

# 6. Kvalita služby (QoS)

PB156: Počítačové síť

Eva Hladká

*Slidy připravil: Tomáš Rebok*

Fakulta informatiky Masarykovy univerzity

jaro 2014

# Struktura přednášky

- 1 Úvod
- 2 Plánování (Scheduling)
  - FIFO (FCFS)
  - Priority Queuing
  - Weighted Fair Queuing
- 3 Formování/omezování toků (Traffic Shaping)
  - Leaky Bucket
  - Token Bucket
- 4 Prevence zahlcení (Congestion Avoidance)
- 5 Kvalita služby v Internetu
  - Integrované služby (Integrated Services)
  - Rozlišované služby (Differentiated Services)
- 6 Rekapitulace

# Struktura přednášky

- 1 Úvod
- 2 Plánování (Scheduling)
  - FIFO (FCFS)
  - Priority Queuing
  - Weighted Fair Queuing
- 3 Formování/omezování toků (Traffic Shaping)
  - Leaky Bucket
  - Token Bucket
- 4 Prevence zahlcení (Congestion Avoidance)
- 5 Kvalita služby v Internetu
  - Integrované služby (Integrated Services)
  - Rozlišované služby (Differentiated Services)
- 6 Rekapitulace

# Úvod I.

- všechny síťové toky jsou v TCP/IP sítích obhospodařovány ekvivalentně
  - žádný není upřednostňován
  - na libovolném z nich může dojít ke ztrátám paketů, zpoždění, duplikaci, přeskládání, snížení přenosové rychlosti, atp.
    - *best-effort service*
- jsou případy, kdy je nezbytné některé toky *upřednostnit (prioritizovat)* před jinými, resp. poskytnout jim definovanou **kvalitu služby (Quality of Service, QoS)**
  - omezit maximální možnou ztrátovost, zpoždění, . . . , garantovat požadovanou přenosovou rychlost, atp.
  - nezbytné zejména pro přenosy reálného času (tzv. *real-time přenosy*)
    - multimediální přenosy, haptická interakce – např. lékařství (operace na dálku), IPTV
    - kritická řídicí data – např. obsluha jaderného reaktoru
    - atp.

## Úvod II.

- základní parametry síťových toků:
  - *spolehlivost (reliability)* – požadavek plné spolehlivosti vs. tolerance definované ztrátovosti
  - *zpoždění (latency, delay)*
  - *rozptyl zpoždění (jitter)*
  - *přenosová kapacita (bandwidth)*
  
- mechanismy zajištění kvality služby:
  - *plánování (Scheduling)*
  - *formování/omezování toků (Traffic Shaping)*
  - *prevence zahlcení (Congestion Avoidance)*
  
- nutno zajistit na více vrstvách (L4, L3, L2)

# Struktura přednášky

- 1 Úvod
- 2 Plánování (Scheduling)
  - FIFO (FCFS)
  - Priority Queuing
  - Weighted Fair Queuing
- 3 Formování/omezování toků (Traffic Shaping)
  - Leaky Bucket
  - Token Bucket
- 4 Prevence zahlcení (Congestion Avoidance)
- 5 Kvalita služby v Internetu
  - Integrované služby (Intergrated Services)
  - Rozlišované služby (Differentiated Services)
- 6 Rekapitulace

# Plánování (Scheduling)

- souvisí s obsluhou vstupních/výstupních front na odesílateli, příjemci a vnitřních síťových prvcích
  - struktura front a způsob manipulace s nimi zásadně ovlivňuje možnosti garance zpoždění přenosu
    - nezbytné kombinovat s dalšími přístupy (Congestion Avoidance, Traffic Shaping)
- základní přístupy k obsluze front:
  - *FIFO (FCFS)*
  - *Priority Queuing*
  - *Weighted Fair Queuing*

