

Equal-Cost Multi-path con FRRouting en una topología Fattree

13 de agosto de 2019

Índice

1. Objetivo	3
1.1. Topologías Fat-tree	3
1.2. Equal-cost Multi-path (ECMP)	4
1.3. FRRouting	5
1.4. Generador de topologías Fat-tree	5
2. Pruebas	7
2.1. Prueba 1	8
2.2. Prueba 2	10
3. Conclusión	14
Referencias	15

1. Objetivo

El objetivo del trabajo es lograr hacer funcionar la implementación de FRRouting de los protocolos IS-IS, OpenFabric y BGP en un data center emulado con mininet (utilizando una topología Fat-tree con $k=4$) y ver que funcione el ECMP (Equal-cost Multi-path) logrando el balanceo de la carga y la redirección de los paquetes cuando un enlace cae.

Para ello vamos a seguir los siguientes pasos:

1. Instalar FRRouting
2. Desarrollar una herramienta para generar topologías Fat-tree en mininet con k arbitrario y que utilice FRRouting
3. Levantar una topología Fat Tree con $k=4$
4. Configurar IS-IS y probar que funcione el Multi-Path
5. Configurar OpenFabric y probar que funcione el Multi-Path
6. Configurar BGP y probar que funcione el Multi-Path

1.1. Topologías Fat-tree

Las topologías del tipo Fat-tree se basan en las topologías del tipo Clos que fueron diseñadas por Charles Clos en la década del 50 para solucionar el problema de cómo manejar el rápido crecimiento de las redes de telefonía. Hoy el problema es el de cómo manejar el rápido crecimiento de las redes en los Data Centers.

En una red tipo Fat-tree todos los routers son iguales (misma cantidad de puertos), de bajo costo y se conectan de manera de tener una gran redundancia.

Una topología Fat-Tree se describe con un solo parámetro “ k ” (que debe ser mayor o igual a 4 y par) y tiene tres niveles de routers: core, aggregation y edge, y se conectan como en la Figura “Fattree con $k=4$ ”. Los routers de aggregation y edge se agrupan en “pods” de k routers cada uno.

El parámetro k define la cantidad de routers de cada tipo, la cantidad de puertos de cada uno de ellos, la cantidad de enlaces de redundancia entre routers y la cantidad de racks que se van a conectar a cada switch de la siguiente manera:

- Habrán k pods
- Cada pod tendrá k routers distribuidos en dos niveles de $k/2$ routers cada uno
- Cada switch edge se conectará a $k/2$ racks y $k/2$ routers aggregation
- Cada switch aggregation se conectará a $k/2$ routers edge y $k/2$ routers core
- Cada switch core se conectará a cada uno de los k pods

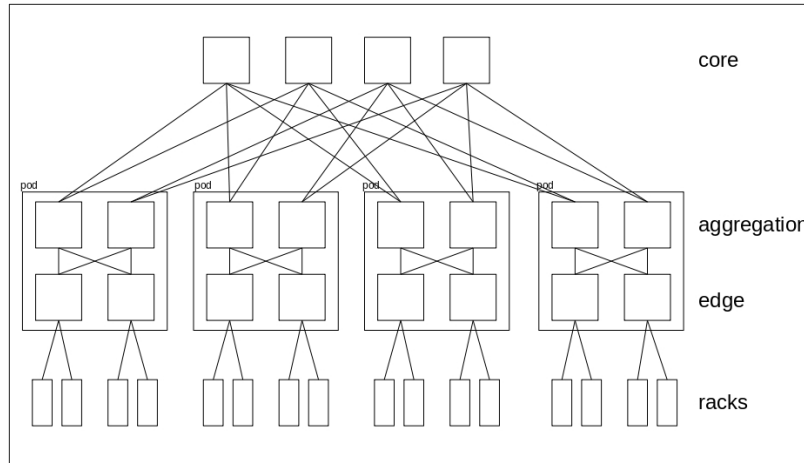


Figura 1: Fattree con $k=4$ generada

De esa manera quedan definidos k caminos entre cada uno de los racks (Ver Figura “ k caminos diferentes en Fattree con $k=4$ ”).

1.2. Equal-cost Multi-path (ECMP)

Los dispositivos de enrutamiento (incluidos los Linux) tienen la capacidad de manejar múltiples rutas de igual costo hacia un destino de red. Esta capacidad está basada en una estrategia de enrutamiento llamada Equal-cost Multi-path, la cual incrementa sustancialmente el ancho de banda al balancear la carga de tráfico sobre esas múltiples rutas.

