Control de Tr´afico en Linux

Sistemas Telem´aticos para Medios Audiovisuales

Departamento de Teor´ıa de la Se˜nal y Comunicaciones y

Sistemas Telem´aticos y Computaci´on

Diciembre de 2020



GSyC - 2020 Control de Tr´afico en Linux 1

©2019 Grupo de Sistemas y Comunicaciones.

Algunos derechos reservados.

Este trabajo se distribuye bajo la licencia

Creative Commons Attribution Share-Alike

disponible en http://creativecommons.org/licenses/by-sa/3.0/es

GSyC - 2020 Control de Tr´afico en Linux 2

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

GSyC - 2020 Control de Tr´afico en Linux 3

Introducci´on: control de tr´afico en Linux

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

GSyC - 2020 Control de Tr´afico en Linux 4

Introducci´on: control de tr´afico en Linux

Control de tr´afico en Linux

tc (*traffic control*) es una herramienta que permite realizar el control de tr´afico en Linux.

tc utiliza disciplinas de colas, denominadas qdisc (*queueing discipline*).

Cada vez que el kernel tiene que enviar un paquete a una interfaz, se encola en la qdisc configurada en dicha interfaz. El kernel trata de sacar el m´aximo n´umero de paquetes de dicha qdisc para entreg´arselos al driver de la tarjeta de red. El tipo de qdisc m´as simple es FIFO (First In First Out). Si el driver de la tarjeta de red est´a ocupado, el paquete se

quedar´a guardado en la cola.

GSyC - 2020 Control de Tr´afico en Linux 5

Elementos del control de tr´afico

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

GSyC - 2020 Control de Tr´afico en Linux 6

Elementos del control de tr´afico

Interfaces de entrada/salida de un router

El control de tr´afico se puede aplicar en la interfaz de entrada de un router o en la interfaz de salida del router.

Cada interfaz de red de un router puede actuar como entrada de paquetes, para los paquetes que se reciben en dicha interfaz, y como interfaz de salida, para los paquetes que se env´ıan a trav´es de dicha interfaz.

**Router,**

Disciplinas%de%colas% en%las%interfaces%de% entrada%

**Nivel,superior:,TCP,,UDP,,etc**

Disciplinas%de%colas% en%las%interfaces%de% salida%

eth0% qdisc eth1% qdisc

ingress

eth1% qdisc ingress

root

**Demul&plexión**

**de,entrada, Reenvío,**

eth0% qdisc

root

GSyC - 2020 Control de Tr´afico en Linux 7

Elementos del control de tr´afico

Elementos del control de tr´afico: Qdisc/Class/Filter

Qdisc (Disciplina de cola): determina qu´e paquetes se reenviar´an antes que otros. Las hay de 2 tipos:

Disciplina de colas sin clases de tr´afico: ejemplo FIFO.

**qdisc** entrada' salida'

Disciplina de colas con clases de tr´afico: ejemplo PRIO

Class (Clase de tr´afico): especifica las diferentes categor´ıas de

tr´afico. Una clase de tr´afico est´a asociada a una disciplina de

cola (qdisc).

Filter (Filtro de tr´afico): identifica el tr´afico que se

corresponde con cada clase (class).

entrada'

salida'

**filter filter**

**class**

**qdisc qdisc**

**filter qdisc**

**class**

GSyC - 2020 Control de Tr´afico en Linux 8

Elementos del control de tr´afico

Configuraci´on de una disciplina de cola a la entrada/salida de un router

En una determinada interfaz se pueden definir:

Disciplinas de cola de entrada: para los paquetes que entran a

trav´es de dicha interfaz.

| tc qdisc add dev <interfaz> ingress ... |
| --- |

Disciplinas de cola de salida: para los paquetes que salen a

trav´es de dicha interfaz.

| tc qdisc add dev <interfaz> root ... |
| --- |

Por defecto, si no se modifican las disciplinas de cola a la entrada y a la salida de un router, el router aplicar´a una variante de FIFO (pfifo\_fast, FIFO mejorado).

GSyC - 2020 Control de Tr´afico en Linux 9

Elementos del control de tr´afico

Descriptor (*handle*) de una disciplina de cola

Cuando se define una disciplina de cola se le asocia un descriptor (*handle*) para poder referenciarla.

El *handle* est´a formado por 2 n´umeros separados por el caracter ”:”, es decir, X:Y

X : es el n´umero mayor.

Y : es el n´umero menor.

Donde X identifica la disciplina de cola e Y hace referencia a un elemento que depende de esa disciplina de cola (flujos, clases).

