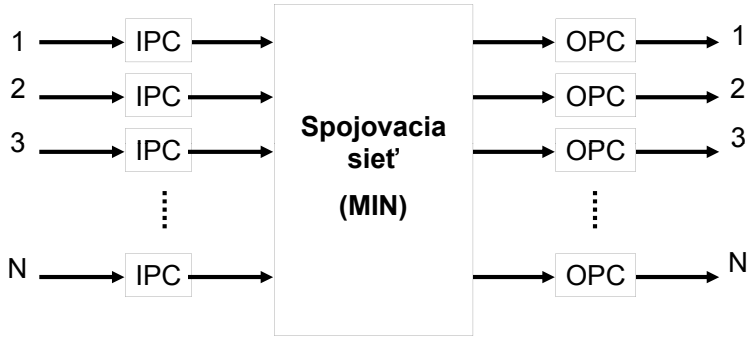
Spôsob radenia paketov, organizácia pamätí, prioritizácia a obsluha radov

doc. Ing. Martin Medvecký, PhD.

Architektúra spojovacieho systému



IPC/OPC Input/Output Port Controller

Charakteristika

- Prepínač NxN
- Predpokladajme:
  - pakety konšt. dĺžky (ATM)
  - vstupy/výstupy prenášajú dáta rovnakou prenosovou rýchlosťou.
- Spojovacia sieť (MIN) môže pracovať M krát rýchlejšie, ako je rýchlosť vstupných/výstupných liniek
  (počas trvania jedného paketu môže byť prenesených M paketov)
- Súčasne môže prísť na vstupy niekoľko paketov, ktoré majú byť smerované na rovnaký výstupný port => **preťaženie**

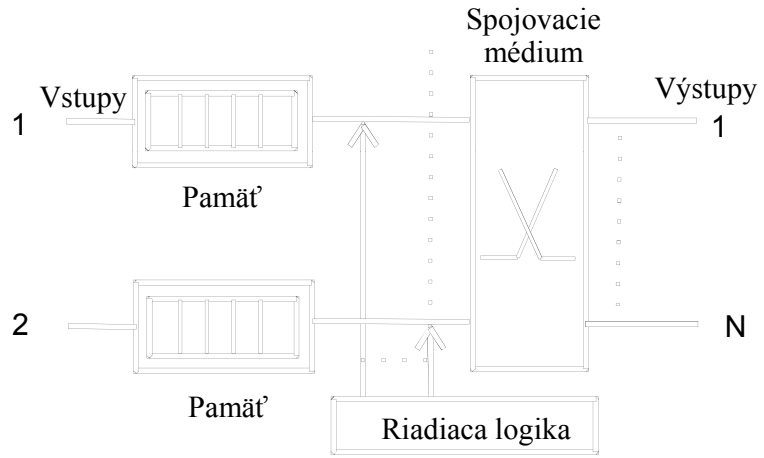
➔ **Sieť musí byť schopná radiť pakety do radu**

MIN bez vyrovnávacích pamätí

- V prípade kolízie paketov sa prenáša iba jeden paket (ostatné sú zahodené).
- **Výpočet priepustnosti prepínača**
  - Predpokladajme, že pakety prichádzajú v časových intervaloch (ATM bunky) s pravdepodobnosťou ρ (Bernoulliho rozdelenie)
  - Pre N -> ∞ platí pre priepustnosť prepínača vzťah

    $(1 - e^{-\rho})$
  - Maximálna priepustnosť (ρ=1) je **0,632**
  - Pravdepodobnosť, že paket bude zahodený je **0,368**

Radenie na vstupe



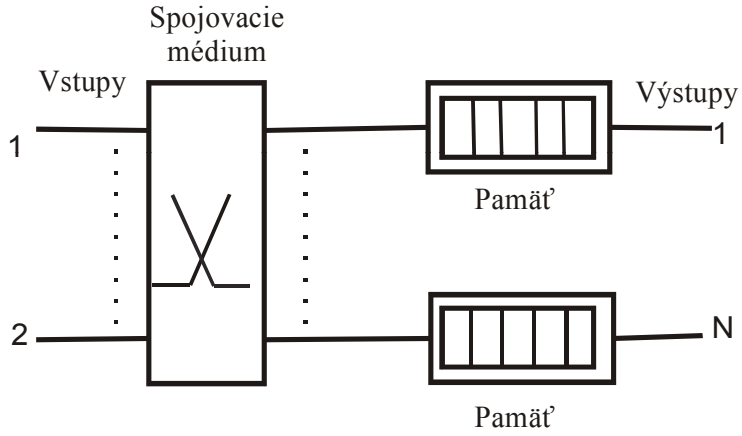
Radenie na vstupe (FIFO buffer)

- Každý vstup obsahuje vyrovnávaciu pamäť typu FIFO
- Slabina – *Head of line (HOL) blocking*
  (blokovanie prvým v rade)
- Max. priepustnosť pre exponenciálne rozdelenie dĺžky paketu a Poissonove rozdelenie pravdepodobnosti príchodu paketov je okolo 0,5

Radenie na vstupe (nie FIFO buffer)

- Pre výber paketov sa využíva oknová metóda, alebo predpovedanie kolízie
- Ak nastane kolízia, na vyslanie sa vyberajú postupne pakety z prvých W paketov *(W je veľkosť okna)* v každom vstupnom rade.
  To sa opakuje dovtedy (max. W krát), kým sa nenájde paket, ktorý môže byť vyslaný –> potláča sa HOL blocking.
- Dosahuje dobré výsledky (najmä pre malé N a veľké W)

