

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SYSTEMY FRONT AKTIVNÍCH PRVKŮ PRO ŘÍZENÉ ODESÍLÁNÍ
PAKETŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ VALA

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SYSTÉMY FRONT AKTIVNÍCH PRVKŮ PRO ŘÍZENÉ ODESÍLÁNÍ PAKETŮ

QUEUING SYSTEMS OF ACTIVE NETWORKS FOR CONTROLLED PACKET SENDING

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ONDŘEJ VALA

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. JAROSLAV KOTON, Ph.D.

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Ondřej Vala

ID: 120807

Ročník: 3

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Systémy front aktivních prvků pro řízené odesílání paketů

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku systémů front aktivních síťových prvků. Popište jejich vlastnosti s ohledem na datový provoz, který je daným systémem front obsluhován. Pozornost pak věnujte také popisu algoritmů, které se využívají v rámci jednotlivých systémů front zajišťující požadovanou kvalitu služby. Ve vhodném simulačním software implementujte dané systémy front a jejich chování analyzujte.

DOPORUČENÁ LITERATURA:

[1] MEDHI, D., RAMASAMY, K.: Network Routing: Algorithms, Protocols, and Architectures, Morgan Kaufmann, ISBN: 978-0-12-088588-6, 2007.

[2] CHAO, H.J., LIU, B.: High Performance Switches and Routers, Wiley-IEEE Press, ISBN: 978-0-4700-5367-6, 2007.

Termín zadání: 9.2.2015

Termín odevzdání: 2.6.2015

Vedoucí práce: doc. Ing. Jaroslav Koton, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce se zabývá popisem systému front hromadné obsluhy pro zajištění kvality služeb v telekomunikačních sítích. Práce se skládá z teoretické a praktické části. V teoretické části jsou popsány typy algoritmů, které se využívají v aktivních prvcích a ochranné mechanismy zahlcení front. V praktické části je rozbor dílčích částí jednoho z vytvořených modelů v programu Matlab - Simulink. Poslední kapitoly v praktické části jsou věnovány porovnání všech systémů z hlediska ztrátovosti paketů a zpoždění paketů ve frontách při konstantní vstupní intenzitě datových toků.

KLÍČOVÁ SLOVA

Teorie front, správa front, kvalita služeb, Matlab, Simevents, aktivní prvek

ABSTRACT

This bachelor thesis is describing queueing systems to provide quality of services in telecommunication networks. The work consists of a theoretical and practical part. The theoretical part describes the types of algorithms, which are used in active components and protective mechanisms against queues congestion. In the practical part, there is the analysis of the real Matlab-Simulink model. The last chapter of the practical part is devoted to the comparison of all the systems in terms of packet loss and packet delays in queues at constant input intensity of data streams.

KEYWORDS

Queueing theory, Congestion Management, Quality of Service, Matlab, Simevents, Active Networks

VALA, Ondřej *Systémy front aktivních prvků pro řízené odesílání paketů*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 67 s. Vedoucí práce byl doc. Ing. Jaroslav Koton, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Systémy front aktivních prvků pro řízení odesílání paketů“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Jaroslavu Kotonovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

OBSAH

Úvod	11
1 Teorie hromadné obsluhy front	12
1.1 Kendallova klasifikace systému hromadné obsluhy	13
1.2 Systém M/M/1	15
2 Správa front	17
2.1 Pasivní správa front	17
2.2 Aktivní správa front	17
2.2.1 RED	18
2.2.2 WRED	19
3 Typy front na aktivních prvcích	20
3.1 FIFO	20
3.2 Prioritní fronty - PQ	21
3.2.1 Striktní prioritní obsluha front - SPQ	21
3.2.2 Spravedlivá obsluha front - FQ	22
3.2.3 Vážená cyklická obsluha - WRR	23
3.2.4 Vážená spravedlivá obsluha - WFQ	24
3.2.5 Systém front s váženou spravedlivou obsluhou řízenou podle tříd - CBWFQ	25
3.2.6 Low latency queueing - LLQ	25
4 Realizace systému hromadné obsluhy v programu Matlab	28
4.1 Výpočet stacionárního režimu	29
4.2 Realizace společných jednotlivých bloků systému hromadné obsluhy	30
4.2.1 Generátor entit (paketů)	30
4.2.2 Aktivní správa front RED	31
4.2.3 Vstupní vyrovnávací paměť	34
4.2.4 Sdružování datových toků do jediného	35
4.2.5 Kanál obsluhy a rychlost linky	36
4.2.6 Váhování tříd v systému CBWFQ	37
4.2.7 Řídící algoritmus CBWFQ - vnější pohled	37
4.2.8 Řídící algoritmus CBWFQ - vnitřní pohled	38
4.3 Vyhodnocení systému	40
5 Závěr	46

Literatura	47
Seznam symbolů, veličin a zkratk	48
Seznam příloh	49
A Příloha	50
A.1 Výsledky simulace	50

SEZNAM OBRÁZKŮ

1.1	Systém hromadné obsluhy	12
2.1	Agresivní zahazování paketů, převzato z [4]	17
2.2	Předčasné zahazování paketů, převzato z [4]	18
2.3	Profil zahazování paketů RED, převzato z [4]	19
2.4	Znázornění parametrů α ve frontě, převzato z [4]	19
3.1	FIFO	20
3.2	Striktní prioritní obsluha front, převzato z [3]	22
3.3	Spravedlivá obsluha front (pakety se stejnou délkou), převzato z [3]	23
3.4	Spravedlivá obsluha front (pakety se rozdílnou délkou), převzato z [3]	23
3.5	WRR, převzato z [3]	24
3.6	CBWFQ, převzato z [3]	26
3.7	Low Latency Queue	26
3.8	Profil zahazování paketů WRED převzato z [3]	27
4.1	Generátor entit (paketů) - nastavení vstupních parametrů	31
4.2	Generátor entit pro jeden datový tok (paketů) - celkový pohled	31
4.3	Výpočet váženého průměru	32
4.4	Řídící člen RED	33
4.5	Pravoúhlý trojúhelník	34
4.6	Profil zahazování RED	34
4.7	Vnitřní struktura řídicího členu profilu zahazování RED	35
4.8	Vstupní vyrovnávací paměť jednoho datového toku.	35
4.9	Sdružování datových toků do jediného - multiplexer	36
4.10	Kanál obsluhy a nastavení rychlosti linky	37
4.11	Realizace váhování tříd podle vzorce 4.6	38
4.12	Vstupní parametry algoritmu CBWFQ	44
4.13	Vstupní blok algoritmu CBWFQ pro definování proměných	45
4.14	Část vnitřní struktury algoritmu CBWFQ v bloku Char	45
A.1	FIFO - Ztrátovost paketů na čase - scénář 1	50
A.2	FIFO - Zpoždění paketů ve frontě - scénář 1	50
A.3	FIFO - Ztrátovost paketů na čase - scénář 2	51
A.4	FIFO - Zpoždění paketů ve frontě - scénář 2	51
A.5	FIFO - Ztrátovost paketů na čase - scénář 3	52
A.6	FIFO - Zpoždění paketů ve frontě - scénář 3	52
A.7	FQ - Ztrátovost paketů na čase - scénář 1	53
A.8	FQ - Zpoždění paketů ve frontě - scénář 1	53
A.9	FQ - Ztrátovost paketů na čase - scénář 2	54
A.10	FQ - Zpoždění paketů ve frontě - scénář 2	54

A.11 FQ - Ztrátovost paketů na čase - scénář 3	55
A.12 FQ - Zpoždění paketů ve frontě - scénář 3	55
A.13 SPQ - Ztrátovost paketů na čase - scénář 1	56
A.14 SPQ - Zpoždění paketů ve frontě - scénář 1	56
A.15 SPQ - Ztrátovost paketů na čase - scénář 2	57
A.16 SPQ - Zpoždění paketů ve frontě - scénář 2	57
A.17 SPQ - Ztrátovost paketů na čase - scénář 3	58
A.18 SPQ - Zpoždění paketů ve frontě - scénář 3	58
A.19 WRR - Ztrátovost paketů na čase - scénář 1	59
A.20 WRR - Zpoždění paketů ve frontě - scénář 1	59
A.21 WRR - Ztrátovost paketů na čase - scénář 2	60
A.22 WRR - Zpoždění paketů ve frontě - scénář 2	60
A.23 WRR - Ztrátovost paketů na čase - scénář 3	61
A.24 WRR - Zpoždění paketů ve frontě - scénář 3	61
A.25 WFQ - Ztrátovost paketů na čase - scénář 1	62
A.26 WFQ - Zpoždění paketů ve frontě - scénář 1	62
A.27 WFQ - Ztrátovost paketů na čase - scénář 2	63
A.28 WFQ - Zpoždění paketů ve frontě - scénář 2	63
A.29 WFQ - Ztrátovost paketů na čase - scénář 3	64
A.30 WFQ - Zpoždění paketů ve frontě - scénář 3	64
A.31 CBWFQ - Ztrátovost paketů na čase - scénář 1	65
A.32 CBWFQ - Zpoždění paketů ve frontě - scénář 1	65
A.33 CBWFQ - Ztrátovost paketů na čase - scénář 2	66
A.34 CBWFQ - Zpoždění paketů ve frontě - scénář 2	66
A.35 CBWFQ - Ztrátovost paketů na čase - scénář 3	67
A.36 CBWFQ - Zpoždění paketů ve frontě - scénář 3	67

SEZNAM TABULEK

3.1	Vliv velikosti paketů na výslednou šířku pásma v modelu WRR . . .	25
4.1	Hodnoty vstupní intenzity pro nastavení scénáře 1	42
4.2	Hodnoty vstupní intenzity pro nastavení scénáře 2	42
4.3	Hodnoty vstupní intenzity pro nastavení scénáře 3	43

ÚVOD

V dnešní době kdy společnost využívá telekomunikační systémy na denní bázi a uživatelé jsou čím dál víc náročnější na kvalitu připojení, je nutné odlišovat jednotlivé datové toky a rozdílně s každým datovým tokem nakládat pro zajištění požadované kvality služeb.

To je zajištěno rozdělení datových toků např. do tříd provozu, zde lze aplikovat prioritní obsluhu pro zajištění požadované kvality služeb na úkor ostatních datových toků.

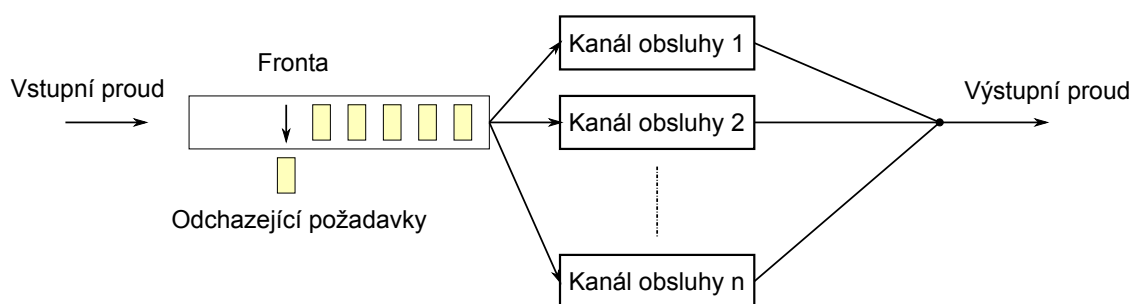
Cílem této bakalářské práce je popsat tyto systémy hromadné obsluhy, které se používají v telekomunikačních technologiích na aktivních prvcích pro odesílání paketů. Popsat jednotlivé algoritmy, které jsou využívány k zajištění kvality na síti. A následně je porovnat při konstantním vstupním toku pro všechny typy obsluh.

Porovnávané vlastnosti byly ztrátovost paketů a zpoždění paketů ve vstupní vyrovnávací frontě.

V úvodních kapitolách této práce jsou popsány teoretické základy SHO, ochranné mechanismy proti zahlcení front. Ve třetí kapitole jsou popsány jednotlivé druhy SHO od těch nejjednodušších po komplexní systémy hromadné obsluhy. Čtvrtá kapitola se věnuje realizaci SHO v programu Matlab.

1 TEORIE HROMADNÉ OBSLUHY FRONT

Systém hromadné obsluhy (SHO) je obslužné zařízení, tvořené jedním až n obslužných kanálů, vykonávající obsluhu přichozích „zákazníků“ s požadavkem na obsluhu. Základní schéma SHO je zobrazeno na obrázku 1.1. Zákazníci v přicházejícím proudu vstupují do systému hromadné obsluhy. Pokud v době, kdy vstoupí zákazník s požadavkem na obsluhu, není volný žádný obslužný kanál, zákazník se zařadí do fronty. Tato fronta vzniká před obslužnými kanály do té doby, než se jednotliví zákazníci dostanou ve frontě na řadu a uvolní se obslužný kanál. Poté, co je zákazník obsloužen, opouští systém hromadné obsluhy.[1][3][5]



Obr. 1.1: Systém hromadné obsluhy

Systém hromadné obsluhy vždy obsahuje následující prvky:

Vstupní zdroj - zdroj požadavku rozlišujeme podle:

- Počtu požadavků - omezený zdroj má konečný počet požadavků (typické pro uzavřené systémy) a je závislý na stavu obslužného systému. Neomezený zdroj má nekonečný počet vstupních požadavků a vstupní proud je nezávislý na stavu obslužného systému.
- Povahy požadavků - deterministické požadavky přicházející do systému pravidelně po sobě. Stochastické požadavky přicházejí do systému náhodně.
- Podle druhu příchodu - stejnorodé požadavky požadují identický druh obsluhy. Různorodé požadavky požadují rozdílné druhy obsluhy.
- Podle příchodu - požadavky přicházející jednotlivě nebo skupinově.
- Podle intenzity vstupu - intenzita vstupních požadavků je konstantní, nebo proměnlivá.

Fronta - se tvoří před kanálem obsluhy, pokud požadavky na obsluhu přicházející do SHO jsou větší než množství, které dokážou obsloužit dané kanály obsluhy. Fronta může mít kapacitu buď neomezenou, nebo omezenou, jejíž kapacita je pevně dána a pokud dojde zaplnění této kapacity, jsou nové přicházející požadavky odmítány.

Požadavky ve frontě se dělí podle trpělivosti na trpělivé požadavky, které čekají ve frontě tak dlouho, než budou obslouženy a netrpělivé požadavky, které odcházejí z fronty, pokud překročí práh trpělivosti. Práh může být například počet požadavků ve frontě, nebo čas strávený ve frontě. Fronty se dělí podle způsobu odchodu z fronty:

- FIFO - první přijde, první odchází
- LIFO - poslední přijde, první odchází
- SIRO - náhodné pořadí odchodu
- PRIO - podle priority
- GE - obecné řazení front

Obslužné kanály - jsou tvořeny jedním až n obslužnými zařízeními. Mohou být řazeny sériově, paralelně, případně jejich kombinací. Doba obsluhy se liší pro každý požadavek, protože požadavky mohou mít deterministický nebo stochastický charakter doby obsluhy. Počet kanálů může být dán buď pevně, nebo adaptivně - v tomto případě se počet obslužných kanálů mění podle potřeby.

1.1 Kendallova klasifikace systému hromadné obsluhy

V roce 1951 D. G. Kendall definoval základy klasifikace SHO podle tří hlavních hledisek ve tvaru A/B/C, tato klasifikace byla později rozšířena na tvar A/B/C/D/E/F, kde

- A - charakterizuje typ pravděpodobností rozdělení vstupních intervalů
- B - charakterizuje typ pravděpodobnostního rozložení doby obsluhy
- C - je počet paralelně uspořádaných obslužných linek
- D - přirozené číslo, které udává maximální počet požadavků v systému
- E - přirozené číslo, které udává max. počet požadavků ve vstupním proudu
- F - typ fronty (FIFO, LIFO, SIRO, PRIO)

Parametr A a B může nabývat následujících hodnot:

- M - exponenciální rozdělení (Poissonův proces)
- D - deterministické rozdělení (konstantní)
- E_k - Erlangovo rozdělení s parametry λ a k
- G - libovolné rozdělení
- N - Gaussovo rozdělení
- U - rovnoměrné rozdělení

Veličiny

Pro potřeby popisu chování systému hromadné obsluhy jsou využívány statické parametry, mezi nejčastěji vyhodnocované patří:

- λ - intenzita vstupního proudu, vyjádřená jako počet požadavků vstupujících do systému za jednotku času
- μ - intenzita obsluhy je počet požadavků zpracovaných za jednotku času
- u - intenzita provozu je poměr požadavků vstupujících do systému ku počtu zpracovaných za jednotku času
- l_q - okamžitý počet požadavků v SHO
- L_q - střední počet požadavků v SHO
- L_w - střední délka fronty
- L_s - střední hodnota obsluhovaných požadavků
- t_q - doba průchodu celým SHO pro jeden konkrétní požadavek
- T_q - střední doba průchodu celým SHO pro jeden konkrétní požadavek
- T_s - střední doba obsluhy
- T_w - střední doba strávená ve frontě
- n - počet obslužných kanálů

Elementární vztahy a pojmy

- Intenzita provozu je dána vztahem (1.1), kde u je nejbližší vyšší celé číslo a udává minimální počet obslužných kanálů pro zachování stacionárního režimu SHO.

$$u = \frac{\lambda}{\mu} \quad (1.1)$$

- Koefficient využití obslužných kanálů, daný vztahem (1.2), je jeden z nejdůležitějších parametrů SHO pro zachování stacionárního režimu. Podmínka pro stacionární režim je $\rho < 1$. Pokud není tato podmínka splněna, narůstá v čase neomezeně délka fronty.

$$\rho = \frac{1}{n} \frac{\lambda}{\mu} \quad (1.2)$$

Poissonův (Markovův) proces

V telekomunikačních sítích má většina příchozích proudů do SHO tento charakter. Vyznačuje následujícími vlastnosti:

- Stacionárnost - pravděpodobnost charakteristiky Poissonova toku se v čase nemění. Pravděpodobnost počtu události je nezávislá na počátku pozorování.

- Regulárnost - pravděpodobnost více než jedné události v malém časovém intervalu Δt je velmi malá. V sledovaném časovém intervalu Δt se vyskytne, buď jediná událost, nebo žádná.
- Nezávislost přírůstku - počet událostí, které se vyskytnou v jednom časovém intervalu, nezávisí na počtu událostí v jiných intervalech.[3][5]

1.2 Systém M/M/1

Systém M/M/1 je základní model SHO, kde požadavky vstupují do systému jednotlivě, nezávisle po sobě. Vstupní tok požadavků má exponenciální charakter (Poissonův proces), což nejlépe odpovídá reálným datovým tokům v telekomunikačních sítích. Jak bylo zmíněno v předchozí kapitole, Poissonův vstupní proud má stacionární charakter, takže volba počátečního okamžiku v čase t_0 nemá žádný význam, důležitá je pouze délka intervalu Δt .

Obslužné kanály, v tomto případě jeden obslužný kanál, má stejný exponenciální charakter obsluhy jako vstupní proud. Doba obsluhy je náhodná hodnota, která je nezávislá na vstupním proudu.