El *handle* de la disciplina de la cola de entrada (*ingress*) es ffff: lo que equivale a ffff:0.

| tc qdisc add dev <interfaz> ingress handle ffff: |
| --- |

T´ıpicamente el *handle* de la disciplina de la cola de salida (*root*) es 1: lo que equivale a 1:0.

| tc qdisc add dev <interfaz> root handle 1: ... |
| --- |

GSyC - 2020 Control de Tr´afico en Linux 10

Elementos del control de tr´afico

Borrado de una disciplina de cola

El borrado de disciplinas de cola en la entrada y en la salida de un router se hace de la siguiente forma:

Borrado de la disciplina de la cola de entrada:

| tc qdisc del dev <interfaz> ingress |
| --- |

Borrado de la disciplina de la cola de salida:

| tc qdisc del dev <interfaz> root |
| --- |

GSyC - 2020 Control de Tr´afico en Linux 11

Control de admisi´on para el tr´afico de entrada

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

GSyC - 2020 Control de Tr´afico en Linux 12

Control de admisi´on para el tr´afico de entrada

Control de admisi´on para el tr´afico de entrada (*policing*)

Objetivo: El control de admisi´on para el tr´afico de entrada se utiliza para garantizar que el router s´olo procesa para un

determinado flujo de paquetes de entrada un m´aximo de

ancho de banda.

El control de admisi´on se basa en el modelo TBF.

Definici´on de la disciplina de cola de entrada. Ejemplo, si

los paquetes se reciben en la interfaz eth0:

| tc qdisc add dev eth0 ingress handle ffff: |
| --- |

Definici´on de filtros con limitaci´on de ancho de banda

para cada flujo: se especifican las caracter´ısticas del tr´afico en funci´on de valores de cabecera IP y se indica el ancho de

banda m´aximo y el tama˜no de la cubeta TBF. El flujo queda

identificado por un n´umero menor :n.

| 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 |
| --- |

GSyC - 2020 Control de Tr´afico en Linux 13

Control de admisi´on para el tr´afico de entrada

Ejemplo: control de admisi´on en el tr´afico de entrada (I)

Supongamos que se quiere separar el tr´afico que se ha

recibido en una determinada interfaz, en varios flujos.

En la disciplina de cola de entrada (ffff:) queda almacenado todo el tr´afico recibido en dicha interfaz.

Ahora es necesario aplicar filtros a ese tr´afico para separarlo en *i* flujos (cada flujo queda identificado por ffff:i) a los que asignaremos el ancho de banda que deseemos.

qdisc ingress

ffff:&

Flujo&ffff:1& 1Mbps&

burst=10k&

Flujo&ffff:2& 512&kbps burst=10k&

Flujo&ffff:3& 256&kbps burst=50k&

Flujo&ffff:4& 128&kbps burst=5k&

GSyC - 2020 Control de Tr´afico en Linux 14

Control de admisi´on para el tr´afico de entrada

Ejemplo: control de admisi´on en el tr´afico de entrada (II)

Se define qdisc para la entrada: ingress

Se definen filtros para clasificar el tr´afico: los paquetes que cumplen una condici´on se les aplica un TBF y quedan clasificados dentro de un determinado flujo (*flowid*):

Condici´on: utilizaremos el filtro u32 que comprueba campos de la cabecera IP de los paquetes. Perfil de tr´afico: se configura con *rate* y *burst*.

Flujo: el tr´afico que est´e dentro de la especificaci´on queda clasificado en un flowid.

Acci´on: el tr´afico que supere la especificaci´on puede reclasificarse pasando al siguiente filtro (*continue*) o descartarse (*drop*).

Los filtros se aplican siguiendo el orden dado por el par´ametro prioridad: primero los filtros cuyo valor de prioridad sea menor (n´umeros de prioridad bajos =*>*mayor prioridad). Si un paquete cumple la condici´on del filtro, se le aplica su clasificaci´on o su acci´on asociada. Si no, se pasa al siguiente filtro por orden de prioridad.

eth0%

**DROP& Filter:&¿condición1?&8>&r=1mbit&&&&b=10k&&&&flowid=:1&**

**Filter:&¿condición2?&8>&r=512kbit&&&b=10k&&flowid=:2&**

**Filter:&¿condición3?&8>&r=256kbit&&&&b=50k&&flowid=:3& Filter:&¿condición4?&8>&r=128kbit&&&&b=5k&&&&flowid=:4& qdisc ingress**

