

PA159 – Kvalita služeb a fronty

13. 11. 2009

Kvalita služeb

- Základní parametry
 - Kapacita linky
 - Zpoždění
 - Rozptyl
- Technické zajištění
 - Kapacita linek
 - * Na lince se data nepředbíhají
 - Vysílající/přijímající
 - Aktivní prvky uvnitř sítě

Kvalita služeb a fronty

- Vysílající
- Na aktivních prvcích
 - Na straně výstupních portů
 - Délka souvisí s kapacitou
- Ovlivňují
 - Zpoždění a jeho rozptyl
 - Řazení paketů na výstupu

Fronty

- FIFO (FCFS)
- Fair Queuing
- Processor Sharing
- Bit-round Fair Queuing
- *Generalized Processor Sharing*
- *Weighted Fair Queuing*

FIFO

- Nejjednodušší uspořádání
 - First In First Out (First Come First Serve)
 - Jedna fronta pro každý výstupní port
- Nevýhody:
 1. Žádná podpora priority
 2. Obecně větší průměrné zpoždění paketů (nerozlišuje mezi dlouhými a krátkými pakety)
 3. Agresivní TCP proudy zvýhodněny
 - Kdo se dostane dřív do fronty, bude dřív poslán

Fair Queuing

- Vícenásobné fronty pro každý výstupní port
 - Rodělení podle vstupních proudů
- Každý příchozí paket umístěn do příslušné fronty
- Fronty obsluhovány pořadě, vždy po jednom paketu
 - Tím je zajištěna „férovost“ obsluhy
- Prázdná fronta se přeskočí

Odstraní většinu nevýhod FIFO front

Avšak: penalizuje krátké pakety

Processor Sharing

- Idealizovaný, prakticky nepoužitelný protokol
- Fronty jako u FIFO
- Namísto paketů posílá bity
- Každá fronta sdílí přesně $1/N$ celkové kapacity

Základní pojmy

(všechny údaje jsou normalizovány na výstupní rychlost datového toku)

cyklus = jeden průchod frontami

$R(t)$ = „virtuální čas“, tedy počet cyklů do času t

$N(t)$ = počet neprázdných front v čase t

P_i^α = doba přenosu paketu i ve frontě α

τ_i^α = čas příchodu paketu i do fronty α

S_i^α = hodnota $R(t)$ na začátku přenosu paketu i

F_i^α = hodnota $R(t)$ na konci přenosu paketu i

Změna virtuálního času (závisí na obsazení front):

$$R'(t) = \frac{d}{dt}R(t) = \frac{1}{\max\{1, N(t)\}}$$

PS – doba přenosu

- Pro velikost paketu s_i^α platí:

$$s_i^\alpha = \int_{S_i^\alpha}^{F_i^\alpha} v^\alpha(t)$$

kde $v^\alpha(t)$ je rychlost přenosu výstupního kanálu.

- Virtuální čas zavádíme proto, že v něm platí:

$$P_i^\alpha = s_i^\alpha$$

tedy doba přenosu je právě rovna velikosti paketu

PS – vývoj v čase

- Vztahy pro jednu frontu:

$$F_i^\alpha = S_i^\alpha + P_i^\alpha$$

$$S_i^\alpha = \max\{F_{i-1}^\alpha, R(\tau_i^\alpha)\}$$

- Umožní spočítat virtuální čas konce přenosu paketu
- Neříká nic o reálném čase (závisí na „plnosti“ front)
- Garantuje zcela férový přístup

Bit-round Fair Queuing

- Emulace PS na úrovni paketů
- Asymptoticky se blíží PS
- Principy
 - Spočítá S_i^α a F_i^α každého příchozího paketu
 - Vybere pro přenos vždy paket s nejnižší hodnotou F_i^α
- Pořadí přenosu paketů (start/konec) různé u PS a BRFQ, avšak asymptoticky se BRFQ chová stejně jako PS
 - Asymptoticky = pakety délky 1 bit

Prioritní fronty

- PS (a BRFQ) „férové“
 - Žádný paket „nepředbíhá“
 - Není rozdíl mezi proudy s dlouhými i krátkými pakety
- Nepodporují prioritu
 - Nemožňují upřednostnění určitých spojení