Pro provoz systému hromadné obsluhy je důležité zachování stacionárního stavu. Toho je docíleno tím, že intenzita vstupního proudu λ je menší než intenzita obsluhy μ . Čím menší je intenzita vstupního proudu, tím méně čeká požadavků ve frontě před kanálem obsluhy. Pokud se situace obrátí a výsledné ρ se začne blížit k jednotkové hodnotě, začne se tvořit fronta s nekonečnou délkou před obslužným kanálem. V případě praktického SHO, který využívá frontu konečné délky, to znamená, že v systému začnou nutně vznikat ztráty, což většinou není žádoucí.[3]

Vzorce

V případě očekávaného stacionárního chování SHO lze využitím intenzity vstupu λ a intenzity obsluhy μ určit střední hodnotu počtu požadavků v systému a počtu požadavků ve frontě následujícími vztahy:

Střední počet požadavků ve systému L_q :

$$L_q = \frac{\rho}{1 - \rho} \quad (1.3)$$

Počet požadavků v frontě L_w :

$$L_w = \frac{\rho^2}{1 - \rho} \quad (1.4)$$

kde ρ je vyjádřeno ve vzorci (1.2)

Z výše uvedených vztahů, lze pak pomocí Litteleových vzorců [9] vyjádřit i průměrnou dobu, kterou požadavek tráví v systému či jen ve frontě:

Doba strávená v systému T_q :

$$T_q = \frac{L_q}{\lambda} \quad (1.5)$$

Střední doba strávena ve frontě T_w :

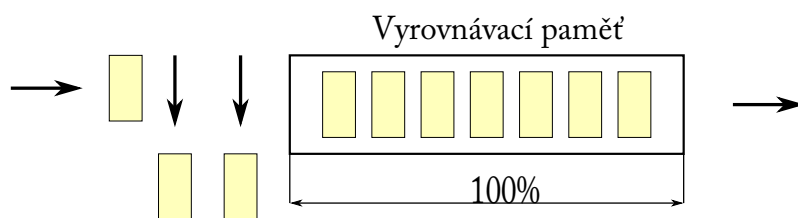
$$T_w = T_q - T_s = T_q - \frac{1}{\mu} \quad (1.6)$$

2 SPRÁVA FRONT

Slouží jako ochranný mechanismus vyrovnávací paměti aktivního prvku a také síťového provozu. Zamezuje přetížení aktivního prvku, pokud jsou příchozí datové toky větší než zvládne aktivní prvek odesílat. Správu front můžeme rozdělit na pasivní a aktivní. Pasivní fronta představuje mechanismus „Tail-Drop“ a jeho modifikace, které jsou popsány v kapitole 2.1, aktivní správa front je popsána v kapitole 2.2. [2][3][4]

2.1 Pasivní správa front

Pasivní správa front pracuje na mechanismu zahazování příchozích paketů „Tail-Drop“ tak, že zahazuje příchozí pakety, pokud je vyrovnávací paměť aktivního prvku plná. Tento způsob zacházení s pakety má nevýhodu hlavně pro spojově orientované přenosy, díky tomu, že jsou zahozené pakety považovány za ztracené a jsou tedy znova vyžádány a odeslány na aktivní prvek. Toto má za následek snížení přenosové rychlosti spojově orientovaných přenosů a další zatížení sítě. Tento jev se nazývá globální synchronizace TCP.



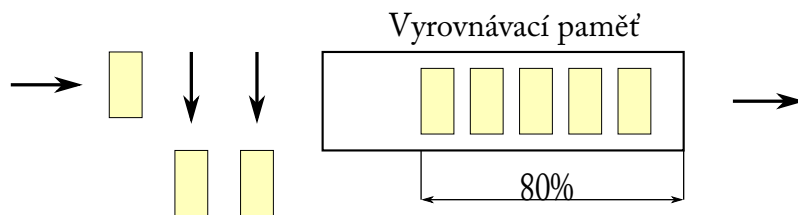
Obr. 2.1: Agresivní zahazování paketů, převzato z [4]

Early dropping

V literatuře je také popisována metoda Early dropping [4], která je jednoduchou modifikací principu Tail-Drop, kdy k zahazování přicházejících paketů nedochází až v okamžiku, kdy je vyrovnávací paměť zcela plná, ale již v okamžiku, kdy je vyrovnávací paměť využita z např. 80%, jak je to uvedeno na obr. 2.2.

2.2 Aktivní správa front

Aktivní správa front je inteligentní mechanismus. Na rozdíl od pasivní správy front, která zahazovala pakety jen podle toho, jak je zaplněna vyrovnávací paměť aktivního prvku, se aktivní správa front snaží tuto situaci eliminovat a předcházet ji. V praxi



Obr. 2.2: Předčasné zahazování paketů, převzato z [4]

se využívají tři typy aktivní správy front: detekce předčasného zahlcení „RED“, jeho modifikace vážené detekce předčasného zahlcení „WRED“ a explicitní signalizace zahlcení „ECN“.

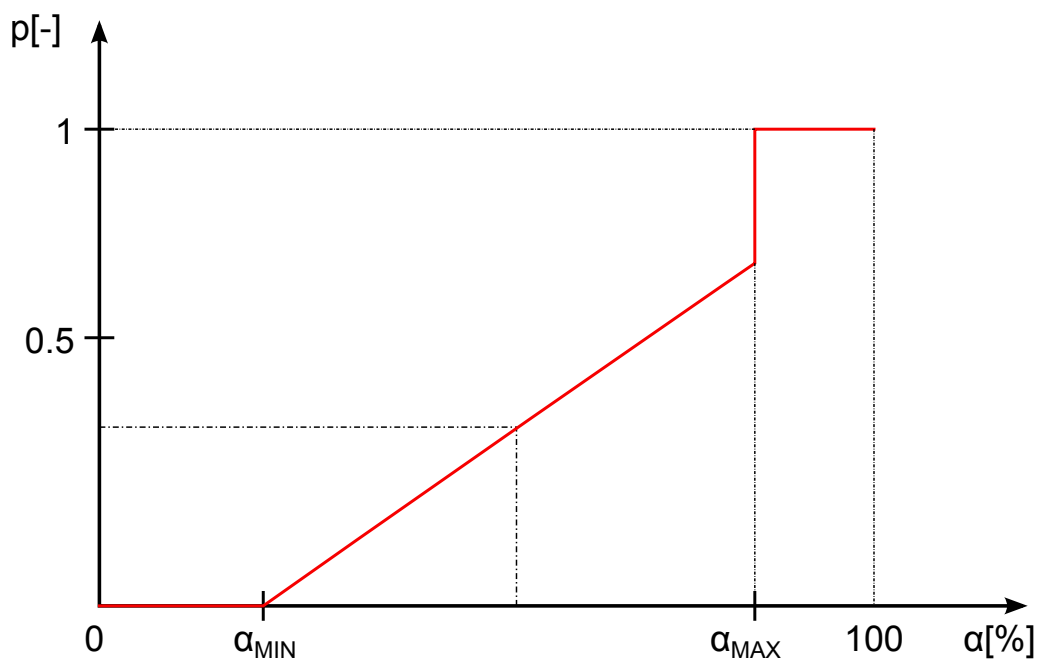
2.2.1 RED

Jak bylo zmíněno v kapitole 2.2, aktivní správa front se snaží předcházet zahlcení vyrovnávací paměti pokud tato situace nastane, začne náhodně zahazovat pakety ve frontě. Metoda RED se skládá ze tří hlavních prvků a to modulu úrovně zahlcení fronty, profilu zahazování paketů a modulu zahazování paketů. Modul zahlcení fronty především sleduje aktuální tok příchozích paketů a zaplnění vyrovnávací paměti, čímž předčasně detekuje možnost zahlcení vyrovnávací paměti. To je realizováno buď sledováním aktuální délky fronty ve vyrovnávací paměti, nebo pokročilou metodou sledování dlouhodobé délky fronty tzn. vážené průměrné délky front. Pomocí jednoduchého vzorce (2.1) lze vypočítat procentuální využití vyrovnávací paměti ze znalosti celkové kapacity paměti B a aktuálního využití kapacity paměti N .

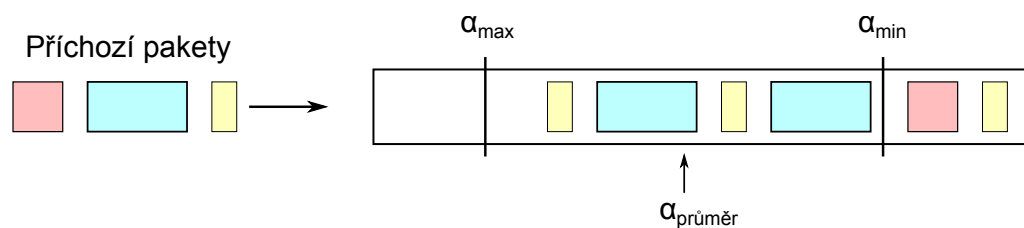
$$\alpha = \frac{N}{B} \cdot 100 \quad (2.1)$$

Další modul je tzn. barvení paketů. Ten funguje na principu přiřazení pravděpodobnosti p zahození každému paketu, tato pravděpodobnost se určuje podle křivky, která je zobrazena na obrázku 2.3, kde pravděpodobnost zahození určuje již zmíněný koeficient α zaplnění fronty. Na horizontální ose jsou dva parametry α_{min} a α_{max} , tyto dva parametry lze zvolit podle profilu pro danou vyrovnávací paměť.

Pokud je fronta zaplněna pod úroveň α_{min} , tak je příchozím paketům přiřazena pravděpodobnost $p = 0$ a metoda RED je nečinná. Pokud parametr α překročí hodnotu α_{min} , je příchozím paketům přidělena pravděpodobnost zahození až do hodnoty α_{max} a pokud hodnota α překročí i tuto hranici, tak se aktivuje zahazování metodou "Tail-Drop", která je popsána v kapitole 2.1.[2][3]



Obr. 2.3: Profil zahazování paketů RED, převzato z [4]



Obr. 2.4: Znázornění parametrů α ve frontě, převzato z [4]

2.2.2 WRED

Umožňuje, na rozdíl od RED, přiřadit v systému front každé frontě jiný profil zahazování, dokonce lze využít tzv. barvení paketů, kde lze pro jednu frontu využít více druhů profilů zahazování. Pakety mají předělenou barvu: zelenou, žlutou a červenou. Na Obr. 3.8 je vidět, že pakety označené zeleně budou mít přiřazenou menší pravděpodobnost zahazení než paket označen žlutou a červenou.

3 TYPY FRONT NA AKTIVNÍCH PRVCÍCH

Tato kapitola popisuje různé typy front, které se uplatňují na aktivních prvcích sítě.

3.1 FIFO

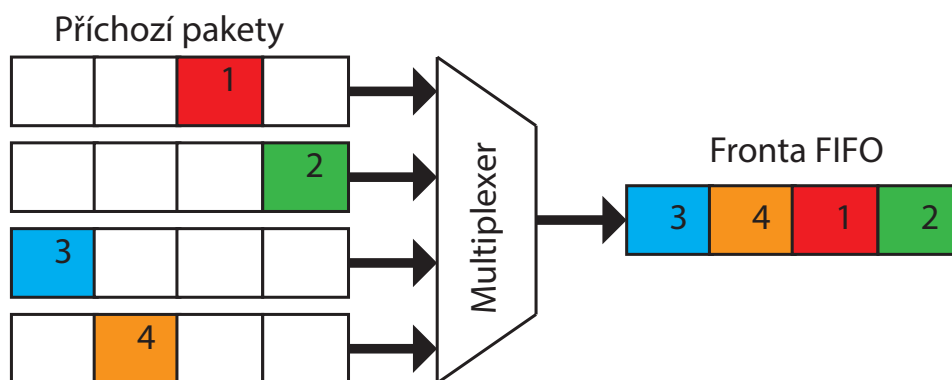
Jak už vyplývá z názvu, jedná se o jednoduchou frontu, kde paket, který přijde do fronty jako první, je také jako první obsloužen a pakety opouští frontu v takovém pořadí v jakém do fronty přišly. Pro svou jednoduchost a malou výpočetní náročnost je velmi lehce implementovatelná. Bohužel tento typ už z principiálního hlediska znemožňuje zajištění jakékoliv kvality služeb na síti (QoS) a vzniká zde předvídatelné zpoždění.

Zpoždění, které vzniká na této frontě, je přímo úměrné maximální kapacitě fronty. Požadované zpoždění lze vypočítat podle následujícího vzorce, kde D udává zpoždění ve frontě v sekundách, B velikost fronty a c je rychlost sítě.

$$D \leq \frac{B}{c} \tag{3.1}$$

Pro zajištění rozlišení provozu lze implementovat aktivní správu front RED, která je popsána v kapitole 2.2.1. Avšak při kombinovaném provozu TCP a UDP segmentů, bude zvýhodněný provoz UDP, který nemá žádné mechanismy propustnosti a uvolněné místo ve frontě okamžitě zaplní.

Tento model je výchozím typem pro řízení odesílání paketů, pokud nebyl přiřazen jiný algoritmus.[4]



Obr. 3.1: FIFO

3.2 Prioritní fronty - PQ

Princip prioritní obsluhy front je dán tím, že je vstupní proud paketů diferenciován do paralelně zapojených front FIFO, které jsou před kanálem obsluhy. Pro tento způsob zacházení je nutné rozlišovat jednotlivé pakety podle jeho atributu (priority), což může být například: z jakého zdroje paket pochází, nebo jakou službu daný paket přenáší. V dalším textu bude pro jednoduchost uváděna jen priorita. V reálné síti je kanál obsluhy například procesor v aktivním prvku. Před procesorem jsou softwarové fronty, které jsou dynamicky přidávány, nebo ubírány podle potřeby, případně podle konkrétního nastavení aktivního prvku.

V dalších kapitolách je většina systémů front popsána na jednoduchém teoretickém modelu, který obsahuje tři fronty, nebo pokud se jedná o systém založený na třídách provozu, je zjednodušen na tři třídy, kde každá třída obsahuje tři fronty. V reálných sítích mohou tyto systémy mít i mnohonásobně větší počet front, než zde demonstrováné příklady. Velikosti teoretických modelů front byly vybrány tak, aby byly přehledné a jednoduše demonstrovaly jejich výhody a nevýhody a také kvůli praktické části této práce, kde se teoretické modely simulují v programu Matlab.

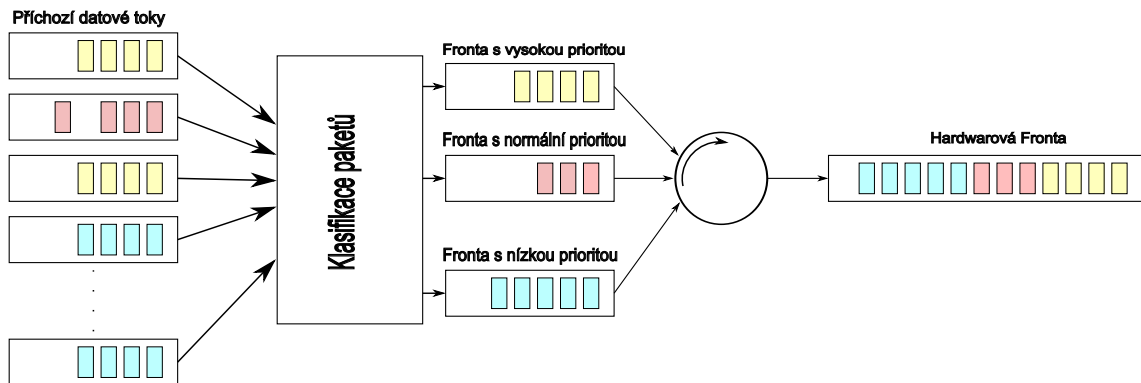
Na rozdíl od FIFO front, prioritní fronty umožňují rozdělení příchozích paketů na výstup aktivního prvku do front 1 až M , tyto fronty jsou většinou tvořeny typem FIFO podle třídy provozu (priority), kterou si nese v hlavičce každý paket.

Pokud se budeme pohybovat v sítích DiffServ, lze počítat s tím, že každý paket byl označován hraničním směrovačem a nese si v hlavičce svou prioritu (DSCP). Zde najdou uplatnění různé prioritní systémy front, které zachází s paketem podle priority, jež mu byla přidělena. Existuje mnoho algoritmů, které jsou popsány v dalších kapitolách. Tento způsob rozdělení paketů je základem pro zajištění kvality služeb na síti, také nazýváno Quality of Service (QoS).

3.2.1 Striktní prioritní obsluha front - SPQ

Tento model vždy upřednostňuje obsluhu paketů s vyšší prioritou nad těmi s prioritou nižší. Příchozí pakety jsou tříděny podle své priority a zařazeny do M front podle své třídy provozu. Pakety s vyšší prioritou mají absolutní přednost před pakety s nižší prioritou, takže paket s nižší prioritou může být obslužen, jen pokud jsou fronty s vyšší prioritou prázdné, což může mít za následek úplné potlačení provozu s nižší prioritou. Dále také mohou být tyto pakety, např. pro příliš velké zpoždění, považovány za ztracené a opětovně poslány, což může vést k dalšímu zatížení sítě. Tento model najde uplatnění zejména tam, kde je požadováno co nejmenší zpoždění pro danou službu, například pro real-time služby (VoIP, video hovory, streaming videa).

Pro lepší názornost je vytvořen obrázek, který znázorňuje teoretický model striktní prioritní obsluhy 3.2. Příchozí pakety jsou rozděleny klasifikátorem paketů do třech front typu FIFO, podle své priority. Jak je psáno v této kapitole, algoritmus obsluhuje tyto tři fronty podle priority, nejdříve obslouží všechny pakety ve frontě s vysokou prioritou, mezitím ostatní fronty čekají na obslužení. Fronty s nižší prioritou jsou obslouženy, jen pokud jsou všechny fronty s vyšší prioritou prázdné.



Obr. 3.2: Striktní prioritní obsluha front, převzato z [3]

3.2.2 Spravedlivá obsluha front - FQ

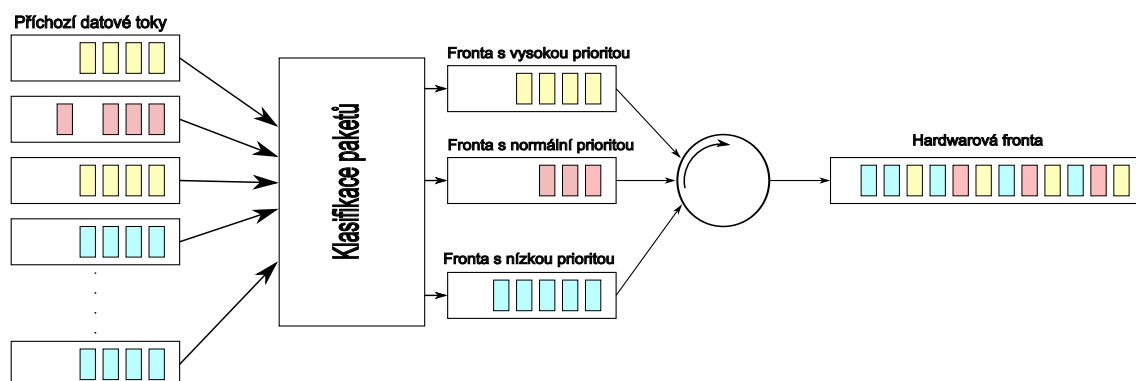
Příchozí pakety jsou rozřazovány do M front podle své třídy provozu. Fronty jsou cyklicky obsluhovány v pevném pořadí. Jak už vyplývá z názvu, každá fronta je obsluhována stejný časový okamžik a je odeslán jen jeden paket v daném časovém úseku. Toto platí pouze za předpokladu, že všechny příchozí pakety mají stejnou velikost, což jde vidět na obrázku 3.3. V reálné síti však tato situace nenastane a pakety mají rozdílnou délku. Díky tomu budou největší šířku pásma zabírat pakety s největší délkou.

Na příkladu (3.2) jsou tři datové toky, v každé frontě jsou pakety s rozdílnou délkou. Ve jmenovateli jsou sečteny všechny odchozí velikosti paketů a v čitateli je velikost paketu z daného datového toku. V třetím datovém toku má paket největší velikost, díky čemuž jsou pakety v toku jedna a dvě ochuzeny a dohromady využívají jen 44,5% šířky pásma. V reálné síti pakety s nejvyšší prioritou bývají zpravidla nejkratší, díky čemuž tento systém front může zvýhodňovat pakety s nižší prioritou nad pakety s tou vyšší. Toto právě demonstruje níže popsáný příklad a také obrázek 3.4, kde jsou příchozí pakety rozdělovány do třech front podle své priority. Jak lze vidět na hardwarové frontě, zabírá největší šířku pásma paket s nejnižší prioritou, druhou největší šířku pásma zabírá paket s normální prioritou a nejmenší šířku pásma zabírá paket s vysokou prioritou.