**DROP&**

**DROP& DROP&**

GSyC - 2020 Control de Tr´afico en Linux 15

Control de admisi´on para el tr´afico de entrada

Ejemplo: control de admisi´on en el tr´afico de entrada (III)

Ejemplo:

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 *\*

match ip dport 80 *\*

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 *\*

match ip dport 21 *\*

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 50k 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 5k drop flowid :4

GSyC - 2020 Control de Tr´afico en Linux 16

Control del tr´afico de salida

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

GSyC - 2020 Control de Tr´afico en Linux 17

Control del tr´afico de salida PFIFO para el tr´afico de salida

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 18

Control del tr´afico de salida PFIFO para el tr´afico de salida

PFIFO para el tr´afico de salida

FIFO con tama˜no de cola dado por el valor limit en n´umero de paquetes:

| tc qdisc add dev eth0 root pfifo limit 5 |
| --- |

Es una disciplina de colas muy sencilla que requiere poco tiempo de computaci´on.

Si hay congesti´on, FIFO perjudica a TCP debido a que una p´erdida de un paquete en TCP activa los mecanismos de

recuperaci´on de p´erdidas (si la ventana est´a llena, no se puede enviar m´as). Una p´erdida de un paquete UDP no tiene ning´un impacto en la tasa de env´ıo para esta comunicaci´on.

Una r´afaga de uno de los flujos que atraviese una cola FIFO puede llenar la cola y perjudicar al resto de los flujos.

GSyC - 2020 Control de Tr´afico en Linux 19

Control del tr´afico de salida PRIO para el tr´afico de salida

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 20

Control del tr´afico de salida PRIO para el tr´afico de salida

PRIO para el tr´afico de salida (I)

PRIO es una disciplina de cola de prioridades que realiza algunas acciones autom´aticamente. El siguiente comando crea la disciplina de cola 1:0 y autom´aticamente crea 3 clases 1:1, 1:2 y 1:3 de tipo PFIFO.

| tc qdisc add dev eth0 root handle 1: prio |
| --- |

entrada'

salida'

**filter filter**

**class&:1& PFIFO& class&:2& PFIFO&**

**filter**

**class&:3& PFIFO&**

**qdisc&PRIO&**

Si se desea crear un n´umero diferente de prioridades se puede usar el par´ametro: bands <n´um\_prio>.

GSyC - 2020 Control de Tr´afico en Linux 21

Control del tr´afico de salida PRIO para el tr´afico de salida

PRIO para el tr´afico de salida (II)

Es necesario especificar los filtros para clasificar el tr´afico dentro de las clases 1:1, 1:2 y 1:3 del ejemplo. Esta clasificaci´on puede hacerse a trav´es del filtro u32 utilizando la direcci´on IP origen:

| tc filter add dev eth0 parent 1:0 prio 1 protocol ip u32 *\* match ip src 11.0.0.1/32 flowid 1:1  tc filter add dev eth0 parent 1:0 prio 2 protocol ip u32 *\* match ip src 11.0.0.2/32 flowid 1:2  tc filter add dev eth0 parent 1:0 prio 3 protocol ip u32 *\* match ip src 11.0.0.3/32 flowid 1:3 |
| --- |

La disciplina de colas de prioridad puede provocar retrasos y p´erdidas de paquetes en la clase menos prioritaria 1:3 si hay mucho tr´afico en 1:1 y 1:2.

GSyC - 2020 Control de Tr´afico en Linux 22

Control del tr´afico de salida Token Bucket Filter (TBF) para el tr´afico de salida

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 23

Control del tr´afico de salida Token Bucket Filter (TBF) para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Disciplina de cola sin clases, trata a todos los paquetes por igual.

TBF limita la velocidad del tr´afico de salida de una

determinada interfaz de un router a trav´es de los par´ametros: ancho de banda: velocidad de generaci´on tokens

tama˜no de la cubeta: para almacenar tokens que permitir´an

enviar r´afagas que superen el ancho de banda.

latencia: tiempo m´aximo de espera de un paquete por un

token.

Ejemplo: Aplicar TBF en la interfaz de salida eth1 de un

router. Ancho de banda 7Mbps , tama˜no de cubeta 10k (en bytes), latencia 50 ms:

| tc qdisc add dev eth1 root handle 1: tbf rate 7Mbit *\* burst 10k latency 50ms |
| --- |

GSyC - 2020 Control de Tr´afico en Linux 24

Control del tr´afico de salida Encadenar disciplinas de colas: TBF y PRIO

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 25

Control del tr´afico de salida Encadenar disciplinas de colas: TBF y PRIO

TBF y PRIO

TBF limita la velocidad del tr´afico de salida de una

determinada interfaz para todos los flujos que lo atraviesan.