El enrutamiento Multi-path puede ser utilizado en conjunto con la mayoría de los protocolos de enrutamiento ya que las decisiones de forwarding son tomadas localmente en cada dispositivo.

La implementación de FRRouting soporta ECMP si es compilada con el parámetro `--enable-multipath=X`, con X mayor a 1.

En el presente trabajo los dispositivos de enrutamiento emulados con mininet son Linux que funcionan como routers. Cabe destacar que en los Linux la implementación del ECMP ha cambiado en el correr del tiempo. Hasta la versión 3.36 y desde la versión 4.4 del kernel de Linux existe una característica que hace que los paquetes enviados a una misma dirección IP salgan todos por la misma interfaz de red (el kernel hace un cache por dirección IP, sin importar cuantas rutas diferentes haya en la tabla de ruteo) para evitar que los paquetes lleguen fuera de orden y que se pueda utilizar multipath en conjunto con servicios anycast (Punto 7 de la sección Networking de [1]). Es por eso que vamos a utilizar un Linux con kernel 4.2.0-16.19 (Ubuntu 15.10) para así poder ver más fácilmente el uso del ECMP ya que esa versión del kernel cuando tiene varias

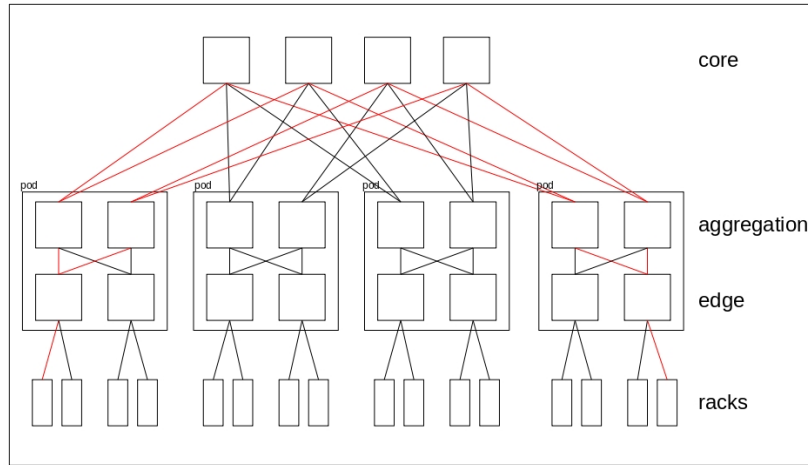


Figura 2: k caminos diferentes en Fattree con k=4

rutas de igual costo hacia un mismo destino de red balancea la carga con una estrategia round-robin.

1.3. FRRouting

FRRouting es una suite de protocolos IP para Linux y Unix e incluye demonios que manejan los protocolos BGP, IS-IS (y su extensión OpenFabric), LDP, OSPF, PIM, y RIP. En nuestro caso vamos a utilizar solamente los demonios bgpd, isisd, y fabricd que manejan los protocolos BGP, ISIS y OpenFabric respectivamente. También utilizamos el demonio zebra, que es el administrador de FRRouting del enrutamiento IP (actualiza la tabla de ruteo del kernel, busca interfaces de red y redistribuye las rutas entre los diferentes protocolos de enrutamiento).

Para instalarlo hay que hacer lo siguiente:

```
# curl -s https://deb.frrouting.org/frr/keys.asc | sudo apt-key add -
# sudo echo deb https://deb.frrouting.org/frr xenial frr-stable > /etc/apt/sources.list.d/frr.list
# sudo apt update && sudo apt install frr frr-pythontools
```

1.4. Generador de topologías Fat-tree

Para generar la topología del laboratorio utilizamos como base la herramienta descargada de [2] la cual crea solamente el esqueleto de la red. Luego la modificamos para que además de generar la topología configure las interfaces de red, les asigne direcciones IP, genere los archivos de configuración de los

demonios y los inicie. Se crearon 3 diferentes herramientas: `fattree_bgp.py`, `fattree_isis.py` y `fattree_openfabric.py`. Todas ellas reciben como único parámetro el número k y la única diferencia entre ellas esta en los archivos de configuración generados y los demonios iniciados, ya que las topologías finales son idénticas.