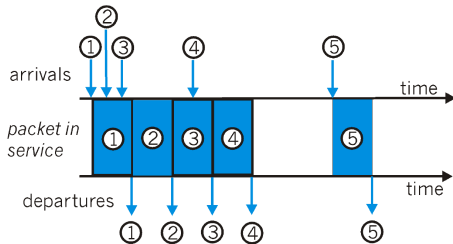
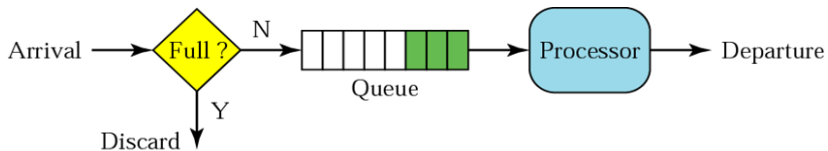
# Plánování (Scheduling) – FIFO (FCFS)

- *FIFO = First In First Out*
  - *FCFS = First Come First Serve*
  - „kdo dříve přijde, je dříve obsloužen“
- nejjednodušší uspořádání
  - využívá pouze jednu frontu pro obsluhu procházejících paketů
- *nevýhody:*
  - žádná podpora priority
  - agresivní proudy zvýhodněny



# Plánování (Scheduling) – FIFO (FCFS)

## Ilustrace

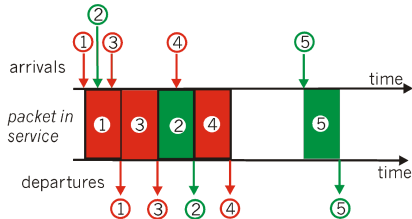
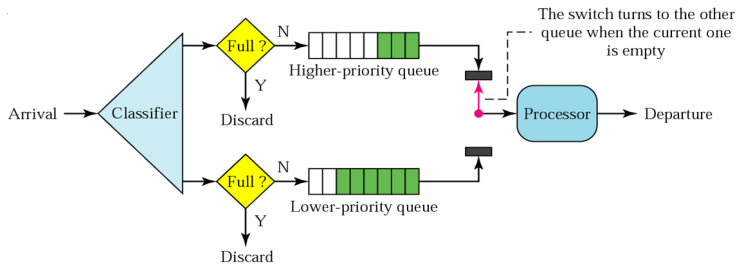


# Plánování (Scheduling) – Priority Queuing

- příchozí pakety jsou zařazeny do prioritních tříd
- každá prioritní třída má svou FIFO frontu
- pakety ve třídě s vyšší prioritou jsou obsluhovány dříve
  - *pakety nižší třídy se neodesílají dokud nejsou odeslány všechny pakety vyšších tříd*
- *výhoda:*
  - síťové toky s vyšší prioritou mají garantovanu přednostní obsluhu (nižší zpoždění generované jejich obsluhou)
- *nevýhoda:*
  - pokud existuje kontinuální tok vyšší priority, pakety nižší priority nebudou nikdy odbaveny
    - = *vyhladovění (starvation)*

# Plánování (Scheduling) – Priority Queuing

## Ilustrace

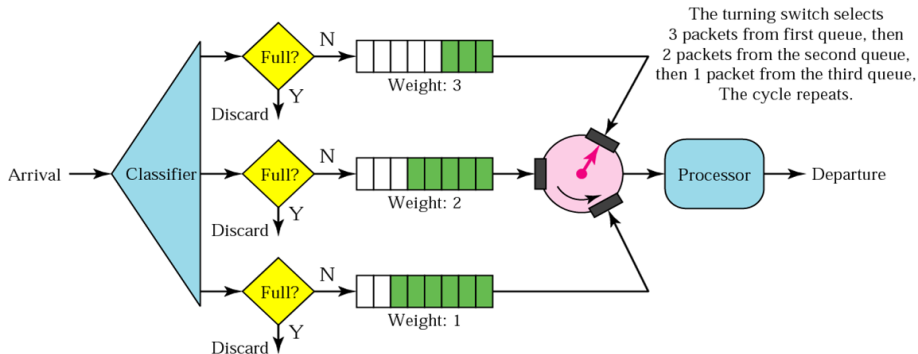


# Plánování (Scheduling) – Weighted Fair Queuing

- pakety opět přiřazovány do prioritních front
- frontám přiřazeny *váhy*
  - pakety opět přiřazovány do prioritních front
  - vyšší váha = vyšší priorita
- fronty obsluhovány střídavě dle přiřazené váhy
  - střídavě = tzv. *Round Robin* mechanismus
  - čím vyšší váha, tím více paketů je z fronty odebráno (a zpracováno)
- *výhoda*:
  - řeší problém vyhladovění předchozího mechanismu