Se puede definir una disciplina de cola PRIO hija de TBF para que el tr´afico adem´as de estar limitado por TBF los paquetes se cursen atendiendo a las prioridades definidas en PRIO:

| tc qdisc add dev eth1 root handle 1: tbf rate 7Mbit *\* burst 10k latency 50 ms  tc qdisc add dev eth1 parent 1:0 handle 10:0 prio  tc filter add dev eth1 parent 10:0 prio 1 protocol ip u32 *\* match ip src 101.0.0.10/32 flowid 10:1  tc filter add dev eth1 parent 10:0 prio 2 protocol ip u32 *\* match ip src 102.0.0.20/32 flowid 10:2  tc filter add dev eth1 parent 10:0 prio 3 protocol ip u32 *\* match ip src 103.0.0.30/32 flowid 10:3 |
| --- |

GSyC - 2020 Control de Tr´afico en Linux 26

Control del tr´afico de salida Hierarchical Token Bucket (HTB)

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 27

Control del tr´afico de salida Hierarchical Token Bucket (HTB)

Hierarchical Token Bucket (HTB) para el tr´afico de salida

TBF se utiliza cuando se desea que todo el tr´afico que sale por una determinada interfaz tenga unas caracter´ısticas

determinadas en cuanto a ancho de banda, tama˜no de cubeta y latencia.

HTB es necesario cuando se quiere clasificar el tr´afico que sale por una determinada interfaz en varias clases, cada uno de ellas con unas caracter´ısticas diferentes de ancho de banda. Si alguna clase no utiliza todo el ancho de banda que se le ha definido, dependiendo de la configuraci´on, se podr´ıa utilizar dicho ancho de banda para alg´un otra clase que lo necesite.

GSyC - 2020 Control de Tr´afico en Linux 28

Control del tr´afico de salida Hierarchical Token Bucket (HTB)

Ejemplo de Hierarchical Token Bucket (HTB) para el tr´afico de salida

Definici´on de HTB con ancho de banda m´aximo de salida de 10Mbps de la interfaz eth0. Se quiere repartir este ancho de banda:

usuarioA (4Mbps): lo usar´a para tr´afico HTTP (3Mbps) y tr´afico de correo (1Mbps)

el usuarioB (6Mbps)

**HTB$**

**1:0$**

filter10Mbps&

**1:1$**

filter

filter

&4&Mbps&

**1:2$ 1:12$**

**1:10$ 1:11$**

&3&Mbps& &1&Mbps&

&6&Mbps&

GSyC - 2020 Control de Tr´afico en Linux 29

Control del tr´afico de salida Hierarchical Token Bucket (HTB)

Ejemplo de Hierarchical Token Bucket (HTB) para el tr´afico de salida

tc qdisc add dev eth0 root handle 1:0 htb

tc class add dev eth0 parent 1:0 classid 1:1 htb rate 10Mbit

tc class add dev eth0 parent 1:1 classid 1:2 htb rate 4Mbit ceil 10Mbit

tc class add dev eth0 parent 1:2 classid 1:10 htb rate 3Mbit ceil 10Mbit tc class add dev eth0 parent 1:2 classid 1:11 htb rate 1Mbit ceil 10Mbit tc class add dev eth0 parent 1:1 classid 1:12 htb rate 6Mbit ceil 10Mbit

tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 *\*

match ip src 1.1.1.1 *\*

match ip dport 80 0xffff *\*

flowid 1:10

tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 *\*

match ip src 1.1.1.1 *\*

match ip dport 25 0xffff *\*

flowid 1:11

tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 *\*

match ip src 2.2.2.2 *\*

flowid 1:12

GSyC - 2020 Control de Tr´afico en Linux 30

Control del tr´afico de salida Hierarchical Token Bucket (HTB)

Unidades para tc

Ancho de banda

| kbps  mbps  kbit  mbit  bps | Kilobytes per second Megabytes per second Kilobits per second Megabits per second Bytes per second |
| --- | --- |

Cantidad de datos

| k  mb o m kbit  mbit  b | Kilobytes  Megabytes Kilobits  Megabits  Bytes |
| --- | --- |

Tiempo

| s, sec o secs  ms, msec o msecs us, usec, usecs | Segundos  Milisegundos  Microsegundos |
| --- | --- |

GSyC - 2020 Control de Tr´afico en Linux 31

Iperf: Generador de tr´afico

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

GSyC - 2020 Control de Tr´afico en Linux 32

Iperf: Generador de tr´afico

Iperf: Generador de tr´afico

Iperf es una aplicaci´on que permite generar tr´afico TCP o UDP y medir el ancho de banda que se est´a utilizando entre dos m´aquinas finales.