Algunas consideraciones respecto a la topología generada:

- Por simplicidad los racks se modelaron como si en cada uno de ellos hubiera un solo host.
- Los routers core se nombran como r1001, r1002, r1003 y así sucesivamente hasta $r(1000+k)$.
- Los routers aggregation se nombran como r2001, r2002, r2003 y así sucesivamente hasta $r(2000 + (k*k/2))$.
- Los routers aggregation se nombran como r3001, r3002, r3003 y así sucesivamente hasta $r(3000 + (k*k/2))$.
- Los hosts se nombran como h001, h002, h003 y así sucesivamente hasta $h(000 + (k*k))$.

La configuración de las interfaces de red es de la siguiente manera:

- Las interfaces en cada router *core* se nombran de izquierda a derecha comenzando con *eth0*, *eth1*, *eth2* y así sucesivamente hasta *eth(k-1)*.
- Las interfaces en los routers *aggregation* y *edge* se nombran de izquierda a derecha y de arriba a abajo comenzando con *eth0*, *eth1*, *eth2* y así sucesivamente hasta *eth(k-1)*.

El direccionamiento IP de las interfaces se realizó de la siguiente manera:

- Las subredes entre los routers core y aggregation tienen el rango $10.0.x.0/24$
- Las subredes entre los routers aggregation y edge tienen el rango $10.1.x.0/24$
- Las subredes entre los routers edge y los hosts tienen el rango $192.168.x.0/24$

donde x se incrementa desde 0 hasta $(k*k)-1$ de izquierda a derecha como muestra la imagen “Direccionamiento IP en Fattree con $k=4$ generada”.

Para el caso de la configuración del `bgpd`, la numeración de los AS se realizó siguiendo los lineamientos de la sección “ASN Numbering” de [3]. Por lo tanto los AS quedan así:

- Los routers core se configuran con el AS 65534
- Los routers aggregation que pertenecen a un mismo pod tienen un mismo AS y se numeran de izquierda a derecha como $65000+nro_pod*k$, siendo el primer pod el $nro_pod=0$
- Los routers edge se numeran como $65000+nro_pod*k+i$, siendo i el número de router edge dentro del pod, incrementándose desde el 1 hasta $k/2$ de izquierda a derecha

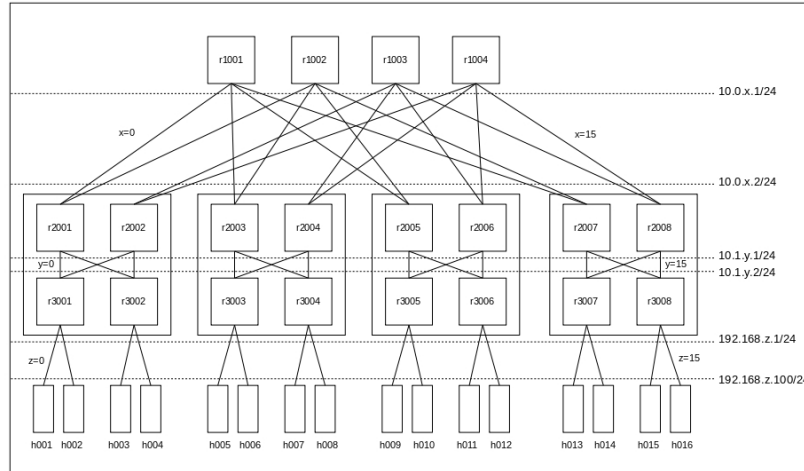


Figura 3: Direccionamiento IP en Fattree con $k=4$ generada

Como ejemplo el resultado para el caso $k=4$ se puede ver en la figura “Fattree con $k=4$ generada”.

Las herramientas están configuradas para que al iniciar corran las dos pruebas (que funciona el balanceo de la carga y que al dar de baja enlaces el enrutamiento continúe por los otros enlaces disponibles) y luego finalicen. Al correr las pruebas se inicia la captura de paquetes en las interfaces de los routers de aggregation que conectan con los routers core (con tcpdump) y se ejecutan pings entre los hosts $h001$ y $h(1000+k*k*k/4)$. Las capturas se guardan en la carpeta pcaps y se pueden ver con el comando `tcpdump -qns 0 -r pcaps/*`.