# Plánování (Scheduling) – Weighted Fair Queuing

## Ilustrace



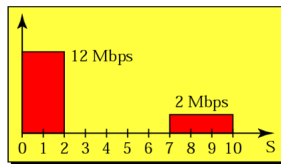
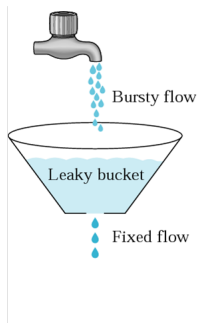
# Struktura přednášky

- 1 Úvod
- 2 Plánování (Scheduling)
  - FIFO (FCFS)
  - Priority Queuing
  - Weighted Fair Queuing
- 3 Formování/omezování toků (Traffic Shaping)**
  - Leaky Bucket
  - Token Bucket
- 4 Prevence zahlcení (Congestion Avoidance)
- 5 Kvalita služby v Internetu
  - Integrované služby (Integrated Services)
  - Rozlišované služby (Differentiated Services)
- 6 Rekapitulace

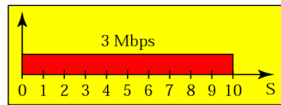
# Formování/omezování toků (Traffic Shaping)

- mechanismus pro řízení množství a rychlosti odesílaných paketů
- řízená kritéria:
  - průměrná rychlost odesílaných paketů (*average rate*) – počet paketů, které lze zpracovat za určitý časový interval (dlouhodobé chování)
  - špičková rychlost odesílaných paketů (*peak rate*) – maximální počet paketů, které lze do sítě zaslat za krátký časový interval (krátkodobé chování)
  - maximální počet paketů, které lze do sítě odeslat najednou (*burst size*) – maximální počet paketů, které lze do sítě zaslat za extrémně krátký časový interval ( $\approx$  najednou)
- navrženy dva mechanismy:
  - *Leaky Bucket* – vyhlazování přenosu (ovlivňuje *average rate*)
  - *Token Bucket* – umožnění krátkodobých špiček (ovlivňuje *peak rate* a *burst size*)

# Traffic Shaping – Leaky Bucket



Bursty data



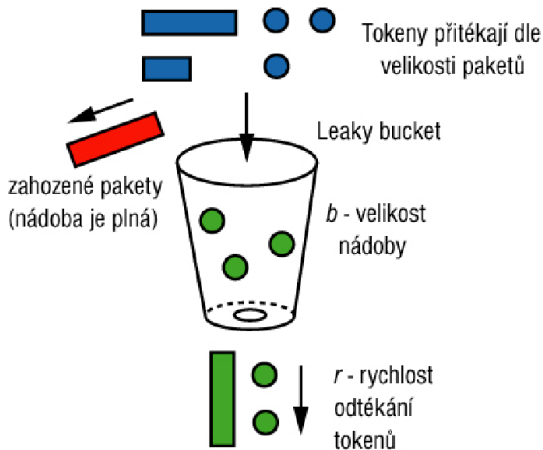
Fixed-rate data

- obdoba „děravého kyblíčku“
  - nezávisle na doplňování tekutiny do kyblíčku, voda dírkou vytéká vždy **konstantní rychlostí**
- využito pro vyhlazování toku
  - nepravidelný tok je průměrován
  - pokud je kyblíček plný, nově přichozí pakety jsou zahozeny