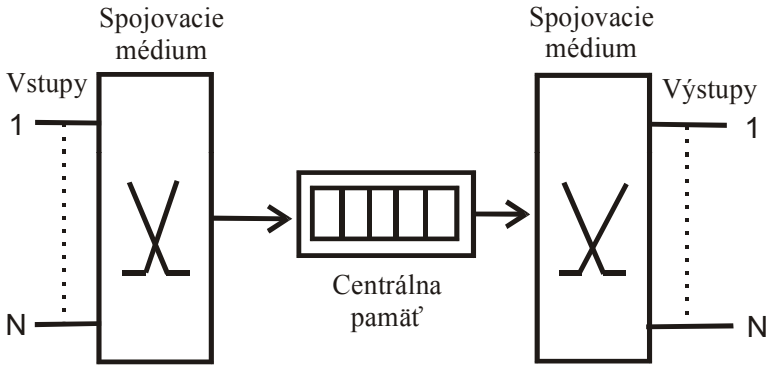
Radenie na výstupe



Radenie na výstupe

- Vyrovnávacie pamäte sú umiestnené na výstupoch
- Spojovacia sieť pracuje M krát rýchlejšie, ako je rýchlosť vstupných liniek
  - Dôsledok – sieťou môže prejsť M paketov určených pre jeden výstup, ktoré musia byť zapísané do výstupného radu.
- Nevyskytuje sa HOL blocking
- Nevýhoda – vyžaduje sa, aby interná rýchlosť spojovania bola vyššia ➔ má dopad na max. veľkosť prepínača

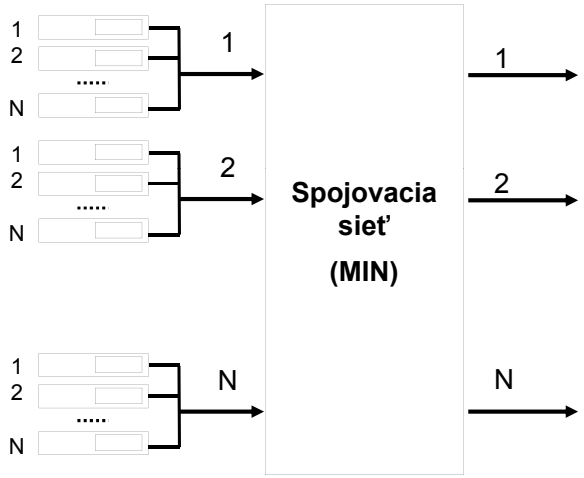
Centrálna zdieľaná pamäť



Centrálna zdieľaná pamäť

- Radenie sa uskutočňuje v centrálnej pamäti, ktorú zdieľajú všetky vstupné porty.
- Centrálna zdieľaná pamäť musí byť schopná uskutočniť N zápisov a N čítaní počas jedného cyklu
- Obmedzuje veľkosť prepínača.

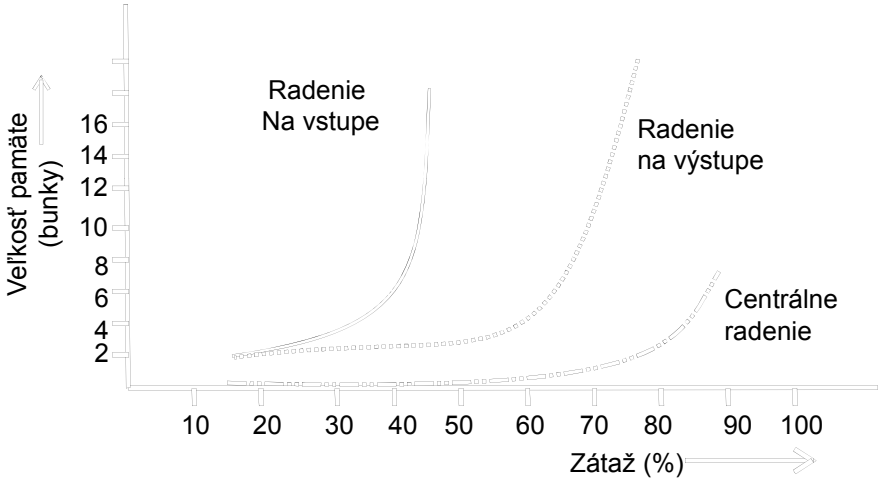
Virtuálne radenie na výstupe



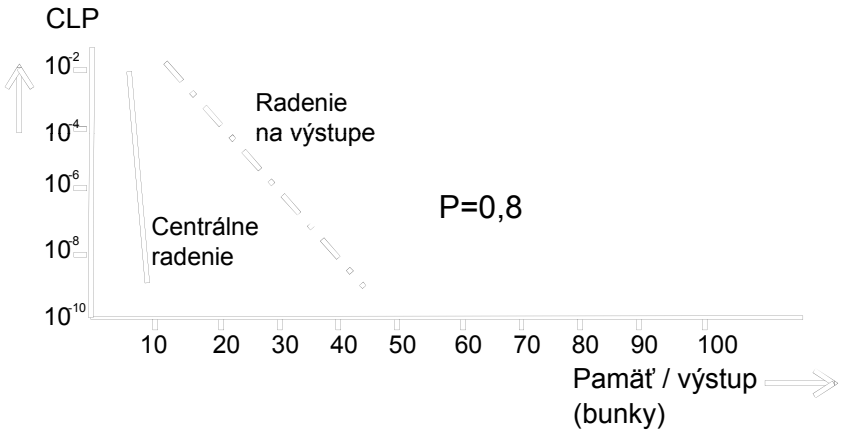
Virtuálne radenie na výstupe

- Netrpí na HOL blocking
- Zachováva škálovateľnosť radenia na vstupe
- Potrebuje dobrý algoritmus na výber paketov, ktoré majú byť vyslané zo vstupných portov na výstupné porty
- Umožňuje dosiahnuť až 99% priepustnosť