2. Pruebas

Tenemos que probar que con cada una de las implementaciones de FRRouting de los protocolos BGP, IS-IS y OpenFabric se cumplan las siguientes características del ECMP:

1. que se balancea la carga entre las diferentes rutas de igual costo definidas en tabla de enrutamiento
2. que al dar de baja un enlace se sigue dirigiendo el tráfico por los restantes enlaces.

Vamos a explicar lo que se hace en detalle en cada una de las pruebas tomando como ejemplo la topología generada con la herramienta con $k=4$.

Se detallan las pruebas realizadas con la herramienta `fattree_bgp.py` que utiliza BGP como protocolo de enrutamiento dado que las pruebas realizadas

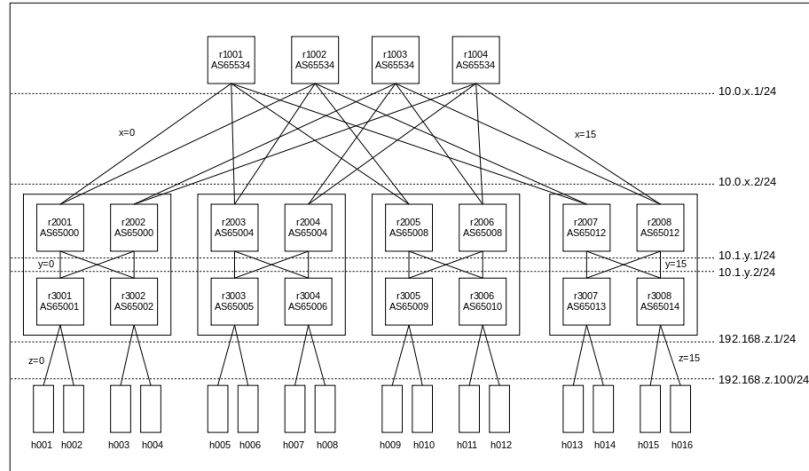


Figura 4: Fattree con k=4 generada

con las otras dos herramientas arrojan idénticos resultados.

2.1. Prueba 1

Para probar que se balancea la carga vamos a ejecutar un ping desde el host h001 al host h016 y vamos a capturar los paquetes (ejecutando `tcpdump -i interface -w pcap/prueba1_router_interface.pcap 'dst host 192.168.15.100 and icmp'`) en las interfaces eth0 y eth1 de los routers aggregation ya que son las interfaces que participan en los 4 caminos disponibles entre los hosts h001 y h016 (ver Figura “Multipath entre h001 y h016”).

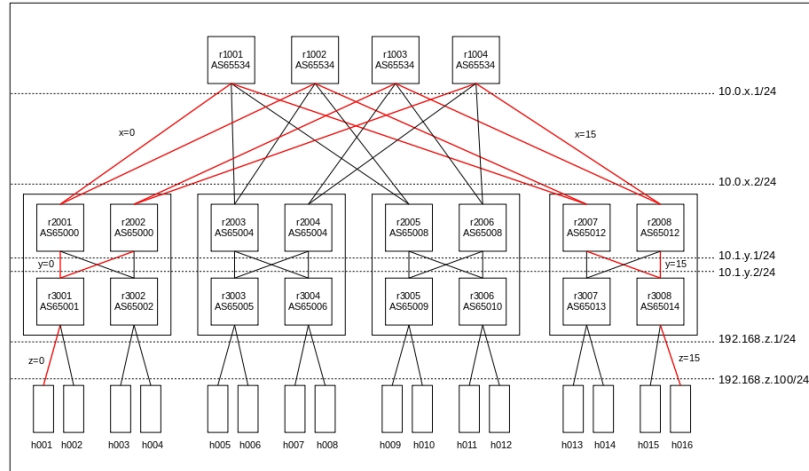


Figura 5: Multipath entre h001 y h016

Lo que deberíamos ver es que los paquetes atraviesan los 4 caminos uno a la vez (aunque no necesariamente en ese orden):