Generalized Processor Sharing

- Založeno na explicitním řízení alokace front
- GPS jako idealizace akceptovaného řešení:
Weighted Fair Queuing
- Rozšíření PS:
 - ϕ^α je *váha* přidělená toku α ; určuje, kolik bitů se má z fronty α přenést v každém cyklu

$$F_i^\alpha = S_i^\alpha + \frac{P_i^\alpha}{\phi^\alpha}$$

$$S_i^\alpha = \max\{F_{i-1}^\alpha, R(\tau_i^\alpha)\}$$

GPS – další vztahy

- Efektivní délka paketu normalizována faktorem $1/\phi^\alpha$
 - Větší váha tedy „zkracuje“ pakety
- Rychlost obsluhy g_i neprázdného toku i je

$$g_i = \frac{\phi_i}{\sum_j \phi_j} C$$

kde C je kapacita výstupní linky a suma jde přes všechny aktivní fronty/toky

GPS – garance zpoždění

Uvažujme několik toků, které jsou dostatečně dlouho nečinné, takže všechny „kyblíčky“ (buckets) jsou plné.

Poté všechny toky začnou vysílat maximální povolenou rychlostí. Síť musí být konfigurována tak, že je schopna těmto kombinovaným požadavkům vyhovět (garance kvality služby pro každý tok). To znamená, že každý garantovaný tok je R_i . Tokeny do „kyblíčků“ jsou přidávány stejnou rychlostí, jakou jsou z nich odebírány, tj. délka fronty není větší než velikost „kyblíčku“.

Maximální zpoždění je tak podíl velikost „kyblíčku“ (B_i) a rychlosti toku.

$$D_i \leq \frac{B_i}{R_i}$$

Wieghted Fair Queuing (WFQ)

- Emulace GPS při přenosu paketů a nikoliv pouze bitů
- Vybrán je vždy paket s nejnižším F_i^α podle rovnice pro GPS.
- Vlastnosti GPS zůstávají (asymptoticky) zachovány:

$$D_i \leq \frac{B_i}{R_i} + \frac{(K_i - 1)L_i}{R_i} + \sum_{m=1}^{K_i} \frac{L_{\max}}{C_m}$$

- kde D_i , R_i a B_i viz výše a pro ostatní parametry platí:
 - K_i = počet uzlů, kterými tok i prochází
 - L_i = délka největšího paketu toku i
 - L_{\max} = délka největšího paketu na všech cestách a uzlech
 - C_m = výstupní kapacita uzlu m

WFQ – diskuse

- Term $\frac{(K_i-1)L_i}{R_i}$ zahrnuje zpoždění každého paketu na každém uzlu
- Term $\sum_{m=1}^{K_i} \frac{L_{\max}}{C_m}$ je důsledkem přenosu paketů a nikoliv bitů.
- Komentář:
 - Vybereme-li k přenosu delší paket a v průběhu přenosu přijde kratší paket, pak při GPS by mohl kratší paket být doručen dříve, než se dokončí přenos dlouhého paketu. Při WFQ je třeba počkat na dokončení přenosu delšího paketu, než může být přenesen i ten kratší
Výsledkem je delší zpoždění, než garantuje GPS.

WFQ – důsledky

- Omezení D_i shora je základem pro poskytnutí garantované služby
 - Je možno zvolit odpovídající vlastnosti směrovače (velikost „kyblíčku“, rychlost přitékání tokenů, ...)
 - Maximální délka fronty je úměrná největšímu zpoždění a blíží se $g_i D_i$ (kde g_i definované výše je rychlost, s jakou odchází tok i).
- **Důsledek:** Je možno rozeznat požadavek, který by nebylo lze uspokojit, a takový požadavek odmítnout. Pro ostatní lze služby garantovat.

