Calidad de Servicio en Linux: DiffServ

Planificación y Despliegue de Redes y Servicios

Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación (GSyC)

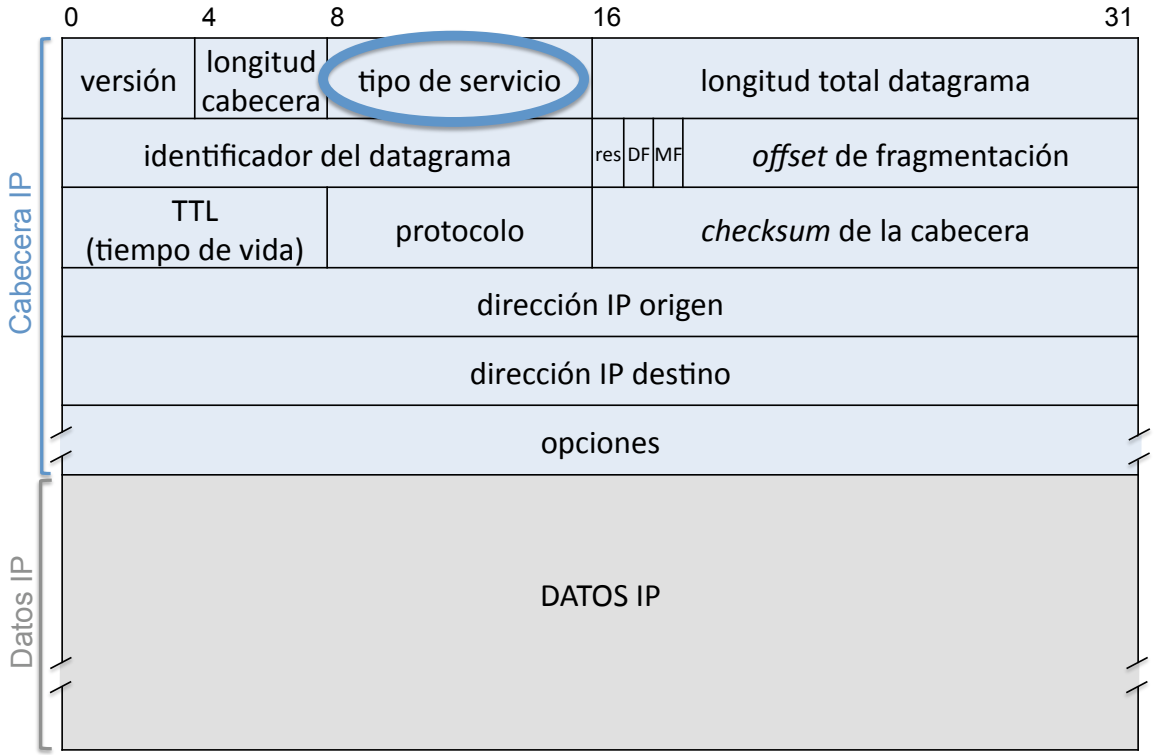
Noviembre de 2018

Contenidos

Contenidos

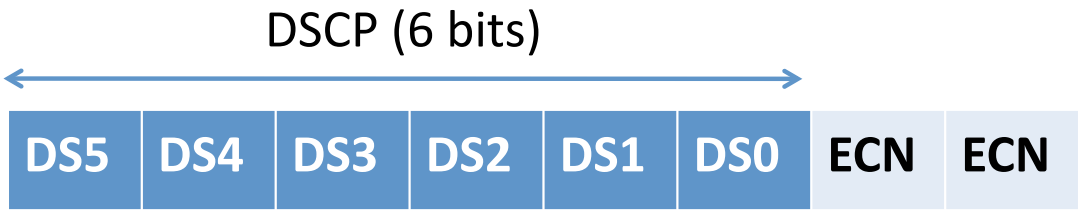
Marcado del tráfico DiffServ en la cabecera IPv4

- Los paquetes se marcan en el campo de 8 bits Type of Service (ToS) de IPv4



Marcado del tráfico DiffServ: campo DSCP

- Se usan 6 bits para identificar *Differentiated Service Code Point (DSCP)* que determinan el comportamiento por salto (PHB, Per-Hop Behavior) que recibirá el paquete en los *routers* de la red DiffServ.
- Quedan los 2 bits menos significativos del campo ToS que no se usan para DiffServ, sino para la notificación de congestión (Explicit Congestion Notification, ECN). ECN es utilizado conjuntamente por los extremos de una conexión TCP y los routers intermedios que utilizan en sus colas *Active Queue Management* (monitorizar el llenado de sus buffers antes de que se produzca la congestión).