Iperf funciona en modo cliente/servidor.

Utilizaremos Iperf para generar tr´afico UDP desde el cliente al servidor durante 10 segundos a una determinada velocidad y medir el ancho de banda que en realidad se ha utilizado.

Arrancaremos Iperf de la siguiente forma:

Servidor

| iperf -u -s |
| --- |

Cliente

| iperf -u -c <dirIPServidor> -b <anchoDeBanda> |
| --- |

Donde <anchoDeBanda> es un n´umero n en bps. Se puede

usar nK (para enviar n kilobit por segundo) o nM (para enviar

n megabit por segundo).

GSyC - 2020 Control de Tr´afico en Linux 33

Iperf: Generador de tr´afico

Resultado de la ejecuci´on de Iperf

Resultado de ejecuci´on de Iperf en 12.0.0.30, en modo servidor:

| iperf -s -u  ---------------------------------------------------  Server listening on UDP port 5001  Receiving 1470 byte datagrams  UDP buffer size: 108 KByte (default)  ---------------------------------------------------  [3] local 12.0.0.30 port 5001 connected with 11.0.0.10 port 32768 [3] 0.0- 9.7 sec 25.2 MBytes 21.8 Mbits/sec 0.289 ms 4940/22916 (22 %) [3] 0.0- 9.7 sec 1 datagrams received out-of-order |
| --- |

Resultado de ejecuci´on de Iperf en 11.0.0.10, en modo cliente:

| iperf -u -c 12.0.0.30 -b 100M  ---------------------------------------------------  Client connecting to 12.0.0.30, UDP port 5001  Sending 1470 byte datagrams  UDP buffer size: 108 KByte (default)  ---------------------------------------------------  [3] local 11.0.0.10 port 32768 connected with 12.0.0.30 port 5001 [3] 0.0- 10.0 sec 32.1 MBytes 26.9 Mbits/sec  [3] Sent 22917 datagrams  [3] Server Report:  [3] 0.0- 9.7 sec 25.2 MBytes 21.8 Mbits/sec 0.289 ms 4940/22916 (22 %) [3] 0.0- 9.7 sec 1 datagrams received out-of-order |
| --- |

GSyC - 2020 Control de Tr´afico en Linux 34

Wireshark

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

GSyC - 2020 Control de Tr´afico en Linux 35

Wireshark

Gr´aficas de ancho de banda (I)

Seleccionar en el men´u de Wireshark Statistics *− >*

IO Graphs.

En el panel inferior, a˜nadir filtros seg´un los criterios que se deseen. Marcar con un tic a la izquierda del filtro para

visualizar los paquetes filtrados en la gr´afica.

Para distinguir el tr´afico de varias fuentes la condici´on de filtrado en Display filter puede incluir la comprobaci´on de la direcci´on IP y puerto TCP/UDP. Ejemplo:

ip.src==11.0.0.1 and udp.dstport==5001

Si no se escribe nada en la condici´on de filtrado, se muestra la gr´afica de todo el tr´afico incluido en la captura.

GSyC - 2020 Control de Tr´afico en Linux 36

Wireshark

Gr´aficas de ancho de banda (II)

Es importante seleccionar adecuadamente las unidades que muestra la gr´afica. Para las pruebas que vamos a realizar en las pr´acticas se recomienda:

Color: Uno diferente para cada filtro

Style: Line

Y Axis: Bits

Interval (com´un para todos los filtros): 1 sec

GSyC - 2020 Control de Tr´afico en Linux 37

Wireshark

Campo DS en una captura de Wireshark

GSyC - 2020 Control de Tr´afico en Linux 38

Wireshark

Gr´aficas de ancho de banda en diferentes clases DiffServ (I)

Seleccionar en el men´u de Wireshark Statistics *− >*

IO Graphs.

En el panel inferior, a˜nadir filtros seg´un los criterios que se deseen. Marcar con un tic a la izquierda del filtro para

visualizar los paquetes filtrados en la gr´afica.

