ANEXO 3

ENRUTAMIENTO EN LA RED DE PRUEBA

1. INTRODUCCIÓN

En este anexo se presentan las consideraciones que se tuvieron para la parte de enrutamiento de la red de laboratorio para pruebas implementada en el presente proyecto.

Para poder medir cada una de las variables de desempeño lo ideal sería mantener el resto de factores constantes, por ejemplo, un tráfico de enrutamiento nulo; sin embargo, dado que el escenario planteado pretende ser un ejemplo en pequeña escala de una red real tal como la de un proveedor de servicios de Internet (ISP) se descartó el uso de rutas estáticas debido a que en una implementación a gran escala sería absurdo, por lo tanto se decidió utilizar un protocolo de enrutamiento que presentara un tráfico con un comportamiento periódico que pudiera considerarse aproximadamente constante en el sentido en que todas las medidas de desempeño realizadas se vean afectadas por él en la misma forma.

Utilizando tres enrutadores software con Quagga sobre Debian GNU/Linux y otros tres enrutadores Cisco 2801 ISR para construir el core de la red, se realizaron las configuraciones básicas para utilizar RIPv2 (*Routing Information Protocol* versión 2), OSPFv2 (*Open Shortest Path First* versión 2) y BGP-4 (*Border Gateway Protocol*), y mediante el analizador de protocolos Wireshark y sus filtros de captura de diferentes protocolos se realizaron gráficos del tráfico entrante y saliente (en Bytes/segundo) a través de las interfaces de uno de los enrutadores (el mismo en todas las mediciones)

2. COMPORTAMIENTO DEL TRÁFICO DE ENRUTAMIENTO

Para la primera prueba se analizó el tráfico de enrutamiento durante un periodo de rollup de 5 minutos para cada protocolo. Todas las gráficas presentadas inician con la captura del primer paquete de dicho protocolo que atraviese una de las interfaces del enrutador en el que se realiza el análisis de tráfico, después de iniciar simultáneamente los demonios del protocolo de enrutamiento y de zebra en los tres enrutadores.

2.1. RIP

Inicialmente se configuró RIPv2 en los tres equipos y se obtuvo la siguiente información utilizando un filtro para capturar solo los paquetes correspondientes a RIP

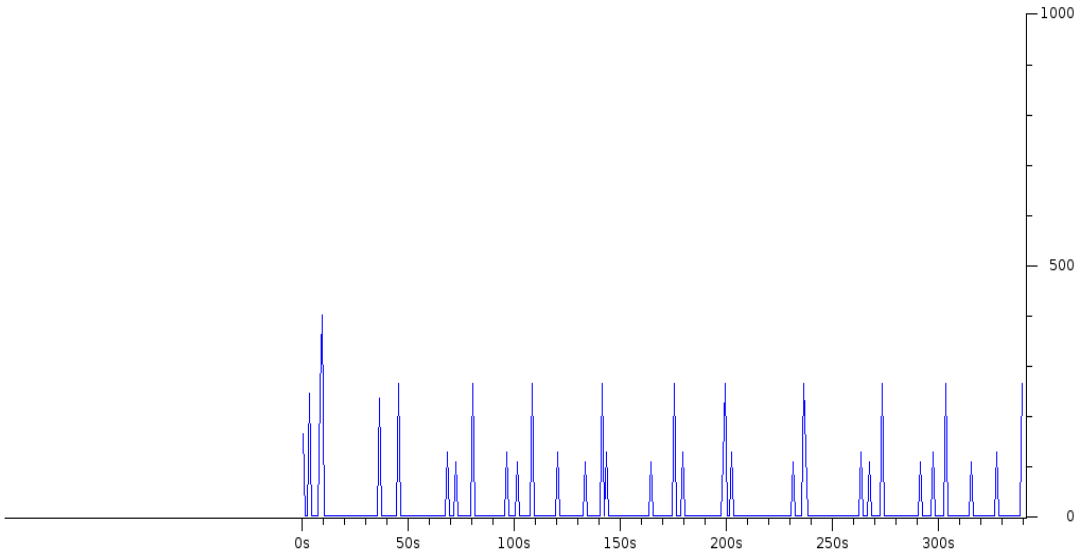

FIGURA A3. 1 TRAFICO RIP

Se observa que el tráfico inicial no es mucho mayor que el generado durante el resto del periodo de medición, sin embargo, se ve un comportamiento más aleatorio, lo cual no permitiría determinar con un buen grado de precisión la manera en que afecta el tráfico de enrutamiento en la medición de desempeño a realizar.