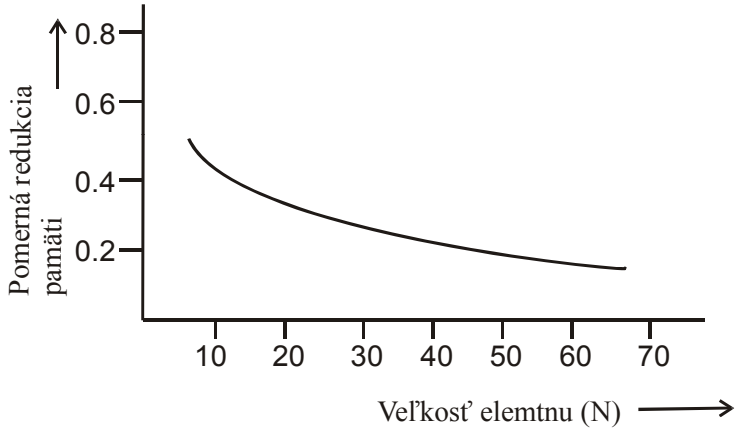
Radenie buniek - porovnanie



Radenie buniek - porovnanie



Radenie buniek - porovnanie



Aktívny manažment radu

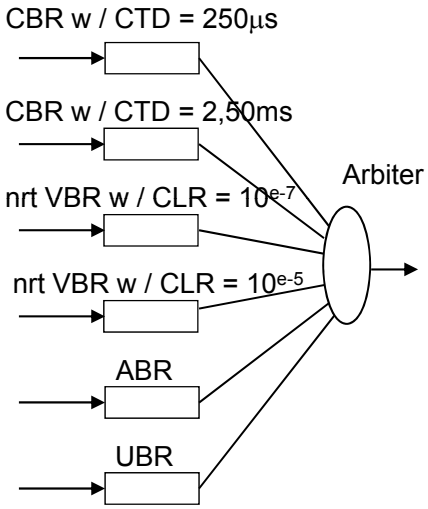
- **QMM (Queue Memory Management)**
  - Kontroluje počet paketov vo výstupnom rade.
  - Vykonáva sa pri operácii zaradenia prichádzajúceho paketu do radu.
- **QSD (Queue Scheduling Disciplines)**
  - Riadi veľkosť šírky prenosového pásma prideleného každej servisnej triede vo výstupnom rade.
  - Vykonáva sa pri výbere paketu z radu a jeho poslaní na výstupnú linku.

QMM zabezpečuje

- Pridávanie paketov do príslušného radu (napr. podľa ich klasifikácie a pod)
- Zahadzovanie paketov, ak je rad plný
- Vyberanie paketov na vysielanie podľa príkazov plánovača *(scheduler)*
- Voliteľne: monitorovanie zaplnenie radov a odstraňovanie paketov ešte pred zaplnením radu, alebo označovanie paketov pre neskoršie odstránenie

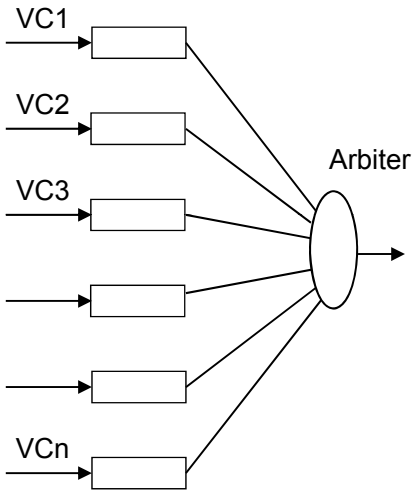
Radenie po skupinách

- Pakety skupiny tokov/volaní patriacich k rovnakej kategórie služieb sú radené spoločne v jednom rade
- Garancie platia pre agregované toky, nie pre individuálne toky/volania.
- Dobrá škálovateľnosť (relatívne malý počet kategórií)



Radenie po tokoch

- Pakety sú radené v individuálnych radoch pre každý tok.
- Garancie platia pre každý individuálny tok.
- Zlá škálovateľnosť (smerovač/prepínač si musí udržiavať stavové informácie o každom individuálnom toku)



Preťaženie siete

- **Preťaženie siete**
  - **Krátkodobé** – spôsobené krátkymi zhlukmi dát z niekoľkých tokov
  - **Dlhodobé** – spôsobené dlhodobým pôsobením všetkých tokov využívajúcich príslušný rad
- **Reakcie QMM**
  - Označovanie paketov
  - Zahadzovanie paketov
  - Informovanie zdroja o preťažení

Zahadzovanie paketov

Výber paketu pre zahodenie

- **Prichádzajúci paket**
  - Jednoduchá realizácia (paket sa nezaradí do radu)
  - Prichádzajúce pakety majú tendenciu mať väčšie oneskorenie a pre real time aplikácie sú „menej užitočné“
- **Paket na čele radu** *(DFF, Drop From Front)*
  - Náročnejšie na manipuláciu s radom,
  - Pre niektorých službách (napr. TCP) – rýchlejšia reakcia na preťaženie siete

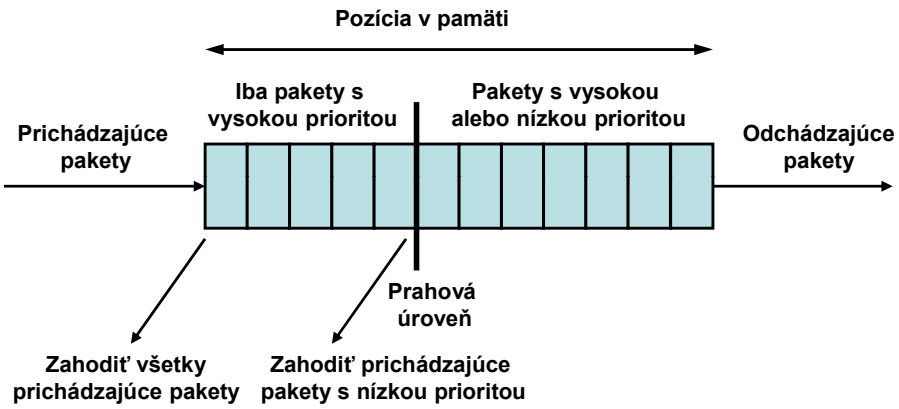
Tail Drop

- Najjednoduchšia technika zahadzovania paketov
- Rad sa plní FIFO princípom, keď je plný, ďalšie prichádzajúce pakety sa automaticky zahadzujú
- Koncové uzly nie sú spravidla o zahadzovaní informované a musia stratu paketu detegovať samostatne
- Môže viesť k tzv. *„globálnej synchronizácií“* (napr. pri TCP), kedy rýchlosť prevádzky osciluje medzi preplnenými a prázdnymi FIFO radmi

Selective Discard

- Policing umožňuje, aby vstupný (ingress) sieťový uzol označoval pakety prekračujúce dohodnuté parametre za pekety s nižšou prioritou
- Možno implementovať v:
  - ATM - CLP bit
  - IP - Diffserv
  - MPLS - Exp bity
- Takto označené pakety sú zahadzované prednostne
- Pri extrémnom preťažení sú zahadzované aj pakety s vyššou prioritou --> potreba implementácie preventívnych mechanizmov

