

# QoS

Richard Cesar (cesarric@fel.cvut.cz)

13. května 2011

# Motivace

- V moderních počítačových sítích dnes existuje mnoho služeb s různými parametry a s různými nároky na síť.
- Pokud nijak neřídíme datové toky na síti, může dojít k řadě problémů jako například prodlužování odezvy sítě (latency), kolísání (jitter) nebo může dojít dokonce k výpadku služeb.

## Příklad - soused "tahač"

Máte souseda se kterým sdílíte internetové připojení. Váš soused si k večeru chce stáhnout film a využije k tomu peer-2-peer spojení například skrze protokol Torrent. Během pár minut získá veškeré přenosové pásmo a pro vás se stane internetové připojení zcela nepoužitelné.

# Co je to QoS ?

- **QoS = Quality Of Service**
- Technika využívaná v počítačových sítích, které jsou založeny na principu přepínání paketů.
- Využívá se k řízení datových toků v síti.
- Cílem je, aby v případě zahlcení sítě nedocházelo k narušení dostupnosti služeb, a aby byly tyto služby vždy poskytovány v patřičné kvalitě vzhledem k jejich charakteru a našim požadavkům.

Klasické příklady využití :

- Chceme upřednostnit některé služby před jinými
- Chceme některé služby částečně nebo zcela omezit
- Chceme regulovat přístup uživatelů ke službám

# Základní principy

- Datové toky se řídí sadou pravidel, které jsou určeny správcem sítě.
- Nejsme schopni ovlivnit datový tok v rámci jednoho lokálního segmentu sítě.
- Pravidla se mohou aplikovat pouze na uzlech, které spojují jednotlivé segmenty sítě na úrovni síťové vrstvy (routery).
- Z principu fungování počítačových sítí jsme schopni ovlivnit pouze data, která sami odesíláme. Nikoliv data, které nám někdo pošle.

## Analogie ke klasické poštovní schránce

Nemůžeme ovlivnit, kolik dopisů a balíků nám někdo pošle. Pošta zkrátka doručí vše, co je nám adresováno. Můžeme ale ovlivnit kolik pošty my sami pošleme.

## Příklad zapojení - internetová gateway

- Typicky máme router a na něm dvě rozhraní. Jedno rozhraní je připojeno k našemu poskytovateli připojení a vede tedy na internet. Druhé rozhraní vede do naší sítě.
- Chceme-li ovlivňovat data **stahovaná** z internetu, musíme aplikovat pravidla na rozhraní jdoucí **do naší sítě**, tedy na rozhraní, skrze které tato data odchází z routeru.
- Chceme-li ovlivňovat data **odesílaná** na internet, musíme aplikovat pravidla na rozhraní jdoucí **na internet**, tedy na rozhraní, skrze které tato data odchází z routeru.

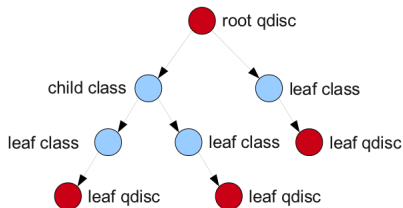


# Implementace v systému Linux

- QoS je v Linuxu implementován přímo v jádře
- Ke konfiguraci se využívají dva základní nástroje, a sice **tc** a **iptables**.
- Nástroje pracují přímo se strukturami v jádře Linuxu a všechny změny jsou tedy provedeny okamžitě po odeslání příkazu.
- Nástroj **tc** (Traffic Control) slouží k definování pravidel QoS.
- Nástroj **iptables** slouží k definování pravidel firewallu, která "označují" jednotlivé pakety pro další zpracování v rámci QoS.

# Implementace v systému Linux

- Pravidla QoS jsou v Linuxu řazena do formy stromu.
- Při zkoumání každého paketu systém prochází strom od kořene a dle pravidel se ho snaží zařadit do konkrétního listu stromu.
- Kořen stromu je navěšen přímo na konkrétní síťové rozhraní, přičemž pro každé rozhraní existuje právě jeden strom pravidel.