Para filtrar una clase de tr´afico puede usarse como condific´on de filtrado alguna de las siguientes (cualquier de ellas servir´ıa para filtrar el tr´afico EF):

ip.dsfield==0xb8

ip.dsfield.dscp==46

ip.dsfield.dscp==0x2e

Para distinguir el tr´afico de varias fuentes la condici´on de

filtrado puede completarse con la comprobaci´on de la direcci´on IP. Ejemplo: ip.dsfield==0x28 and ip.src==11.0.0.1

GSyC - 2020 Control de Tr´afico en Linux 39

Wireshark

Gr´aficas de ancho de banda en diferentes clases DiffServ GSyC - 2020 Control de Tr´afico en Linux 40

DiffServ

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

GSyC - 2020 Control de Tr´afico en Linux 41

DiffServ Marcar el tr´afico

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 42

DiffServ Marcar el tr´afico

Marcado del tr´afico DiffServ en la cabecera IPv4

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

0 4 8 16 31

!"#$%&'(( )\*'+%,-.(

/01"/"#0(( 23\*(."($"#!%/%\*( )\*'+%,-.(,\*,0)(.0,0+#040(

%."'25/0.\*#(.")(.0,0+#040( #"$(67(87( *!"#$%&*."(9#0+4"',0/%&'(

::;((

P

I

<2"43\*(."(!%.0=( 3#\*,\*/\*)\*( *'($')#\*+*(."()0(/01"/"#0(

a

r

e

c

e

.%#"//%&'(>?(\*#%+"'(

b

a

C

.%#"//%&'(>?(."$2'\*(

\*3/%\*'"$((

PI

s

o

ta

D

6@:AB(>?(

GSyC - 2020 Control de Tr´afico en Linux 43

DiffServ Marcar el tr´afico

Marcado del tr´afico 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 recibir´a 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 notificaci´on de congesti´on (Explicit Congestion Notification, ECN). ECN es utilizado conjuntamente por los extremos de una conexi´on TCP y los routers intermedios que utilizan en sus colas *Active Queue Management* (monitorizar el llenado de sus buffers santes de que se produzca la congesti´on).

DSCP%(6%bits)%

**DS5$ DS4$ DS3$ DS2$ DS1$ DS0$ ECN$ ECN$**

Prio:%%

GSyC - 2020 Control de Tr´afico en Linux 44

DiffServ Marcar el tr´afico

Comportamiento por salto (PHB)

Cada clase de tr´afico (DSCP) est´a 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 reenv´ıo:

Reserva de recursos: buffer y ancho de banda.

Caracter´ısticas del tr´afico: retardo y p´erdidas.

DiffServ no especifica las caracter´ısticas concretas del tr´afico en cada una de las clases, s´olo proporciona la clasificaci´on con diferentes c´odigos DSCP.

GSyC - 2020 Control de Tr´afico en Linux 45

DiffServ Marcar el tr´afico

PHBs recomendados

Aunque potencialmente pueden definirse 64 clases de tr´afico 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 p´erdidas, baja latencia, bajo jitter

VA (Voice Admit): DSCP 44=101100*B*

Similar a EF, pero puede incluir un mecanismo de control de admisi´on

AF (Assured Forwarding):

Se proporciona cierta garant´ıa de entrega mientras el tr´afico 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 p´erdida de los paquetes. La

combinaci´on de las 4 clases con las 3 probabilidades de p´erdida dan 12

PHBs: AF11, AF12, AF13, AF21, AF22, AF23, AF31, AF32, AF33, en los

que se cumple que la probabilidad de p´erdida 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 tr´afico que no es DiffServ)

GSyC - 2020 Control de Tr´afico en Linux 46

DiffServ Marcar el tr´afico

Marcas en los paquetes: Valores DS y DSCP m´as 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).

**Probabilidad de**

**DSCP (6bits) DS (8 bits)**

**Clase DF** 000 000 = 0x00 = 0 0000 0000 = 0x00

**descarte de tráfico**

**Clase AF1**

**Clase AF2**

**Clase AF3**

**Clase AF4**

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 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 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 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

GSyC - 2020 Control de Tr´afico en Linux 47

DiffServ DSMARK

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 48

DiffServ DSMARK

DSMARK: qdisc

DSMARK es la qdisc que se utiliza para la clasificaci´on de paquetes seg´un 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.

Clasificaci´on del tr´afico seg´un el campo DS:

Routers del n´ucleo de DiffServ: clasifican los paquetes

atendiendo al contenido de DS que traen para realizar su

tratamiento seg´un diferentes disciplinas de cola.