Además, una red trabajando con RIP no es muy escalable y su tiempo de convergencia normalmente es bastante alto por lo cual se descarta este protocolo y se pasa a explorar OSPF y BGP que en la actualidad son los protocolos utilizados generalmente en los ISP.

2.2. OSPF

Posteriormente en la misma red se realizó una configuración básica análoga a la de RIP, pero esta vez con OSPFv2 y se obtuvo la siguiente información,

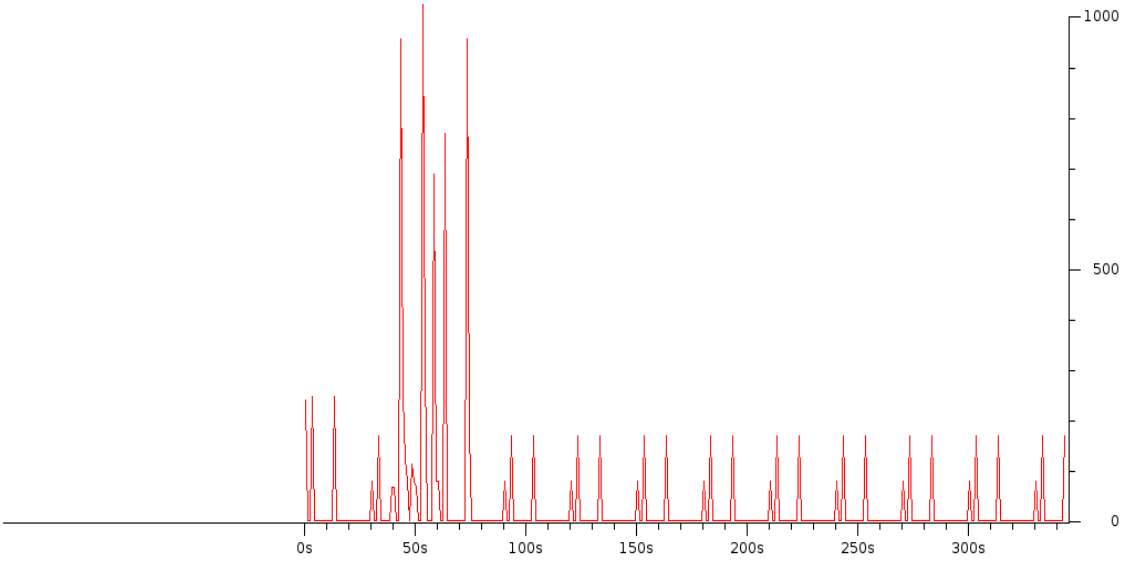

FIGURA A3. 2 TRAFICO OSPF

En este gráfico se puede observar un tráfico inicial bastante alto ya que OSPF intercambia la topología completa de la red, luego, se observa que es más bajo que en el caso de RIP y además tiene un comportamiento caracterizable dado que el tráfico de enrutamiento que pasa a través de estas interfaces consiste de 5 paquetes equivalentes a 416 Bytes en cada lapso de 30 segundos.

Aunque la configuración por defecto de OSPF en Quagga tiene un intervalo de envío de mensajes "Hello" de 10 segundos, para esta prueba se fijó a 30 segundos con el fin de poder realizar una comparación más apropiada dado que RIP por defecto envía mensajes de actualización en este mismo tiempo.

2.3. BGP

Por último se configuró BGP como IGP y se realizaron las mismas mediciones obteniendo el siguiente gráfico,