Selective Discard



Early/Partial Packet Discard

- Vhodné špeciálne pre ATM
- Strata jednej bunky má za následok stratu celého paketu => Efektívnejšie je robiť zahadzovanie na úrovni rámcov ako na úrovni buniek.
- Na označenie poslednej bunky AAL5 PDU slúži PTI (Payload Type Indicator) v ATM bunke

Early/Partial Packet Discard

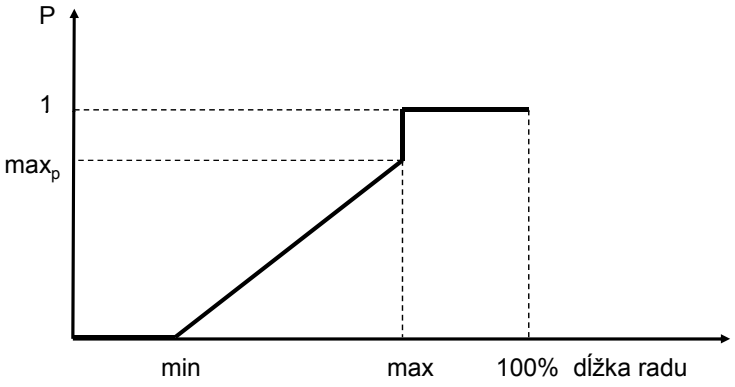
- **Early Packet Discard (EPD**)
  - Zariadenie v stave preťaženia zahadzuje všetky bunky z AAL5 PDU
  - Ak bol povolený zápis paketu do pamäte, rezervuje sa pamäťová kapacita pre všetky bunky z paketu.
- **Partial Packet Discard (PPD)**
  - Ak musí zariadenie zahodiť bunky zo stredu paketu, sú zahodené aj všetky nasledujúce bunky daného paketu.
  - Aplikuje sa, keď niektoré bunky z paketu už boli zapísané do pamäte.

Random Early Detection (RED)

- QMM technika často implementovaná na IP smerovačoch
- Predpokladá spoluprácu s algoritmami pre kontrolu toku technikou predchádzania zahltenia (obsahuje ju napr. TCP)
- RED používa tzv. *packet drop* profil – vyjadruje závislosť medzi pravdepodobnosťou zahodenia prichádzajúceho paketu a zaplnením radu.

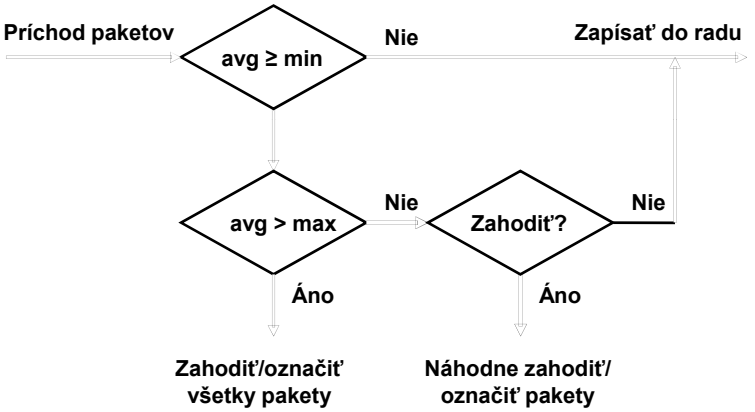
Random Early Detection (RED)

Pravdepodobnosť zahodenia paketu
(Packet drop profil)



Random Early Detection (RED)

Rozhodovací algoritmus



Random Early Detection (RED)

Výhody:

- Zavedenie RED nevyžaduje modifikáciu TCP protokolu
- Prístup k riešeniu zahltenia je proaktívny a nemalo by nikdy dôjsť k úplnému zaplneniu radu a následnému tail drop zahadzovaniu
- Zachováva sa podpora zhlukovej prevádzky, keďže ide o FIFO radenie. Pakety sú vysielané v tom poradí, v akom prišli. (Nevýhodou môže byť, že niektoré pakety zo zhlukovej prevádzky budú zahodené.)
- RED podporuje TCP, lebo nezahadzuje zhluky paketov z jediného TCP toku v dôsledku preplnenia

Random Early Detection (RED)

Výhody *(pokr.)*:

- RED umožňuje držať naplnenosť radu pod určitou úrovňou a pomáha k lepšiemu využitou prenosovej kapacity výstupnej linky. Množstvo paketov držaných v rade nie je ani príliš malé, čo by mohlo viesť k poddimenzovaniu výstupnej kapacity a ani sa neblíži k plnej kapacite, keď by sa museli zahodiť všetky prichádzajúce pakety z množstva TCP spojení, ktoré by následne znížili rýchlosť.
- RED podporuje teoreticky férové zahadzovanie paketov medzi jednotlivými TCP spojeniami – pričom si nemusí udržiavať stavovú informáciu pre každé spojenia.

Random Early Detection (RED)

Nevýhody:

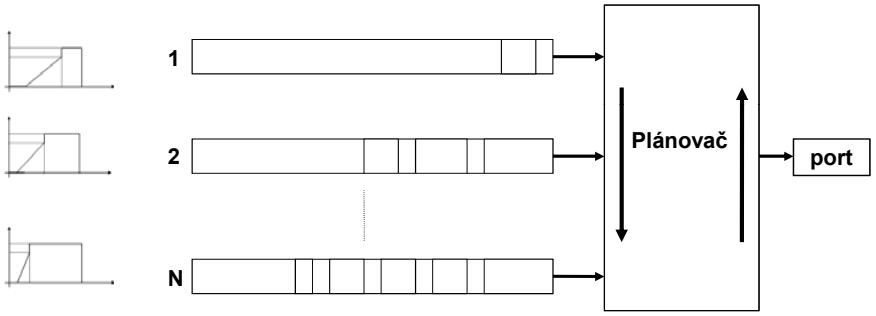
- RED môže byť dosť ťažké nastaviť tak, aby dával očakávané výsledky
- RED pracuje dobre iba s TCP protokolom a s jeho algoritmami na predchádzanie zahlteniu.
- Zahadzovanie paketov nie je veľmi efektívny signál oznamujúci zahltenie. Plytvajú sa pri ňom sieťové prostriedky (paket musí byť znovu preposlaný).