GSyC - 2020 Control de Tr´afico en Linux 49

DiffServ DSMARK

DSMARK: qdisc y sus clases

Una qdisc DSMARK crea autom´aticamente 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 m´axima 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 n´umero de clases igual a 23 = 8

podr´a definir como m´aximo 7 valores DSCP diferentes, uno para cada una de sus clases:

| tc qdisc add dev eth1 root handle |
| --- |

Una vez creada una qdisc DSMARK se habr´an creado autom´aticamente sus clases asociadas. Cada clase tiene un descriptor X:Y, donde: X es el valor del n´umero mayor de la clase qdisc a la que pertenece

Y es el n´umero menor que distingue a cada una de las clases

Por ejemplo, para definir como m´aximo 7 clases de tr´afico dentro de una qdisc 1:0 se usar´an los siguientes descriptores: 1:1, 1:2, 1:3, 1:4, 1:5, 1:6,

1:0 dsm

1:7. El n´umero mayor coincide en todas ellas, ya que todas pertenecen a la qdisc 1.

GSyC - 2020 Control de Tr´afico en Linux 50

DiffServ Router Frontera

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 51

DiffServ Router Frontera

Router Frontera: marcado de paquetes

Para marcar el tr´afico se usa la disciplina DSMARK:

| tc qdisc add dev eth1 root handle 1:0 dsmark |
| --- |

Esta disciplina ha creado 7 clases, se desea que los paquetes de cada clase queden marcados en el campo DS (6 bits ´m´as significativos). Para ello, se realizan las 2 siguientes operaciones en orden:

1 operaci´on & (AND) de bits con una m´ascara para borrar los bits DSCP (6 bits) y mantener ECN (2 bits)

2 operaci´on | (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 |
| --- |

indi

mask

mask#0x03#

0000#0011#

DS#=#0000#0000#=#0x00# &#

value#0x68#

0110#1000#

DS#=#0000#0000#=#0x00# I#

DS#=#0110#1000#=#0x68#

GSyC - 2020 Control de Tr´afico en Linux 52

DiffServ Router Frontera

Campo tc index

Al recibirse un paquete IP en un router Linux, 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 ingres´o en la cola qdisc ingress:

Paquetes(tc\_index=1(

**Filter flowid=:1'**

**CONTINUE'**

**Filter flowid=:2'**

**CONTINUE'**

**Filter flowid=:3'**

**DROP'**

Paquetes(tc\_index=2( Paquetes(tc\_index=3( Paquetes(tc\_index=4(

**Filter flowid=:4' Qdisc ingress**

**DROP'**

GSyC - 2020 Control de Tr´afico en Linux 53

DiffServ Router Frontera

Campo tc index en la disciplina ingress

| tc qdisc add dev eth0  tc filter add dev eth0  protocol ip prio 4  match ip src 11.0.0.100  police rate 1mbit  tc filter add dev eth0  protocol ip prio 5  match ip src 11.0.0.100  police rate 512kbit  tc filter add dev eth0  protocol ip prio 6  match ip src 11.0.0.100  police rate 256kbit  tc filter add dev eth0  protocol ip prio 6  match ip src 0.0.0.0/0  police rate 128kbit |
| --- |

i

Si un paquete queda clasificado en flowid :2, el campo tc\_index almacenar´ıa el valor 2.

Esta informaci´on es la que utiliza posteriormente el filtro tcindex dentro de otras qdisc de dicho router para dar tratamiento diferenciado a los paquetes.

u32

burst

u32

bur

u32

bur

u32

bur

GSyC - 2020 Control de Tr´afico en Linux 54

DiffServ Router Frontera

Filtro tcindex en el Router Frontera

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  tc class change dev eth1 classid 1:1 dsmark  tc class change dev eth1 classid 1:2 dsmark  tc class change dev eth1 classid 1:3 dsmark  tc class change dev eth1 classid 1:4 dsmark  tc filter add dev eth1 parent 1:0 protocol  handle 1 tcindex classid 1:1  tc filter add dev eth1 parent 1:0 protocol  handle 2 tcindex classid 1:2  tc filter add dev eth1 parent 1:0 protocol  handle 3 tcindex classid 1:3  tc filter add dev eth1 parent 1:0 protocol  handle 4 tcindex classid 1:4 |
| --- |

indi

mask mask mask mask

ip pri ip pri ip pri ip pri

GSyC - 2020 Control de Tr´afico en Linux 55

DiffServ Router Frontera

Camino de un paquete en un Router Frontera

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 tr´afico en las clases de DSMARK, que marcar´an los paquetes con un determinado valor de DS.

