Implementação de QoS em um roteador Linux

Redes Multimídia

Prof. Emerson Ribeiro de Mello

Instituto Federal de Santa Catarina – IFSC
campus São José
mello@ifsc.edu.br

28 de setembro de 2011

Sumário

1. Teoria sobre o TC
2. Prática com o TC
3. Classificação e marcação com iptables

Linux atuando como um roteador

O pacote **iproute2**

Consiste em uma coleção de ferramentas para controlar o tráfego em redes TCP/IP

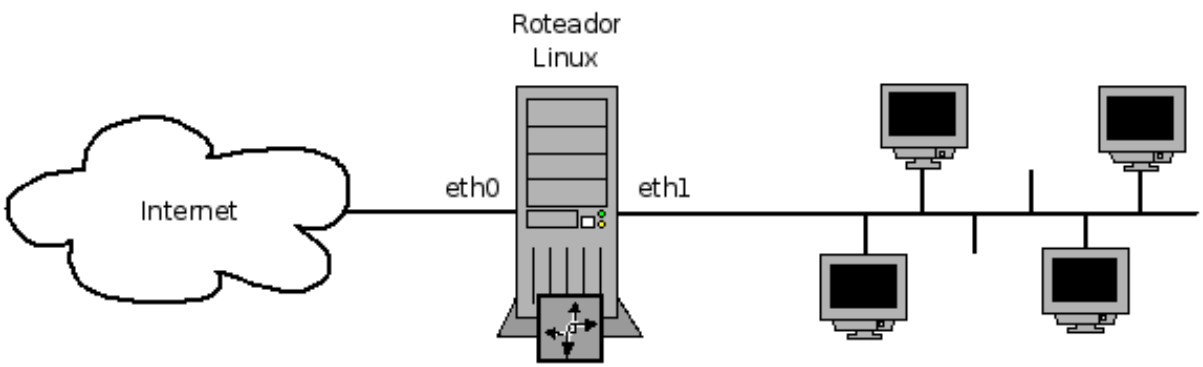
- A ferramenta *Traffic Control* (**tc**) é destinada ao controle de tráfego, permitindo a aplicação de políticas de controle para Qualidade de Serviço
- Permite a combinação de disciplinas de enfileiramento, classes e filtros para os pacotes destinados à **interface de saída**