Comportamiento por salto (PHB)

- Cada clase de tráfico (DSCP) está asociado a un comportamiento por salto (*PHB, Per Hop Behavior*).
- El PHB determina el tipo de tratamiento que se le va a dar al paquete en el reenvío:
  - Reserva de recursos: buffer y ancho de banda.
  - Características del tráfico: retardo y pérdidas.
- DiffServ no especifica las características concretas del tráfico en cada una de las clases, sólo proporciona la clasificación con diferentes códigos DSCP.

PHBs recomendados

- Aunque potencialmente pueden definirse 64 clases de tráfico diferente, y puede establecerse un PHB para cada una de ellas, la RFC recomienda al menos los siguientes PHB:
  - **EF** (Expedited Forwarding): DSCP 46=101110_B
    - Bajas pérdidas, baja latencia, bajo jitter
  - **VA** (Voice Admit): DSCP 44=101100_B
    - Similar a EF, pero puede incluir un mecanismo de control de admisión
  - **AF** (Assured Forwarding):
    - Se proporciona cierta garantía de entrega mientras el tráfico no exceda una tasa acordada.
    - Define 4 clases difentes: AF1-AF4. Dentro de cada clase se crean 3 PHBs, cada uno con diferente probabilidad de pérdida de los paquetes. La combinación de las 4 clases con las 3 probabilidades de pérdida dan 12 PHBs: AF11, AF12, AF13, AF21, AF22, AF23, AF31, AF32, AF33, en los que se cumple que la probabilidad de pérdida de AFX1 $<$ AFX2 $<$ AFX3 para cada una de las cuatro clases.
  - **CS** (Class Selector): usa los 3 primeros bits DSCP=XXX000 para definir prioridades (compatible con el antiguo ToS).
    - Menor prioridad = 000000 a mayor prioridad = 111000
  - **DF** (Default Forwarding): DSCP 0=000000_B
    - IP Best Effort (compatible con tráfico que no es DiffServ)

Marcas en los paquetes: Valores DS y DSCP más comunes

- El valor DS incluye los 8 bits que viajan en el campo TOS de la cabecera IP. Estos 8 bits son incluyen DSCP (6 bits) + ECN (2 bits).

		DSCP (6bits)	DS (8 bits)	Probabilidad de descarte de tráfico
Clase DF		000 000 = 0x00 = 0	0000 0000 = 0x00	
Clase AF1	AF11	**001** 010 = 0x0a = 10	0010 1000 = 0x28	Baja
	AF12	**001** 100 = 0x0c = 12	0011 0000 = 0x30	Media
	AF13	**001** 110 = 0x0e = 14	0011 1000 = 0x38	Alta
Clase AF2	AF21	**010** 010 = 0x12 = 18	0100 1000 = 0x48	Baja
	AF22	**010** 100 = 0x14 = 20	0101 0000 = 0x50	Media
	AF23	**010** 110 = 0x16 = 22	0101 1000 = 0x58	Alta
Clase AF3	AF31	**011** 010 = 0x1a = 26	0110 1000 = 0x68	Baja
	AF32	**011** 100 = 0x1c = 28	0111 0000 = 0x70	Media
	AF33	**011** 110 = 0x1e = 30	0111 1000 = 0x78	Alta
Clase AF4	AF41	**100** 010 = 0x22 = 34	1000 1000 = 0x88	Baja
	AF42	**100** 100 = 0x24 = 36	1001 0000 = 0x90	Media
	AF43	**100** 110 = 0x26 = 38	1001 1000 = 0x98	Alta
Clase EF		101 110 = 0x2e = 46	1011 1000 = 0xb8	

