo **100.1.0.8/30**, uno dentro del AS-5001 para conectar al cliente **C1** con el router **Ra** con el prefijo **192.168.1.0/30** y por ultimo uno dentro del AS-5002 para conectar al cliente **C2** con el router **Rb** con el prefijo **192.168.2.0/30**. Elegimos usar prefijos **/30** para estas conexiones ya que para conexiones punto a punto alcanzaría.

A continuación mostramos una tabla con la numeración IP de los diferentes ASs.
AS-5000 AS-5001 AS-5002

Host	Interfaz	Direccion IP	Direccion MAC	Mascara de red	Broadcast
C1	eth0	192.168.1.1	00::01:04	255.255.255.252	192.168.1.3
Ra	eth0	192.168.1.2	00::01:01	255.255.255.252	192.168.1.3
	eth1	100.1.0.6	00::01:02	255.255.255.252	100.1.0.7
	eth2	100.1.0.1	00::01:03	255.255.255.252	100.1.0.3
C2	eth0	192.168.2.1	00::02:04	255.255.255.252	192.168.2.3
Rb	eth0	192.168.2.2	00::02:01	255.255.255.252	192.168.2.3
	eth1	100.1.0.10	00::02:02	255.255.255.252	100.1.0.11
	eth2	100.1.0.2	00::02:03	255.255.255.252	100.1.0.3
R1	eth0	100.0.0.3	00::00:01	255.255.255.0	100.0.0.255
	eth1	100.1.0.5	00::00:02	255.255.255.252	100.1.0.7
R2	eth0	100.0.0.4	00::00:03	255.255.255.0	100.0.0.255
	eth1	100.1.0.9	00::00:04	255.255.255.252	100.1.0.11
LB1	eth0	100.0.0.5	00::00:05	255.255.255.0	100.0.0.255

	eth1	10.0.0.5	00::00:06	255.255.255.0	10.0.0.255
LB2	eth0	100.0.0.6	00::00:07	255.255.255.0	100.0.0.255
	eth1	10.0.0.6	00::00:08	255.255.255.0	10.0.0.255
S11	eth0	10.0.0.1	00::00:11	255.255.255.0	10.0.0.255
S12	eth0	10.0.0.2	00::00:12	255.255.255.0	10.0.0.255
S21	eth0	10.0.0.3	00::00:21	255.255.255.0	10.0.0.255
S22	eth0	10.0.0.4	00::00:22	255.255.255.0	10.0.0.255

Los dominios de colisión dentro del AS-5000 se mantienen igual que en la primera parte del obligatorio. La única diferencia es que para los routers **R1** y **R2** se agregan nuevas interfaces de red por las cuales se conecta al AS-5000 con los AS-5001 y AS-5002 como es solicitado.
En la siguiente imagen se puede ver la configuración.



3.2. Configuración de BGP

Para lograr la conectividad entre los diferentes AS's tuvimos que configurar en los routers **Ra**, **Rb**, **R1** y **R2** el protocolo de ruteo inter-AS **BGP**. Para esto seguimos como ejemplo los laboratorios de netkit netkit-lab_bgp-simple-peering y netkit-lab_bgp-announcement. En los ejemplos, en cada router se especifica que router de que AS es vecino asi como las subredes para las cuales ofrece conectividad a sus vecinos. Para esto utilizamos los siguientes comandos: **`router bgp <nroAS>`** indica que **`<nroAS>`** es el número de AS del router. **`network <net>`** indica que el router ofrece conectividad hacia la red **`<net>`**, es decir, garantiza la entrega de mensajes que tengan como destino un host de la red **`<net>.`**
`neighbor <dirIP> remote-as <nroAS>` se utiliza para indicar la dirección IP de un router vecino, ya sea que esté en el mismo AS o en otro AS.

Finalmente para completar la conectividad entre AS's, se configuró en los routers **R1** y **R2** el protocolo de ruteo intra-AS **RIP**. Esto fue necesario ya que no pudimos conseguir que se comunicaran las rutas aprendidas por BGP ante la pérdida de algún enlace. Para esto

```
! configuracion de rip
router rip
redistribute connected
! que distribuya por rip las rutas aprendidas por
redistribute bgp
! indico subred manejada
network 100.0.0.0/24
```

Dado que LB1 y LB2 pueden lograr conectividad hacia afuera del AS-5000 a través de R1 o R2, se decidió que ambos ejecuten el protocolo RIP para no depender de una ruta por defecto. Configurar una ruta por defecto, por ejemplo de LB1 a R1, implicaría que si R1 si "cae" LB1 ya no podría salir hacia afuera del AS-5000, cuando en realidad puede hacerlo a via R2.