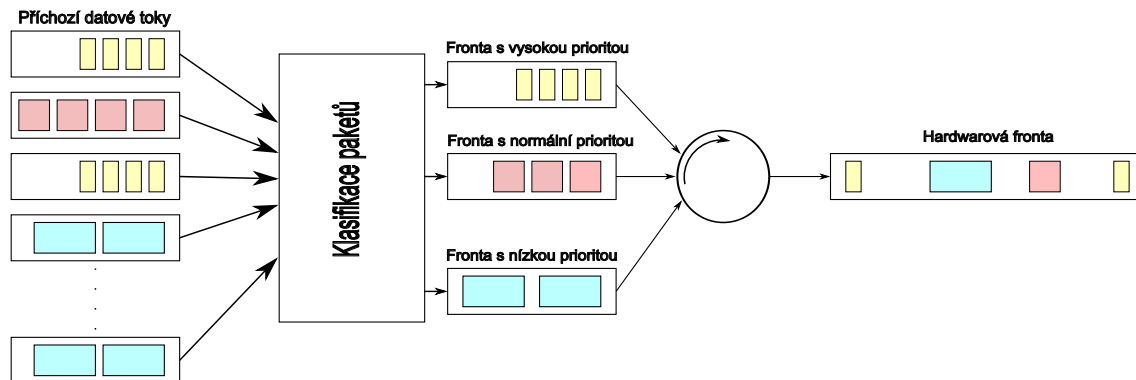
$$\text{Datový tok 1} = \frac{100\text{B}}{100\text{B} + 300\text{B} + 500\text{B}} = \frac{100\text{B}}{900\text{B}} \cong 11,11\% \quad (3.2)$$

$$\text{Datový tok 2} = \frac{300\text{B}}{100\text{B} + 300\text{B} + 500\text{B}} = \frac{300\text{B}}{900\text{B}} \cong 33,33\%$$

$$\text{Datový tok 3} = \frac{500\text{B}}{100\text{B} + 300\text{B} + 500\text{B}} = \frac{500\text{B}}{900\text{B}} \cong 55,55\%$$



Obr. 3.3: Spravedlivá obsluha front (pakety se stejnou délkou), převzato z [3]



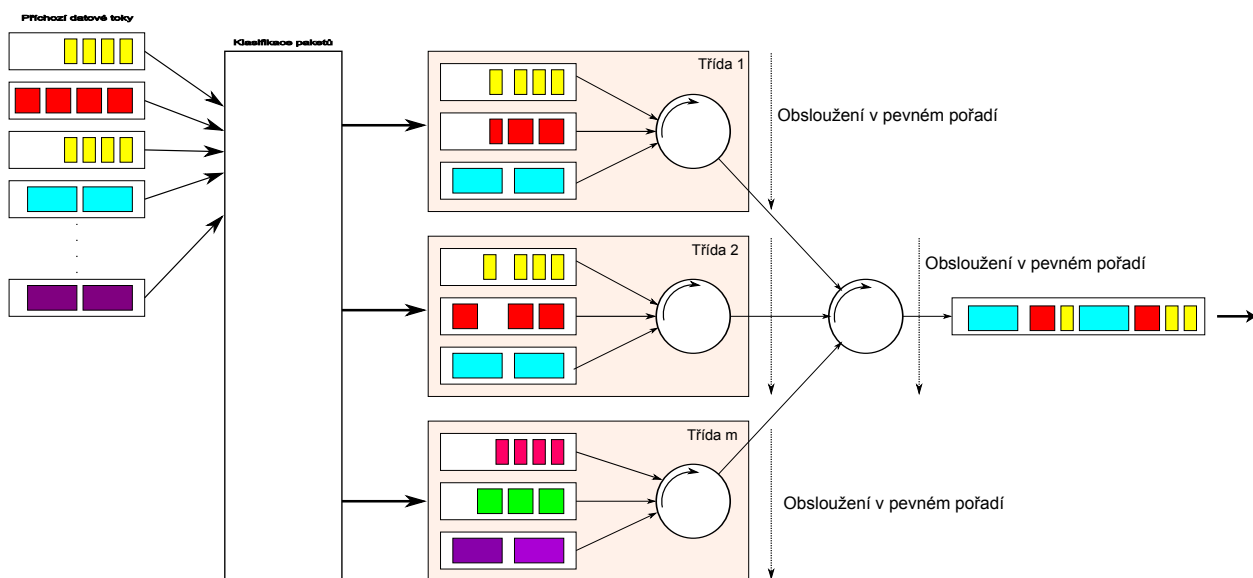
Obr. 3.4: Spravedlivá obsluha front (pakety se rozdílnou délkou), převzato z [3]

3.2.3 Vážená cyklická obsluha - WRR

Vážená cyklická obsluha, neboli řízení front podle třídy (CBQ), funguje na principu dvojúrovňového cyklického obsluhování Round-Robin, což znázorňuje Obr. 3.5. Příchozí pakety jsou klasifikátorem paketů přiděleny do třídy provozu a poslány do fronty v přidělené třídě. Každé třídě lze přiřadit váhový faktor, který garantuje v procentuálním vyjádření výslednou šířku pásma na výstupní lince z aktivního prvku. Váhový faktor udává jak často budou obsluhovány fronty z dané třídy. Součet všech

váhových faktorů musí být roven 100% z celkové šířky pásma. Obsluha je vykonávána dvojúrovňově, v první úrovni jsou cyklicky obsluhovány jednotlivé třídy. V druhé úrovni jsou cyklicky obsluhovány fronty z dané třídy.

Nevýhodou cyklické obsluhy je, stejně jako u FQ to, že při obsluze front je postupně obslužen pouze jeden paket z každé fronty. Pakety však mají rozdílnou délku, díky tomu obsluha každého paketu trvá jinou dobu. Tedy nejdelší paket je obsluhován nejdéle a tím zabírá největší šířku pásma. Daná situace je uvedena na příkladu 3.3 [3]



Obr. 3.5: WRR, převzato z [3]

Třídě provozu číslo jedna je přiřazen váhový faktor 33%, ta odpovídá třídě provozu s nejvyšší prioritou. Třídě provozu číslo dvě je přiřazen váhový faktor 42% a poslední třídě provozu je předěleno zbývajících 25%. Zbývajících devět datových toků je vypočteno v Tab. 3.1

$$\text{Datový tok 1} = \frac{80\text{B}}{80\text{B} + 50\text{B} + 30\text{B}} * w_i = \frac{80\text{B}}{160\text{B}} * 33\% \cong 16,5\% \quad (3.3)$$

3.2.4 Vážená spravedlivá obsluha - WFQ

Příchozí pakety jsou rozděleny do M front klasifikátorem paketů podle své priority. Každé frontě je přidělen váhový faktor, který určuje v procentuálním vyjádření váhu pro danou frontu a ta udává využití šířky pásma pro danou frontu opět v procentech. Součet všech váhových faktorů se musí rovnat 100%. Vážená spravedlivá obsluha je na první pohled velice podobná WRR. Ale, na rozdíl od WRR a FQ, řeší hlavní nevýhodu obou zmíněných systému front a to, že pakety neodesílá jako celek, nýbrž

Tab. 3.1: Vliv velikosti paketů na výslednou šířku pásma v modelu WRR

Datový tok	Třída provozu	Přidělena šířka pásma třídy [%]	Velikost paketů [Byte]	Výsledná přidělená [%] šířka pásma
1	1	33	80	16,5
2	1	33	50	10,31
3	1	33	30	6,19
4	2	42	200	20
5	2	42	100	10
6	2	42	120	12
7	3	25	300	5,36
8	3	25	200	3,57
9	3	25	500	8,93
10	3	25	400	7,14

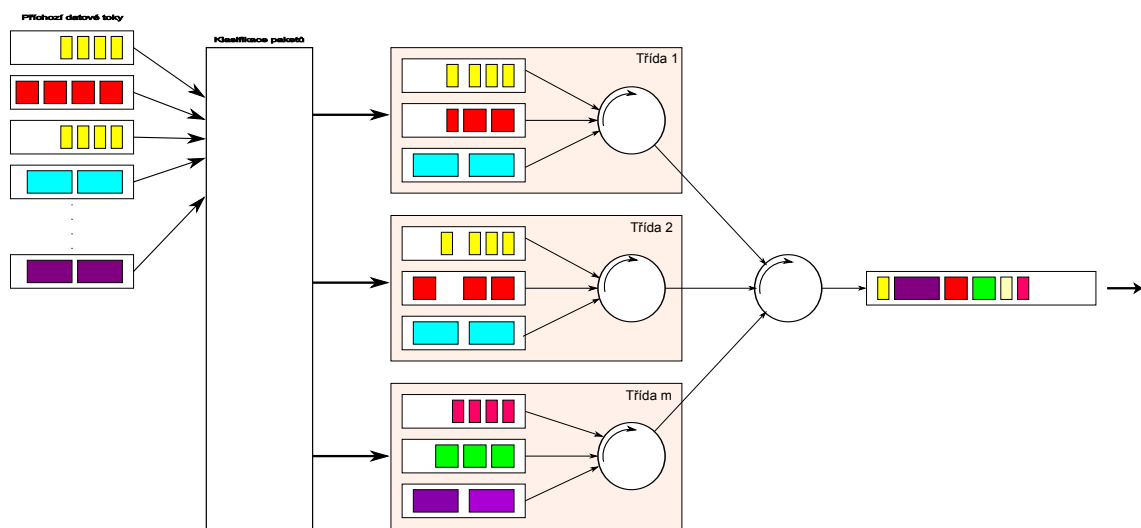
bit po bitu. Tento způsob se nazývá vážená bitová cyklická obsluha. Fronty jsou cyklicky obsluhovány, jenže pakety nejsou odesílány jako celek, ale po bitech do bloku, kde čekají než budou kompletní a odchází do HW fronty. Čas, který stráví paket v tomto bloku, je přímo úměrný jeho velikosti.

3.2.5 Systém front s váženou spravedlivou obsluhou řízenou podle tříd - CBWFQ

Tento systém je rozšířením WFQ o třídy provozu. Příchozí pakety jsou klasifikátorem rozděleny do jedné až m tříd provozu. Třída je tvořena n frontami. Obsluhování v tomto systému je dvojúrovňové. První úrovní jsou obsluhovány třídy cyklicky, v druhé úrovní jsou jednotlivé fronty ve třídě m obsluhované pomocí WFQ. K jednotlivým třídám lze přiřadit váhovou hodnotu, ta udává jakou šířku z celkové šířky pásma linky zabere v procentech. Součet všech váhových hodnot musí být 100%. Rovněž lze váhovat jednotlivé datové toky uvnitř třídy.

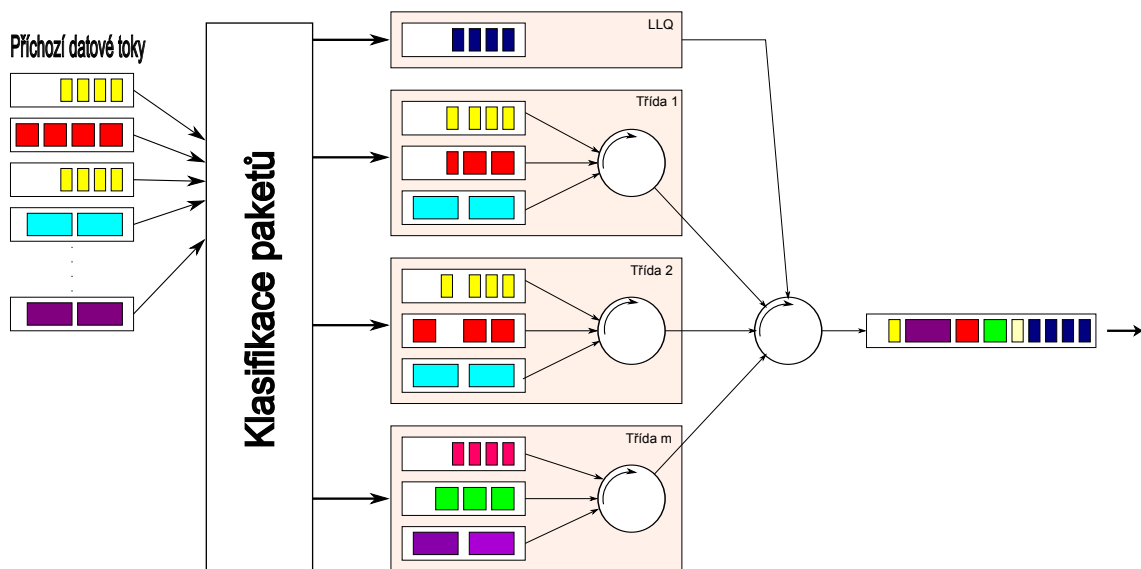
3.2.6 Low latency queueing - LLQ

Systém LLQ je obdobný jako systém front s váženou spravedlivou obsluhou řízenou podle tříd, avšak k tomuto systému je přidána striktní prioritní fronta. Pracuje na totožném principu, jak bylo popsáno v kapitole 3.2.5, jen do té doby, dokud ve frontě LLQ není žádný paket. Pokud do fronty LLQ vstoupí paket, systém CBWFQ

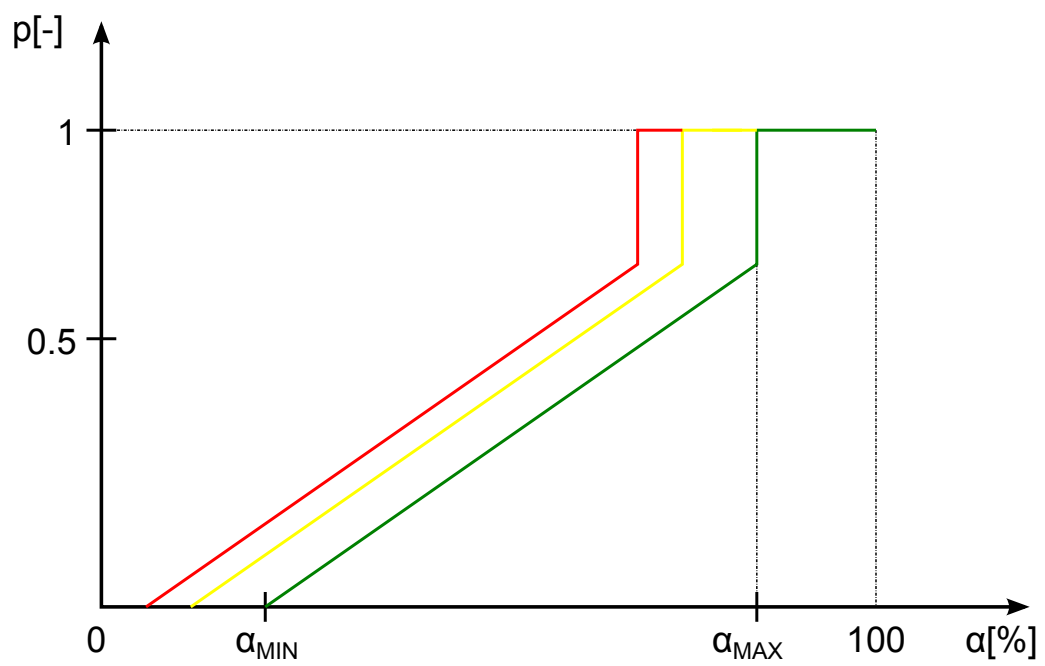


Obr. 3.6: CBWFQ, převzato z [3]

přestane pracovat a jsou obslouženy všechny pakety ve frontě LLQ, poté se vrací systém do režimu CBWFQ.



Obr. 3.7: Low Latency Queue



Obr. 3.8: Profil zahazování paketů WRED převzato z [3]

4 REALIZACE SYSTÉMU HROMADNÉ OBSLUHY V PROGRAMU MATLAB

Tato kapitola se věnuje praktické části bakalářské práce, v níž se ověřují teoretické předpoklady vlastností front na aktivních prvcích, které byly popsány v kapitole 3. K tomu byla využita nástavba Matlabu - Simulink ve verzi 2015a, kde lze pomocí bloků z knihovny Simevents simulovat diskrétní systémy. Pro spuštění nasimulovaných modelů je zapotřebí knihovny Simevents a Stateflow.

V rámci bakalářské práce bylo nasimulováno šest typů hromadné obsluhy front. Jednotlivé modely vytváří spojovací pole, které tvoří devět nezávislých vstupních datových toků, kde každý datový tok má svou vstupní paměť o vyhrazené kapacitě, zde požadavky čekají do té doby, než jsou vyzvány k obsluze. Poté požadavek vstupuje do multiplexeru, který sdružuje devět datových toků do jediného a zde je obslužen kanálem obsluhy. Pro lepší pochopení můžeme kanál obsluhy popsat jako procesor v aktivním prvku. Čas strávený v kanálu obsluhy je přímo úměrný velikosti požadavku, tedy delší požadavky budou obsluhovány déle než požadavky, které jsou kratší, což má zásadní vliv u mnoha způsobů obsluhy, které tento parametr buď ignorují, nebo s tímto faktorem naopak pracují. Multiplexer je řízen algoritmy a to základním modelem FIFO popsáným v kapitole 3.1, striktní prioritní obsluhou front popsanou v kapitole 3.2.1, spravedlivou obsluhou popsanou v kapitole 3.2.2, váženou cyklickou obsluhou popsanou v kapitole 3.2.3, váženou spravedlivou obsluhou popsanou v kapitole 3.2.4 a váženou spravedlivou obsluhou řízenou podle tříd 3.2.5. Do každého modelu je implementována aktivní správa front typu RED, ta chrání vstupní paměti před zahlcením a pokud by hrozilo zaplnění vstupní paměti, příchozí požadavky zahodí a tyto požadavky do paměti nevstoupí.

Pro porovnání rozdílných systémů hromadné obsluhy je důležité, aby měl vstupní proud stejné parametry pro všechny simulované systémy a všechny systémy pracovaly ve stacionárním režimu. K tomu to účelu byl vzat jako referenční model FIFO. V tomto modelu byly definovány vstupní parametry jako intenzita dílčích datových toků, dále byla jednotlivým požadavkům na obsluhu přiřazena délka požadavku a jejich priorita, rovněž byla zvolena celková kapacita vstupní paměti. Pro modely kde jednotlivé datové toky využívají vlastní vstupní paměti, byla tato celková kapacita spravedlivě rozdělena, v úvahu se musel brát fakt, že program Matlab v prostředí Simulink nebere v potaz parametr velikosti požadavku a každý požadavek zabírá v paměti jednu buňku. Generátor paketů je zobrazen na obr. ???. Všechny systémy hromadné obsluhy využívaly stejnou šířku pásma, která byla stanovena na 1Mbit/s.

Sledované parametry obou zmíněných systémů jsou ztrátovost paketů a doba jakou požadavek strávil ve frontě, než byl obslužen. Zpoždění vlivem obsluhy a

velikostí vstupní vyrovnávací paměti je popsán ve kapitole 3.1. Ztrátovost požadavků (paketů) je vyjádřena v procentech z celkového počtu vygenerovaných paketů.

4.1 Výpočet stacionárního režimu

Šířka pásma je pro oba systémy 1Mbit/s, jelikož velikost požadavků je vyjádřena v Byte, musí se šířka pásma převést na veličinu B/s, což je v tomto případě 125kB/s a rychlost R_o vyjadřuje, jakou dobu bude trvat obsloužit paket závislosti na jeho velikosti dle vzorce 4.1

$$R_o = \frac{\text{délka paketů}}{\text{rychlost linky}} \quad (4.1)$$

Každá třída provozu má vlastní generátor paketů zobrazený na obr. 4.2, kde jako atribut je každému paketu přidělována jeho délka. K tomu je využit blok Event-Based random Number, který generuje náhodnou hodnotu délky paketů s exponenciálním rozložením, což nejlépe odpovídá reálné síti. Hlavním parametrem generátoru je střední hodnota Mean, která udává střední hodnotu pravděpodobnosti délky paketů. Pro nejvyšší prioritu je nastaven na 80, což můžeme považovat za 80 Byte. Obdobně jsou nastaveny zbývající osm generátorů pro ostatní priority. Tyto hodnoty byly zvoleny tak, aby odrážely skutečnost podle [8].

Příklad výpočtu

Pro názornost, je matematicky vyjádřen postup dosažení ideální hodnoty intenzity vstupního proudu pro danou prioritní linku tak, aby byl součet koeficientů využití obslužného kanálů $\rho < 0,8$.