Linux atuando como um roteador

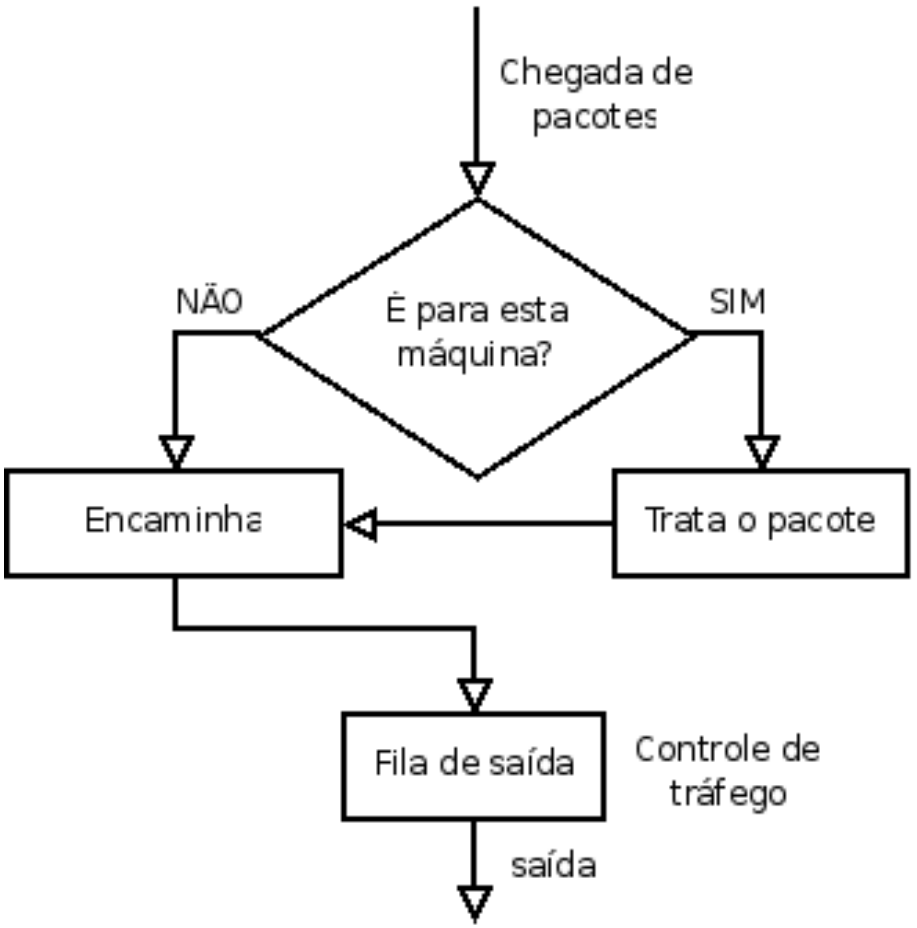
O pacote **iproute2**

Consiste em uma coleção de ferramentas para controlar o tráfego em redes TCP/IP

- A ferramenta *Traffic Control* (**tc**) é destinada ao controle de tráfego, permitindo a aplicação de políticas de controle para Qualidade de Serviço
- Permite a combinação de disciplinas de enfileiramento, classes e filtros para os pacotes destinados à **interface de saída**

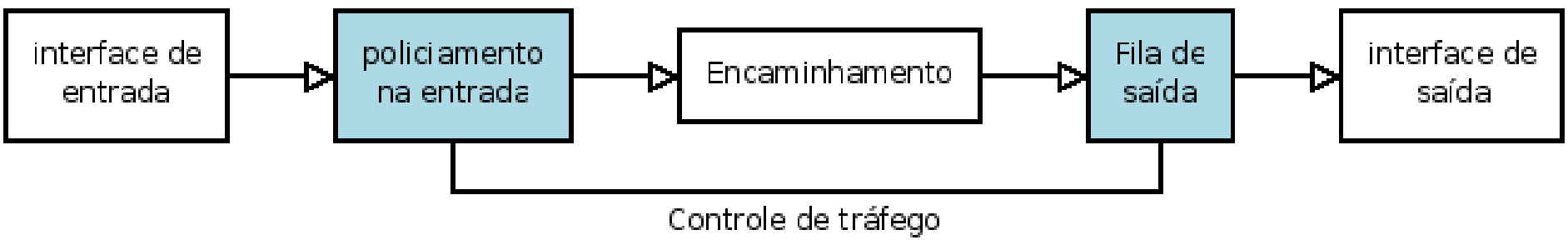


Funcionamento do código de rede do Linux



Aplicação do controle de tráfego

- Policiamento de pacotes no ingresso
  - Pacotes indesejáveis são descartados
- Controle de filas na interface de saída
  - Pacotes podem ser descartados, atrasados ou priorizados



Componentes do Controle de Tráfego

- **Queueing disciplines – qdisc**
  - Disciplinas de enfileiramento
  - Indica como os pacotes na fila serão tratados. Ex: FIFO
- **Classes**
  - Forma de classificação e priorização de pacotes.
- **Filters**
  - Determina para quais classes os pacotes deverão ser encaminhados
- **Policiers**
  - Controla o tráfego que passa pelos filtros para evitar abusos

Componentes do Controle de Tráfego (cont.)

- Uma **qdisc** pode:
  - Armazenar pacotes
  - Conter **classes**

Componentes do Controle de Tráfego (cont.)

- Uma **qdisc** pode:
  - Armazenar pacotes
  - Conter **classes**
- Uma **classe** não armazena pacotes.
  - Dentro de uma classe só pode existir uma **qdisc**

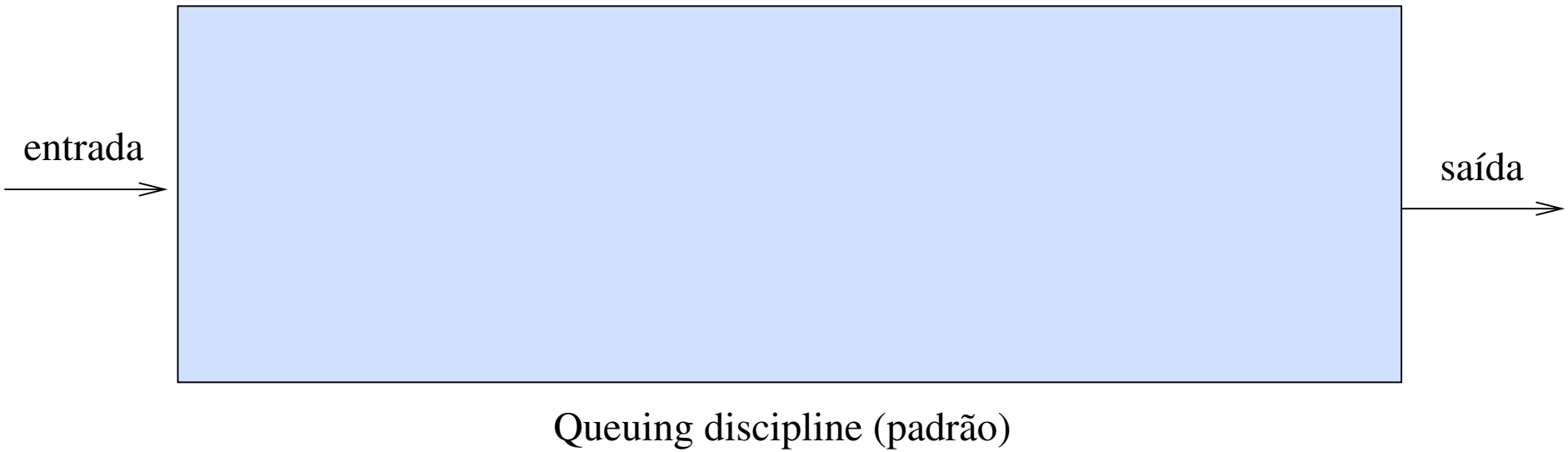
Componentes do Controle de Tráfego (cont.)