3.3. Pruebas de conectividad

Aparte de probar que los clientes podían acceder a las páginas web que les correspondía por la ruta esperada, se plantearon las siguientes pruebas.

1. Dar de baja la interfaz que conecta Ra con R1, con la intención de que los paquetes enviados por C1 hacia LB1 pasen por Rb y luego R2. Se comprobó que el resultado fuera el esperado mirando la ruta con traceroute, también se comprobó que la tabla routing de Ra se haya actualizado (ver archivo Parte1Test1.png). Se comprueba que al volver levantar la interfaz el sistema se comporta como lo hacía originalmente.

2. Dar de baja la interfaz que conecta R2 con Rb. El objetivo es el mismo que en el caso anterior pero probando C2 nada más que la desconexión se da desde el lado del AS-5000. Los resultados son los esperados, ver archivo Parte2Test2.png
3. Dar de baja la interfaz de R1 que conecta al interior del AS 5000. El objetivo de la prueba es el mismo que en el primer caso. No pudimos comprobar este caso. Si bien la interfaz no aparece en la tabla de routing de R1, este sigue anunciando la subred 100.0.0.0/24 ya que esta configurado de esa forma. No encontramos una forma de configurar BGP para que R1 y R2 dejen de anunciar la subred a la que han perdido acceso.

4. Implementación de la Alta Disponibilidad

En esta parte nos centraremos solamente en el AS-5000, ya que es donde se realizan cambios en cuanto a la configuración y desde el punto de vista de los otros AS's no hay cambios. a continuación mostramos una tabla donde se muestra como quedo el direccionamiento IP y direcciones MAC de los balanceadores de carga LB1 y LB2 y de los switches SW11, SW12, SW21 y SW22.

Host	Interfaz	Direccion IP	Direccion MAC	Mascara de red	Broadcast
LB1	eth0.10	100.0.0.5	00::00:05	255.255.255.0	100.0.0.255
	eth0.20	10.0.0.5	00::00:06	255.255.255.0	10.0.0.255
LB2	eth0.10	100.0.0.6	00::00:07	255.255.255.0	100.0.0.255
	eth0.20	10.0.0.6	00::00:08	255.255.255.0	10.0.0.255
SW11	eth0	-	00::11:01	-	-
	eth1	-	00::11:02	-	-
	eth2	-	00::11:03	-	-
	eth3	-	00::11:04	-	-
SW12	eth0	-	00::12:01	-	-
	eth1	-	00::12:02	-	-
	eth2	-	00::12:03	-	-
	eth3	-	00::12:04	-	-
SW21	eth0	-	00::21:01	-	-

	eth1	-	00::21:02	-	-
	eth2	-	00::21:03	-	-
	eth3	-	00::21:04	-	-
SW22	eth0	-	00::22:01	-	-
	eth1	-	00::22:02	-	-
	eth2	-	00::22:03	-	-
	eth3	-	00::22:04	-	-

4.1. Esquema de VLANs

Para seguir la línea por la que veníamos trabajando en las partes anteriores, nos pareció correcto separar nuestra topología en dos VLANs, un poco como se veía venir dado el nombre de los hubs que se mostraban en las partes anteriores, VLAN1 y VLAN2.
Una VLAN es la que comunica los routers R1 y R2 con los balanceadores de carga LB1 y LB2 para la cual usamos la etiqueta de VLAN “10” (cables negros), y la otra VLAN es la que conecta los balanceadores de carga LB1 y LB2 con los servidores web S11, S12, S21 y S22 para la cual usamos la etiqueta de VLAN “20” (cables rojos). Los cables azules son los enlaces trunk, por los cuales puede viajar tráfico de las dos VLANs. En la siguiente sección se comenta más sobre estos enlaces.
En las pruebas realizadas que mencionamos más adelante se tomara alguna captura de tráfico para mostrar dónde se pueda apreciar la etiqueta de VLAN en la cabecera ethernet.