Při znalosti velikosti paketů v jednotlivých prioritách lze vypočítat jakou dobu bude přibližně trvat obsluha jednotlivých paketů pomocí vzorce (4.1), kde obsluha paketu s nejvyšší prioritou trvá $6,4 \cdot 10^{-4} s$. Z čehož lze snadno vypočítat kolik paketů zvládne kanál obsluhy obsloužit za minutu, neboli intenzitu obsluhy podle vzorce (4.3). Pro linku s nejvyšší prioritou byl zvolen dílčí koeficient využití obslužných kanálů $\rho=0,3$, pro normální prioritu $\rho=0,2$ a pro nízkou prioritu $\rho=0,3$. Koeficienty jsou zvoleny tak, aby jejich součet dával $\rho=0,8$, což je hraniční hodnota pro zachování stacionárního režimu obsluhy front. Tato hodnota byla schválně zvolena proto, že mohou vznikat situace, kdy systém pracuje na hraně stacionárního režimu a příchozí proud může být větší, než zvládne kanál obsluhy obsloužit. A v této situaci se nejvíce projeví různý způsob obsluhování paketů a lze sledovat zejména ztrátovost paketů a čas strávený ve frontě před kanálem obsluhy.

Dále je vypočtena intenzita obsluhy podle vzorce (4.3), která vyjadřuje kolik paketů s velikostí 80B zvládne kanál obsluhy obsloužit. A v posledním kroku (4.4) je nastavení vstupního parametru, tedy s jakou četností za sekundu budou přicházet pakety do SHO.

$$R_o = \frac{80}{125000} = 6,4 \cdot 10^{-4} s \quad (4.2)$$

$$\mu = \frac{60}{R_o} = \frac{60}{6,4 \cdot 10^{-4}} = 93750 \quad (4.3)$$

$$\lambda = \frac{60}{\rho \cdot \mu} = \frac{60}{0,3 \cdot 93750} = 2,13 \cdot 10^{-3} s \quad (4.4)$$

Obdobně jsou vypočteny parametry pro zbývající prioritní linky provozu.

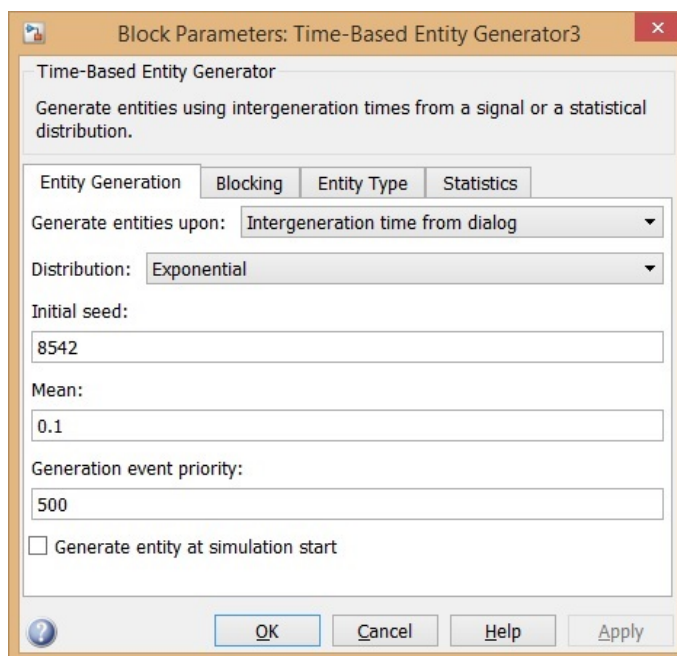
4.2 Realizace společných jednotlivých bloků systému hromadné obsluhy

V této kapitole jsou popsány jednotlivé bloky obecného modelu SHO, které jsou využity pro všechny typy obsluhy. Podkapitoly jsou řazeny postupně, jak požadavky vznikají až po jejich obsloužení, nebo zahození. První blok je Generátor požadavků v Matlabu označovány jako entita. Entita je generována za určitých podmínek s určitými parametry, podrobně je modul popsán v podkapitole 4.2.1. V druhé podkapitole je popsána realizace aktivní správy front 4.2.2, která pracuje paralelně vůči bloku vstupní vyrovnávací paměti. Předposlední blok je multiplexer, který sdružuje devět datových toků, podle algoritmů popsanych v kapitole 3. Za multiplexerem jsou entity obsluhovány kanálem obsluhy s rychlostí linky. [6] [7]

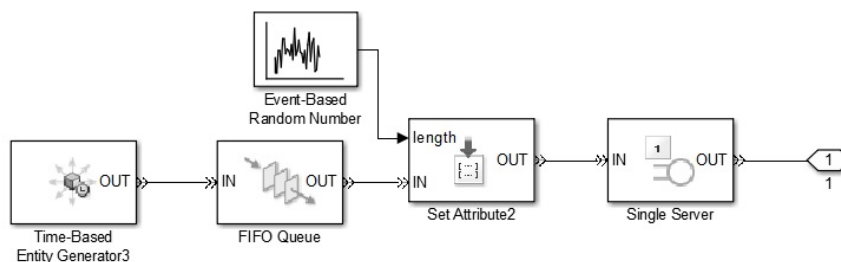
4.2.1 Generátor entit (paketů)

Vstupující požadavky do SHO jsou generovány nezávisle pro každý datový tok, neboť každý datový tok má vlastní generátor dat. Generátor entit (paketů) se skládá z pěti bloků vyobrazených na Obr. 4.2, kde jsou generovány entity, kterým jsou postupně přidávány atributy. Hlavní základní blok generátoru je generátor náhodných entit řízený podle času v modelu uveden pod názvem „Time-Based Entity Generator“, ten na základě vstupního parametru „Mean“ vytváří množství entit za jednotku času s rozptylem značeným. „Initial seed“. Předposlední parametr je v jakém rozložení budou požadavky generovány, zde pro všechny SHO bylo zvoleno exponenciální rozložení, které nejlépe kopíruje reálný provoz na síti. Detail generátoru

je vyobrazen na Obr. 4.1. V záložce „Entity Type“ je zvolen typ entit na standard, ten přiděluje každé vygenerované entitě atribut „priority“, což je pevně stanovené číslo priority pro každý datový tok, je to obdoba například značkování paketů na hraničním směrovači. [7]



Obr. 4.1: Generátor entit (paketů) - nastavení vstupních parametrů



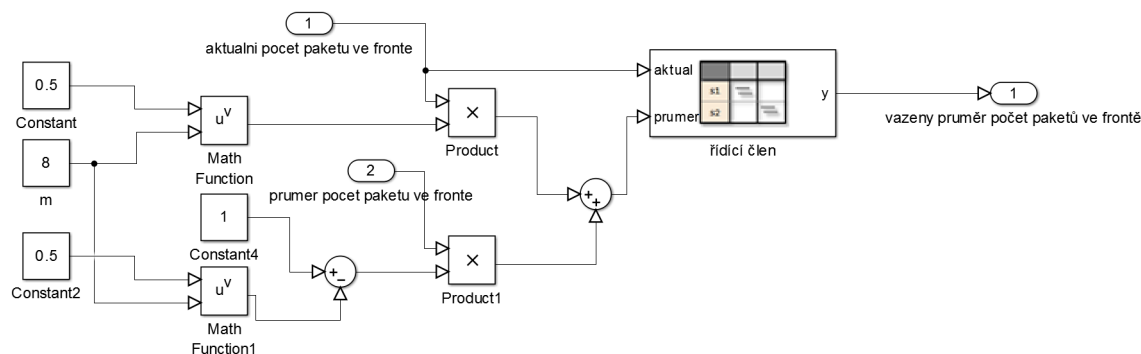
Obr. 4.2: Generátor entit pro jeden datový tok (paketů) - celkový pohled

4.2.2 Aktivní správa front RED

V rámci aktivní správy front byl do každého modelu SHO implementován systém předčasné detekce zahlcení „RED“. Celý systém se skládá ze dvou modulů pro každý dílčí datový tok. První modul sleduje dva parametry vytížení kapacity vstupní paměti a to parametr $\#n$, ten vyjadřuje aktuální počet požadavků ve vstupní paměti a

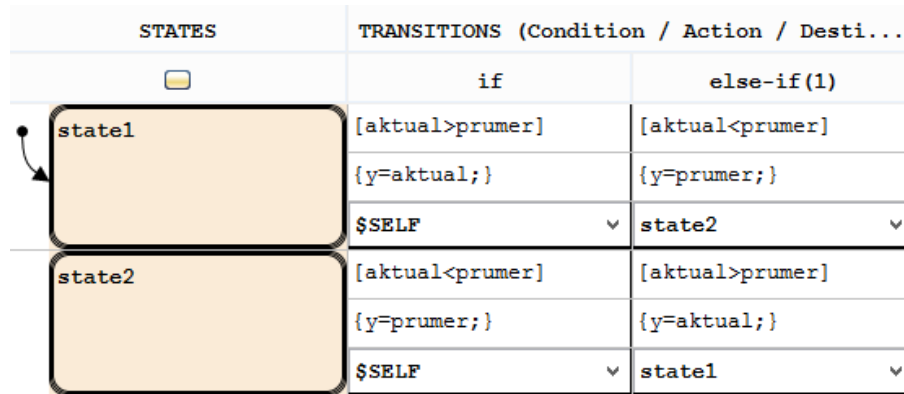
parametr $\#len$, ten udává průměrný počet požadavků v paměti. Pomocí vzorce 4.5 získáme jednotkové číslo, které zahrnuje oba parametry a výsledné číslo vyjadřuje vážený průměr zaplnění vstupní paměti, výsledné číslo je velice podobné parametru $\#len$ jen s tím rozdílem, že pomocí konstanty m , lze měnit profil výpočtu váženého průměru. To znamená, že čím je konstanta m nižší, tím výsledný průměr rychleji reaguje na případné změny zaplnění paměti a předchází zaplnění fronty. Ovšem mohou nastat situace, kdy je fronta během krátkého okamžiku zcela zaplněna a podle výše zmíněného přepočtu by výsledný vážený průměr nestačil zareagovat na tuto událost a proto je vložen do modelu řídicí člen, který porovnává aktuální zaplnění fronty s výše uvedeným váženým průměrem. Pokud nastane situace kdy aktuální zaplnění fronty je větší než vážený průměr zaplnění tak řídicí mechanismus přepne na výstup hodnotu s aktuálním zaplněním. Tím se rapidně zvedne pravděpodobnost zahození přicházejících požadavků do fronty. Pokud aktuální počet požadavků ve frontě klesne pod vážený průměr požadavků ve frontě, řídicí člen přepne na výstup vážený průměr. Pro přehlednost je na Obr. 4.4 zobrazen řídicí člen, který je z knihovny Stateflow a funguje na principu podmínek „if“ a „else-if“. Vzorec 4.5 je zobrazen na Obr. 4.3 pomocí bloků v programu Simulink.

$$\text{vážený průměr} = \#n \cdot \left(\frac{1}{2}\right)^m + \#len \cdot \left(1 - \frac{1}{2}^m\right) \quad (4.5)$$



Obr. 4.3: Výpočet váženého průměru

Do druhého bloku aktivní správy na Obr. 4.6 „RED“ vstupuje vážený průměr z bloku prvního, který je podělen maximální kapacitou fronty, toto číslo je vynásobeno stem a výsledné číslo udává z kolika procent je zaplněna fronta. Tento parametr vstupuje do řídicího členu jako proměnná „a“ před vstupem je signál rozdvojen, poté je od procentuálního zaplnění fronty odečten spodní práh, který byl zvolen 30% a následně je číslo vyděleno stem. Toto číslo je vynásobené $\tan \alpha$, kde úhel α je zvolená strmost profilu zahazování, v tomto případě byl zvolen uhel 35° . Násobení touto konstantou při znalosti strany c v pravoúhlém trojúhelníku obr. 4.5 získáme



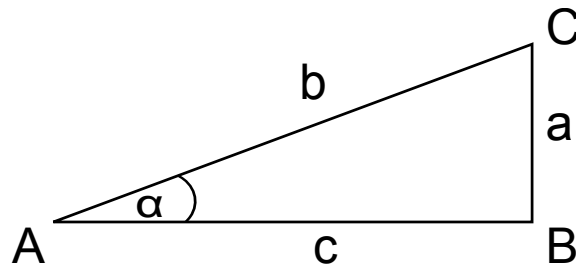
Obr. 4.4: Řídící člen RED

stranu a , jež výsledné číslo odpovídá míře pravděpodobnosti zahození při zvoleném profilu zahazování příchozích paketů. Výsledné číslo vstupuje do řídicího členu jako proměnná „*pocet*“. Další parametry vstupující do řídicího členu je spodní a horní práh zaplnění fronty v procentech značeny jako „*min*“ a „*max*“ v tomto případě je to 30% a 90%.

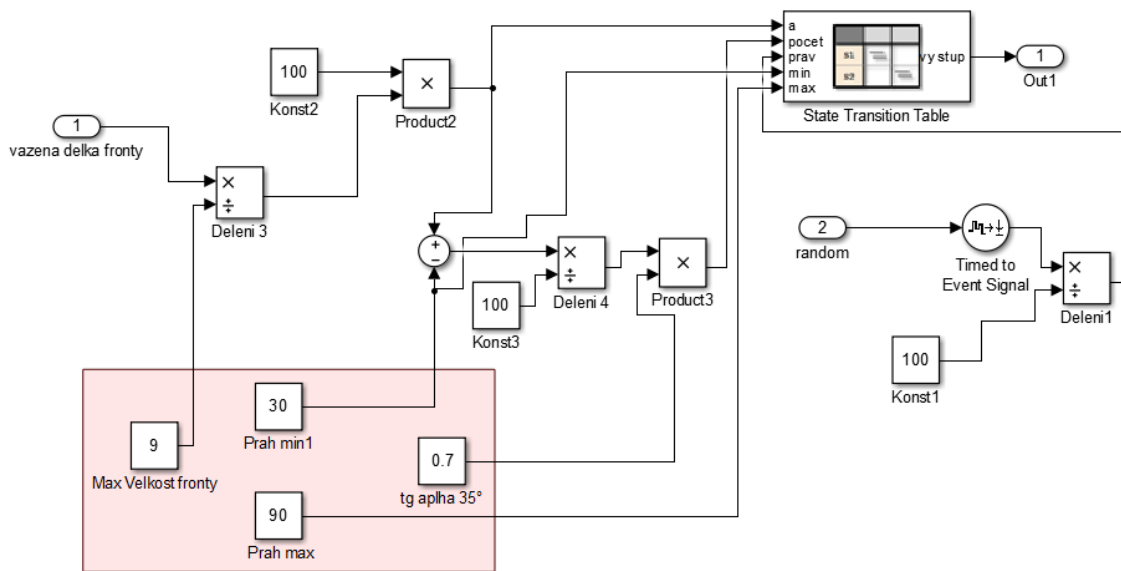
Poslední proměnná která vstupuje do tohoto bloku je náhodné číslo z generátoru od 0 do 100, které je použito pro náhodné zahazování požadavků před vstupem do fronty, ovšem toto číslo je poděleno stem tak, aby výsledné číslo odpovídalo pravděpodobnosti od 0 do 1. Výsledný parametr vstupuje do řídicího členu jako proměnná „*prav*“.

Algoritmus v řídicím členu je detailně vyobrazen na Obr. 4.7 pracuje s výše uvedenými proměnnými, kde porovnává proměnnou „ a “ se spodním a horním prahem zaplnění vstupní paměti. Tímto mohou nastat v řídicím členu tři stavy. První stav: proměnná „ a “ je menší než spodní práh „*min*“. Na výstupu řídicího členu bude jednička, která udává to, že příchozí pakety do vstupní vyrovnávací paměti nebudou zahazovány. Druhý stav: proměnná „ a “ je větší, než horní stanovený práh „*max*“. Na výstupu řídicího členu bude dvojka, která udává, že všechny příchozí pakety do vstupní vyrovnávací paměti budou zahozeny. Třetí stav nastane pokud zaplnění paměti je v rozmezí od 30 do 90 procent, zde dochází k výpočtu pravděpodobnosti zahození na základě porovnání těchto dvou proměnných - pravděpodobností zahození, v modelu označovanou jako „*pocet*“ s náhodným generovaným číslem označovaným jako „*prav*“. Zde mohou nastat dvě situace. V prvním případě je náhodné číslo menší než pravděpodobnost zahození, tím je splněna podmínka pravděpodobnosti a příchozí paket bude zahozen. V druhém případě je náhodné číslo větší než pravděpodobnost zahození, tím pravděpodobnost zahození není splněna a příchozí paket nebude zahozen a vstoupí do paměti. Tímto algoritmem je zabezpečeno náhodné zahazování vstupních požadavků do fronty tak, aby bylo předcházeno úplnému za-

hlčení. [6] [7]



Obr. 4.5: Pravoúhlý trojúhelník



Obr. 4.6: Profil zahazování RED

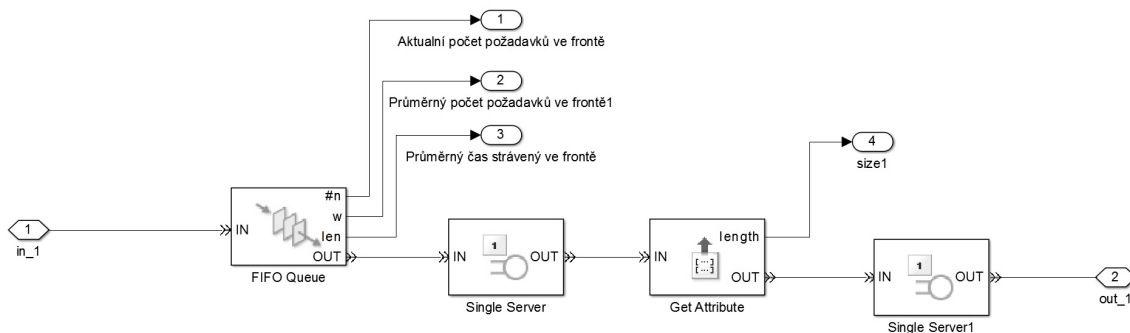
4.2.3 Vstupní vyrovnávací paměť

Pokud aktivní správa front RED propustila vstupní požadavek do SHO a kanál obsluhy je využíván jinými požadavky, požadavek se umístí do vstupní vyrovnávací paměti. Mimo systém obsluhy FIFO je pro každý datový tok alokována vlastní vyrovnávací paměť o dané kapacitě. Paměť je většinou tvořena typem FIFO detailně popsána v kapitole 3.1. Zde požadavek čeká ve frontě na řadu do doby než bude vyzván multiplexerem a následně obslužen. V bloku FIFO jsou sledované parametry „#n“ - aktuální počet požadavků ve frontě, „len“ - průměrný počet požadavků ve frontě, „w“ - průměrný čas strávený ve frontě. Za frontou je kanál obsluhy s nastavenou dobou obsluhy na nula, na požadavek nemá žádný vliv, funguje jen na principu

Řídící člen					
Block: cbWFQ_red2/Red1/Řídící člen					
1	state1	[a<min]	[a>max]	[a>min && a<max && prav > pocet]	[a>min && a<max && prav < pocet]
		{vystup = 1;}			
		\$SELF	state2	state3	state4
2	state2	[a>max]	[a<min]	[a>min && a<max && prav > pocet]	[a>min && a<max && prav < pocet]
		{vystup = 2;}			
		\$SELF	state1	state3	state4
3	state3	[a>min && a<max && prav > pocet]	[a>min && a<max && prav < pocet]	[a>max]	[a<min]
		{vystup = 1;}			
		\$SELF	state4	state2	state1
4	state4	[a>min && a<max && prav < pocet]	[a>min && a<max && prav > pocet]	[a>max]	[a<min]
		{vystup = 2;}			
		\$SELF	state3	state2	state1

Obr. 4.7: Vnitřní struktura řídicího členu profilu zahazování RED

paměti o kapacitě jednoho požadavku, ale zároveň pracuje jako brána, požadavek v paměti je blokován do té doby než se vstupní port na multiplexeru otevře a následně kanál obsluhy požadavek uvolní a projde přes multiplex. Před „*Single Server*“ je blok „*Get Attribute*“, který pracuje s kanálem obsluhy a sleduje atribut délky požadavku umístěný v tomto bloku. S tímto parametrem pracují algoritmy zohledňující délku paketu např. WFQ a CBWFQ. [7]

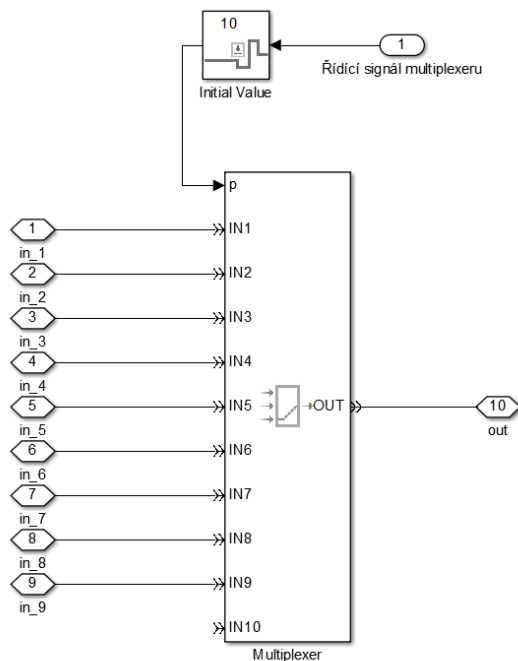


Obr. 4.8: Vstupní vyrovnávací paměť jednoho datového toku.