- Uma **qdisc** pode:
  - Armazenar pacotes
  - Conter **classes**
- Uma **classe** não armazena pacotes.
  - Dentro de uma classe só pode existir uma **qdisc**
- Ao criar uma **qdisc** composta por algumas **classes**, as **qdisc** destas classes terão por padrão a disciplina `FIFO`
  - É possível alterar para outras disciplinas

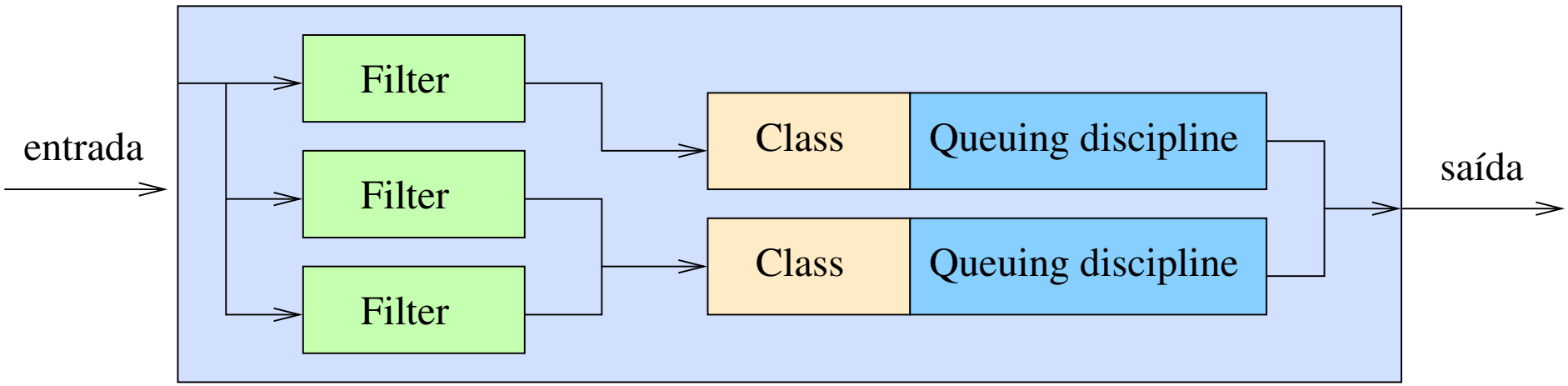
Componentes do Controle de Tráfego (cont.)

- Uma **qdisc** pode:
  - Armazenar pacotes
  - Conter **classes**
- Uma **classe** não armazena pacotes.
  - Dentro de uma classe só pode existir uma **qdisc**
- Ao criar uma **qdisc** composta por algumas **classes**, as **qdisc** destas classes terão por padrão a disciplina `FIFO`
  - É possível alterar para outras disciplinas
- O objetivo dos **filtros** é de classificar os pacotes, encaminhando-os para as respectivas **classes** de acordo as informações contidas no cabeçalho IP

Componentes do Controle de Tráfego (cont.)



Componentes do Controle de Tráfego (cont.)



Queuing discipline (padrão)

Componentes do Controle de Tráfego (cont.)