- Camino 1: ... r2001-eth0 ↔ r1001-eth0 ↔ r1001-eth3 ↔ r2007-eth0 ...
- Camino 2: ... r2002-eth0 ↔ r1003-eth0 ↔ r1003-eth3 ↔ r2008-eth0 ...
- Camino 3: ... r2001-eth1 ↔ r1002-eth0 ↔ r1002-eth3 ↔ r2007-eth1 ...
- Camino 4: ... r2002-eth1 ↔ r1004-eth0 ↔ r1004-eth3 ↔ r2008-eth1 ...

```

reading from file pcaps/prueba1_r2001_eth0.pcap, link-type EN10MB (Ethernet)
14:35:41.296675 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 1, length 64
14:35:45.296211 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 5, length 64
14:35:49.296401 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 9, length 64
14:35:53.296195 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 13, length 64
reading from file pcaps/prueba1_r2001_eth1.pcap, link-type EN10MB (Ethernet)
14:35:43.296144 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 3, length 64
14:35:47.296187 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 7, length 64
14:35:51.296284 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 11, length 64
14:35:55.296431 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 15, length 64
reading from file pcaps/prueba1_r2002_eth0.pcap, link-type EN10MB (Ethernet)
14:35:42.296214 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 2, length 64
14:35:46.296295 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 6, length 64
14:35:50.296462 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 10, length 64
14:35:54.296388 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 14, length 64
reading from file pcaps/prueba1_r2002_eth1.pcap, link-type EN10MB (Ethernet)
14:35:44.296282 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 4, length 64
14:35:48.296460 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 8, length 64
14:35:52.296285 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 12, length 64

```

```

14:35:56.296406 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 16, length 64
reading from file pcaps/prueba1_r2003_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba1_r2003_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba1_r2004_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba1_r2004_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba1_r2005_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba1_r2005_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba1_r2006_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba1_r2006_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba1_r2007_eth0.pcap, link-type EN10MB (Ethernet)
14:35:41.296683 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 1, length 64
14:35:45.296225 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 5, length 64
14:35:49.296415 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 9, length 64
14:35:53.296207 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 13, length 64
reading from file pcaps/prueba1_r2007_eth1.pcap, link-type EN10MB (Ethernet)
14:35:43.296158 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 3, length 64
14:35:47.296202 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 7, length 64
14:35:51.296298 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 11, length 64
14:35:55.296446 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 15, length 64
reading from file pcaps/prueba1_r2008_eth0.pcap, link-type EN10MB (Ethernet)
14:35:42.296231 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 2, length 64
14:35:46.296311 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 6, length 64
14:35:50.296478 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 10, length 64
14:35:54.296405 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 14, length 64
reading from file pcaps/prueba1_r2008_eth1.pcap, link-type EN10MB (Ethernet)
14:35:44.296298 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 4, length 64
14:35:48.296473 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 8, length 64
14:35:52.296293 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 12, length 64
14:35:56.296420 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16262, seq 16, length 64

```

Se puede observar en las capturas que los paquetes efectivamente transitan por los 4 caminos siguiendo una estrategia round-robin (observar los número de secuencia del ping).

2.2. Prueba 2

Para probar que al dar de baja un enlace el tráfico de paquetes sigue encaminándose por los enlaces restantes vamos ejecutar pings desde el host h001 al host h016 e ir bajando los enlaces r2001-eth0↔r1001-eth0, r2001-eth1↔r1002-eth0 y r2002-eth0↔r1003-eth0, y vamos a capturar los paquetes en las interfaces eth0 y eth1 de los routers aggregation ya que son las interfaces que participan en los 4 caminos disponibles entre los hosts h001 y h016.

```

Analizando la corrida 16291 *****
reading from file pcaps/prueba2_r2001_eth0.pcap, link-type EN10MB (Ethernet)
14:37:01.412904 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16291, seq 1, length 64
14:37:05.412159 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16291, seq 5, length 64
14:37:09.412191 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16291, seq 9, length 64
14:37:13.412167 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16291, seq 13, length 64
reading from file pcaps/prueba2_r2001_eth1.pcap, link-type EN10MB (Ethernet)
14:37:03.412112 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16291, seq 3, length 64
14:37:07.412122 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16291, seq 7, length 64