4.2.4 Sdružování datových toků do jediného

Zde dochází k slučování devíti datových toků do jediného pomocí bloku „Input Switch“ vyobrazen na Obr. 4.9, který je řízen portem p . Port p je řízen pomocí čísel od 1 do 10 typu itenger. Desátý vstupní port je využíván k blokaci zbývajících devíti vstupních portů, protože na řídicím portu musí být celou dobu přiveden vstupní signál. Při výpočtech je potřeba blokovat všechny vstupní porty a kdyby se na řídicím signálu měl udržovat signál od 1 do 9, tak multiplexer by měl vždy otevřený

jeden vstupní port, což je velice nežádoucí a tento efekt by zkreslovat výsledné data. A proto je zde desátý port na který není přiveden žádný datový tok a je užíván pro blokaci vstupních portů při analýze datových toků. V řídicím signálu je ještě vklíněn blok „Initial Value“, který slouží pro počáteční hodnotu v čase nula v řídicím signále, než bude přiveden řídicí signál. [7]

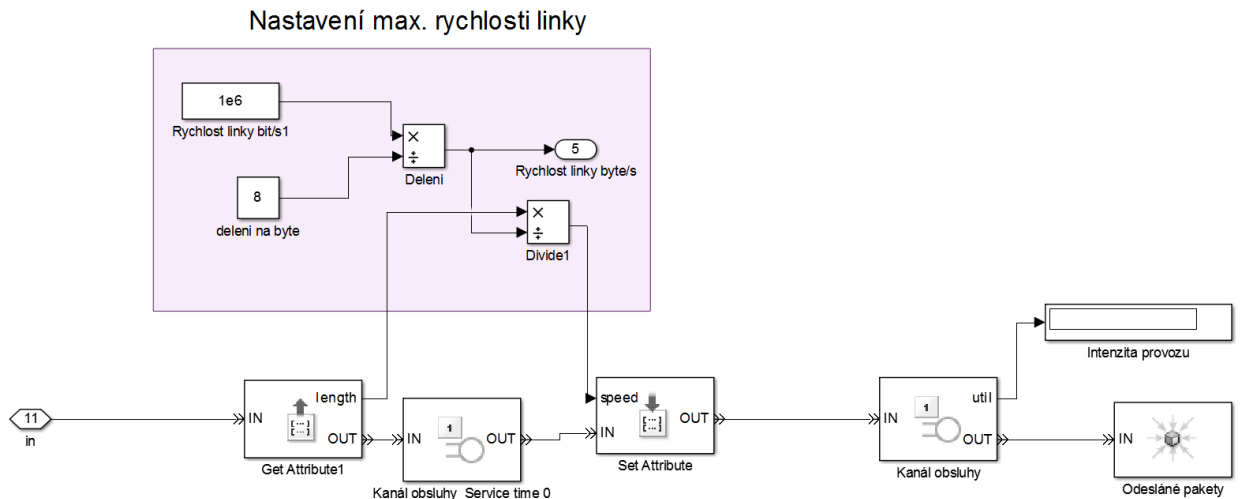


Obr. 4.9: Sdružování datových toků do jediného - multiplexer

4.2.5 Kanál obsluhy a rychlost linky

Nastavení rychlosti linky lze realizovat změnou bloku „Rychlost linky bit/s“, kde rámci testů byla zvolena rychlost 1Mbit/s . Tato hodnota je podělena osmi z důvodu, že velikost paketů je vyjádřena byte. Výsledná hodnota je odesílaná do řídicího algoritmu pro výpočet váhování tříd, nebo front.

Vstupní signál přichází z výstupu multiplexeru a je zjišťována délka každého požadavku, ta je podělena rychlostí linky v byte/s a výsledné číslo udává jak dlouho bude konkrétní požadavek obsluhován. Tato hodnota je každému požadavku zapsána jako atribut pomocí bloku „Set Atribut“ jako atribut „speed“. Poté signál vstupuje do hlavního kanálu obsluhy, kde servisní čas obslužení se určuje podle zmíněného atributu speed. V kanálu je sledován parametr *utilization*, který udává jak vytížený kanál obsluhy je. Vytížení může nabývat hodnot od 0 do 1, kde do 0,8 můžeme systém považovat ve stacionárním režimu. Do posledního bloku „Odeslané pakety“ vstupují obslužené požadavky a zde zaniknou. [7]



Obr. 4.10: Kanál obsluhy a nastavení rychlosti linky

4.2.6 Váhování tříd v systému CBWFQ

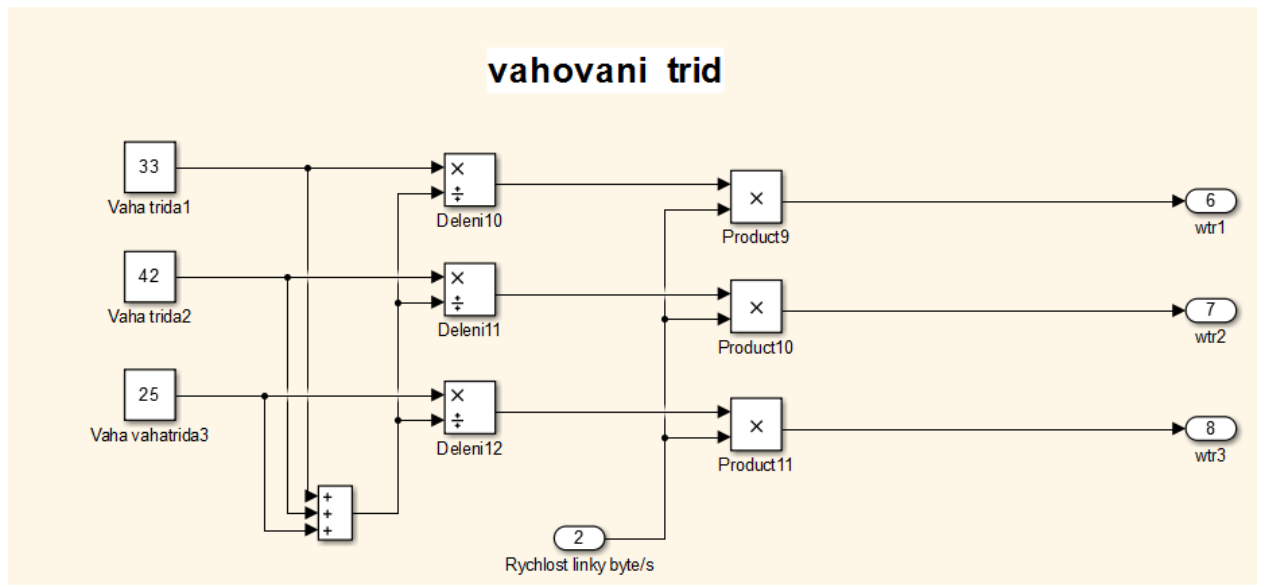
Pro zajištění přidělení části šířky pásma je nutno realizovat váhování datových tříd, popřípadě datových toků uvnitř třídy. Podle vzorce (4.6) lze vypočítat váhu pro danou třídu. Ve zlomku v čitateli je požadovaná váha třídy, v jmenovateli se suma všech vah tříd a celý zlomek je násoben celkovou rychlostí linky, tím je docíleno, že součet přidělených vah nebude větší než celková rychlost linky. Na Obr. 4.11 je realizace vzorce 4.6 pomocí bloků z programu Simulink.

Obdobně je provedeno váhování datových tříd uvnitř třídy, jen s tím rozdílem, že místo celkové rychlosti linky je zlomek ve vzorci 4.6 násoben přidělené části šířky pásma např. w_{tr1} [7]

$$w_{tr1} = \frac{Váha\ třída1}{\sum Váha\ třída} \cdot Rychlost\ linky \quad (4.6)$$

4.2.7 Řídící algoritmus CBWFQ - vnější pohled

K vytvoření řídicího algoritmu CBWFQ zobrazen na Obr. 4.12 a popsán v kapitole 3.2.5 byl využit blok „Chart“ z knihovny Stateflow [6], kde lze velmi intuitivně pomocí bloků vytvořit komplexní algoritmus, konkrétně zde CBWFQ. Algoritmus pracuje s těmito vstupními proměnnými $\#n$ - (počet požadavků ve frontě), wtr - (přidělena váha dané třídy), $size$ - velikost požadavku čekajícího ve frontě před multiplexerem, w - přidělená váha datovému toku ve třídě, $length$ - velikost výstupního požadavku z multiplexeru. Jediný výstupní signál je y , který je přiváděn jako řídicí signál na multiplexer popsáný v kapitole 4.2.4



Obr. 4.11: Realizace váhování tříd podle vzorce 4.6

4.2.8 Řídící algoritmus CBWFQ - vnitřní pohled

Do vnitřní struktury bloku Char vstupují proměnné popsané v kapitole 4.2.7. Vnitřní struktura bloků se skládá z „state“ bloků, tyto bloky fungují na principu základového modulu kterému lze přiřadit n číslovaných sekvencích podmínek a příkazu ve schématu vyjádřenými šipkami, kde na začátku šipky je přiřazeno číslo které udává pořadí vykonání příkazů na základě splnění podmínky uvedené v hranatých závorkách. Pokud je podmínka v hranatých závorkách splněna, vykoná se příkaz uveden ve složených závorkách. Pokud podmínka splněna nebyla je testována další podmínka s číslem o jednu vyšší. Tyto podmínky mohou být vykonávané do té doby, dokud podmínka vyhovuje, nebo mohou být vykonány pouze jednou, a nebo mohou po vykonání ukazovat na jiný blok State. Pro lepší názornost na konci odstavce je uveden konkrétní příklad podmínky a příkazu. Podmínka říká, pokud x_1 je menší než x_2 a zároveň i_1 je menší než w_1 , tak x_1 inkrementován o jedničku a y se bude rovnat dvěma. [6]

```
[x1<x2 && i1<w1]
{x1=x1+1, y=2;}
```

V následující větách je demonstrován způsob obsluhy CBWFQ pomocí části algoritmu, který je vyobrazen na Obr. 4.14. Tato část obsluhuje jen jeden datový tok uvnitř třídy a je zcela dostačující zde popisovat pouze tuto část, protože zbývajících osm řídicích datových toků jsou totožné, jen pracují s rozdílnými hodnotami proměnných.

Řídící blok se skládá ze dvou bloků State, kde blok f1 můžeme nazvat řídicí a blok bit1 slouží k vyčítání po bitech. Do bloku f1 vstupují dva spouštěcí signály první je veden z počátečního bloku ten je použit pouze na začátku celé simulace, kde definuje počáteční hodnoty na nulu pro proměnné $i1$ až $i9$, $b1$ až $b9$ a $k1$ až $k3$, které jsou využity pro inkrementaci. Druhý příchozí signál přichází konkrétně pro tento případ z třetí třídy a to v případě kdy proměnná $k3$ je větší než přiřazená váha pro celou třídu $wtr1$. Proměnná $k3$ je inkrementována o velikost požadavků $length$ z celé třídy, které byly obslouženy multiplexerem. Před blokem f1 druhého vstupního signálu je vložen rozvojovací člen, ten bude popsán později kvůli návaznosti.

Příchozím signálem je blok „State“ aktivován a v pevném pořadí začne testovat podmínky od 1 do n . První podmínka se ptá, zda proměnná $size1$ je větší než $w1$ a zároveň $k1$ je menší než $wtr1$ a zároveň $x1$ je větší než nula. Pokud podmínka je splněna, řídicí signál vstupuje do bloku bit1, proměnné $b1$ je přiřazena hodnota $size1$ a na výstupní proměnné y je hodnota deset. Obdobně jsou cyklicky testovány ostatní podmínky v bloku f1. Detailní vysvětlivky proměnných jsou uvedeny za tímto odstavcem.

Seznam proměnných, které jsou použity v modelu CBWFQ

- $size$ - velikost požadavku na vstupu multiplexeru.
- $length$ - velikost požadavků na výstupu multiplexeru.
- w - přidělena váha datovému toku v třídě v bytech
- wtr - přidělena váha všem datovým tokům ve dané třídě front vyjádřena bytech
- i - lokální proměnná sloužící pro inkrementaci o velikost požadavků odeslaných z daného datového toku
- k - lokální proměnná sloužící pro inkrementaci o velikost požadavků odeslaných z celé třídy front
- x - udává počet požadavků ve vstupní vyrovnávací paměti pro každý datový tok
- b - lokální proměnná sloužící pro dekrementaci pokud požadavek je větší než přiřazená váha w . Proměnné b je přiřazena velikost čekajícího požadavku $size$ a je dekrementována o velikost přidělené váhy datového toku w do té body než proměnná b je nulová, poté je až požadavek obsloužen multiplexer
- y - výstupní řídicí proměnná multiplexeru

Typy podmínek, které jsou použity v modelu CBWFQ

- < - menší než
- > - větší než
- <= - menší, nebo rovno
- >= - větší, nebo rovno

&& - a zároveň
|| - a nebo
== - rovná se

Se znalostí všech proměnných je za tímto odstavcem vypsána sekvence podmínek v bloku f1. Číslování odpovídá posloupnosti testování podle Obr. 4.14

1. `[size1>w1 && k1<wtr1 && x1>0]`
`{b1=size1, y=10}`
2. `[i1<w1 && x1 > 0 && k1<wtr1]`
`{y = 1, i1=i1+length, k1=k1+length;}`
3. `[k1>=wtr1 || (x1==0 && x2==0 && x3==0)]`
`{y=10, k1=0, i1=0;}`
4. `[b2>0 && k1<wtr1]`
`{y=10, i1=0;}`
5. `[(i1>=w1 || x1 ==0) && k1<wtr1]`
`{y =10, i1 =0;}`

4.3 Vyhodnocení systému

V rámci porovnání všech druhů obsluhy byly vytvořeny tři scénáře, které definují tři různé nastavení intenzity dílčích datových toků. Charakter přidělování vstupní intenzity je rozdělení vstupních požadavků tak, aby se výsledné zatížení obslužného kanálu rovnalo $\rho = 0,95$. Tato hodnota vyšla jako nejvhodnější. Jelikož hodnota vzešla z mnoha testovacích scénářů v rámci přípravy bakalářské práce, kde scénáře při nižším vytížení kanálu vykazovali velmi malou ztrátovost vstupujících požadavků, zejména modely CBFWQ a WFQ, kde ztrátovost byla nulová i při rapidně snížení kapacity front. Naopak při vytížení kanálu na 95% je ztrátovost velmi vysoká, zejména v modelech FQ, SPQ a WRR. Ztrátovost dosahovala i 90% avšak jako pozitivum lze velmi lehce poukázat na nedostatky různých systému obsluhy. V rámci hodnocení nasimulovaných výsledků se musí brát potaz, že model nerozlišuje příchozí požadavky z hlediska spojově orientovaných spojení (TCP) a nespojovaných orientovaných spojení (UDP) a všechny příchozí požadavky nemají žádné zpětné mechanismy.

Rozložení vstupní intenzity podle scénáře 1 je: druhému datovému toku v první třídě, (což jsou toky jedna až tři) je přiřazena hodnota vstupní intenzity taková, aby

vytížila kanál obsluhy na 30% při obslužení všech příchozích požadavků. Ve druhé třídě, (což jsou datové toky čtyři až šest) je přidělena hodnota vstupní intenzity taková, aby vytížila kanál obsluhy na 10% při obslužení všech příchozích požadavků. Stejně je nastaven osmý datový tok ve třetí třídě. Zbylé procenta jsou rovnoměrně rozloženy mezi zbývající šest datových toků. Přehledně získání vstupní hodnoty „Mean“ je uvedeno v tabulce Tab. 4.1. Obdobně je nastaveny zbývající dva scénáře, jen s tím rozdílem, že vytížení jednoho datového toku na 30% je nastaveno ve druhé třídě pro scénář 2 a ve třetí třídě pro scénář 3. Tabulky s výslednými vstupními hodnoty „Mean“ jsou v Tab. 4.2 a Tab. 4.3

Provnání dvou modelů: FQ a CBFWQ V rámci přehlednosti jsou zde popsány rozdíly mezi spravedlivou obsluhou a systémem front s váženou spravedlivou obsluhou řízenou podle tříd pro první scénář.

Z grafu A.7 systému spravedlivou obsluhou je zcela patrné, že vstupní intenzita převyšuje kapacitu jakou kanál obsluhy může zvládnout. Nejvíce se projevuje ztrátovost ve druhém datovém toku, to je způsobené hlavně vstupní intenzitou, která by měla zatěžovat kanál obsluhy na 30% v případě obslužení všech vstupních požadavků. Druhý hlavní důvod tak vysoké ztrátovosti vyjádřeno procentech je velikost požadavků v daném datovém toku. V tomto případě je průměrná velikost v druhém datovém toku 50B. Tato skutečnost zcela výstižně demonstruje nevýhodu typu obsluh (SPQ,FQ,WRR) které nepracují s délkou požadavku. Za povšimnutí stojí třetí datový tok (Priorita3) i když nastavení vstupní intenzity má přidělovat výstupní šířku pásma cca. 6,6% tak ztrátovost se pohybuje kolem 50%. To je způsobeno rovněž velikostí požadavků a to z důvodu, že jedná o 30B požadavek. Ze vzorce 3.2 plyne, že reálná výstupní šířka pásma třetího datového toku (Priorita3) jen z 1,5%. Obdobně je na tom se ztrátovostí pátý datový tok. Zbylé datové toky se pohybují pod 10% ztrátovosti vstupních požadavků z důvodu, že se jedná o mnohonásobně větší požadavky v řádu stovek bytů dle Tab. 4.1.

Na Obr. A.8 je znázorněno zpoždění požadavků v sekundách ve vstupní vyrovnávací frontě. Tato závislost udává časový údaj, jak dlouho se průměrně zdržely požadavky ve vstupní vyrovnávací paměti, než opustili frontu. Vzniklé zpoždění je zapříčiněno třemi důvody. První důvod je vytížení kanálu obsluhy nad míru, kdy kanál nestíhá požadavky obsluhovat a fronta se začne plnit. Druhý důvod je kapacita fronty. V kapitole 3.1 je popsán vliv zpoždění na velikost kapacity fronty. Třetí důvod je velikost požadavků. Požadavek, který má velikost 50B má kapacitu na dvacet požadavků ve frontě o velikosti 1000B. Z toho plyne, že požadavky ve frontě čekají déle, než opustí frontu. Tento jev se nejvíce projeví v systému obsluhy FQ, kde v jediném cyklu je obslužen jeden požadavek z každé fronty.

V grafu je vidět, že nejdéle čekal třetí datový tok a to přibližně půl sekundy.

Příčina takového zpoždění je zapříčiněná všemi třemi důvody, které byly popsány v minulém odstavci. Ostatní datové toky oproti třetímu toku zpozdily minimálně.

Systém CBWFQ obr. A.31 a obr. A.32 si vedl o několik tříd lépe při stejném nastavení vstupních intenzit. Ztrátovost ve všech datových tocích byla pod 6%. To bylo docílené hlavně pomocí váhování tříd a zejména algoritmus zohledňuje velikost požadavků.