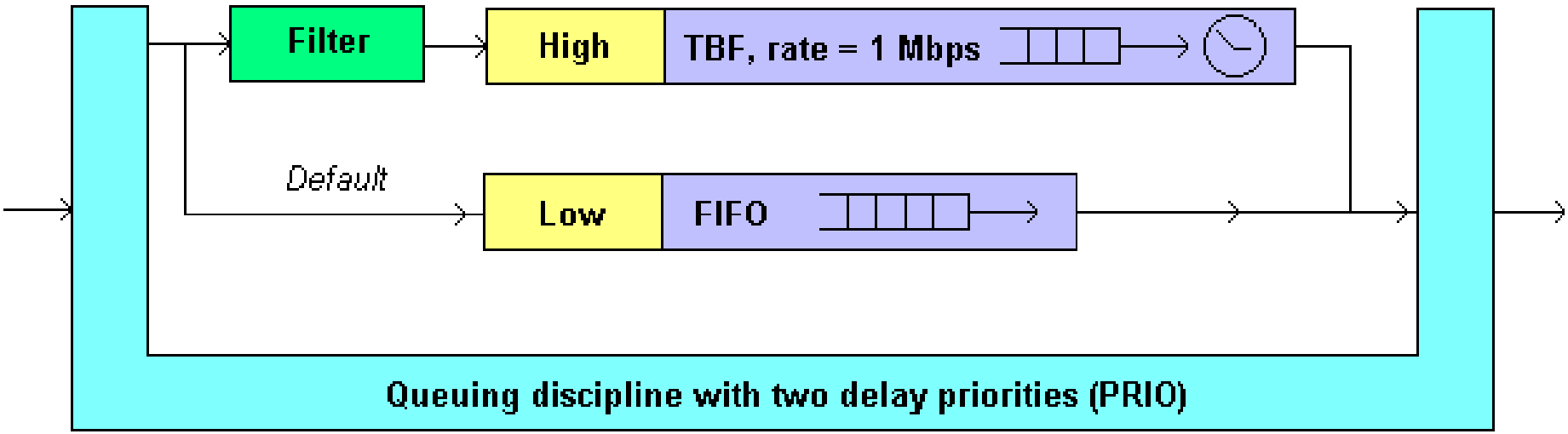


Figure 2.1.8

TBF = *Token Bucket Filter* (Balde de Fichas)

Algoritmos para enfileiramento de pacotes no Linux (qdisc)

Sem classificação (não possui classes)

- First-In-Firt-Out (FIFO) – Primeiro que chega, é o primeiro que sai
- Token Bucket Flow (TBF) – Balde de fichas
- Stochastic Fair Queueing (SFQ) – Enfileiramento justo estocástico
- Random Early Detection (RED) – Detecção aleatória antecipada
- Diff-Serv Marker (DS_MARK) – Marcação DiffServ

Algoritmos para enfileiramento de pacotes no Linux (qdisc)

Com classificação (possui classes e permite o uso de filtros)

- Priority Queueing (PRIO) – Enfileiramento com prioridades
- Class Based Queueing (CBQ) – Enfileiramento baseado em classes
- Hierarchical Token Bucket (HTB) – Balde de fichas hierárquico

Disciplina padrão do Linux: **pfifo_fast**

- Pacotes são armazenados em 3 "bandas" (`0,1,2`)
  - Enquanto houver pacotes aguardando na banda 0, pacotes na banda 1 não serão processados. O mesmo ocorre da banda 1 para a banda 2
  - Cada banda possui enfileiramento `FIFO`
- A marcação no campo **Tipo de Serviço** do cabeçalho IP indica em qual banda o pacote será armazenado
  - Pacotes que requerem atraso mínimo são direcionados para a banda 0
- Possui semelhanças com a disciplina **PRIO**, porém a **pfifo_fast** é *sem classificação*

pfifo_fast – Como os pacotes são priorizados

```
   0     1     2     3     4     5     6     7
+-----+-----+-----+-----+-----+-----+-----+-----+
|                 |                       |     |
|   PRECEDENCIA   |          TOS          |  0  |
|                 |                       |     |
+-----+-----+-----+-----+-----+-----+-----+-----+

# os 4 bits do TOS

Binario Decimal Significado
----------------------------------------
1000      8     Minimize delay (md)
0100      4     Maximize throughput (mt)
0010      2     Maximize reliability (mr)
0001      1     Minimize monetary cost (mmc)
0000      0     Normal Service
```

pfifo_fast – Como os pacotes são priorizados

TOS	Bits	Significado	Prioridade	Banda
0x0	0	Normal Service	0 Best Effort	1
0x2	1	Minimize Monetary Cost	1 Filler	2
0x4	2	Maximize Reliability	0 Best Effort	1
0x6	3	mmc+mr	0 Best Effort	1
0x8	4	Maximize Throughput	2 Bulk	2
0xa	5	mmc+mt	2 Bulk	2
0xc	6	mr+mt	2 Bulk	2
0xe	7	mmc+mr+mt	2 Bulk	2
0x10	8	Minimize Delay	6 Interactive	0
0x12	9	mmc+md	6 Interactive	0
0x14	10	mr+md	6 Interactive	0
0x16	11	mmc+mr+md	6 Interactive	0
0x18	12	mt+md	4 Int. Bulk	1
0x1a	13	mmc+mt+md	4 Int. Bulk	1
0x1c	14	mr+mt+md	4 Int. Bulk	1
0x1e	15	mmc+mr+mt+md	4 Int. Bulk	1