Fronty – shrnutí

- Struktura front a způsob manipulace s nimi zásadně ovlivňuje možnosti garance zpoždění
- Ukázali jsme, že je možno shora omezit maximální možné zpoždění
- Nutno kombinovat s dalšími postupy

Problém ochrany proti zahlcení

- TCP a podobné algoritmy reagují na přetížení sítě
- To může být pozdě, navíc použití klasického TCP mechanismu má určité nedostatky:
 - Ztracené pakety musí být znovu přeneseny (další zvýšení zátěže, vysoké zpoždění)
 - Fenomén *globální synchronizace*:
 - * Při přetížení směrovače se začnou ztrácet pakety všech TCP proudů
 - * Všechny přejdou na slow start a tím začne být síť nedostatečně využívána
 - * To všechny rozeznejí a síť opět (synchronně) přetíží

Proaktivní ochrana

- Možné řešení:
 - Zvětšení front (buferů): dlouhodobě neúčinné (zvýšení zpoždění, není horní limit)
 - Včasná detekce zahlcení a výběr jednoho (postupně více) TCP proudů, které budou zpomaleny: *prokativní ochrana proti zahlcení*
- Odstranění globální synchronizace, lepší využití sítě

Random Early Detection

- Zahazuje pakety dříve, než je fronta zcela zaplněna
- Jednoduchý princip po příchodu paketu:
 - Pokud je ve frontě dostatek místa, paket je zpracován
 - Pokud je fronta delší než TH_{\min} ale kratší než TH_{\max} , pak se některé pakety zahodí
 - Pokud je fronta delší než TH_{\max} , pak se zahodí každý příchozí paket
- Náhodným zahozením paketu se předchází zahlcení

Random Early Detection – Algoritmus

Spočti průměrnou velikost fronty avg

if $avg < TH_{min}$

ulož paket do fronty

else if $TH_{min} \leq avg \leq TH_{max}$

spočti pravděpodobnost P_a

s pravděpodobnosti P_a zahod packet

else ulož paket do fronty % Pravděpodobnost $1 - P_a$

else if $TH_{max} \leq avg$

zahod packet

Random Early Detection – *avg*

- Průměrná velikost fronty (*avg*):

- if fronta je neprázdná

$$avg = (1 - w_q)avg + w_qq$$

else

$$m = f(time - q_time)$$

$$avg = (1 - w_q)^m avg$$

- kde

q_time = čas, kdy se fronta vyprázdnila

w_q = váha fronty

q = aktuální délka fronty

time = aktuální čas

f(t) = lineární funkce času *t*

RED – doplnění

- Váha fronty (w_q) filtruje dočasná přetížení (doporučená hodnota 0,002) – zabraňuje přehnaně rychlé reakci na nával (burst)
- Pravděpodobnost P_a je počítána postupně:

- Nejprve se spočte P_b :