Contenidos

DSMARK: qdisc

- DSMARK es la qdisc que se utiliza para la clasificación de paquetes según la arquitectura de DiffServ y el marcado del campo DS.
- Marcado del campo DS:
  - **Routers frontera de DiffServ**: identifican clases en los paquetes atendiendo a valores de su cabecera IP y marcan adecuadamente el campo DS.
- Clasificación del tráfico según el campo DS:
  - **Routers del núcleo de DiffServ**: clasifican los paquetes atendiendo al contenido de DS que traen para realizar su tratamiento según diferentes disciplinas de cola.

DSMARK: qdisc y sus clases

- Una qdisc DSMARK crea automáticamente un conjunto de clases para asignar a cada una de ellas un valor diferente en el campo DSCP.
- Al crear la qdisc DSMARK es necesario especificar la cantidad máxima de valores DSCP que se van a usar para clasificar los paquetes. Este valor debe ser una potencia de 2.
  - Por ejemplo, la siguiente qdisc con número de clases igual a $2^3 = 8$ podrá definir como máximo 7 valores DSCP diferentes, uno para cada una de sus clases:

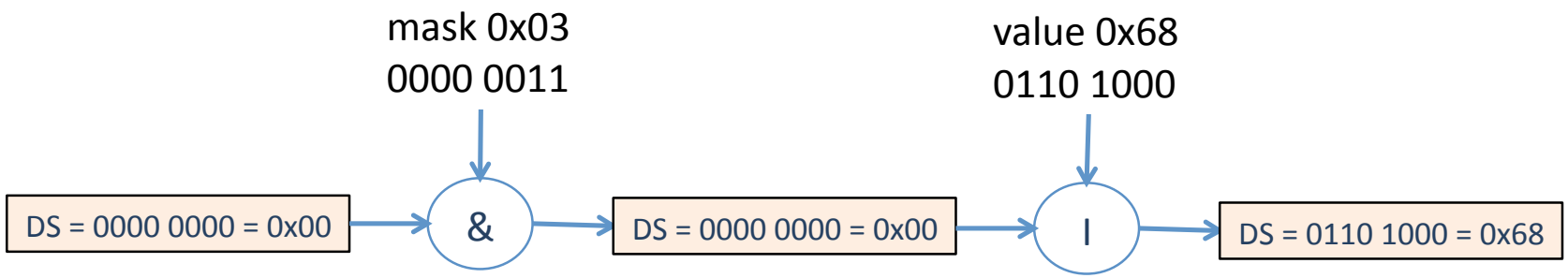
```
tc qdisc add dev eth1 root handle 1:0 dsmark indices  8
```

- Una vez creada una qdisc DSMARK se habrán creado automáticamente sus clases asociadas. Cada clase tiene un descriptor X:Y, donde:
  - X es el valor del número mayor de la clase qdisc a la que pertenece
  - Y es el número menor que distingue a cada una de las clases
- Por ejemplo, para definir como máximo 7 clases de tráfico dentro de una qdisc 1:0 se usarán los siguientes descriptores: 1:1, 1:2, 1:3, 1:4, 1:5, 1:6, 1:7. El número mayor coincide en todas ellas, ya que todas pertenecen a la qdisc 1.

Contenidos

Filtro tcindex: Router Frontera (I)

- Para marcar un valor DS en un paquete IP se utiliza una clase DSMARK en la que se realizan las 2 siguientes operaciones en orden:
  1. operación & (AND) de bits con una máscara para mantener los bits del campo DS que no se quiere modificar. Así, la máscara 0x03 mantendrá los 2 bits menos significativos del valor DS (en los que viaja ECN)
  2. operación | (OR) de bits con el valor DS que se desee asignar.