4.2. Enlaces trunks

Antes de mencionar cuales son los enlaces trunk, nos pareció buena idea comentar un poco algunos conceptos relacionados con las VLANs para asi entender mejor que cuales son los enlaces trunk.
Una VLAN es un método utilizado para configurar diferentes redes lógicas independientes en una misma red física, de forma que estos segmentos lógicos no intercambien datos de forma innecesaria, a menos que se realizará una configuración especial para permitir el intercambio de datos entre los diferentes segmentos lógicos, por ejemplo utilizando el método de router on a stick visto en el teórico.
Para poder lograr esta separación lógica, se utiliza el protocolo IEEE 802.1Q que permite que diferentes redes compartan de forma transparente el mismo medio físico sin que haya problemas de interferencia. Los enlaces por los que se produce este tipo de tráfico son los que se conocen como enlaces trunks. Hablando mal y pronto, se podría decir que son los enlaces

por los que viaja la etiqueta de VLAN. Para esta etiqueta, el protocolo añade 4 bytes a la cabecera ethernet los cuales son utilizados para identificar a qué VLAN pertenecen las tramas.

En nuestra configuración, los enlaces que conectan los balanceadores de carga LB1 y LB2 con los switches SW11 y SW21 respectivamente necesariamente van a tener que ser enlaces trunks ya que estos dispositivos son los que comunican las dos VLANs configuradas.
De forma de poder brindar un servicio de alta disponibilidad, los enlaces que interconectan los switches también deberían ser enlaces trunks, de forma de dar tolerancia de falla de un enlace de forma que el servicio siga funcionando.

En conclusión, los enlaces trunk de nuestra implementación son los enlaces que interconectan los switches y los enlaces que conectan los balanceadores de carga LB1 y LB2 con los switches SW11 y SW21 respectivamente. En la imagen adjunta más arriba serían los enlaces azules.

4.3. Cambios en la configuración de los dispositivos

Primero que nada, como los dispositivos que brindan el servicio de alta disponibilidad son switches (dispositivos de capa 2), en esta parte empiezan a tomar importancia las direcciones MAC. No es que antes no se usarán, es que en partes anteriores sólo trabajamos a nivel de capa 3. Si bien ya las habíamos configurado antes, en esta parte es donde empiezan a tener más utilidad y se realizan pruebas que apuntan a ver como los switches aprenden la información necesaria para encaminar las tramas.

Los dispositivos que cambiaron su configuración fueron los balanceadores de carga LB1 y LB2. El cambio que tuvieron es que antes tenían dos interfaces físicas, una para conectarse con lo que sería la VLAN1 y otra para conectarse con lo que sería la VLAN2. Ahora estos dispositivos solo cuentan con una interfaz física, a través de la cual pasa tráfico dirigido a las dos VLANs. Para solucionar esto configuramos dos interfaces virtuales y a cada una le asignamos las mismas direcciones IP que teniamos configuradas antes en cada balanceador, de forma que cada interfaz virtual pertenece a una VLAN diferente.

Por otro lado se agregaron los dispositivos SW11, SW12, SW21 y SW22. Estos dispositivos solo llegan a capa 2 por lo que no conocen de direcciones IP, por eso que es que ahora empiezan a tener importancia las direcciones MACs de los dispositivos.
En estos dispositivos fue necesaria la diferenciación del tráfico de las VLANs de forma que no hubiera interferencia, es decir, se utilizó el concepto de bridge para aislar el tráfico de cada VLAN de forma que las tramas solo sean replicadas por las salidas que tiene sentido replicar.

Para las configuraciones de esta parte utilizamos los siguientes comandos:
Con el comando **`vconfig add ethX tagVLAN`**, con este comando se crea la interfaz virtual **`ethX.tagVLAN`** con la cual después utilizando el comando **`ifconfig`** podemos asignar una

direccion IP a la misma. De esta manera es que configuramos las dos interfaces virtuales, cada una perteneciendo a una diferente VLAN.