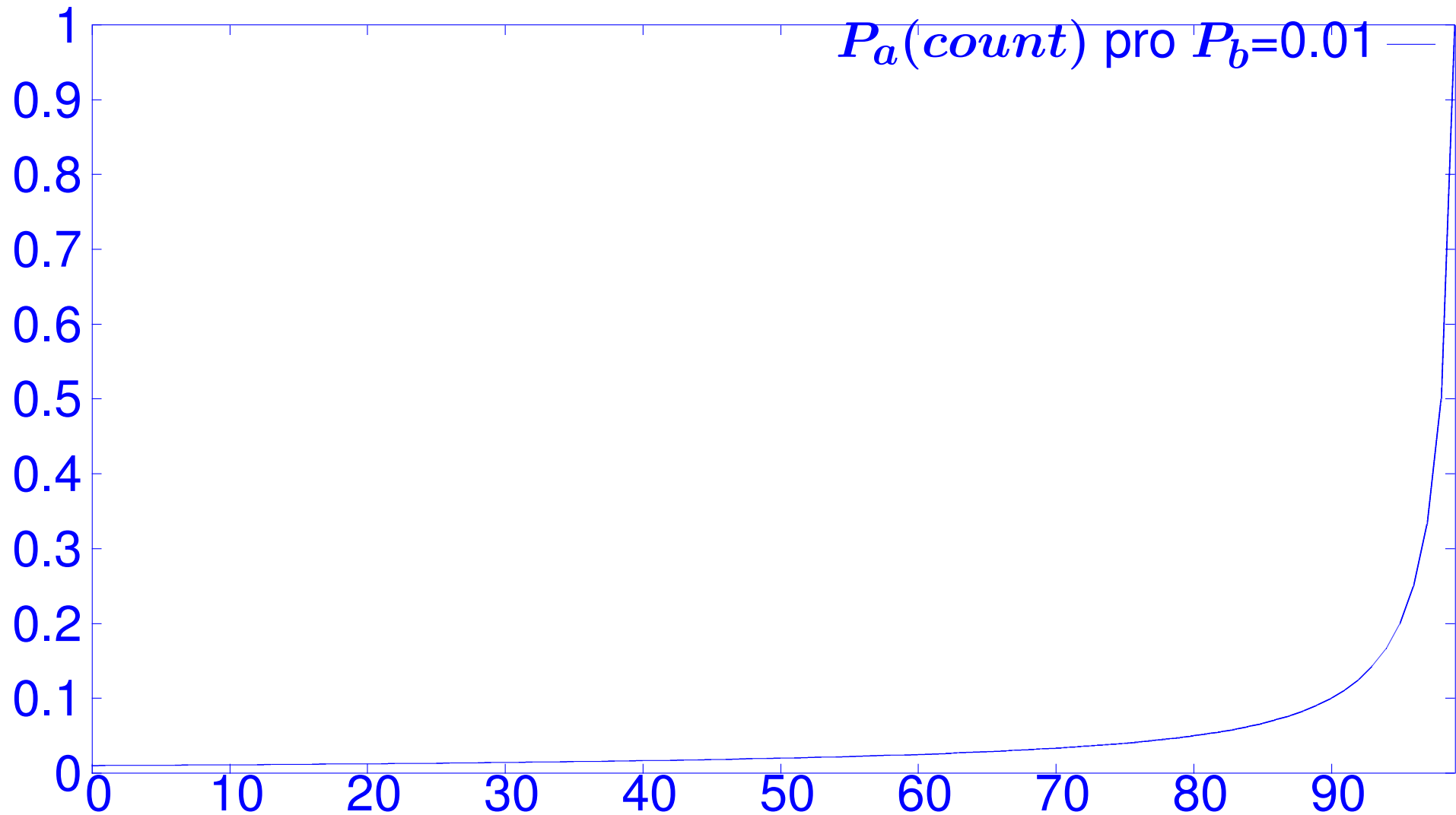
$$P_b = \frac{avg - TH_{\min}}{TH_{\max} - TH_{\min}} \times P_{\max}$$

- Z něj se spočte pravděpodobnost zahození paketu P_a :

$$P_a = \frac{1}{\frac{1}{P_b} - count}$$

kde *count* je délka serie nezahozených paketů a P_{\max} je pravděpodobnost zahození paketu při frontě délky TH_{\max} (obvykle 1)

Pravděpodobnost jako funkce paketů ve frontě



RED – vlastnosti P_a

- Graf P_a jako funkce *count*:
 - Velmi malá hodnota pro většinu hodnot *count*
 - Extrémně rychle roste jak se *count* blíží $\frac{1}{P_b} - 1$
 - *count* tuto hodnotu nemůže překročit
- Garantuje vysokou rovnoměrnost zahazování paketů
- Při zatížení lepší vlastnosti než prosté zahazování paketů při plné frontě

Zajištění zdrojů

- Řízení zdrojů
 - Technické předpoklady nestačí
 - Je nutno koordinovat požadavky na zdroje
- Nezbytné signalizační/řídící protokoly (control protocol)

Sestavení relace

- Součástí rezervace zdrojů (resource reservation)
- Musí zajistit dostatek prostředků pro garanci služby
- Musí spolupracovat s dynamickým směrováním (IP netvoří okruhy)
- Princip *měkkého stavu* (soft state): stavová informace se drží pouze krátkou dobu, pak musí být znovu potvrzena
- Řešení pro Internet
 - RSVP (Resource ReSerVation Protocol)
 - DiffServ (Differentiated Services)