**Router'Linux'**

**Filter**

**flowid=:1'**

**CONTINUE'**

**Filter**

**flowid=:2'**

Paquetes(( tc\_index=1(

Paquetes( tc\_index=2(

E( N( C( A(

Paquetes( marcados( con(

tc\_index

Filter fw

tc\_index=1(

Filter fw(( tc\_index=2(

**Classid=1: qdisc 1'DS=0x28'**

**Classid=1: qdisc 2'DS=0x48'**

**Filter**

**CONTINUE'**

Paquetes(

Filter fw

**Classid=1: qdisc**

**flowid=:3' Filter**

**flowid=:4'**

**DROP' DROP'**

tc\_index=3( Paquetes( tc\_index=4(

M( I(

tc\_index=3(

Filter fw

tc\_index=4(

**3'DS=0x68'**

**Classid=1: qdisc 4'DS=0x88'**

**Qdisc ingress**

N( A

**Qdisc'DSMARK'**

Interfaz(de(entrada( Interfaz(de(salida(

M(

GSyC - 2020 Control de Tr´afico en Linux 56

DiffServ Router Frontera

Filtro tcindex: Router Frontera (VI) **ffff:0$ ingress**

**11.0.0.100$ 1mbit$**

**Resto$$**

**128kbit$**

**11.0.0.100$**

**256kbit$**

**11.0.0.100$**

**512kbit$**

**ffff:1$ ffff:2$ ffff:3$ ffff:4$**

handle=1) tcindex

handle=2) tcindex

handle=3) tcindex

handle=4) tcindex

ENCAMINAMIENTO) **dsmark 1:0$**

**handle=1$**

**tcindex**

**handle=2$**

**tcindex**

**handle=4$**

**tcindex**

**handle=3$**

**tcindex**

**1:1$ 1:2$ 1:3$ 1:4$**

**0x28$**

**0x48$ 0x68$ 0x88$**

GSyC - 2020 Control de Tr´afico en Linux 57

DiffServ Router N´ucleo

Contenidos

1 Introducci´on: control de tr´afico en Linux

2 Elementos del control de tr´afico

3 Control de admisi´on para el tr´afico de entrada

4 Control del tr´afico de salida

PFIFO para el tr´afico de salida

PRIO para el tr´afico de salida

Token Bucket Filter (TBF) para el tr´afico de salida

Encadenar disciplinas de colas: TBF y PRIO

Hierarchical Token Bucket (HTB)

5 Iperf: Generador de tr´afico

6 Wireshark

7 DiffServ

Marcar el tr´afico

DSMARK

Router Frontera

Router N´ucleo

GSyC - 2020 Control de Tr´afico en Linux 58

DiffServ Router N´ucleo

Campo tc index

En el router del n´ucleo, 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 par´ametro: set tc index

| tc qdisc add dev eth1 root handle |
| --- |

1:0 dsm

GSyC - 2020 Control de Tr´afico en Linux 59

DiffServ Router N´ucleo

Disciplinas de cola encadenadas DSMARK+HTB

Como se copia el campo DS entero, el filtro deber´a utilizar s´olo los 6 bits m´as significativos del campo DS, es decir, el DSCP, para clasificar los paquetes. Es necesario extraer esos 6 bits (operaci´on & con mask 1111 1100 = 0xfc y desplazamiento a la derecha 2 bits) del campo DSCP:

| tc filter add dev eth1 parent 1:0 protocol  tcindex mask 0xfc shift 2 |
| --- |

A continuaci´on se puede usar otra disciplina de cola encadenada con DSMARK para realizar la planificaci´on de paquetes que se desee, por ejemplo HTB:

| tc qdisc add dev eth1 parent 1:0 handle  tc class add dev eth1 parent 2:0 classid  tc class add dev eth1 parent 2:1 classid  # Tr´afico AF11=> DS=0x28 =00101000 =>  tc filter add dev eth1 parent 2:0 protocol  handle 0x0a tcindex classid 2:10 |
| --- |

ip pri

2:0 htb

2:1 htb

2:10 htb

DSCP=00101

ip pri

GSyC - 2020 Control de Tr´afico en Linux 60

DiffServ Router N´ucleo

Router del N´ucleo

Ejemplo completo:

tc qdisc add dev eth1 root handle 1:0 dsm

tc filter add dev eth1 parent 1:0 protoco

tcindex mask 0xfc shift 2

tc qdisc add dev eth1 parent 1:0 handle 2

tc class add dev eth1 parent 2:0 classid

tc class add dev eth1 parent 2:1 classid

rate 800kbit ceil 1Mbit

# Tr´afico AF11=> DS=0x28 =00101000 => DS

tc filter add dev eth1 parent 2:0 protoco

handle 0x0a tcindex classid 2:10

GSyC - 2020 Control de Tr´afico en Linux 61