```

```
14:37:11.412170 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 11, length 64
14:37:15.412199 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 15, length 64
reading from file pcaps/prueba2_r2002_eth0.pcap, link-type EN10MB (Ethernet)
14:37:02.412247 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 2, length 64
14:37:06.412466 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 6, length 64
14:37:10.412257 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 10, length 64
14:37:14.412377 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 14, length 64
reading from file pcaps/prueba2_r2002_eth1.pcap, link-type EN10MB (Ethernet)
14:37:04.412144 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 4, length 64
14:37:08.412355 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 8, length 64
14:37:12.412315 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 12, length 64
14:37:16.412137 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 16, length 64
reading from file pcaps/prueba2_r2003_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2003_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2004_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2004_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2005_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2005_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2006_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2006_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2007_eth0.pcap, link-type EN10MB (Ethernet)
14:37:01.412917 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 1, length 64
14:37:05.412174 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 5, length 64
14:37:09.412205 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 9, length 64
14:37:13.412180 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 13, length 64
reading from file pcaps/prueba2_r2007_eth1.pcap, link-type EN10MB (Ethernet)
14:37:03.412128 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 3, length 64
14:37:07.412138 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 7, length 64
14:37:11.412186 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 11, length 64
14:37:15.412213 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 15, length 64
reading from file pcaps/prueba2_r2008_eth0.pcap, link-type EN10MB (Ethernet)
14:37:02.412262 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 2, length 64
14:37:06.412481 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 6, length 64
14:37:10.412273 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 10, length 64
14:37:14.412390 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 14, length 64
reading from file pcaps/prueba2_r2008_eth1.pcap, link-type EN10MB (Ethernet)
14:37:04.412159 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 4, length 64
14:37:08.412369 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 8, length 64
14:37:12.412329 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 12, length 64
14:37:16.412151 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16291, seq 16, length 64
*****
```

```
Analizando la corrida 16304 *****
reading from file pcaps/prueba2_r2001_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2001_eth1.pcap, link-type EN10MB (Ethernet)
14:37:21.441748 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 1, length 64
14:37:23.440108 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 3, length 64
14:37:25.440141 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 5, length 64
14:37:27.440206 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 7, length 64
14:37:29.440006 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 9, length 64
14:37:31.440251 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 11, length 64
14:37:33.440011 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 13, length 64
14:37:35.440232 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 15, length 64
reading from file pcaps/prueba2_r2002_eth0.pcap, link-type EN10MB (Ethernet)
14:37:22.440758 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 2, length 64
```

```

14:37:26.440105 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 6, length 64
14:37:30.439990 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 10, length 64
14:37:34.440225 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 14, length 64
reading from file pcaps/prueba2_r2002_eth1.pcap, link-type EN10MB (Ethernet)
14:37:24.440092 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 4, length 64
14:37:28.440038 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 8, length 64
14:37:32.440001 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 12, length 64
14:37:36.439975 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 16, length 64
reading from file pcaps/prueba2_r2003_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2003_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2004_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2004_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2005_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2005_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2006_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2006_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2007_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2007_eth1.pcap, link-type EN10MB (Ethernet)
14:37:21.441761 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 1, length 64
14:37:23.440122 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 3, length 64
14:37:25.440154 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 5, length 64
14:37:27.440219 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 7, length 64
14:37:29.440019 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 9, length 64
14:37:31.440265 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 11, length 64
14:37:33.440024 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 13, length 64
14:37:35.440246 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 15, length 64
reading from file pcaps/prueba2_r2008_eth0.pcap, link-type EN10MB (Ethernet)
14:37:22.440772 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 2, length 64
14:37:26.440118 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 6, length 64
14:37:30.440002 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 10, length 64
14:37:34.440240 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 14, length 64
reading from file pcaps/prueba2_r2008_eth1.pcap, link-type EN10MB (Ethernet)
14:37:24.440106 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 4, length 64
14:37:28.440053 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 8, length 64
14:37:32.440015 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 12, length 64
14:37:36.439989 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16304, seq 16, length 64
*****

```

Analizando la corrida 16317 *****

```

reading from file pcaps/prueba2_r2001_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2001_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2002_eth0.pcap, link-type EN10MB (Ethernet)
14:37:41.466955 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 1, length 64
14:37:43.464996 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 3, length 64
14:37:45.464276 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 5, length 64
14:37:47.464425 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 7, length 64
14:37:49.464258 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 9, length 64
14:37:51.464203 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 11, length 64
14:37:53.464161 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 13, length 64
14:37:55.464524 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 15, length 64
reading from file pcaps/prueba2_r2002_eth1.pcap, link-type EN10MB (Ethernet)
14:37:42.465967 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 2, length 64
14:37:44.464650 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 4, length 64
14:37:46.464418 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 6, length 64
14:37:48.464291 IP 192.168.0.100 > 192.168.15.100: ICMP echo request , id 16317, seq 8, length 64