# Traffic Shaping – Leaky Bucket

Přirázování tokenů odcházejícím paketům

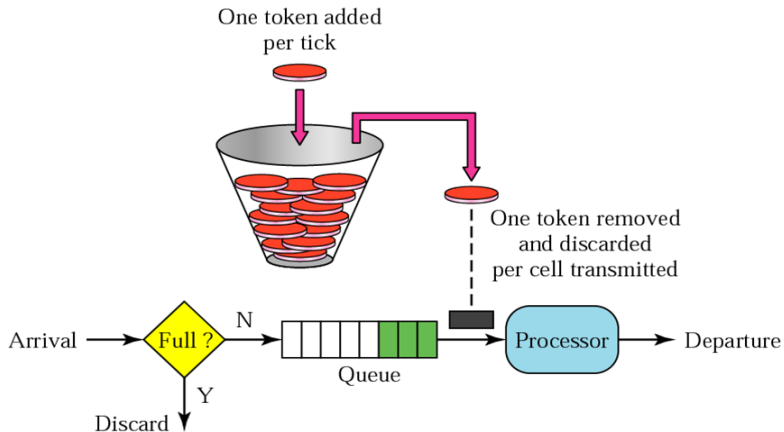


# Traffic Shaping – Token Bucket

- *Leaky Bucket* je velmi restriktivní – penalizuje nečinné uzly
  - nedovoluje nashromáždit uzlu tokeny v okamžiku nečinnosti (a použít je pro budoucí špičkový přenos)
- **Token Bucket**
  - umožňuje shromažďování tokenů v okamžiku nečinnosti uzlu
  - za každou přenesenou datovou buňku (např. 1 B, 100 B, atp.) je z kyblíčku odebrán token
  - velikost kyblíčku ovlivňuje velikost krátkodobých špiček
- možná kombinace obou přístupů
  - umožnění vstupních špiček spolu s vyhlazováním toku
  - *Otázka: V jakém pořadí (Token Bucket vs. Leaky Bucket)?*

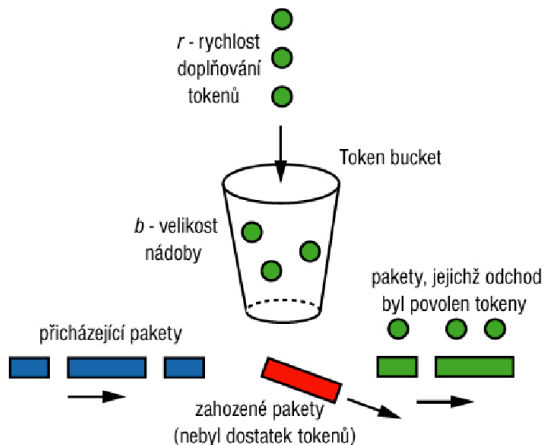
# Traffic Shaping – Token Bucket

## Ilustrace



# Traffic Shaping – Token Bucket

Přirázování tokenů odcházejícím paketům



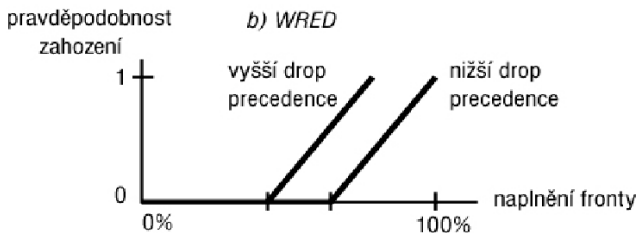
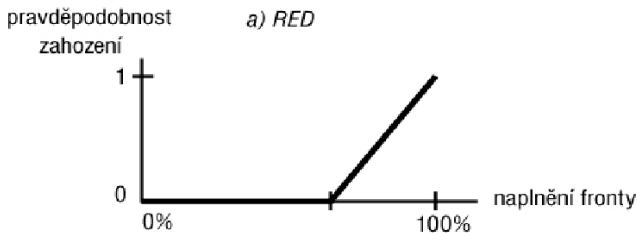
# Struktura přednášky

- 1 Úvod
- 2 Plánování (Scheduling)
  - FIFO (FCFS)
  - Priority Queuing
  - Weighted Fair Queuing
- 3 Formování/omezování toků (Traffic Shaping)
  - Leaky Bucket
  - Token Bucket
- 4 **Prevence zahlčení (Congestion Avoidance)**
- 5 Kvalita služby v Internetu
  - Integrované služby (Intergrated Services)
  - Rozlišované služby (Differentiated Services)
- 6 Rekapitulace