En los switches fue necesario utilizar el concepto de bridges para aislar el tráfico de cada VLAN de forma que no fuera replicado por interfaces de forma innecesaria.
Para configurar los bridges utilizamos el comando **`brctl addbr <br>`**, de esta manera se crea un bridge con la etiqueta **`br`**, nos queda agregar las interfaces correspondientes en los bridges, para esto usamos el comando **`brctl addif <br> <ethX>`**, con esto agregamos la interfaz **`ethX`** al bridge **`br`**, independientemente de si la interfaz **`ethX`** es una interfaz física (eth0) o virtual (eth0.10).
Por último con el comando **`brctl stp br10 on`** se levanta el protocolo STP el cual fue necesario utilizar para evitar los ciclos de redundancia en la topología, es decir, para evitar que una trama no quedara en un loop infinito de reenvios en el anillo de switches. El protocolo STP consigue crear un árbol de cubrimiento de la topología, evitando los ciclos. Este árbol de cubrimiento se consigue mediante el intercambio de mensajes entre los dispositivos donde se ponen de acuerdo en un nodo raíz y luego deciden por cuál interfaz llegan a ese nodo raíz con menor costo (puertos raíz).
Para evitar los ciclos de redundancia, con el protocolo se marcan algunos enlaces como no activos, de forma que cuando una trama llega por ese enlace, esa trama es descartada. Solo se procesan las tramas enviadas por el mismo protocolo. Cabe mencionar que este árbol de cubrimiento es independiente para cada bridge configurado en los switches, es decir, se configura un árbol activo para la VLAN 10 y otro árbol activo para la VLAN 20.

4.4. Pruebas del sistema

Las pruebas realizadas en esta parte apuntan más que nada a probar la alta disponibilidad del sistema por un lado, y por otro lado ver que el tráfico de una VLAN no llegue a los nodos de la otra VLAN. Esto es porque desde el punto de vista del AS-5001 y del AS-5002 no hay cambios en el funcionamiento del sistema, por lo que los resultados de las pruebas deberían ser los mismos que en las partes anteriores.

Para ver el funcionamiento de las tablas MAC en los switches, tomamos una captura del tráfico en SW22, SW21 y S22 luego de ejecutar un ping desde S21 hacia S22 (ver archivos tablasMacSW22.cap, tablasMacSW21.cap y tablasMacS22.cap respectivamente). En las capturas se ve como S1 tiene que mandar una trama ARP que se reciben SW21 y SW22 y S22. En la respuesta se ve como S22 no tiene que averiguar la MAC de respuesta ya que la recibió un mensaje (el echo request de S21).

Como prueba básica realizamos una consulta http mediante lynx a www.redes2015.net desde el cliente C1 con la infraestructura completa como está planteada en la letra del laboratorio realizando una captura en LB1. Es interesante ver la captura en LB1 pues es el host que realiza el cambio de vlan, es decir, el cambio de etiqueta vlan en la trama ethernet.

Analizando la captura entregada en *parte3-LB1-infraestructura-inicial.cap* con whireshark pudimos comprobar que efectivamente el correcto etiquetado. A modo de resumen con tshark obtuvimos:

ip.src	ip.dst	vlan
192.168.1.1	100.0.0.5	10
10.0.0.5	10.0.0.2	20
10.0.0.2	10.0.0.5	20
100.0.0.5	192.168.1.1	10

A efectos de comprobar el correcto funcionamiento de las tablas de direcciones mac de los switches tomamos una captura de pantalla (Ver tablasSwitchesPrePostPing.png) donde se pueden ver las direcciones MAC que están almacenadas al comienzo del laboratorio, y luego de ejecutar un ping desde S11 a S22.
Se puede apreciar que antes de ejecutar el comando ping, en las tablas de direcciones MAC de los switches SW12 y SW22 no están las direcciones MAC 00::00:11 y 00::00:22 qué son las direcciones MAC asignadas a los servidores S11 y S22. luego de ejecutar el comando link, estas direcciones MAC aparecen en las tablas de direcciones MAC.

Otras de las pruebas consistió en realizar una consulta HTTP desde el cliente C1 a www.redes2015.net habiendo dado de baja los enlaces de SW11 a SW12 (enlace1) y luego el enlace de SW12 a SW22 (enlace2). Se entregan capturas de ambas pruebas, las cuales alcanzaron el resultado esperado: cuando se bajó la interfaz eth2 de SW11, no se perdió conectividad de los servidores S11 y S12, pudiendo constatarse tanto en las capturas (carpeta *parte3SinSW11eth2*) como en la salida del comando wget. Cuando se bajó la interfaz eth1 de SW12, los servidores S11 y S12 perdieron conectividad, se puede apreciar en la captura (*parte3SinSW11eth2SW12eth1*) de LB1 que no puede establecer conexión TCP. Analizando las capturas de SW22, podemos ver que la conexión se establece a través de LB2; en la salida de wget se evidencia que la salida viene dada por S21 y S22.