- Por ejemplo: para que la clase 1:3 se marque con DS=0x68 (AF31) conservando los bits ECN:

```
tc class change dev eth1 classid 1:3 dsmark mask 0x3 value 0x68
```

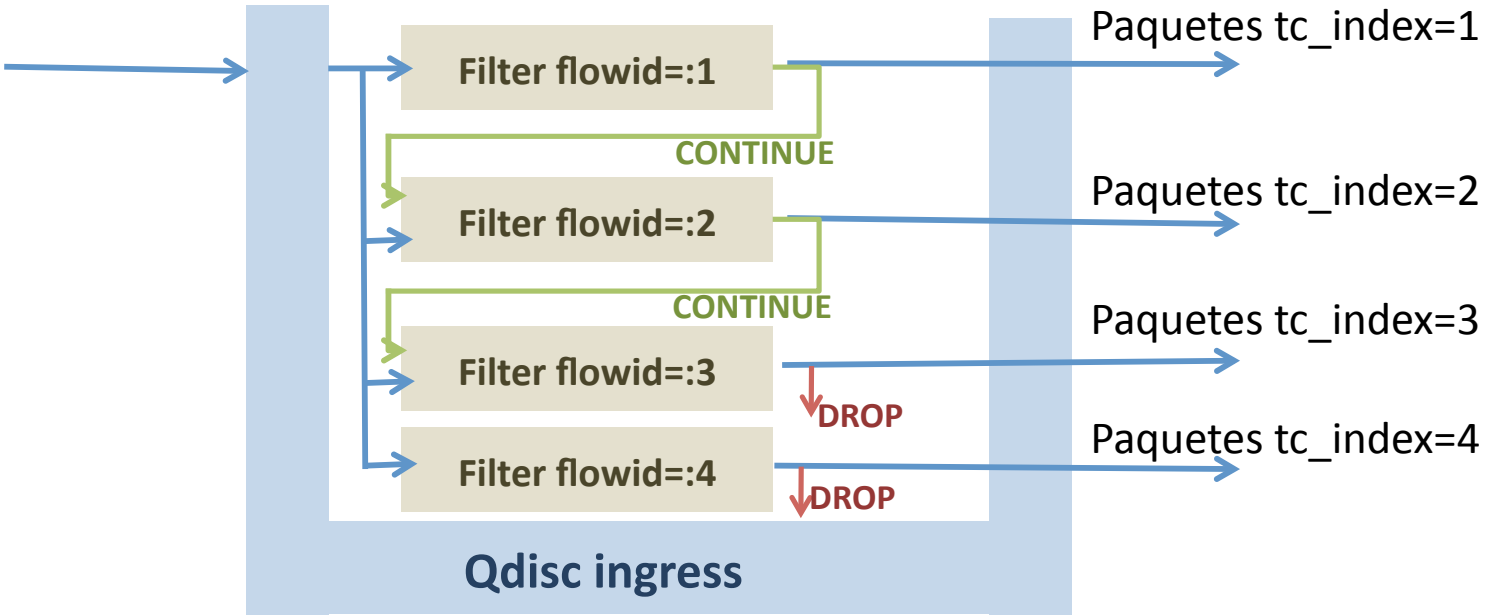
Filtro tcindex: Router Frontera (II)

- Al recibirse un paquete IP en un router, se almacena en un buffer.
- Cuando un paquete atraviesa la disciplina de cola de entrada `qdisc ingress`, el buffer que almacena el paquete IP dentro del kernel del router Linux tiene un campo `tc_index` en el que se almacena el identificador de flujo en el que fue clasificado dicho paquete cuando ingresó en la cola qdisc ingress:
  - si un paquete queda clasificado en `flowid :2`, el campo `tc_index` almacenaría el valor 2.
- Esta información es la que utiliza posteriormente el filtro tcindex dentro de otras qdisc de dicho router para dar tratamiento diferenciado a los paquetes.

```
tc qdisc add dev eth0 ingress handle ffff:

tc filter add dev eth0 parent ffff: \
    protocol ip prio 4 u32 \
    match ip src 11.0.0.100/32 \
    police rate 1mbit burst 10k continue flowid :1

tc filter add dev eth0 parent ffff: \
    protocol ip prio 5 u32 \
    match ip src 11.0.0.100/32 \
    police rate 512kbit burst 10k continue flowid :2

tc filter add dev eth0 parent ffff: \
    protocol ip prio 6 u32 \
    match ip src 11.0.0.100/32 \
    police rate 256kbit burst 10k drop flowid :3

tc filter add dev eth0 parent ffff: \
    protocol ip prio 6 u32 \
    match ip src 0.0.0.0/0 \
    police rate 128kbit burst 10k  drop flowid :4
```