# Prevence zahlčení (Congestion Avoidance)

- standardní chování front:
  - fronta se plní, žádné pakety nejsou zahazovány
  - jakmile je fronta plná, nově přichozí pakety jsou zahozeny
  - *nevýhody*:
    - nemožnost pružné reakce na blížící se zaplnění fronty (odesílatel se o tomto stavu s předstihem nedozví)
    - může dojít k synchronizaci zahlčení mezi více směrovači a vytváření vln, kdy dojde k zahlčení a kdy naopak nejsou cesty využity (důsledek chování TCP protokolu)
- *Random Early Detection (RED)*
  - přesáhne-li zaplnění fronty určitou mez, začne směrovač zahazovat pakety náhodně vybraných toků
    - ⇒ odesílatel daného toku sníží rychlost odesílání (= reakce na ztrátu)
  - pravděpodobnost zahození paketu se zvyšuje se zvyšujícím se zaplněním fronty
  - odstraňuje problém globální synchronizace
- *Weighted Random Early Detection (WRED)*
  - totéž, co RED, avšak pravděpodobnost zahození paketu závisí také na prioritě paketu (resp. toku, kterému přísluší)

## RED vs. WRED



# Struktura přednášky

- 1 Úvod
- 2 Plánování (Scheduling)
  - FIFO (FCFS)
  - Priority Queuing
  - Weighted Fair Queuing
- 3 Formování/omezování toků (Traffic Shaping)
  - Leaky Bucket
  - Token Bucket
- 4 Prevence zahlcení (Congestion Avoidance)
- 5 Kvalita služby v Internetu**
  - Integrované služby (Integrated Services)
  - Rozlišované služby (Differentiated Services)
- 6 Rekapitulace



# Kvalita služby v Internetu

pro Internet navrženy dva mechanismy zajištění kvality služby:

- *Integrované služby (Intergrated Services)*
  - založeno na oznamování požadovaných parametrů QoS a rezervaci nezbytných zdrojů na vnitřních prvcích sítě (po cestě k příjemci)
- *Rozlišované služby (Differentiated Services)*
  - založeno na značkování paketů (přiřazování paketů do definovaných tříd provozu a jejich prioritní obsluha na vnitřních prvcích sítě)

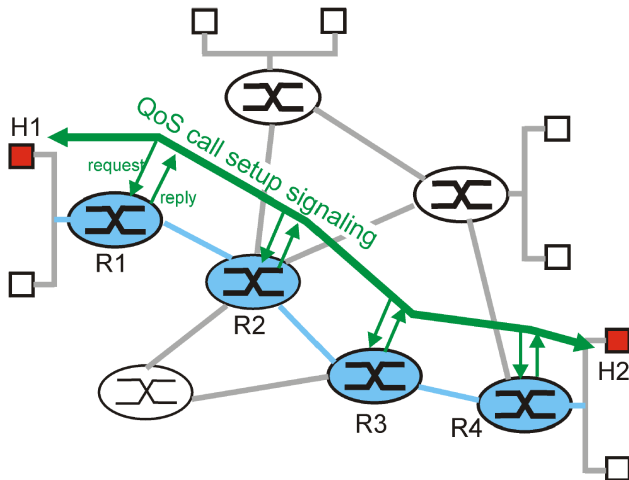
# Integrované služby (Integrated Services)

## Integrované služby (Integrated Services)

- aplikace oznámí počítačové síti své kvalitativní požadavky na přenos dat
- síť ověří, zda jsou k dispozici požadované prostředky a rozhodne, zda požadavkům vyhoví  
(fáze řízení přístupu (*Admission Control*))
- pokud nelze požadavkům vyhovět, spojení je zamítnuto
  - aplikace se může rozhodnout, zda požádá o méně náročné QoS
- pokud lze požadavkům vyhovět, informuje síť všechny komponenty po cestě k příjemci, ať rezervují odpovídající objem prostředků (velikost front, jejich prioritu, atp.)
  - nutnost využití rezervačního protokolu
    - např. *Resource reSerVation Protocol (RSVP)* (RFC 2205) či protokolu *YESSIR (YEt another Sender Session Internet Reservations)*
- *zásadní nevýhoda:*
  - nutnost udržování stavu na vnitřních prvcích sítě ( $\Rightarrow$  problémy se škálovatelností)

