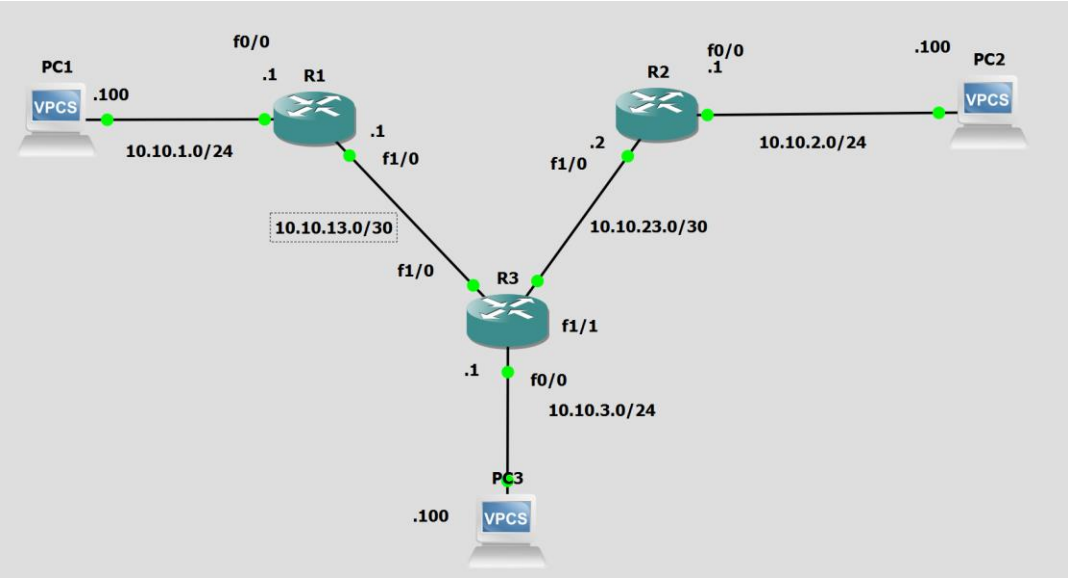
Precedencia de paquetes en QoS con DiffSer

Martínez Coronel Brayan Yosafat

Servicios diferenciados (DiffServ) es un nuevo modelo en el cual el tráfico es procesado a través de sistemas intermedios con prioridades relativas en base al campo Tipo de servicios (ToS). Definido en RFC 2474 y RFC 2475, el estándar DiffServ reemplaza la especificación original para definir la prioridad del paquete descrita en RFC 791.

DiffServ aumenta el número de niveles de prioridad definibles al reasignar los bits de un paquete de IP para que se les haga una marcación prioritaria. La arquitectura DiffServ define el campo DiffServ (DS), que reemplaza el campo ToS de IPv4 para tomar decisiones de comportamiento por salto (PHB), sobre la clasificación de paquetes y las funciones de condicionamiento del tráfico, tales como medición, marcado, forma y vigilancia. Los RFC no dictan la manera de implementar PHB; esta responsabilidad es del vendedor. Cisco implementa técnicas de colocación en cola que pueden basar su PHB en la precedencia de IP o en el valor DSCP del encabezado IP de un paquete. Sobre la base de la precedencia DSCP o IP, el tráfico se puede clasificar en una clase de servicio determinada. A los paquetes incluidos en una clase de servicio se los trata del mismo modo.

No es suficiente conocer las características del software, se necesita saber a qué plataformas son aplicables. Las funciones de QoS dependen mucho de la plataforma sobre la que se aplique. Algunas características se pueden aplicar solo a los enrutadores, otras solo a los switch y, algunos pueden ser diferentes entre miembros de la misma familia. Por ejemplo, no todas las características de QoS disponibles para Cisco 3560 son válidas para Cisco 3550.



Los routers usan OSPF:

```
Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O       10.10.1.0/24 [110/3] via 10.10.23.1, 00:28:50, FastEthernet1/0
C       10.10.2.0/24 is directly connected, FastEthernet0/0
O       10.10.3.0/24 [110/2] via 10.10.23.1, 00:28:50, FastEthernet1/0
O       10.10.13.0/30 [110/2] via 10.10.23.1, 00:28:50, FastEthernet1/0
C       10.10.23.0/30 is directly connected, FastEthernet1/0
```

Se aplican los siguientes comandos en el R1, el resto son muy parecidos:

```
class-map match-all OSPF
match protocol ospf
class-map match-all MATCH_HTTP
match access-group 105
class-map match-all ICMP_A_NUCLEO
match precedence 1
class-map match-all HTTP_A_NUCLEO
match precedence 3
class-map match-all MATCH_ICMP
match access-group 101
R1#show running-config | section policy-map
policy-map DESDE_HOST
class MATCH_ICMP
set precedence 1
class MATCH_HTTP
set precedence 3
policy-map A_NUCLEO
class ICMP_A_NUCLEO
bandwidth 8
police cir 8000
conform-action transmit
exceed-action drop
class HTTP_A_NUCLEO
```

```
bandwidth 10000
class OSPF
set precedence 7
priority 1000


access-list 101 permit icmp any any
access-list 101 remark "match icmp"
access-list 105 remark "match http"
access-list 105 permit tcp any any eq www

interface FastEthernet0/0
ip address 10.10.1.1 255.255.255.0
duplex auto
speed auto
service-policy input DESDE_HOST
end

interface FastEthernet2/0
ip address 10.10.13.1 255.255.255.0
duplex auto
speed auto
service-policy output A_NUCLEO
end
```

Estos comandos se aplican a los otros dos routers cambiando nada más la interfaz a la que se liga la política. A_NUCLEO signfica que es una interfaz que va a otro router, mientras que DESDE_HOST significa que conecta con una VPC. Para comprobar que se han realizado correctamente estas políticas, hacemos un ping del router 1 al 2.

Este ping va a pasar el límite de paquetes, y por tanto el router tendrá un exceso, la acción de este exceso es que lo elimine, con esto tendremos paquetes que sí tienen respuesta, y otros que son eliminados en la interfaz que sale al router 3. La mitad de los paquetes se habrá perdido en la interfaz f1/0, y esto lo comprobaremos con otro comando en esa interfaz:

```
R1#ping
Protocol [ip]:
Target IP address: 10.10.23.2
Repeat count [5]: 100
Datagram size [100]: 1400
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.10.13.1
Type of service [0]: 32
Set DF bit in IP header? [no]: yes
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 100, 1400-byte ICMP Echos to 10.10.23.2, timeout is 2 seconds:
Packet sent with a source address of 10.10.13.1
Packet sent with the DF bit set
!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.
!.!.!.!.!.!.!.!.!.!.!.!.!.!.!.
Success rate is 50 percent (50/100), round-trip min/avg/max = 24/54/88 ms
```

```
R1#show policy-map interface f1/0 output class ICMP_A_NUCLEO
 FastEthernet1/0

  Service-policy output: A_NUCLEO

    Class-map: ICMP_A_NUCLEO (match-all)
      100 packets, 141400 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match:  precedence 1
      Queueing
        Output Queue: Conversation 265
        Bandwidth 8 (kbps)Max Threshold 64 (packets)
        (pkts matched/bytes matched) 50/70700
        (depth/total drops/no-buffer drops) 0/0/0
      police:
          cir 8000 bps, bc 1500 bytes
        conformed 50 packets, 70700 bytes; actions:
          transmit
        exceeded 50 packets, 70700 bytes; actions:
          drop
        conformed 0 bps, exceed 0 bps
```