Filtro tcindex: Router Frontera (III)

- Cuando un paquete atraviesa la disciplina de cola de entrada, el buffer que contiene el paquete tiene un campo donde se ha almacenado el descriptor del flujo al que pertenece: `tc_index`.



- Ejemplo: Los paquetes clasificados dentro del `flowid=:1` llevarán en el campo `tc_index=1`

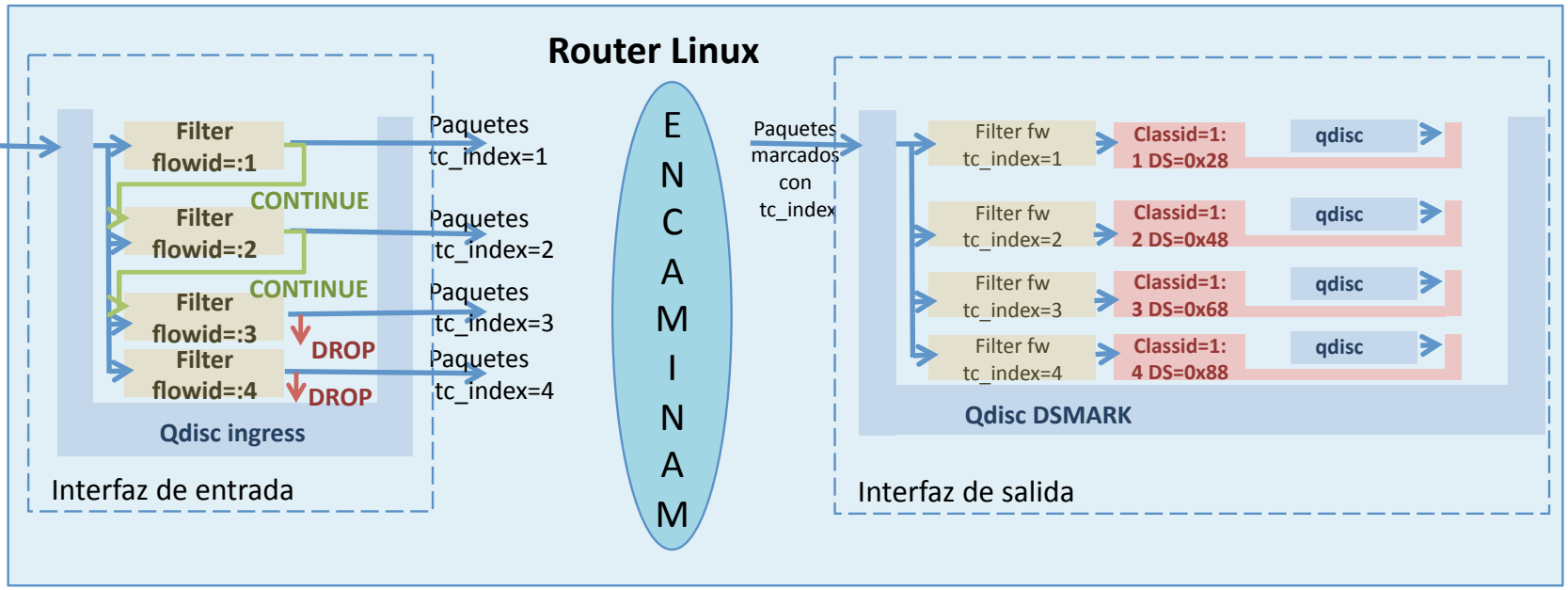
Filtro tcindex: Router Frontera (IV)