# Integrované služby (Integrated Services)

Ilustrace rezervace prostředků

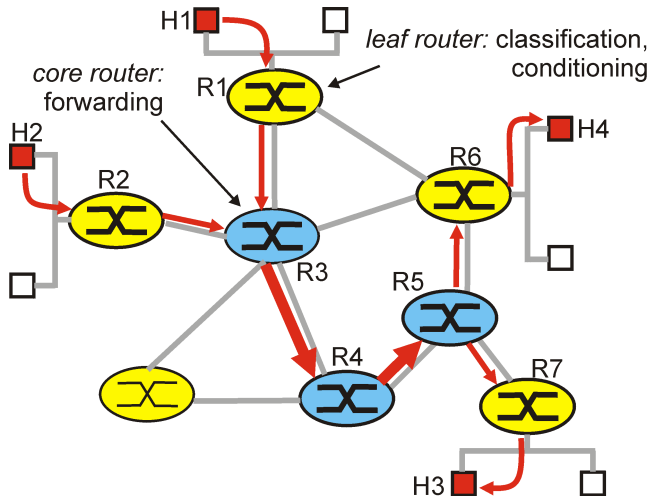


# Rozlišované služby (Differentiated Services)

- ne vždy je zapotřebí přesná definice požadovaných parametrů QoS
  - často si vystačí s garancí, že se kvalita přenosu výrazně nezhorší při změně zatížení sítě
- ⇒ **Rozlišované služby (Differentiated Services)**
  - počítačové sítě se neoznamují žádné požadavky na kvalitu přenosu
    - není nutné využití rezervačních protokolů
  - každý paket vstupující do sítě je označen značkou, která určuje kvalitativní třídu přenosu
    - označení paketu provedeno pouze na vstupu do sítě
    - značka umístěna do pole *Type of Service (IPv4)* či *Traffic Class (IPv6)*
    - na základě přiřazené třídy je s paketem na vnitřních prvcích sítě zacházeno (vnitřní prvky jen čtou přiřazenou značku)
  - *výhoda:*
    - jednoduché (jak pro implementaci na prvcích sítě, tak pro aplikaci)
    - žádné stavové informace na vnitřních prvcích sítě (⇒ dobrá škálovatelnost)
    - žádné úvodní zpoždění přenosu dané nutností rezervace požadovaných zdrojů

# Rozlišované služby (Differentiated Services)

## Ilustrace klasifikace paketů



# Struktura přednášky

- 1 Úvod
- 2 Plánování (Scheduling)
  - FIFO (FCFS)
  - Priority Queuing
  - Weighted Fair Queuing
- 3 Formování/omezování toků (Traffic Shaping)
  - Leaky Bucket
  - Token Bucket
- 4 Prevence zahlcení (Congestion Avoidance)
- 5 Kvalita služby v Internetu
  - Integrované služby (Integrated Services)
  - Rozlišované služby (Differentiated Services)
- 6 Rekapitulace

# Rekapitulace – kvalita služeb (QoS)

- v TCP/IP sítích jsou všechny toky standardně obhospodařovány ekvivalentně
  - tzv. *best-effort service*
- prioritizace toků (zajištění požadované QoS) se provádí s využitím:
  - *plánování toků* – přístupy k obsluze front paketů na odesílateli, příjemci a vnitřních prvcích
  - *formování/omezování toků* – řízení množství a rychlosti odesílaných dat
  - *prevence zahlcení* – mechanismus zahazování paketů v případě zahlcení (či blížícího se zahlcení)
- pro Internet byly navrženy dva mechanismy pro zajištění QoS:
  - *Integrované (IntServ) vs. Rozlišované (DiffServ) služby*