Weighted RED (WRED)

- Rozširuje RED o možnosť priradiť rôzne drop profily rôznym typom prevádzky
- Využitie:
  - Viac rôznych profilov v rámci jedného radu.
  - Rôzne profily pre jednotlivé rady
    (ak existuje viac radov, napr. existuje samostatný rad pre každú triedu služby)

Weighted RED (WRED)

Aplikovanie rôznych profilov pre rôzne rady

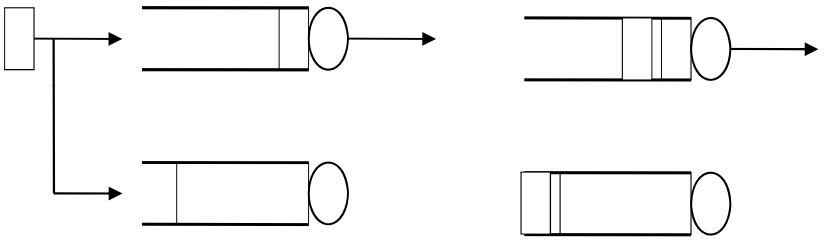


Adaptive Virtual Queue (AVQ)

- Nepočíta pravdepodobnosť zahodenia paketu, ale určuje kapacitu virtuálneho radu.
- Spravuje tzv. virtuálny rad, ktorého kapacita je nižšia ako skutočná kapacita výstupnej linky. Keď je paket zaradený do reálneho radu, je aktualizovaný aj virtuálny rad.
- Keď virtuálny rad pretečie:
  - pakety sú zahodené alebo označené,
  - na každej linke je prepočítaná virtuálna kapacita, aby sa zabezpečilo, že celkový dátový tok smerovaný na linku dosahuje požadovanú úroveň využitia linky.

Adaptive Virtual Queue (AVQ)

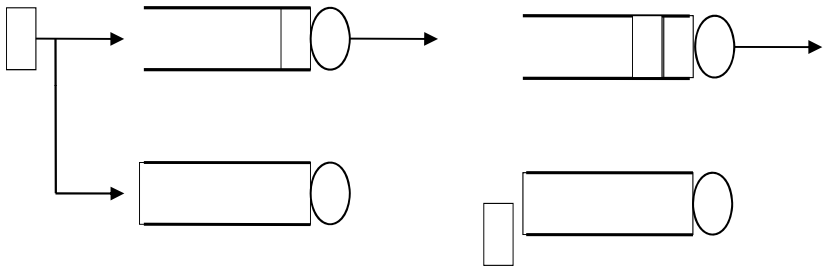
Zapísanie paketu do virtuálneho radu



Reálny rad Virtuálny rad

Adaptive Virtual Queue (AVQ)

Nezapísanie paketu do virtuálneho radu



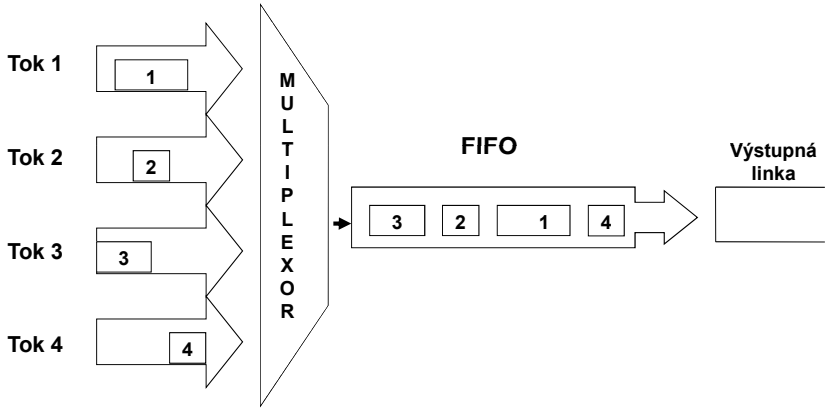
Pred príchodom paketu Po príchode paketu

Reálny rad Virtuálny rad Označený paket

Adaptive Virtual Queue (AVQ)

- AVQ rozhoduje o označení/zahodení na základe rýchlosti prichádzajúcich dát a nie na základe dĺžky radu,
  ➔ zabezpečuje to rýchlu reakciu systému.
- Namiesto dĺžky radu reguluje využitie linky
  ➔ dosahuje vyššiu odolnosť a stabilitu pri výskyte extrémne krátkych tokov dát, alebo pri meniacom sa počte dlhých tokov.
- Dĺžka radu pri AVQ je v porovnaní s inými metódami pomerne nízka a výrazne sa nezvyšuje ani pri rastúcom zaťažení siete.
  ➔ Využitie linky sa stabilne pohybuje okolo požadovanej úrovne (teda nie je maximálne)
  ➔ Straty sú nižšie ako pri iných algoritmoch.

FIFO



FIFO

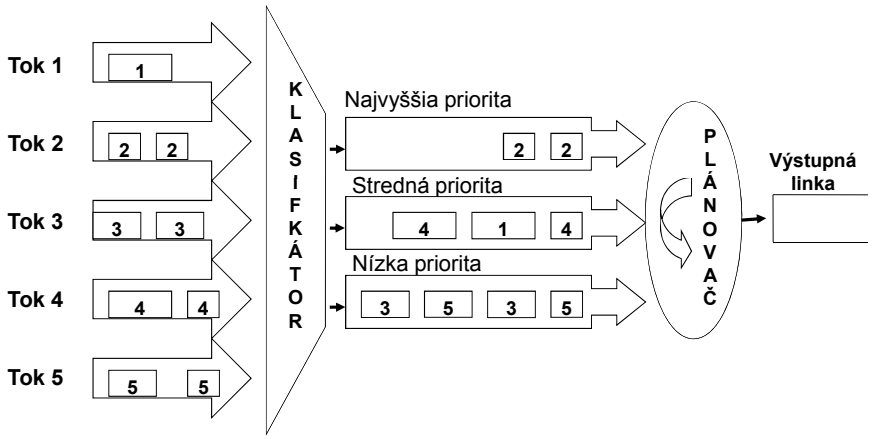
Výhody:

- Oneskorenie závisí len od veľkosti radu
- Nízka výpočtová náročnosť
- Zachováva zhlukový charakter vstupnej premávky

Nevýhody:

- Neumožňuje diferencovaný prístup k paketom rôznych tried
- Nevhodný pre real-time a garantované služby
- Oneskorenie všetkých paketov rastie úmerne s nárastom zaťaženia siete
- Zvýhodňuje UDP toky pred TCP tokmi
- Nepriamo podporuje zhlukové toky - môžu pohltiť väčšinu kapacity FIFO radu

Priority Queuing (PQ)



Priority Queuing (PQ)

Rozoznávame dva modely implementácie PQ:

- **PQ so striktnou prioritou** *(Strict PQ)* - pakety s vyššou prioritou sú vždy odoslané pred paketmi s nižšou prioritou.