- EL filtro `tcindex` utiliza el valor almacenado en `tc_index` del buffer que contiene un paquete IP para clasificarlo dentro de una clase en la disciplina de cola de salida, en este caso DSMARK.
- Ejemplo:
  - qdisc DSMARK con 3 diferentes valores de marcado

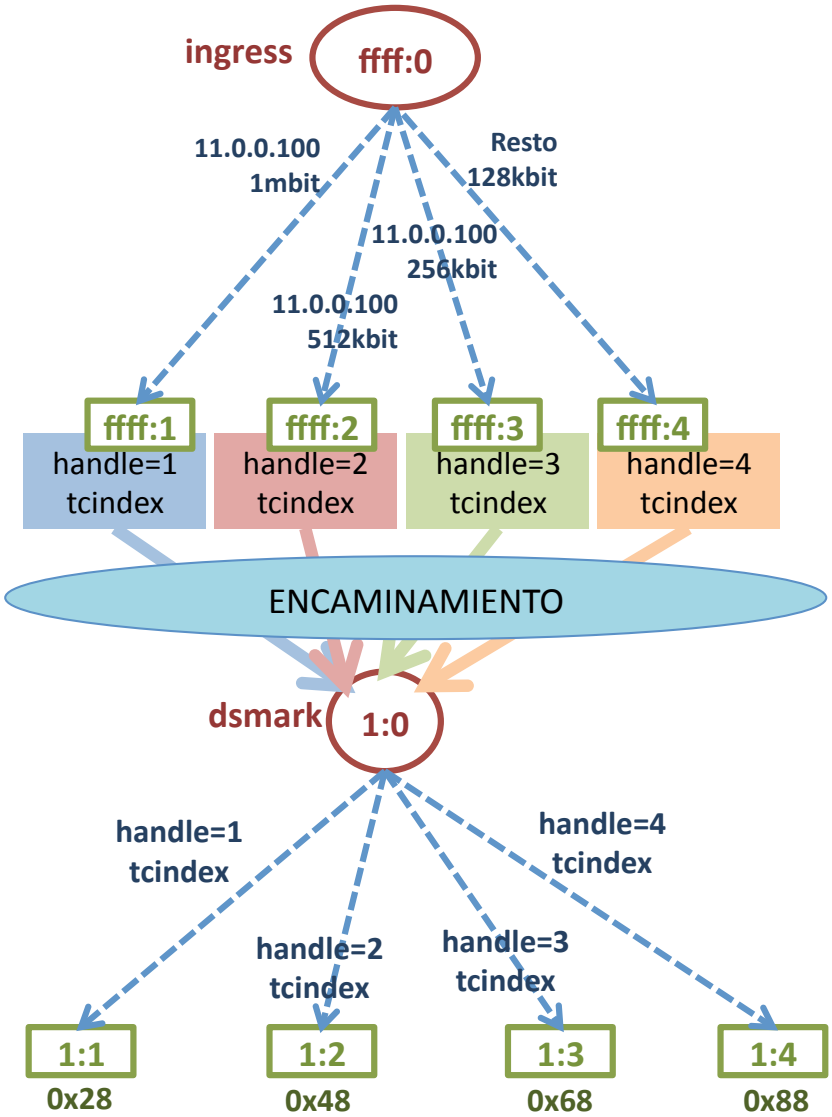
```
tc qdisc add dev eth1 root handle 1:0 dsmark indices 8

tc class change dev eth1 classid 1:1 dsmark mask 0x3 value 0x28
tc class change dev eth1 classid 1:2 dsmark mask 0x3 value 0x48
tc class change dev eth1 classid 1:3 dsmark mask 0x3 value 0x68
tc class change dev eth1 classid 1:4 dsmark mask 0x3 value 0x88

tc filter add dev eth1 parent 1:0 protocol ip prio 1 \
        handle 1 tcindex classid  1:1
tc filter add dev eth1 parent 1:0 protocol ip prio 1  \
        handle 2 tcindex classid  1:2
tc filter add dev eth1 parent 1:0 protocol ip prio 1  \
        handle 3 tcindex classid  1:3
tc filter add dev eth1 parent 1:0 protocol ip prio 1  \
        handle 4 tcindex classid  1:4
```

Filtro tcindex: Router Frontera (V)

- Los paquetes entran en el router y atraviesan la cola ingress, donde se rellena el campo `tc_index`.
- Dentro de la cola DSMARK en la interfaz de salida, se comprueba el valor de `tc_index` y se clasifica el tráfico en las clases de DSMARK, que marcarán los paquetes con un determinado valor de DS.