Stochastic Fair Queueing (SFQ)

Enfileiramento justo estocástico

Projetado para garantir o acesso justo aos recursos. Cada fluxo possui uma quota para transmitir

Stochastic Fair Queueing (SFQ)

Enfileiramento justo estocástico

Projetado para garantir o acesso justo aos recursos. Cada fluxo possui uma quota para transmitir

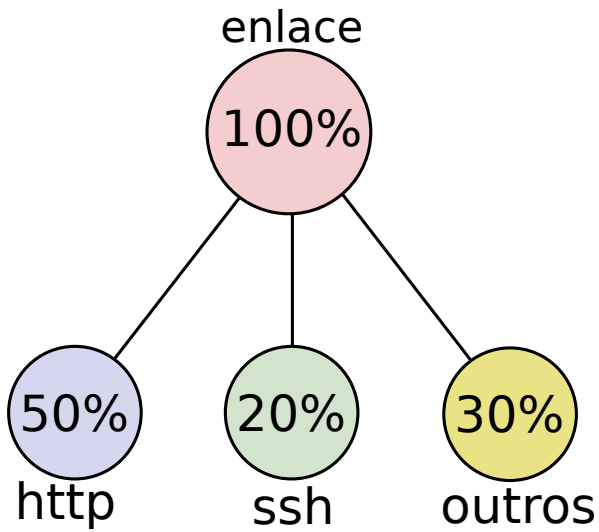
- Existe um número pequeno de filas **FIFO**
  - As filas são atendidas através da varredura cíclica
- A aplicação de uma função de resumo (hash) sobre cada pacote indica em qual fila este deverá ser armazenado
  - São usadas as informações: endereço de origem, porta de origem e endereço de destino
- É possível que pacotes oriundos de diferentes origens possam ser armazenados em uma mesma fila
  - Torna a solução escalável