Obdobně lze porovnat výsledky simulace zbývajících systémů ve třech scénářích. Všechny výsledky jsou uvedeny v příloze A.1

Tab. 4.1: Hodnoty vstupní intenzity pro nastavení scénáře 1

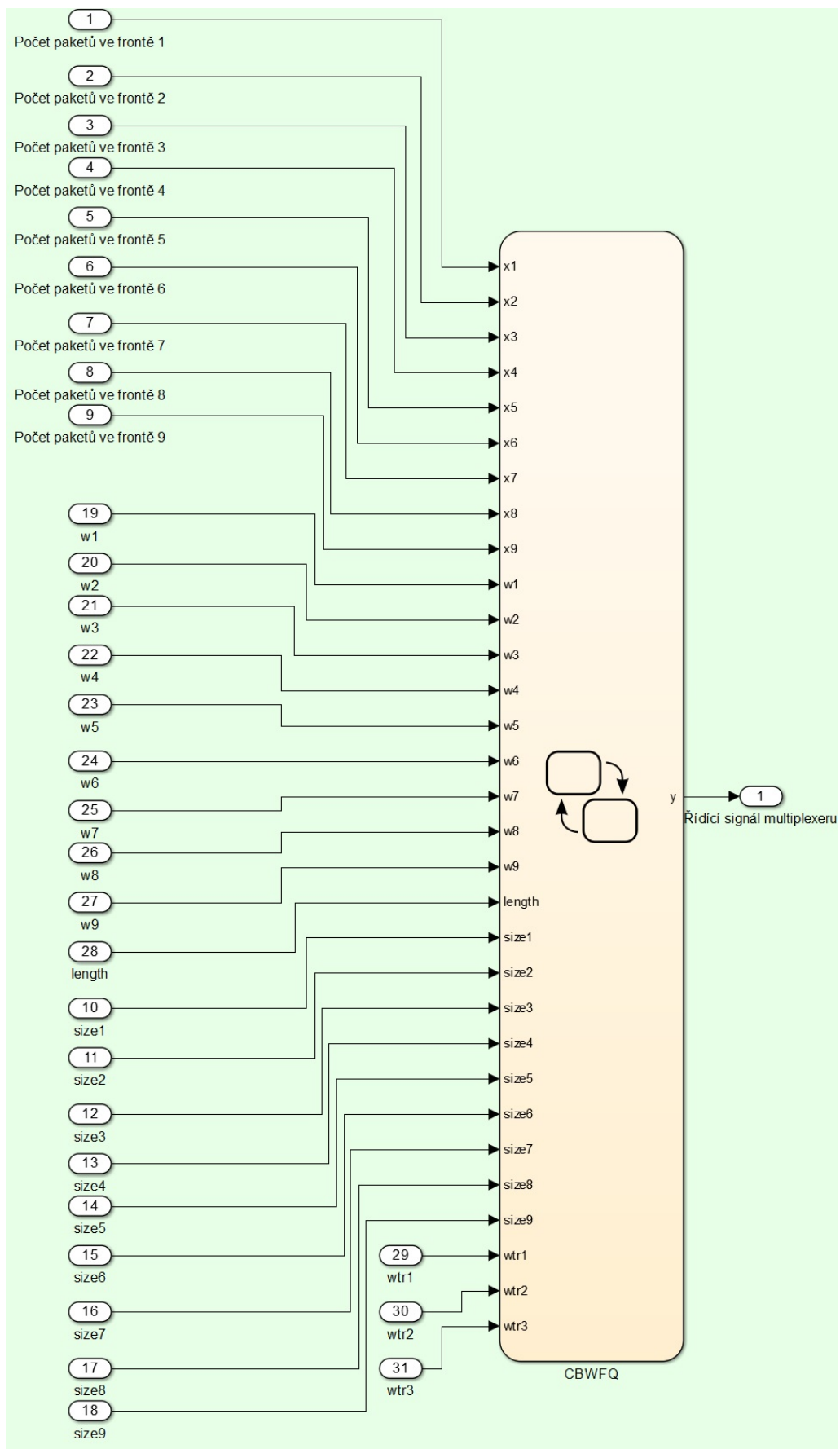
Velikost paketů [Byte]	Čas obslužení jediného paketu	Počet paketů obslužených za jednotku času	Požadovaná intenzita vstupního proudu	Hodnota Mean
80	6,40E-04	93750	0,066	0,009697
50	4,00E-04	150000	0,3	0,001333
30	2,40E-04	250000	0,066	0,003636
200	1,60E-03	37500	0,066	0,024242
100	8,00E-04	75000	0,1	0,008
120	9,60E-04	62500	0,066	0,014545
300	2,40E-03	25000	0,066	0,036364
200	1,60E-03	37500	0,1	0,016
500	4,00E-03	15000	0,066	0,060606

Tab. 4.2: Hodnoty vstupní intenzity pro nastavení scénáře 2

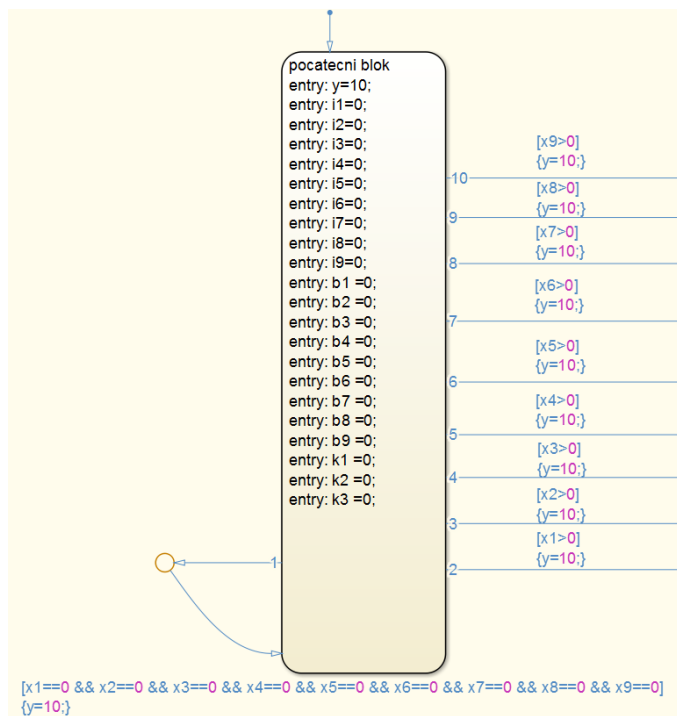
Velikost paketů [Byte]	Čas obslužení jediného paketu	Počet paketů obslužených za jednotku času	Požadovaná intenzita vstupního proudu	Hodnota Mean
80	6,40E-04	93750	0,066	0,009697
50	4,00E-04	150000	0,1	0,004
30	2,40E-04	250000	0,066	0,003636
200	1,60E-03	37500	0,066	0,024242
100	8,00E-04	75000	0,1	0,008
120	9,60E-04	62500	0,066	0,014545
300	2,40E-03	25000	0,066	0,036364
200	1,60E-03	37500	0,3	0,005333
500	4,00E-03	15000	0,066	0,060606

Tab. 4.3: Hodnoty vstupní intenzity pro nastavení scénáře 3

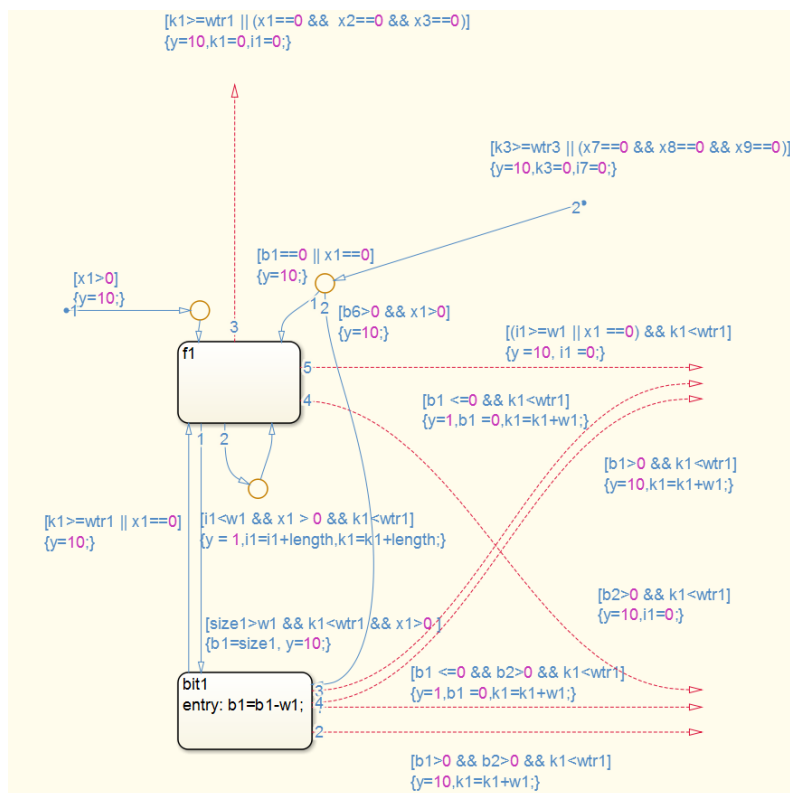
Velikost paketů [Byte]	Čas obslužení jediného paketu	Počet paketů obslužených za jednotku času	Požadovaná intenzita vstupního proudu	Hodnota Mean
80	6,40E-04	93750	0,066	0,009697
50	4,00E-04	150000	0,1	0,004
30	2,40E-04	250000	0,066	0,003636
200	1,60E-03	37500	0,066	0,024242
100	8,00E-04	75000	0,1	0,008
120	9,60E-04	62500	0,066	0,014545
300	2,40E-03	25000	0,066	0,036364
200	1,60E-03	37500	0,3	0,005333
500	4,00E-03	15000	0,066	0,060606



Obr. 4.12: Vstupní parametry algoritmu CBWFQ



Obr. 4.13: Vstupní blok algoritmu CBWFQ pro definování proměných



Obr. 4.14: Část vnitřní struktury algoritmu CBWFQ v bloku Char

5 ZÁVĚR

Cílem této bakalářské práce bylo nastínit problematiku systému front v aktivních prvcích s důrazem na popis algoritmů, ztrátovosti paketů a vznikajícího zpoždění paketů ve frontách.

V prvních kapitolách jsou rozvedena teorie front, správy front a popis jednotlivých druhů obsluhy. Ve čtvrté kapitole jsou rozepsány jednotlivé části modelu realizované v prostředí Simulink. V kapitole 4.3 bylo demonstrováno porovnání systému spravedlivé obsluhy se systémem front s váženou spravedlivou obsluhou řízenou podle tříd s následným na vysvětlení výsledků z grafů z nichž na první pohled vyplývá rapidní rozdíl mezi těmito dvěma systémy obsluhy.

V rámci bakalářské bylo vytvořeno šest modelů obsluhy, které se využívají pro zajištění kvality služeb v telekomunikačních sítích v aktivních prvcích a jejich následné vzájemné porovnání při konstantních vstupních parametrech ve třech scénářích. Všechny modely byly vytvořeny v simulačním programu Matlab-Simulink s využitím knihoven Simevents a Stateflow určených pro diskrétní simulace.

U modelů byla porovnávána ztrátovost paketů s využitím mechanismu předčasného náhodného zahazování RED a čas, který pakety strávily ve frontách než byly obslouženy. Výsledky těchto porovnání jsou zhodnoceny v kapitole 4.3 a v příloze A.1 jsou uvedeny všechny grafické výsledky provedených simulací.

Vytvořené modely potvrzují teoretické předpoklady, největším rozdílem ve výsledcích mezi jednotlivými modely byla vlastnost modelu zda zohledňují velikosti paketů. Jediný model, který teoretické předpoklady nesplnil byl FIFO, zde docházelo k blokaci vstupních paketů na vstupu multiplexeru z důvodu kolize, kdy jednotlivé požadavky přicházely ve stejný čas na většinu vstupních portů a propuštěn mohl být jen jeden. To se odrazilo v nízké ztrátovosti paketů i zpoždění paketů ve frontě.

LITERATURA

- [1] COOPER, R.B, *Introduction to queueing theory..* 2d ed. New York: North Holland, c1981, 347 p. ISBN 04-440-0379-7.
- [2] CHAO, H.J., LIU, B. *High Performance Switches and Routers.* Wiley-IEEE Press, c2007, 788 s. ISBN 978-0-12-088588-6.
- [3] KOTON, J, *Moderní síťové technologie..* Skripta VUT Brno, c2013, 191s.
- [4] MEDHI, D., RAMASAMY, K. *Network routing: alogorithms, protocols, and architectures..* Amsterdam: Elsevier, c2007, 788 s. ISBN 978-0-12-088588-6.
- [5] ŠEDA, M. *Modely hromadné obsluhy.*[online]. s. 33 [cit. 12.10.2014]. Dostupné z URL: <http://web2.vslg.cz/fotogalerie/acta_logistica/2011/2_cislo/3_seda.pdf>.
- [6] The MathWorks, Inc. *Stateflow User's Guide.*[online]. s. 12-2 [cit. 23.05.2015]. Dostupné z URL: <http://www.mathworks.com/help/pdf_doc/stateflow/sf_ug.pdf>.
- [7] The MathWorks, Inc. *SimEvents Getting Started Guide.*[online]. s. 12-2 [cit. 23.05.2015]. Dostupné z URL: <http://www.mathworks.com/help/pdf_doc/simevents/simevents_gs.pdf>.
- [8] Cisco *Voice Over IP - Per Call Bandwidth Consumption.*[online]. s. 12 [cit. 23.05.2015]. Dostupné z URL: <<http://www.cisco.com/c/en/us/support/docs/voice/voice-quality/7934-bwidth-consume.html>>.
- [9] Little, J. *A proof for the queuing formula: $L = \lambda W$.*[online]. s. 383-387 [cit. 23.05.2015]. Dostupné z URL: <<http://www.cs.bilkent.edu.tr/~tugrul/CS518/Papers/little.pdf>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

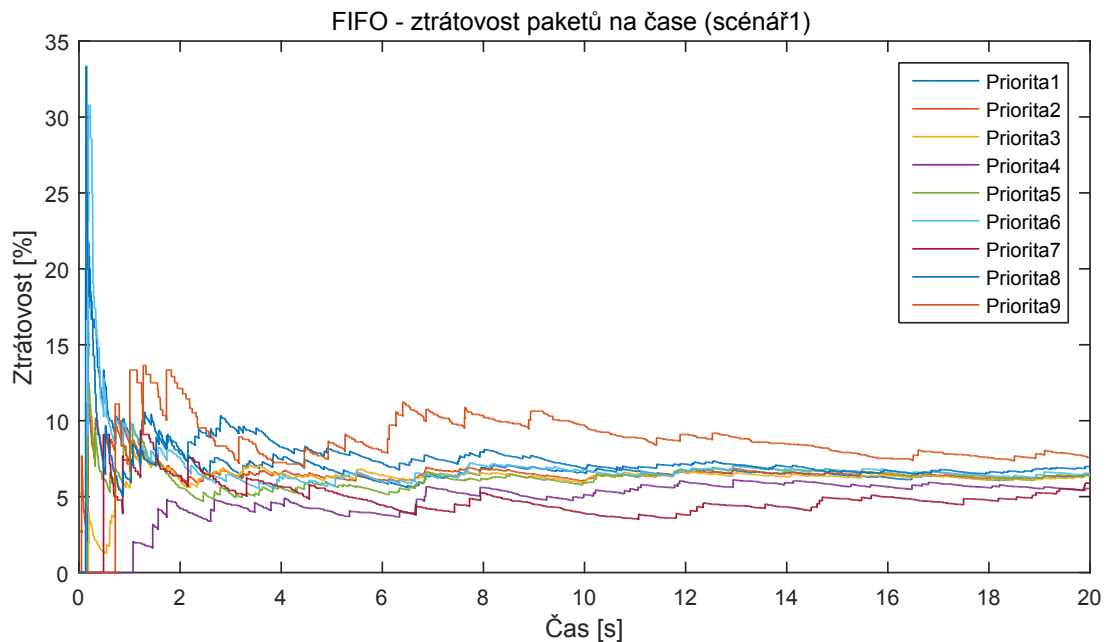
ECN	Explicit Congestion Notification
FQ	Fair Queuing
QoS	Quality of Service
RED	Random Early Detection
SHO	System hromadné obsluhy
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
WFQ	Weighed Fair Queuing
WRED	Weighted Random Early Detection
WRR	Weighed Round Robin