- **PQ s nastaviteľnou šírkou pásma** *(Rate-controlled PQ)* **–** pakety s vyššou prioritou sú uprednostnené pred paketmi s nižšou prioritou iba ak prevádzka vo vyššej priorite nepresahuje stanovenú úroveň (napr. 20% výstupnej šírky pásma).

Priority Queuing (PQ)

Výhody:

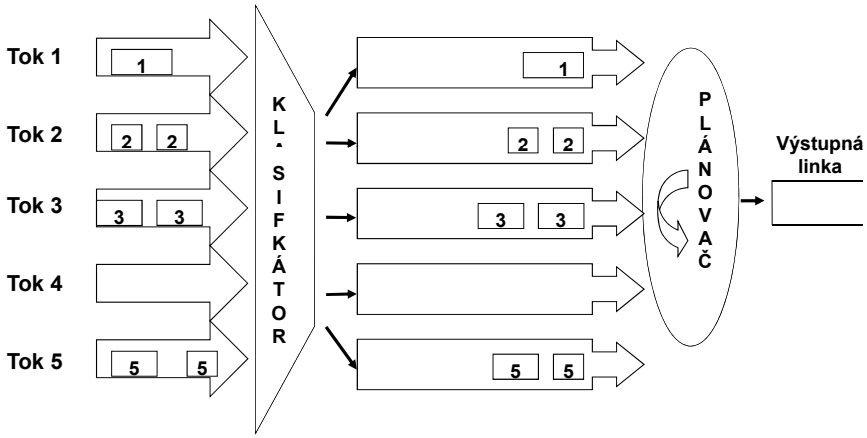
- Nízka výpočtová náročnosť - vhodné aj pre softvérové smerovače.
- PQ umožňuje zaviesť diferencovanie prevádzky. *Prevádzka s vyššou prioritou (napr. real-time prevádzka citlivá na oneskorenie) môže byť uprednostnená pred best-effort prevádzkou.*
- Vie zabezpečiť stabilitu siete počas zahltenia priradením najvyššej priority riadiacim signálom siete.

Priority Queuing (PQ)

Nevýhody:

- Ak sa tok s vyššou prioritou približuje ku kapacite linky alebo ju presahuje, dochádza k výraznému oneskoreniu alebo až zastaveniu služby s nižšou prioritou.
- Triedy s rovnakou prioritou sú smerované rovnako, ako keby bolo použité FIFO radenie.
- Nerieši problém férovosti medzi TCP a UDP.
  Ak je TCP v triede s vyššou prioritou a neobmedzí sa, snaží sa využiť celú kapacitu linky na úkor UDP.

Fair Queuing (FQ)



Fair Queuing (FQ)

Výhody:

- Extrémne zhlukové toky, alebo toky ktoré sa chovajú zle, nemajú žiadny vplyv na QoS poskytovanú ostatným tokom, pretože toky sú navzájom izolované.

Fair Queuing (FQ)

Nevýhody:

- Toky sú obsluhované rovnako ➔ nie je možné zvýhodniť jeden pred ostatnými a tak zabezpečiť QoS.
- Rovnaká obsluha je zachovaná iba pri paketoch presne rovnakej veľkosti ➔toky s dlhšími paketmi dostanú väčšiu časť odchodzej kapacity ako toky s krátkymi paketmi.
- Je závislý na poradí príchodu paketov.
- Fair queuing nemá jednoduchý mechanizmus na podporu real-time služieb.

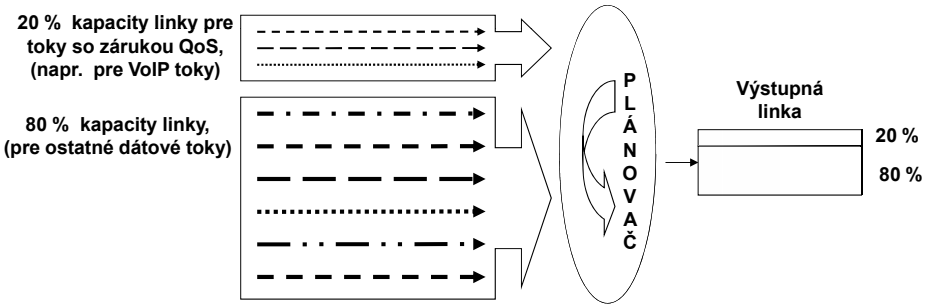
Fair Queuing (FQ)

Nevýhody *(pokr.)*:

- Vo veľkých IP sieťach s desať tisícmi spojení by bolo potrebné obsluhovať pre každé spojenie samostatný FIFO rad ➔ neúnosné, spomaľujúce a ťažké na efektívnu implementáciu.
- FQ predpokladá, že vieme jednoducho a presne klasifikovať pakety do tokov. ➔ možné zneužitie otvorením viacerých tokov

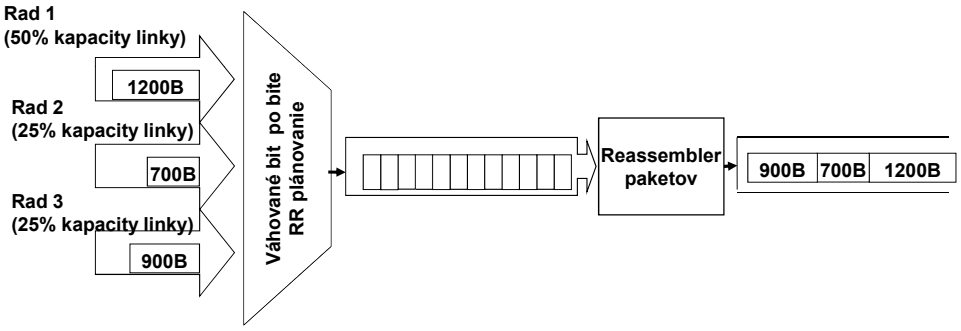
Class-based Fair Queuing (FQ)