Priority Queueing (PRIO) - Enfileiramento prioritário

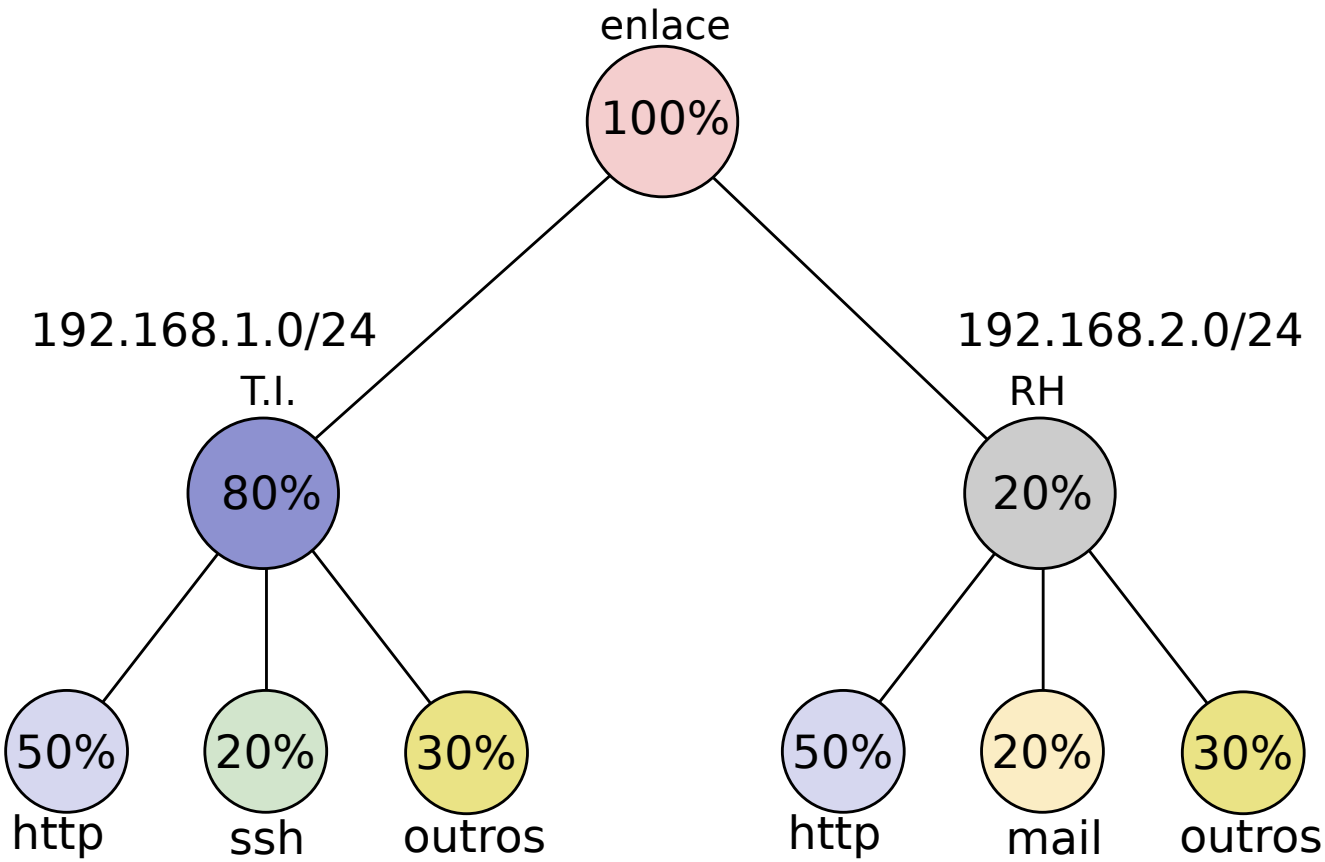
- Constituído por diversas filas, cada qual com uma prioridade diferente
- Após a classificação os pacotes são adicionados em filas FIFO
- Vantagem:
  - Uma forma simples para implementar diferenciação de pacote
- Desvantagens:
  - Pacotes de filas com alta prioridade podem impedir o serviço de pacotes com menor prioridade

Hierarchical Token Bucket (HTB) - Balde de fichas hierárquico

- Constituído por baldes de fichas organizados em uma hierarquia
- Classes HTB possui dois parâmetros principais:
  - **rate**: taxa de transmissão garantida
  - **ceil**: taxa máxima concedida a classe
  - (*rate* – *ceil*): é a taxa de transmissão "emprestada" da classe pai



Hierarchical Token Bucket (HTB) (cont.)



Aplicativo **tc**: Exemplos

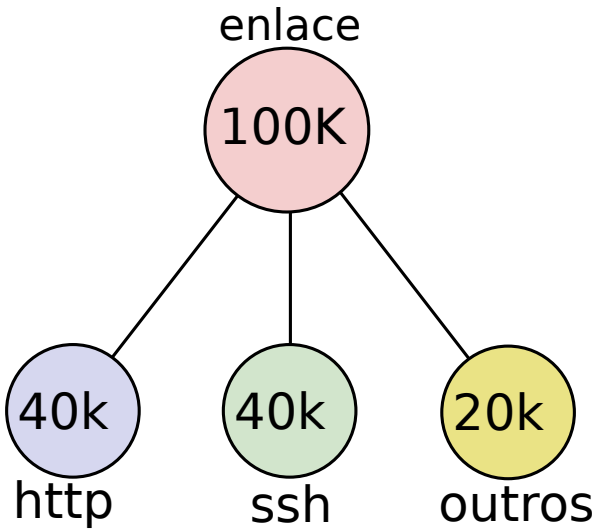
```
1  # para mostrar a disciplina utilizada
2  tc qdisc show dev eth0
3
4  # Adicionando uma fila padrao na raiz da interface de rede
5  tc qdisc add dev eth0 root handle 1:0 htb default 12
6
7  # alterando uma disciplina
8  tc qdisc change dev eth0 root tbf rate 220 latency 50ms burst 1540
9
10 # Monitorar estado
11 tc -s qdisc show dev eth0
12 # Apaga a disciplina
13 tc qdisc del dev eth0 root
```

Parâmetros para o algoritmo HTB – linha 5

- `handle 1:0` - Identificador da `qdisc` (1) e (0) identificador da `classe` pertencente a `qdisc`
- `default 12` - Classe padrão para receber pacotes não classificados

Exemplo 1: Aplicações com diferentes taxas

- Enlace com **100Kbps** de taxa de transmissão
  - **40Kbps** para pacotes WWW
  - **40Kbps** para pacotes SSH
  - **20Kbps** para outros