SEZNAM PŘÍLOH

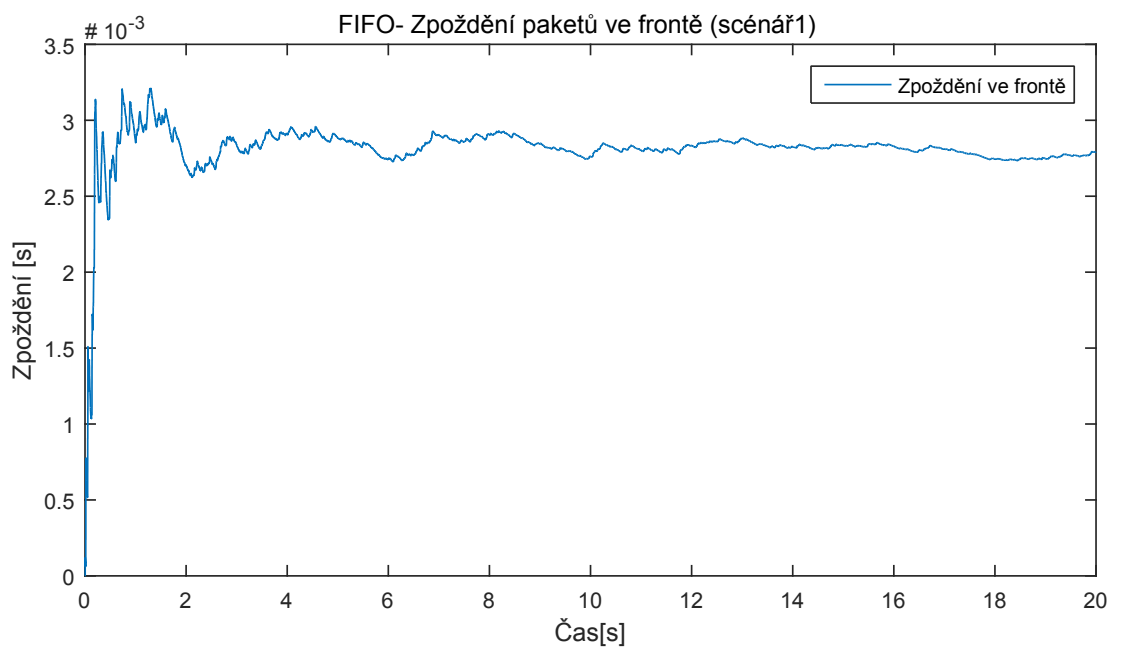
A Příloha	50
A.1 Výsledky simulace	50

A PŘÍLOHA

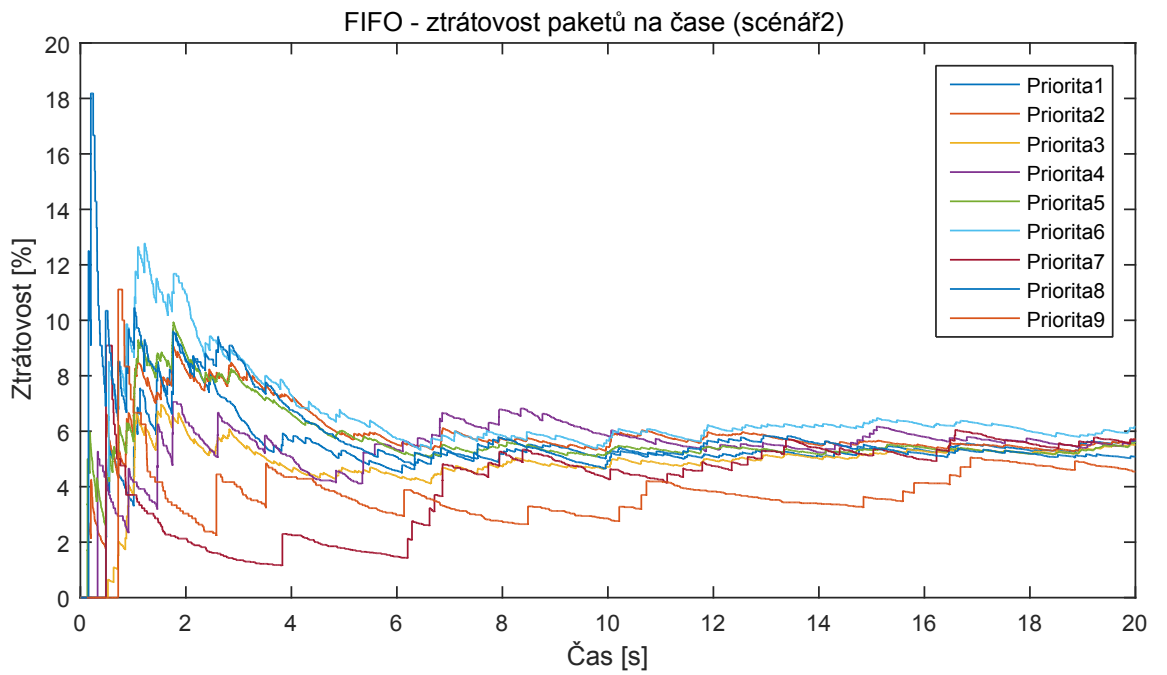
A.1 Výsledky simulace



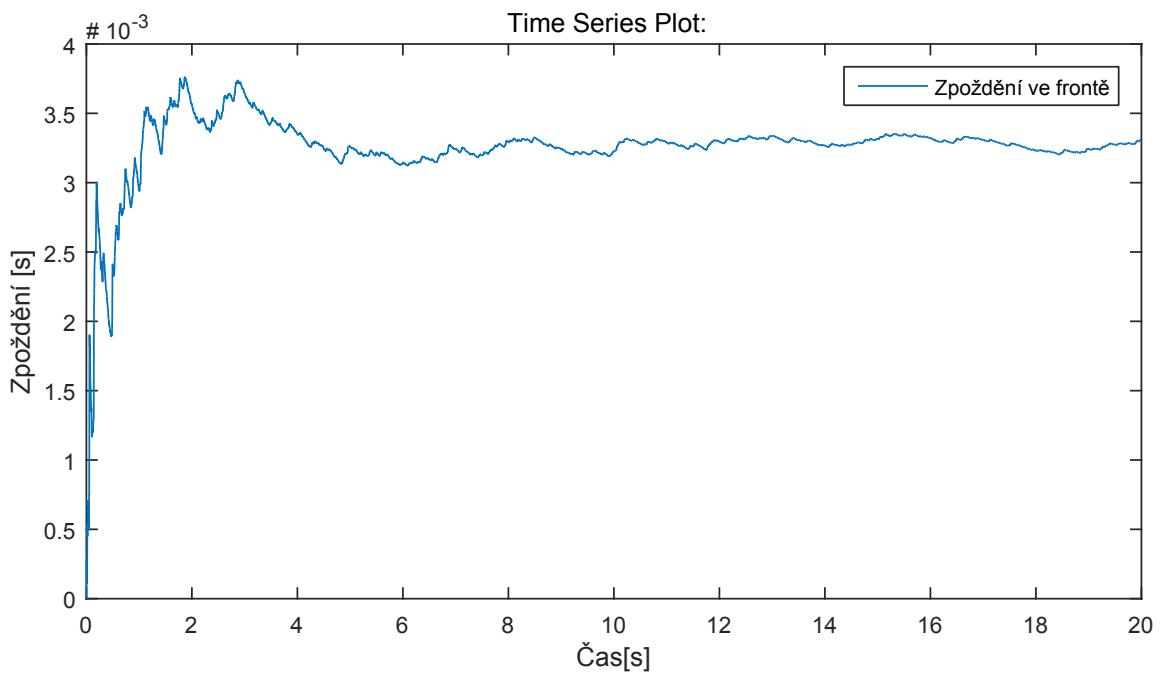
Obr. A.1: FIFO - Ztrátovost paketů na čase - scénář 1



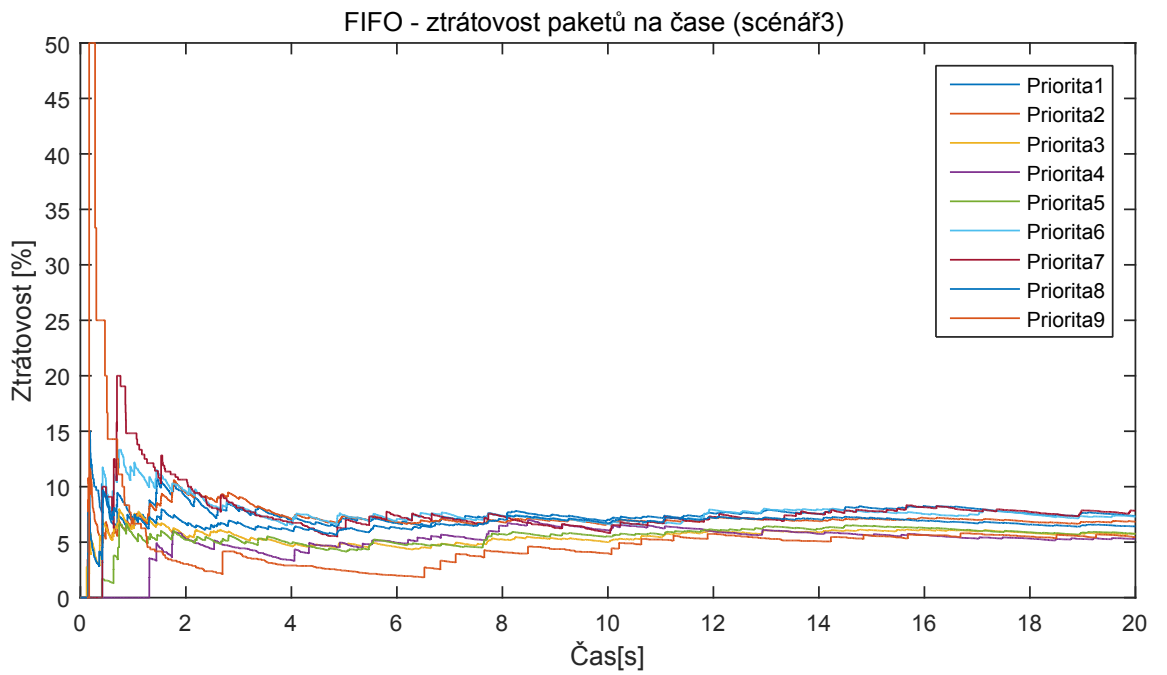
Obr. A.2: FIFO - Zpoždění paketů ve frontě - scénář 1



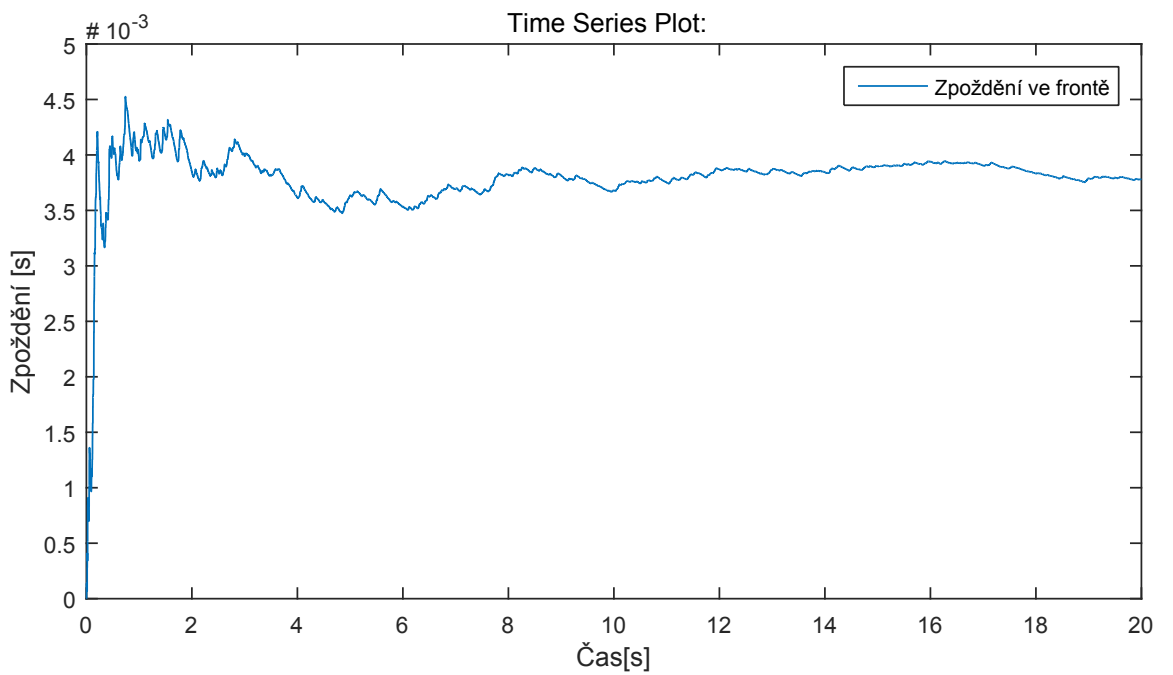
Obr. A.3: FIFO - Ztrátovost paketů na čase - scénář 2



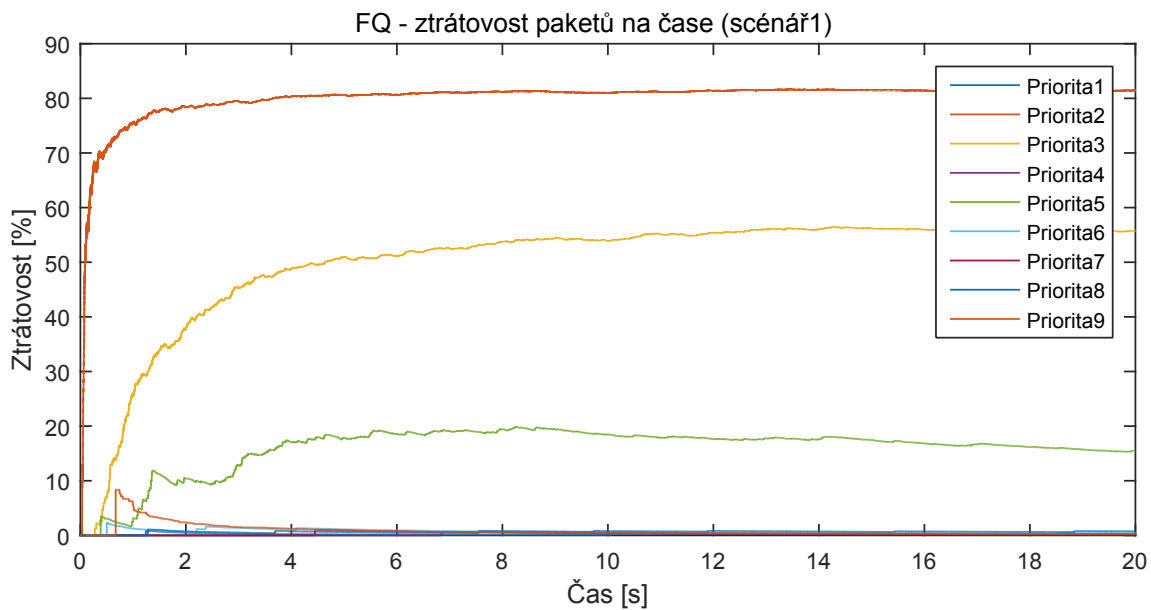
Obr. A.4: FIFO - Zpoždění paketů ve frontě - scénář 2



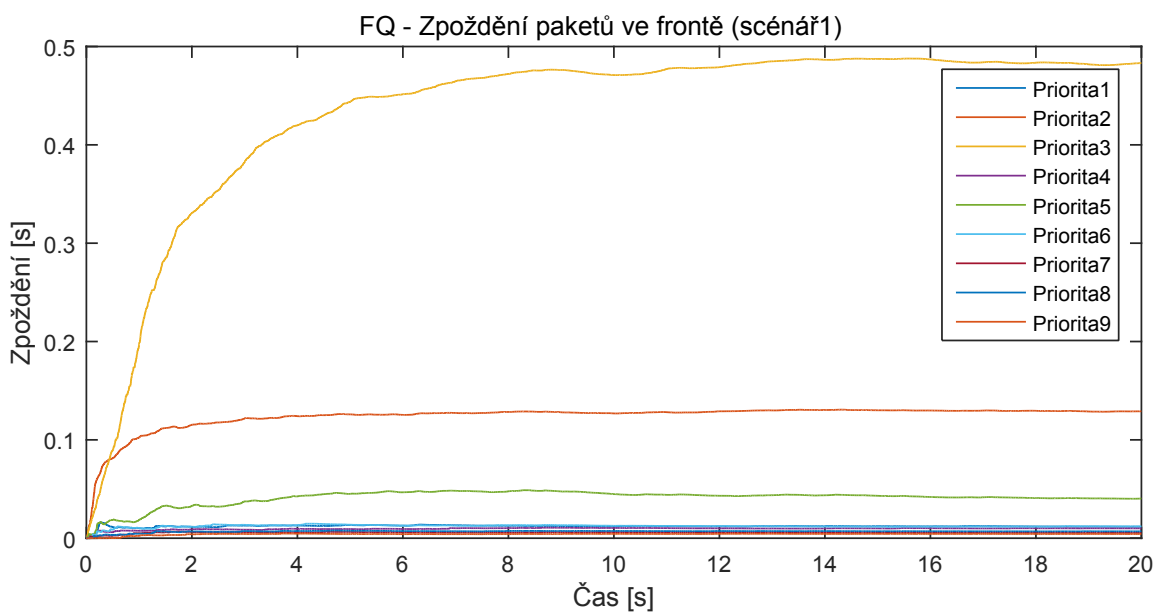
Obr. A.5: FIFO - Ztrátovost paketů na čase - scénář 3



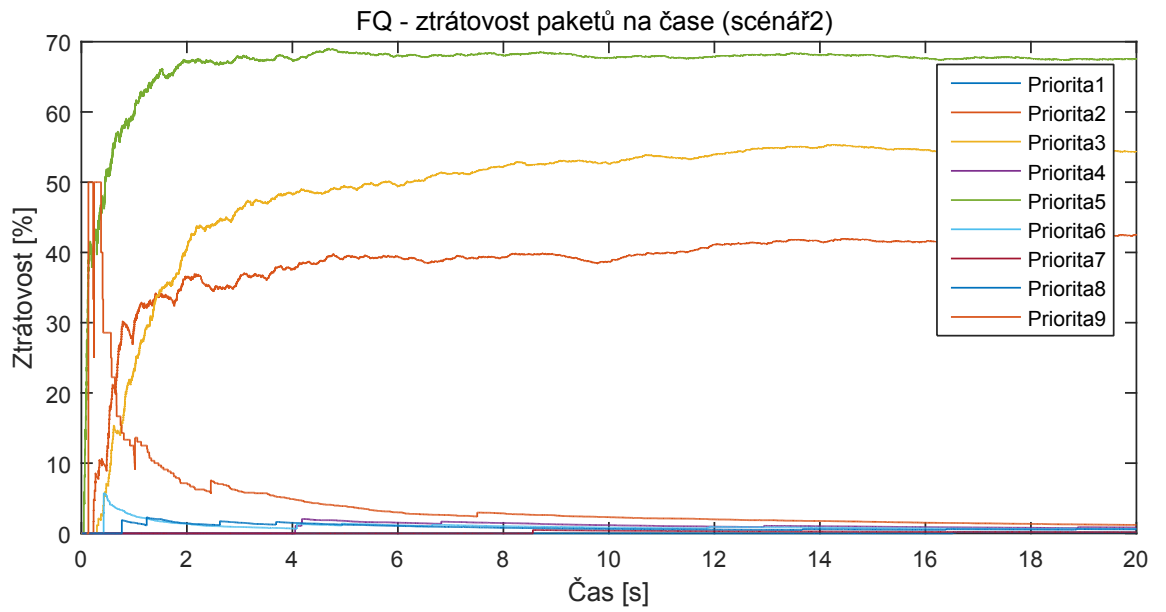
Obr. A.6: FIFO - Zpoždění paketů ve frontě - scénář 3



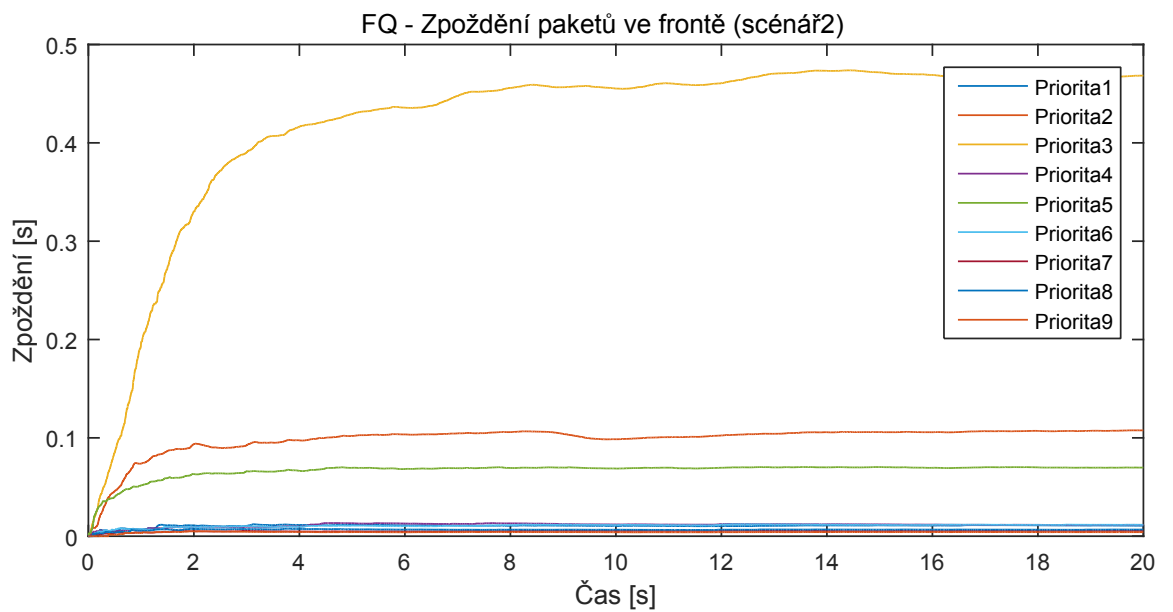
Obr. A.7: FQ - Ztrátovost paketů na čase - scénář 1



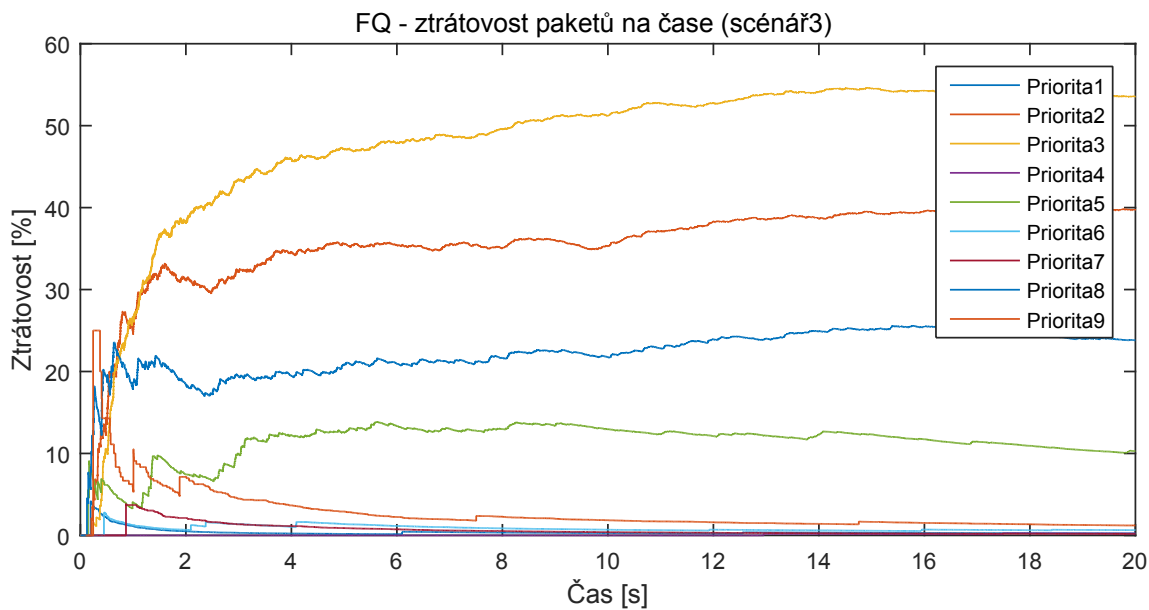
Obr. A.8: FQ - Zpoždění paketů ve frontě - scénář 1