```

```

14:37:50.464211 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 10, length 64
14:37:52.464107 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 12, length 64
14:37:54.464527 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 14, length 64
14:37:56.468230 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 16, length 64
reading from file pcaps/prueba2_r2003_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2003_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2004_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2004_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2005_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2005_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2006_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2006_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2007_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2007_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2008_eth0.pcap, link-type EN10MB (Ethernet)
14:37:41.466968 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 1, length 64
14:37:43.465009 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 3, length 64
14:37:45.464290 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 5, length 64
14:37:47.464442 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 7, length 64
14:37:49.464272 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 9, length 64
14:37:51.464218 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 11, length 64
14:37:53.464174 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 13, length 64
14:37:55.464539 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 15, length 64
reading from file pcaps/prueba2_r2008_eth1.pcap, link-type EN10MB (Ethernet)
14:37:42.465983 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 2, length 64
14:37:44.464670 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 4, length 64
14:37:46.464434 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 6, length 64
14:37:48.464306 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 8, length 64
14:37:50.464227 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 10, length 64
14:37:52.464123 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 12, length 64
14:37:54.464541 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 14, length 64
14:37:56.468244 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16317, seq 16, length 64
*****

```

Analizando la corrida 16330 *****

```

reading from file pcaps/prueba2_r2001_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2001_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2002_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2002_eth1.pcap, link-type EN10MB (Ethernet)
14:38:01.496144 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 1, length 64
14:38:02.496301 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 2, length 64
14:38:03.496452 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 3, length 64
14:38:04.496328 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 4, length 64
14:38:05.496372 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 5, length 64
14:38:06.496343 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 6, length 64
14:38:07.496177 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 7, length 64
14:38:08.495931 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 8, length 64
14:38:09.496932 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 9, length 64
14:38:10.497174 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 10, length 64
14:38:11.497074 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 11, length 64
14:38:12.496852 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 12, length 64
14:38:13.496889 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 13, length 64
14:38:14.496893 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 14, length 64
14:38:15.496793 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 15, length 64
14:38:16.496689 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 16, length 64

```

```

reading from file pcaps/prueba2_r2003_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2003_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2004_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2004_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2005_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2005_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2006_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2006_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2007_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2007_eth1.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2008_eth0.pcap, link-type EN10MB (Ethernet)
reading from file pcaps/prueba2_r2008_eth1.pcap, link-type EN10MB (Ethernet)
14:38:01.496158 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 1, length 64
14:38:02.496317 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 2, length 64
14:38:03.496468 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 3, length 64
14:38:04.496344 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 4, length 64
14:38:05.496388 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 5, length 64
14:38:06.496359 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 6, length 64
14:38:07.496193 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 7, length 64
14:38:08.496001 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 8, length 64
14:38:09.496947 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 9, length 64
14:38:10.497190 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 10, length 64
14:38:11.497090 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 11, length 64
14:38:12.496867 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 12, length 64
14:38:13.496904 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 13, length 64
14:38:14.496909 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 14, length 64
14:38:15.496808 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 15, length 64
14:38:16.496705 IP 192.168.0.100 > 192.168.15.100: ICMP echo request, id 16330, seq 16, length 64
*****

```

Se puede ver que inicialmente (corrida 16291) el tráfico sigue los 4 caminos de la Prueba 1 pero al dar de baja el enlace r2001-eth0↔r1001-eth0 el tráfico comienza a ir solamente por los caminos 2, 3 y 4 (corrida 16304). Luego al dar de baja el enlace r2001-eth1↔r1002-eth0 el tráfico comienza a ir por los caminos 2 y 4 (corrida 16317). Y por último al dar de baja el enlace r2002-eth0↔r1003-eth0 el tráfico comienza a ir solamente por el camino 4 (corrida 16330).

3. Conclusión

Se lograron cumplir los objetivos de hacer funcionar la implementación de FRRouting de los protocolos IS-IS, OpenFabric y BGP en un data center emulado con mininet (utilizando una topología Fat-tree con k=4) y ver que funcione el ECMP (Equal-cost Multi-path) logrando el balanceo de la carga y la redirección de los paquetes cuando un enlace cae.

Referencias

- [1] RikvanRiel. Kernelnewbies. https://kernelnewbies.org/Linux_4.4.
- [2] Andreas Pantelopoulos. mininet-fattree.
<https://github.com/panandr/mininet-fattree>.
- [3] Dinesh G. Dutt. *BGP in the Data Center*. O'Reilly Media, Inc.