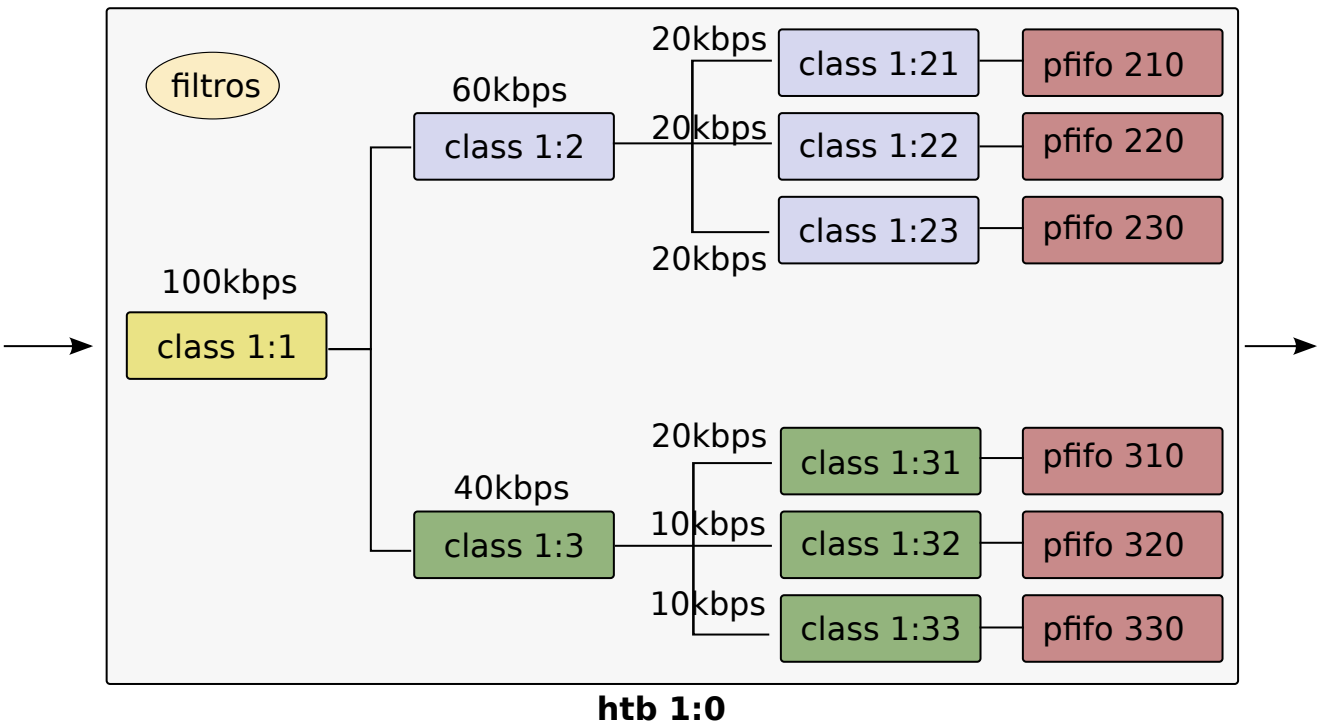


Exemplo 1: (cont.)

```
14  # adicionando a qdisc HTB na raiz da interface
15  tc qdisc add dev eth0 root handle 1:0 htb default 12

17  # criando as classes
18  # classe pai
19  tc class add dev eth0 parent 1:0 classid 1:1 htb rate 100kbps ceil 100kbps

21  # classes filhas
22  tc class add dev eth0 parent 1:1 classid 1:10 htb rate 40kbps ceil 100kbps
23  tc class add dev eth0 parent 1:1 classid 1:11 htb rate 40kbps ceil 100kbps
24  tc class add dev eth0 parent 1:1 classid 1:12 htb rate 20kbps ceil 100kbps

26  # filtros
27  tc filter add dev eth0 protocol ip parent 1:0 prio 1 u32 match ip src 192.168.1/24 match ip dport
        22 0xffff flowid 1:10
28  tc filter add dev eth0 protocol ip parent 1:0 prio 1 u32 match ip src 192.168.1/24 match ip dport
        80 0xffff flowid 1:11
```

Exemplo 2: HTB com dois níveis de hierarquia

- Filtros Máquina 1
  - SSH $\hookrightarrow$ 1:21
  - WWW $\hookrightarrow$ 1:22
  - Outros $\hookrightarrow$ 1:23
- Filtros Máquina 2
  - SSH $\hookrightarrow$ 1:31
  - WWW $\hookrightarrow$ 1:32
  - Outros $\hookrightarrow$ 1:33