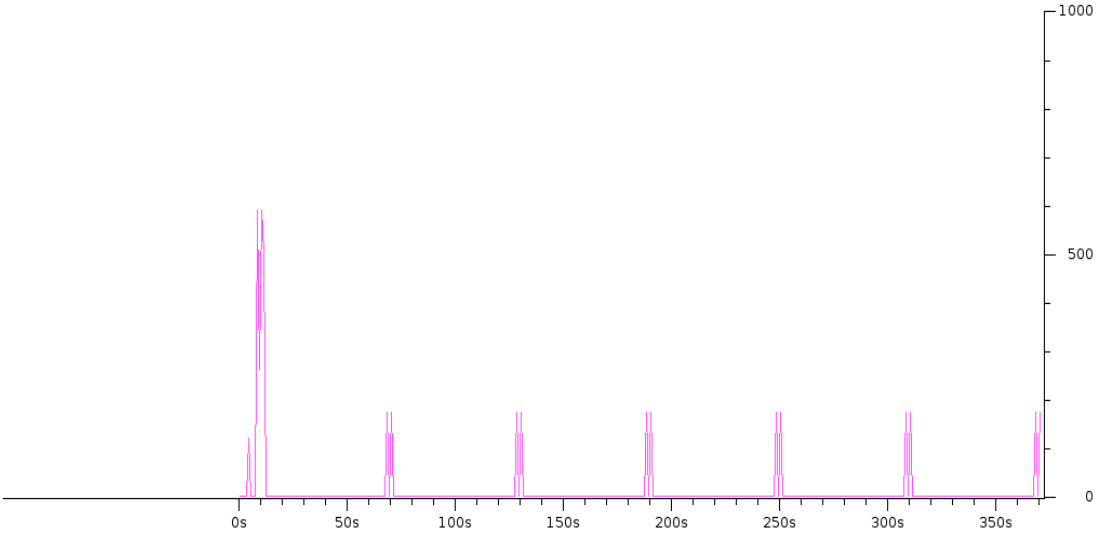

FIGURA A3. 3 TRAFICO BGP

Observando que el tráfico generado por este protocolo de enrutamiento es mucho menor que en los dos anteriores, tanto en la fase inicial como en el resto del tiempo de medición; este tráfico consiste en el envío de la base de datos completa de la red inicialmente y posteriormente de mensajes "Keep Alive" que se generan cada 60 segundos en la configuración por defecto.

3. TIEMPO DE CONVERGENCIA

Adicionalmente se realizó una prueba para comparar el tiempo de convergencia de OSPF y BGP al inhabilitar el enlace por el cual se alcanzaba la red 172.16.2.0 y obligando a los enrutadores a actualizar sus tablas de enrutamiento; en las siguientes gráficas se puede apreciar el envío de mensajes ICMP (echo request en color verde y echo reply en negro) de la red 172.16.5.0 a la red 172.16.6.0 que inicialmente se llevaba a cabo a través del router 3, pero al desconectar la interfaz que conecta el router 1 y el router 2 se debe enviar a través del router 3.

Cuando los enrutadores se encontraban trabajando con OSPF (con el intervalo entre mensajes "Hello" por defecto de 10 segundos) el tiempo de convergencia fue de aproximadamente 3 segundos, se perdió solamente un paquete y para el siguiente a éste el retardo subió de 0.2ms a 2.6ms, lo que hace que en la gráfica no sea visible la variación en el envío de los mensajes ICMP.

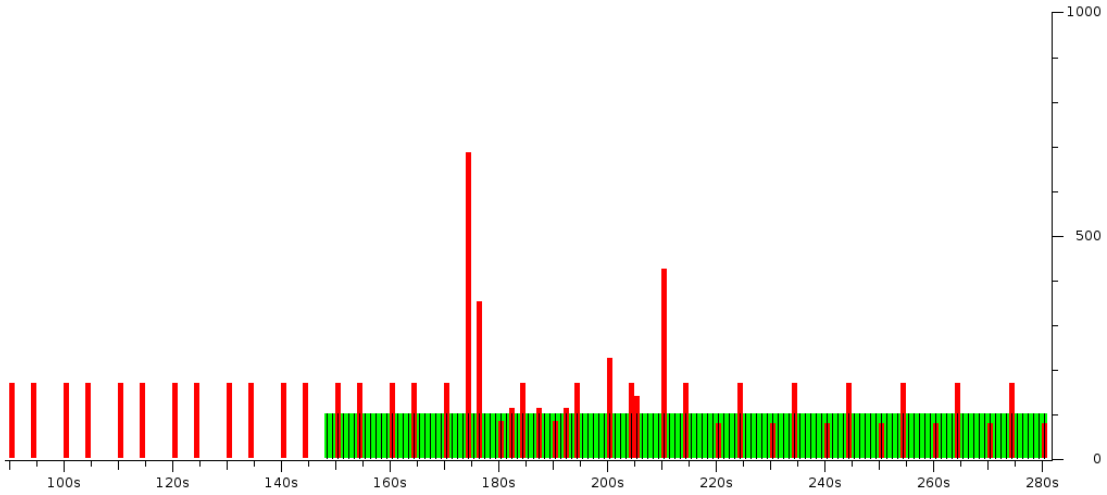

FIGURA A3. 4 TIEMPO DE CONVERGENCIA DE OSPF

Sin embargo se puede observar que alrededor de los 170 segundos (momento en que se inhabilitó el enlace) se disparó el tráfico de enrutamiento (en color rojo) informando la nueva topología completa de la red; además después de que se ha normalizado el comportamiento de la misma se observa menos tráfico que antes de la perturbación dado que uno de los enlaces del enrutador 2 (donde se realiza la medida del tráfico) está caído.