Filtro tcindex: Router Frontera (VI)



Contenidos

Filtro tcindex: Router del Núcleo (I)

- En el **router del núcleo**, el paquete llega con el valor DS establecido.
- Si el router no tiene configurada una disciplina de cola ingress, inicialmente no se rellena el campo `tc_index` del buffer del kernel del Linux que contiene el paquete.
- La propia qdisc DSMARK puede copiar el campo DS que lleva la cabecera IP del paquete en el campo `tc_index` de la estructura de datos del kernel del Linux que almacena el paquete utilizando el siguiente parámetro: set_tc_index

```
tc qdisc add dev eth1 root handle 1:0 dsmark indices 8 set_tc_index
```

Filtro tcindex: Router del Núcleo (II)

- Como se copia el campo DS entero, el filtro deberá utilizar sólo los 6 bits más significativos del campo DS, es decir, el DSCP, para clasificar los paquetes. Es necesario extraer esos 6 bits (operación & con mask 1111 1100 = 0xfc y desplazamiento a la derecha 2 bits) del campo DSCP:

```
tc filter add dev eth1 parent 1:0 protocol ip prio 1  \
     tcindex mask 0xfc shift 2
```

- A continuación se puede usar otra disciplina de cola encadenada con DSMARK para realizar la planificación de paquetes que se desee, por ejemplo HTB:

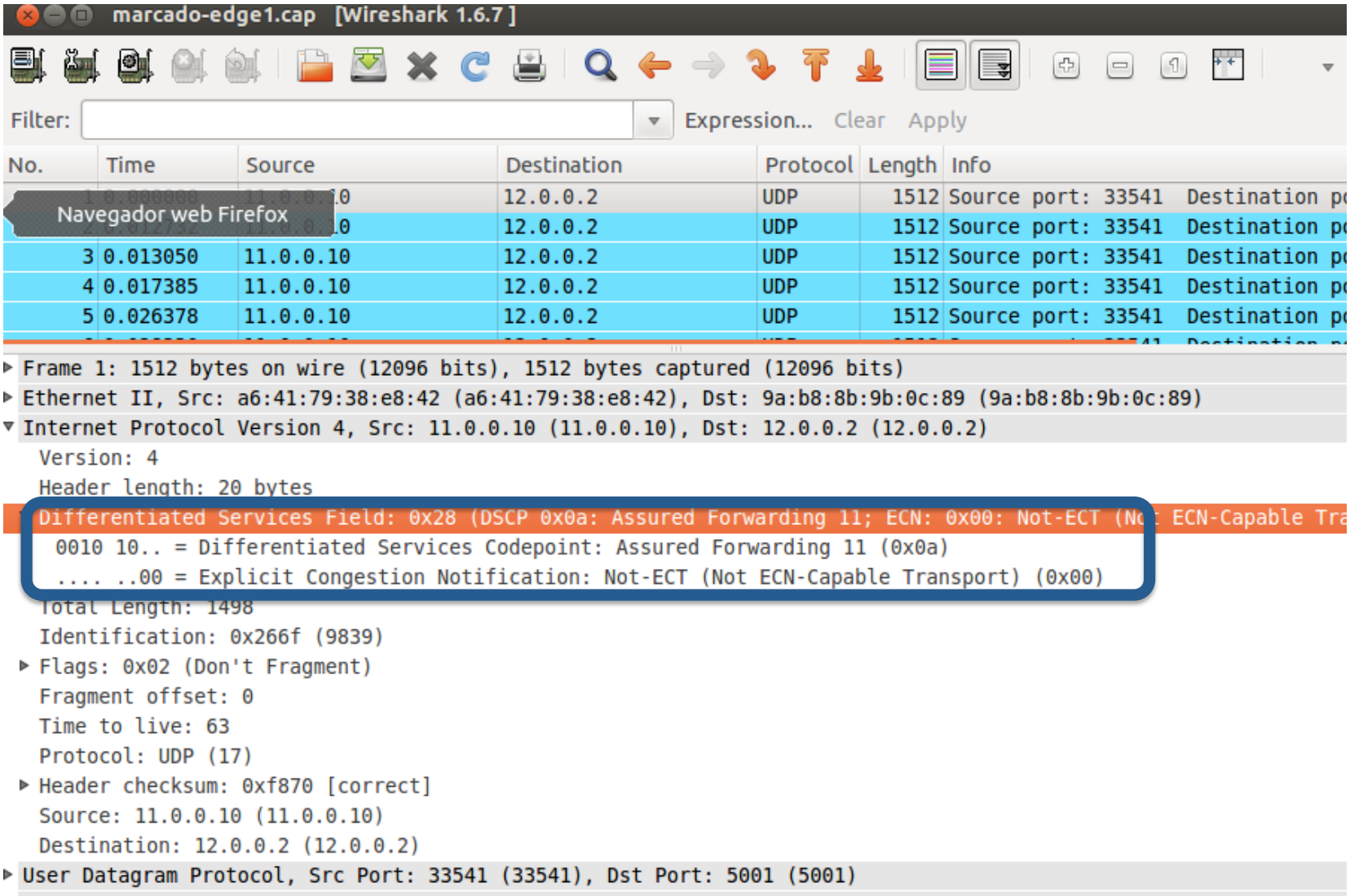
```
tc qdisc add dev eth1 parent 1:0  handle 2:0 htb

tc class add dev eth1 parent 2:0 classid 2:1 htb rate 1Mbit
tc class add dev eth1 parent 2:1 classid 2:10 htb rate 800kbit ceil 1Mbit

# Tráfico AF11=> DS=0x28 =00101000 =>  DSCP=001010=0x0a
tc filter add dev eth1 parent 2:0 protocol ip prio 1 \
     handle 0x0a tcindex classid 2:10
```

Filtro tcindex: Router del Núcleo (III)

- Ejemplo completo:

```
tc qdisc add dev eth1 root handle 1:0 dsmark indices 8 set_tc_index

tc filter add dev eth1 parent 1:0 protocol ip prio 1  \
      tcindex mask 0xfc shift 2

tc qdisc add dev eth1 parent 1:0 handle 2:0 htb

tc class add dev eth1 parent 2:0 classid 2:1 htb rate 1Mbit
tc class add dev eth1 parent 2:1 classid 2:10 htb \
     rate 800kbit ceil 1Mbit

# Tráfico AF11=> DS=0x28 =00101000 =>  DSCP=001010=0x0a
tc filter add dev eth1 parent 2:0 protocol ip prio 1 \
      handle 0x0a tcindex classid 2:10
```

Campo DS en una captura de Wireshark



Gráficas de ancho de banda en diferentes clases DiffServ (I)

- Seleccionar en el menú de Wireshark `Statistics` $->$ `IO Graphs`.
- En el panel inferior, añadir filtros según los criterios que se deseen. Marcar con un tic a la izquierda del filtro para visualizar los paquetes filtrados en la gráfica.
- Para filtrar una clase de tráfico puede usarse como condificón de filtrado alguna de las siguientes (cualquier de ellas serviría para filtrar el tráfico EF):
  - `ip.dsfield==0xb8`
  - `ip.dsfield.dscp==46`
  - `ip.dsfield.dscp==0x2e`
- Para distinguir el tráfico de varias fuentes la condición de filtrado puede completarse con la comprobación de la dirección IP. Ejemplo: `ip.dsfield==0x28 and ip.src==11.0.0.1`

Gráficas de ancho de banda en diferentes clases DiffServ