Vylepšenie - rozčlenenie kapacity linky podľa tried



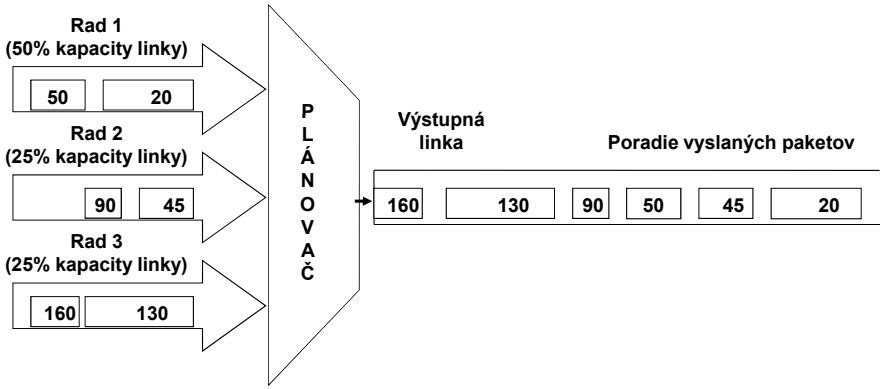
Weighted Fair Queuing (WFQ)

WbbRR s reasemblerom paketov



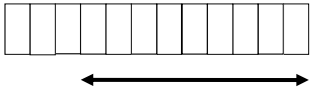
Weighted Fair Queuing (WFQ)

Princíp Weighted Fair Queuing metódy

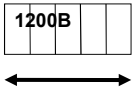


Weighted Fair Queuing (WFQ)

Spôsob počítania PFT



PFT



Čas potrebný na odoslanie paketu

Weighted Fair Queuing (WFQ)

Výhody:

- Každý rad má garantovanú minimálnu výstupnú kapacitu nezávisle na správaní ostatných radov.
- Pri kombinácii s riadením prevádzky na okraji siete, WFQ zaručuje rovnomerné rozdelenie výstupnej kapacity, vzhľadom na váhu jednotlivých radov, s obmedzením oneskorením.

Weighted Fair Queuing (WFQ)

Nevýhody:

- Výpočet a zoradenie paketov podľa časov je výpočtovo náročné
- Silná zhluková prevádzka môže ovplyvniť ostatné toky v tom istom rade.
- Výpočtová zložitosť obmedzuje použitie WFQ v rýchlych zariadeniach s veľkým počtom tried.
- Aj napriek garantovanému oneskoreniu môžu byť iné algoritmy lepšie pre svoj jednoduchší výpočet.
- Pri N radoch s váhami $w_1,w_2,...,w_N$ bude i-temu radu pridelená približne rýchlosť

$$R_i = \frac{R.w_i}{\sum_{i=1}^{N} w_i}$$

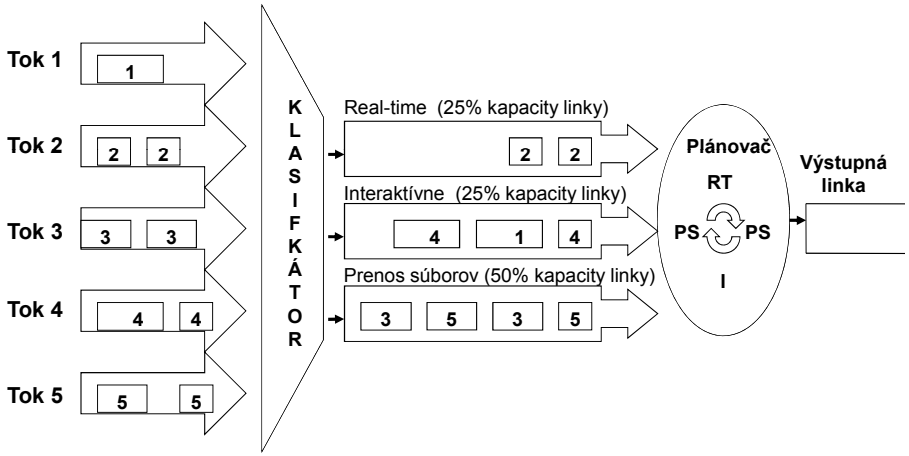
Weighted Fair Queuing (WFQ)

- **Vylepšenia WFQ**:
  - **Class-based WFQ (CBWFQ):**
    - Používa užívateľom definované triedy na základe rôznych parametrov ako je protokol, vstupné rozhranie, nastavenie prioritných bitov v IP protokole.
    - Pre každú triedu je vyhradený rad s pridelenými parametrami ako šírka pásma, váha a maximálna veľkosť paketu a ich počet v rade.
    - Pri prekročení tohto limitu dochádza k zahodeniu paketov.
  - **Self-clocking Fair Queuing (SCFQ):**
    - Zjednodušuje výpočtovú náročnosť času, podľa ktorého sa pakety odosielajú, čo znižuje oneskorenie paketov.

Weighted Fair Queuing (WFQ)

- **Vylepšenia WFQ**:
  - **Worst-case Fair Weighted Fair Queuing (WF^2Q):**
    - Približuje sa viac ku GPS *(General Processor Sharing)*.
    - Neuvažuje len čas, za ktorý by sa paket odoslal v GPS, ale aj čas príchodu paketu.
  - **Worst-case Fair Weighted Fair Queuing+ (WF^2Q+)**:
    - Implementuje novú funkciu virtuálnych hodín.
    - Je výpočtovo menej náročný a presnejší.

Weighted Round Robin (WRR)



Weighted Round Robin (WRR)

Výhody:

- WRR môže byť implementovaný hardvérovo, takže môže byť použitý na vysokorýchlostných zariadeniach chrbticovej siete.
- WRR vykonáva hrubú reguláciu percenta výstupnej kapacity pridelenej jednotlivým službám.
- V každom kole sa z každého FIFO radu vyšle aspoň jeden paket, t.j. nemôže dôjsť k „vyhladovaniu“.

Weighted Round Robin (WRR)

Nevýhody:

- Správne rozdelenie výstupnej kapacity dosiahneme len pri zachovaní rovnakej veľkosti paketov.
  (Vhodné pre siete s fixnou dĺžkou paketov ako je napr. ATM.)