# Komponenty QoS

- V zásadě můžeme v rámci stromu pravidel vytvářet tři komponenty : **qdisc**, **class**, **filter**
- Stavěním jednotlivých komponent říkáme systému, jaké plánovací algoritmy mají být použity, určujeme jejich konfiguraci a říkáme, za jakých podmínek mají být použity.

## qdisc

- Plní funkci plánovače.
- Každé síťové rozhraní má alespoň jeden qdisc libovolného typu (defaultně typu First-In-First-Out).

```
tc qdisc add dev eth0 root handle 1: htb default 15
```



# Komponenty QoS

## class

- Existují pouze uvnitř komponenty qdisc.
- Mohou obsahovat další podtřídy nebo právě jeden vnořený qdisc.
- Třída může obsahovat filtry, které rozhodují o dalším zařazení klasifikovaných dat do nižších tříd, případně rozhodují o zahození, či reklasifikaci.
- Takzvané leaf-class neobsahují již žádné potomky typu class. Musí však obsahovat právě jeden qdisc, který se nazývá leaf-qdisc.

```
tc class add dev eth0 parent 1:0 classid 1:1 htb rate 256kbit ceil 512kbit
```

# Komponenty QoS

## filter

- Plní roli klasifikátoru.
- Nejkomplexnější komponenta QoS.
- Může být připojen ke qdisc nebo ke class.
- Umožňuje vytvářet podmínky na nejrůznější vlastnosti klasifikovaných paketů (porty, adresy, TOS flagy).

```
tc filter add dev eth0 protocol ip parent 10: prio 1  
u32 match ip dport 22 0xffff flowid 10:1
```

Syntaxe filtru může být někdy velmi složitá. Nicméně existuje alternativa ...

# Klasifikace pomocí iptables

- Volitelně můžeme použít ke klasifikaci nástroj iptables.
- Pravidla vkládáme do tabulky s názvem **mangle**.
- Cíl každého pravidla musí být metoda **CLASSIFY**, které pak můžeme určit přímo ID třídy, do které má daný paket zapadnout.

```
iptables -t mangle -A POSTROUTING -o eth0 -d  
192.168.69.5 -j CLASSIFY --set-class 1:10
```

# Příklad konfigurace

```
tc qdisc add dev eth0 root handle 1: htb default 13
tc class add dev eth0 parent 1: classid 1:1 htb rate 2048kbit ceil
2048kbit
tc class add dev eth0 parent 1:1 classid 1:10 htb rate 512kbit ceil
2048kbit prio 0
tc class add dev eth0 parent 1:1 classid 1:11 htb rate 512kbit ceil
2048kbit prio 1
tc class add dev eth0 parent 1:1 classid 1:12 htb rate 512kbit ceil
2048kbit prio 2
tc class add dev eth0 parent 1:1 classid 1:13 htb rate 512kbit ceil
2048kbit prio 2
tc qdisc add dev eth0 parent 1:10 handle 10: sfq perturb 10
tc qdisc add dev eth0 parent 1:11 handle 11: sfq perturb 10
tc qdisc add dev eth0 parent 1:12 handle 12: sfq perturb 10
tc qdisc add dev eth0 parent 1:13 handle 13: sfq perturb 10
iptables -t mangle -A POSTROUTING -o eth0 -d 192.168.1.0/26 -j CLASSIFY
--set-class 1:10
iptables -t mangle -A POSTROUTING -o eth0 -d 192.168.1.64/26 -j CLASSIFY
--set-class 1:11
iptables -t mangle -A POSTROUTING -o eth0 -d 192.168.1.128/26 -j
CLASSIFY --set-class 1:12
iptables -t mangle -A POSTROUTING -o eth0 -d 192.168.1.192/26 -j
CLASSIFY --set-class 1:13
```

Dotazy ???

Děkuji za pozornost

- BROWN, Martin A. The Linux Documentation Project : Traffic Control, [online]. 2006 [cit. 2011-05-22]. <<http://tldp.org/HOWTO/Traffic-Control-HOWTO/>>
- HUBERT, Bert - GRAF, Thomas - MAXWELL, Greg - VAN MOOK, Remco. Linux Advanced Routing & Traffic Control, [online]. 2006 [cit 2011-05-22]. <<http://lartc.org/howto/>>