Posteriormente se realiza la misma prueba con BGP con un intervalo entre mensajes "Keep Alive" de 10 segundos al igual que entre los "Hello" de OSPF, y se obtiene la siguiente información,

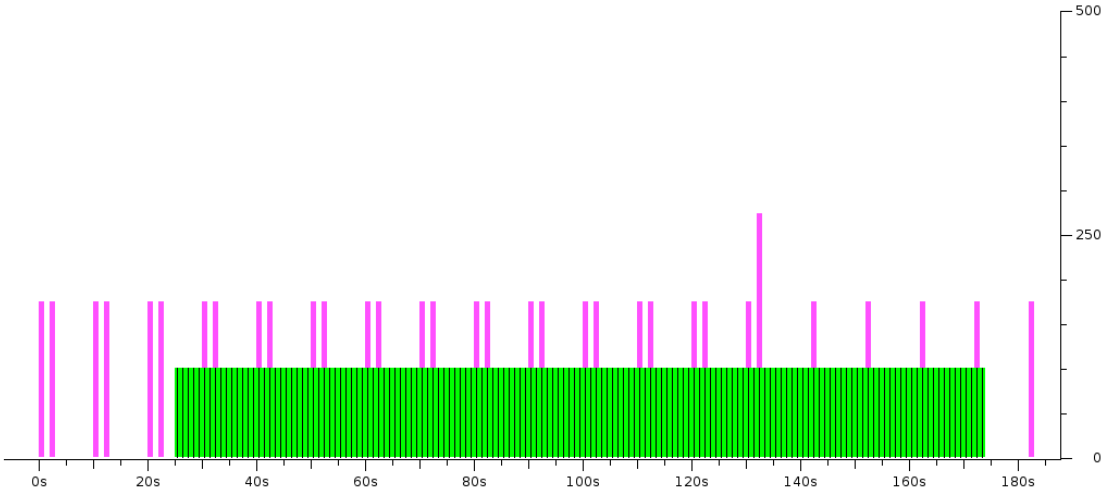

FIGURA A3. 5 TIEMPO DE CONVERGENCIA DE BGP

Una vez el comportamiento de la red se ha estabilizado se empiezan a enviar mensajes ICMP de ping y se rompe el enlace alrededor de los 130 segundos; inmediatamente se aprecia el intercambio de información de enrutamiento (en color morado) adicional que consiste en un mensaje de "Update" informando de la ruta caída a eliminar y la nueva ruta hacia la red 172.16.6.0. Se demora aproximadamente un segundo para poder continuar el envío de información ocasionando que no se pierda ningún paquete.

Después de comprobar el comportamiento estable de BGP mientras la topología de la red no cambie y su rápida convergencia implicando menor tráfico incluso frente a OSPF en caso de presentarse modificaciones a la misma, y teniendo en cuenta otros factores como su alta escalabilidad y el manejo de sistemas autónomos, se decidió utilizar BGP-4 como protocolo de enrutamiento en el backbone de la red del escenario planteado, en el que se evaluará el desempeño de una red IP con servicios de seguridad proporcionados por IPsec.

4. ARCHIVOS DE CONFIGURACIÓN

En esta sección se incluyen los archivos de configuración definitivos con los cuales se desarrolló el trabajo de grado.

Contenido del archivo /etc/quagga/daemons en los tres enrutadores software:

```
zebra=yes
bgpd=yes
ospfd=no
ospf6d=no
ripd=no
ripngd=no
isisd=no
```

A continuación se muestra el contenido del archivo /etc/quagga/zebra en el equipo de backbone1; la configuración de este archivo en los otros dos equipos es la misma, con excepción del nombre del equipo.

```
! -*- zebra -*-
!
! zebra sample configuration file
!
! $Id: zebra.conf.sample,v 1.1.1.1 2002/12/13 20:15:30
paul Exp $
!
hostname backbone1
password zebra
enable password zebra
log file /var/log/quagga/zebra.log
!
! Interface's description.
!
interface lo
interface eth0
interface eth1
interface eth2
!
!interface sit0
! multicast
!
! Static default route sample.
!
!ip route 0.0.0.0/0 203.181.89.241
!
access-list vtylist permit 127.0.0.1/32
access-list vtylist deny any
!
ip forwarding
!
line vty
 access-class vtylist
!
```

Y la configuración en el archivo /etc/quagga/bgpdv de este equipo es la siguiente:

```
!
! Zebra configuration saved from vty
!   2010/04/27 21:10:58
!
hostname backbone1
password zebra
enable password zebra
log stdout
!
router bgp 100
 bgp router-id 172.16.5.1
 network 172.16.1.0/24
 network 172.16.3.0/24
 network 172.16.5.0/24
 timers bgp 10 30
```

```
 neighbor 172.16.1.2 remote-as 200
 neighbor 172.16.3.1 remote-as 300
 neighbor 172.16.5.2 remote-as 100
!
line vty
!
```