Exemplo 2: HTB com dois níveis de hierarquia

- 1º nível de hierarquia de classes

```
30 tc qdisc add dev eth0 root handle 1:0 htb
31 tc class add dev eth0 parent 1:0 classid 1:1 htb rate 100kbps
32 tc class add dev eth0 parent 1:1 classid 1:2 htb rate 60kbps ceil 100kbps
33 tc class add dev eth0 parent 1:1 classid 1:3 htb rate 40kbps ceil 100kbps
```

- 2º nível de hierarquia de classes

```
36 tc class add dev eth0 parent 1:2 classid 1:21 htb rate 20bkps ceil 60kbps
37 tc class add dev eth0 parent 1:2 classid 1:22 htb rate 20bkps ceil 60kbps
38 tc class add dev eth0 parent 1:2 classid 1:23 htb rate 20bkps ceil 60kbps
```

```
41 tc class add dev eth0 parent 1:3 classid 1:31 htb rate 20kbps ceil 40kbps
42 tc class add dev eth0 parent 1:3 classid 1:32 htb rate 10kbps ceil 40kbps
43 tc class add dev eth0 parent 1:3 classid 1:33 htb rate 10kbps ceil 40kbps
```

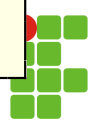
Exemplo 2: HTB com dois níveis de hierarquia

- Adicionando **qdisc pfifo** nas classes

```
44 tc qdisc add dev eth0 parent 1:21 handle 210: pfifo limit 10
45 tc qdisc add dev eth0 parent 1:22 handle 220: pfifo limit 10
46 tc qdisc add dev eth0 parent 1:23 handle 230: pfifo limit 10

48 tc qdisc add dev eth0 parent 1:31 handle 310: pfifo limit 10
49 tc qdisc add dev eth0 parent 1:32 handle 320: pfifo limit 10
50 tc qdisc add dev eth0 parent 1:33 handle 330: pfifo limit 10
```

- Criando os filtros

```
53 tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 match ip dst 192.168.1.1 match ip
       sport 22 0xfff flowid 1:21
54 tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 match ip dst 192.168.1.1 match ip
       sport 80 0xfff flowid 1:22
55 tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 match ip dst 192.168.1.1 flowid
       1:23

57 tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 match ip dst 192.168.1.2 match ip
       sport 22 0xfff flowid 1:31
58 tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 match ip dst 192.168.1.2 match ip
       sport 80 0xfff flowid 1:32
59 tc filter add dev eth0 parent 1:0 protocol ip prio 1 u32 match ip dst 192.168.1.2 flowid
       1:33
```



Classificação de pacotes com iptables

- O iptables pode ser usado na classificação dos pacotes, assumindo assim a tarefa dos **filtros do tc**
- No exemplo abaixo, pacotes com destino a porta 22 (ssh) são colocados na classe **1:10**

```
60  # adicionando HTB na raiz do dispositivo
61  tc qdisc add dev eth0 root handle 1:0 htb default 11

63  # limitando trafego para a interface
64  tc class add dev eth0 parent 1:0 classid 1:1 htb rate 100kbps
65  # adicionando as classes folhas
66  tc class add dev eth0 parent 1:1 classid 1:10 htb rate 20kbps ceil 100
        kbps
67  tc class add dev eth0 parent 1:1 classid 1:11 htb rate 80kbps ceil 100
        kbps

69  # classificando os pacotes com iptables (tabela mangle)
70  iptables -t mangle -A POSTROUTING -p udp --dport 22 -j CLASSIFY --set-
        class 1:10
```

Marcação de pacotes com iptables

- O iptables pode ser usado para marcar pacotes e tal marcação pode então ser usada pelo filtro `fw` do `tc`
- No exemplo abaixo, pacotes com destino a porta 22 (ssh) são marcados com o valor 2 e um filtr do tc irá consumir tal marcação

```
71 # adicionando HTB na raiz do dispositivo
72 tc qdisc add dev eth0 root handle 1:0 htb
73 # limitando trafego para a interface
74 tc class add dev eth0 parent 1:0 classid 1:1 htb rate 100kbps
75 # adicionando as classes folhas
76 tc class add dev eth0 parent 1:1 classid 1:10 htb rate 20kbps ceil 100
       kbps

78 # adicionando filtro para pacotes marcados com o valor 2
79 tc filter add dev eth0 parent 1:0 protocol ip handle 2 fw flowid 1:10

81 iptables -t mangle -A FORWARD -i eth0 -p tcp --dport 22 -j MARK --set-
       mark 2
```