Deficit Weighted Round Robin (DWRR)

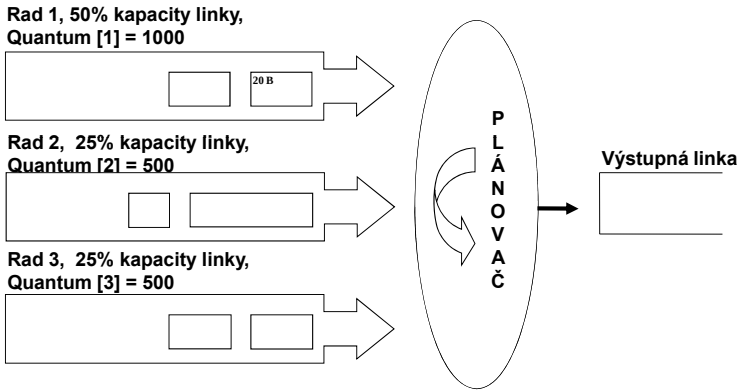
- Má odstrániť nedostatky WRR aj WFQ
- Definuje niekoľko tried služieb, ktoré majú vlastné FIFO rady.
- Pre každú triedu sú definované nasledovné parametre:
  - **Váha [F]** – pre každú triedu (rad F) definované percento z výstupnej kapacity linky
  - **DeficitCounter [F]** – premenná, ktorá pre každú triedu udáva, koľko bajtov z nej bolo v jednom kole DWRR poslaných
  - **Quantum [F]** – max počet bajtov, ktoré môže daný FIFO rad vyslať v jenom kole DWRR. Kvantum je odvodené od váhy pridelenej triede.

Deficit Weighted Round Robin (DWRR)

Každý rad F je obslúžený nasledovne:

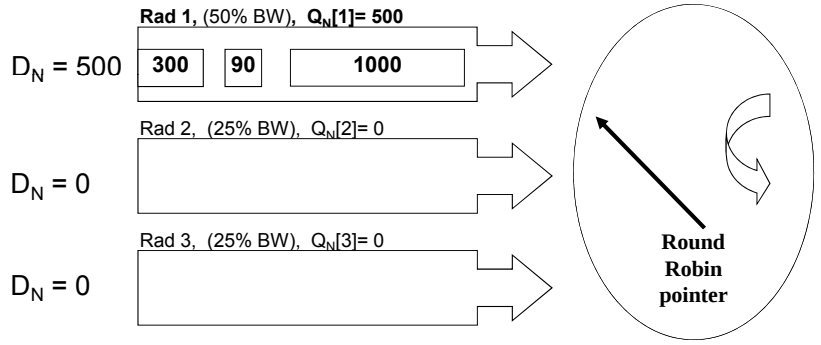
1. nastaví sa DeficitCounter:
   DeficitCounter[F] = DeficitCounter[F] + Quantum[F]
2. z danej triedy F sa pošle toľko paketov, aby suma ich dĺžok bola nanajvýš DeficitCounter[F].
   Nech je to B bajtov.
3. DeficitCounter[F] = DeficitCounter[F] – B
4. Obslúži sa ďalší rad F

Deficit Weighted Round Robin (DWRR)



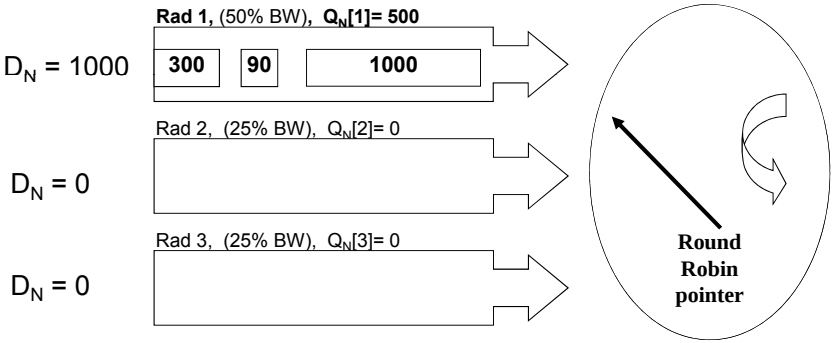
Deficit Weighted Round Robin (DWRR)

1. RR kolo



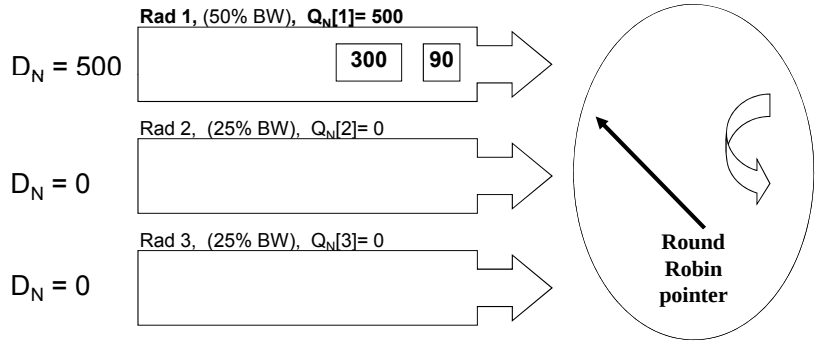
Deficit Weighted Round Robin (DWRR)

2. RR kolo



Deficit Weighted Round Robin (DWRR)

3. RR kolo



Deficit Weighted Round Robin (DWRR)

Výhody:

- Rady sa navzájom neovplyvňujú.
- Zaručuje presné pridelenie výstupnej kapacity aj pri paketoch rôznej dĺžky.
- DWRR zaručuje, že každý rad má prístup k výstupnej kapacite. Nemôže dôjsť k „vyhladovaniu“.
- Jednoduchý a výpočtovo nenáročný algoritmus ➔ možno ho nasadiť na vysokorýchlostné linky

Deficit Weighted Round Robin (DWRR)

Nevýhody:

- Nevie presne garantovať oneskorenie.
- Pri odosielaní paketov kým DeficitCounter nie je menší ako veľkosť paketu na začiatku radu, spôsobuje kolísanie oneskorenia (jitter). To sťažuje implementáciu real-time prevádzky.
- Toky v rámci radu sa navzájom ovplyvňujú .