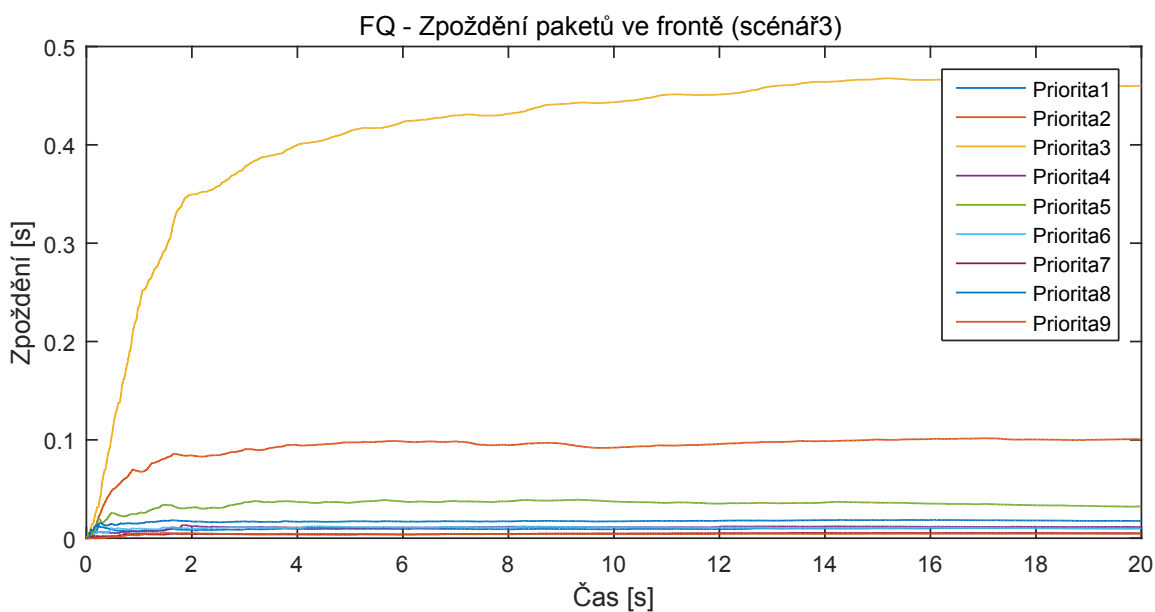
Obr. A.9: FQ - Ztrátovost paketů na čase - scénář 2



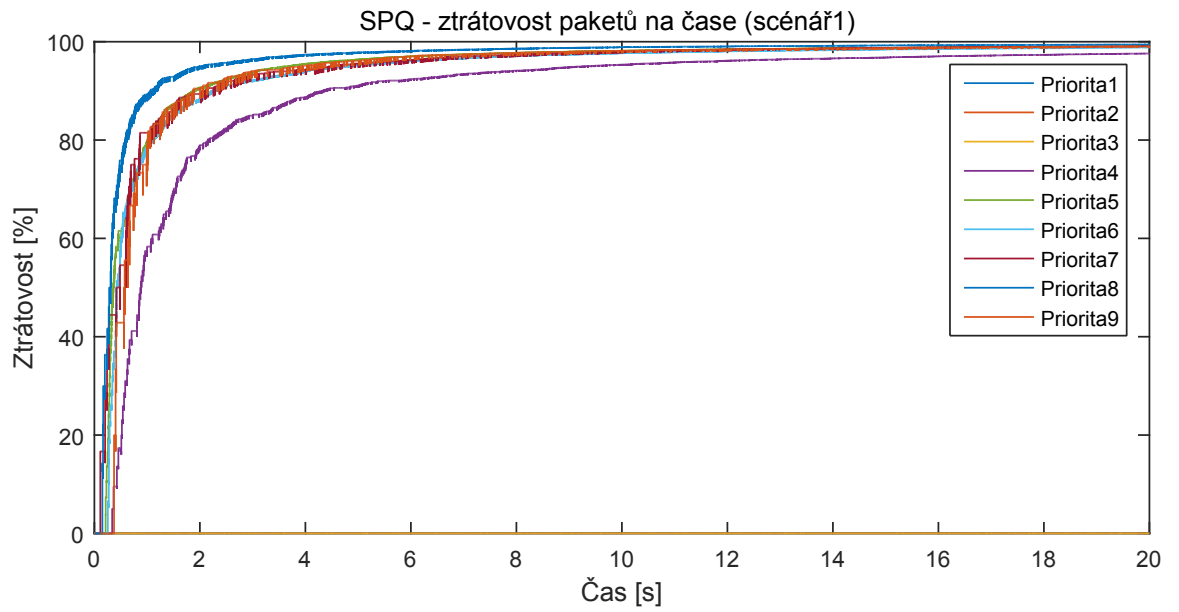
Obr. A.10: FQ - Zpoždění paketů ve frontě - scénář 2



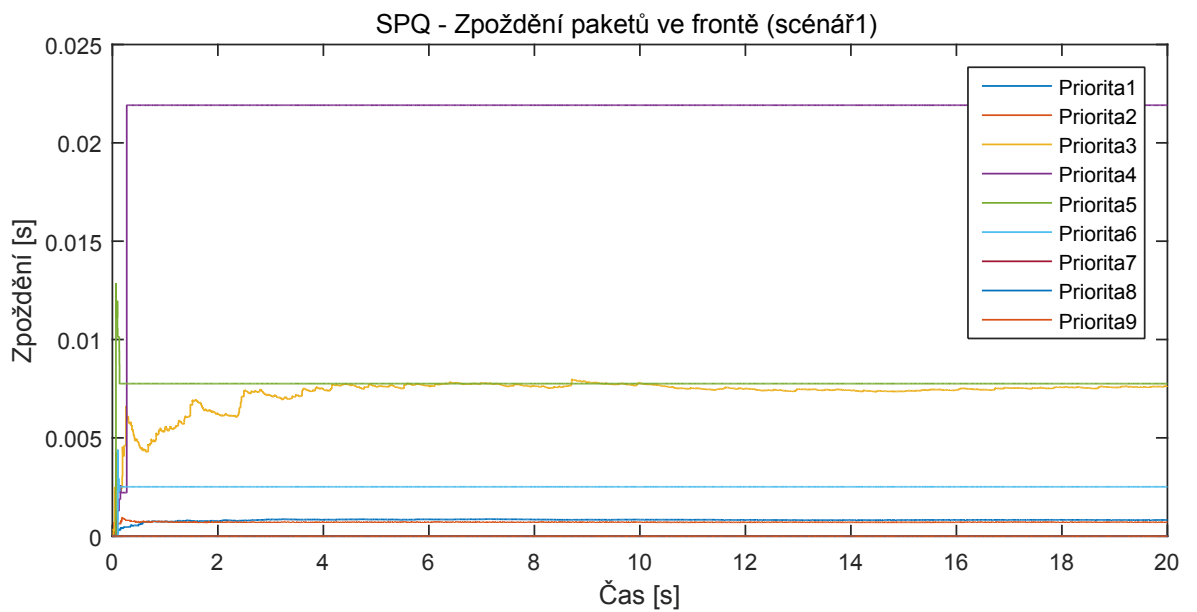
Obr. A.11: FQ - Ztrátovost paketů na čase - scénář 3



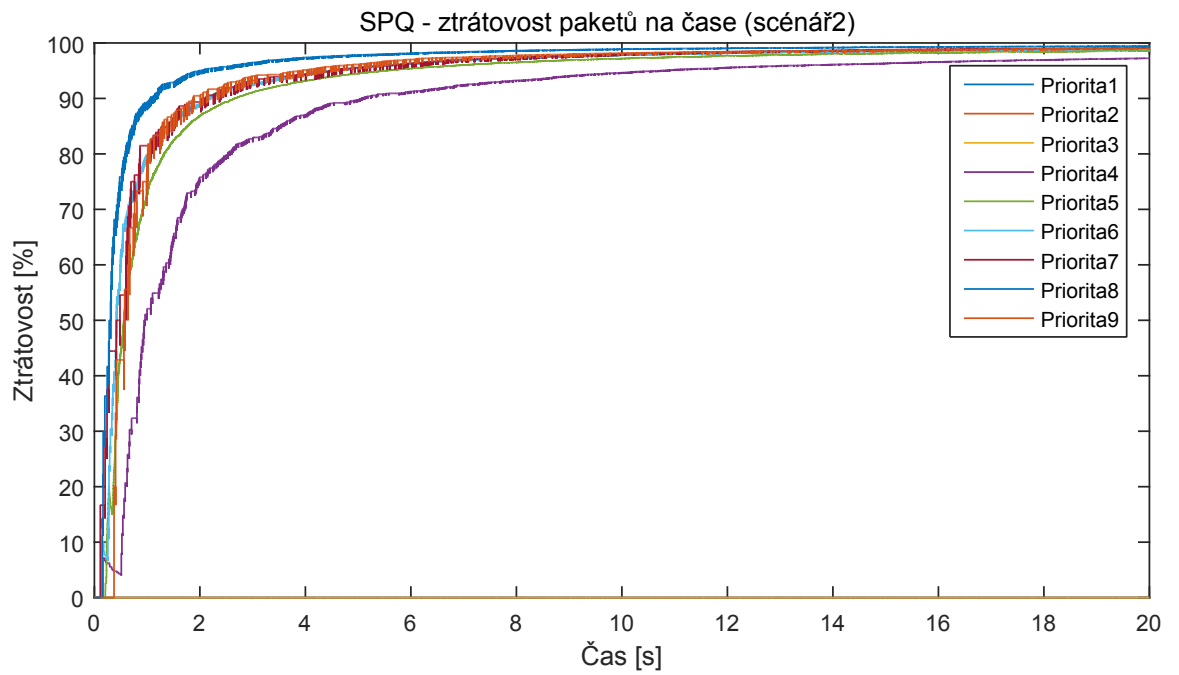
Obr. A.12: FQ - Zpoždění paketů ve frontě - scénář 3



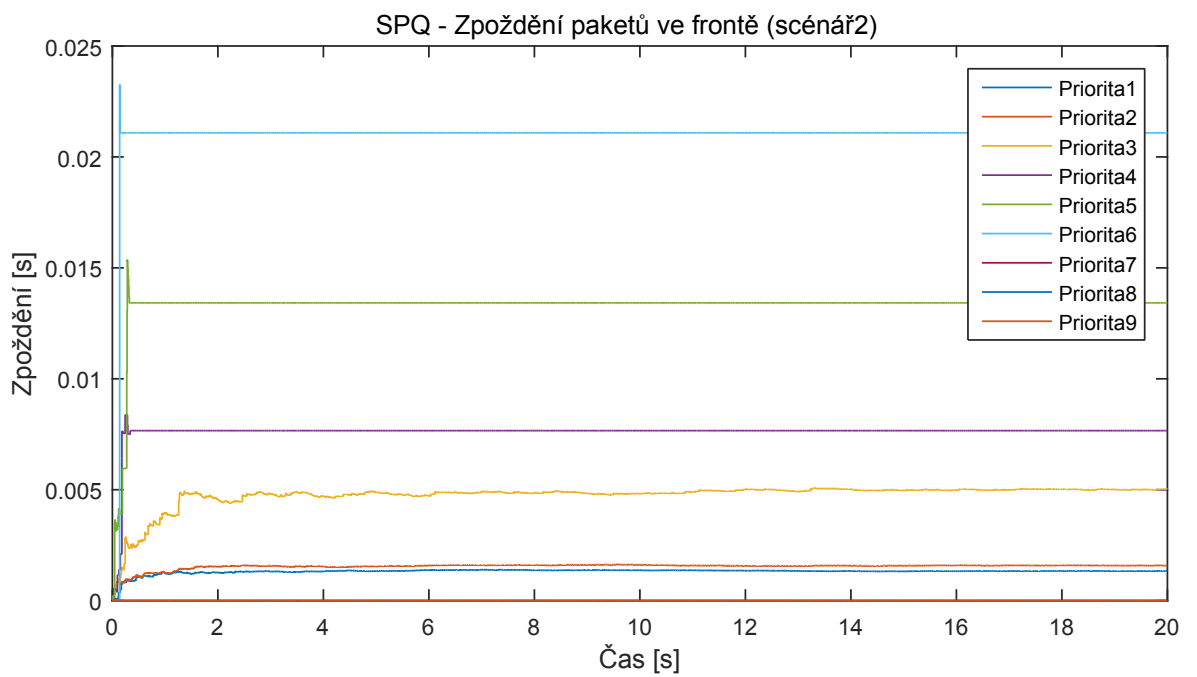
Obr. A.13: SPQ - Ztrátovost paketů na čase - scénář 1



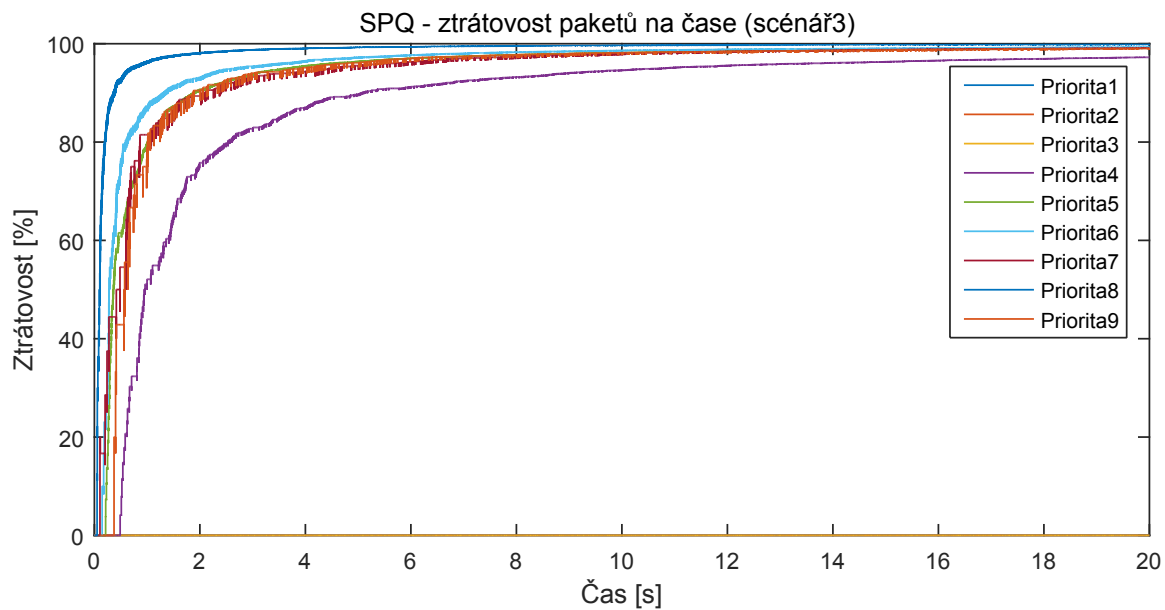
Obr. A.14: SPQ - Zpoždění paketů ve frontě - scénář 1



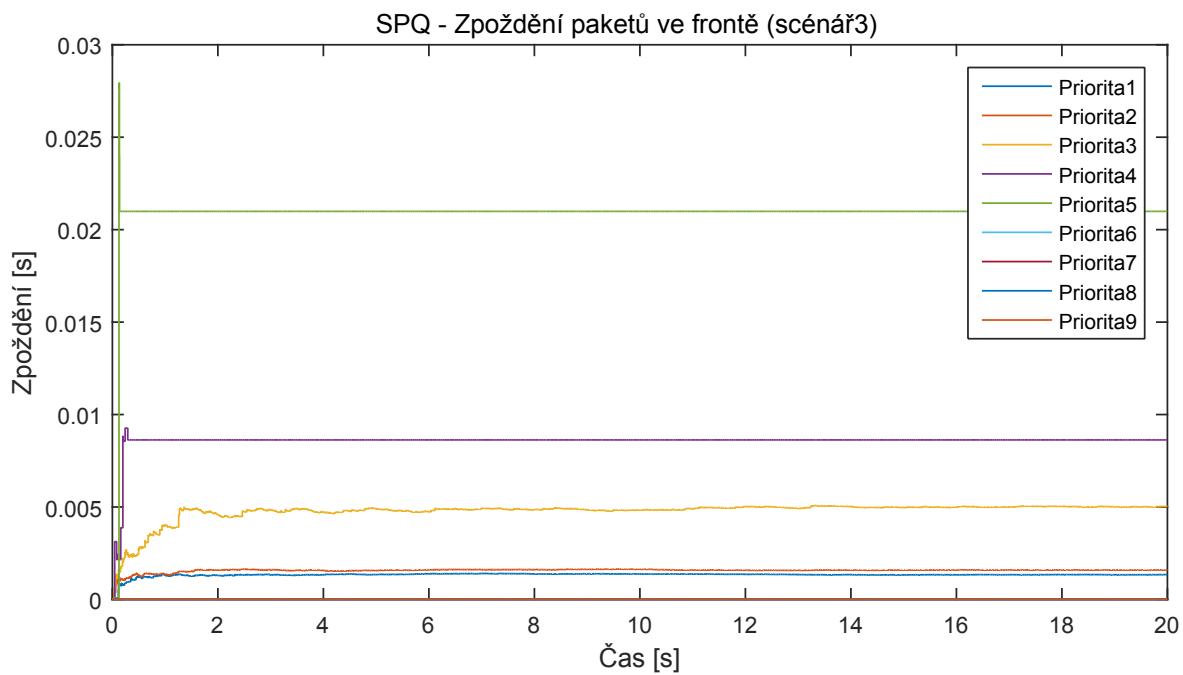
Obr. A.15: SPQ - Ztrátovost paketů na čase - scénář 2



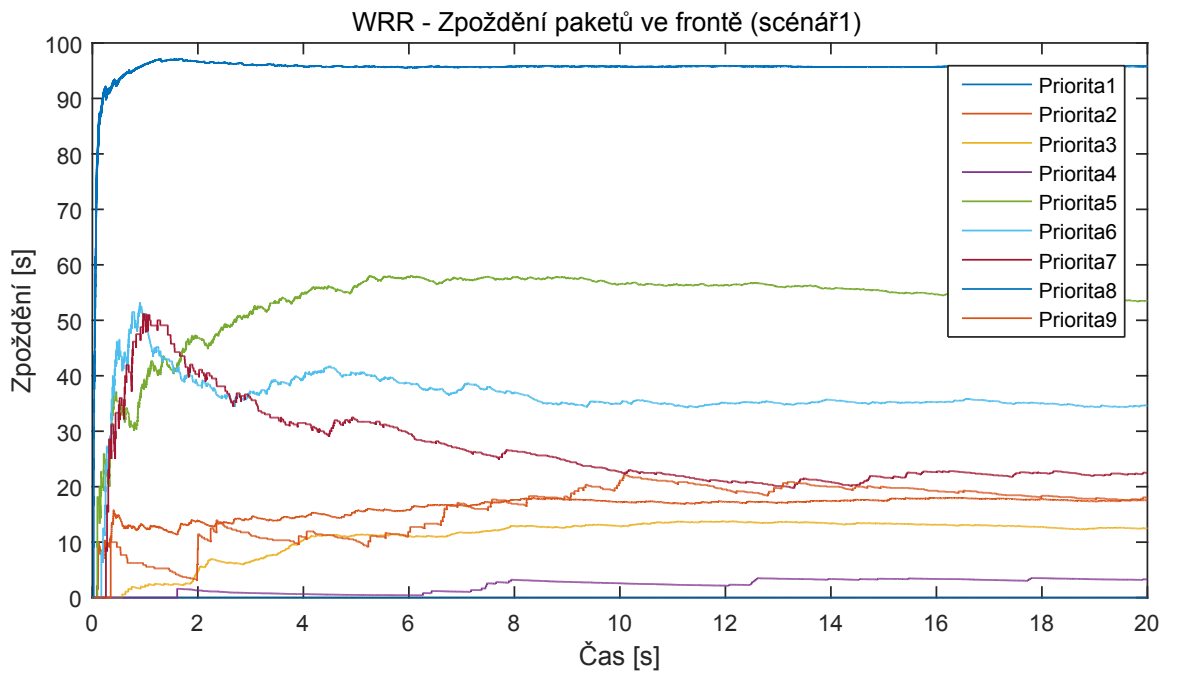
Obr. A.16: SPQ - Zpoždění paketů ve frontě - scénář 2



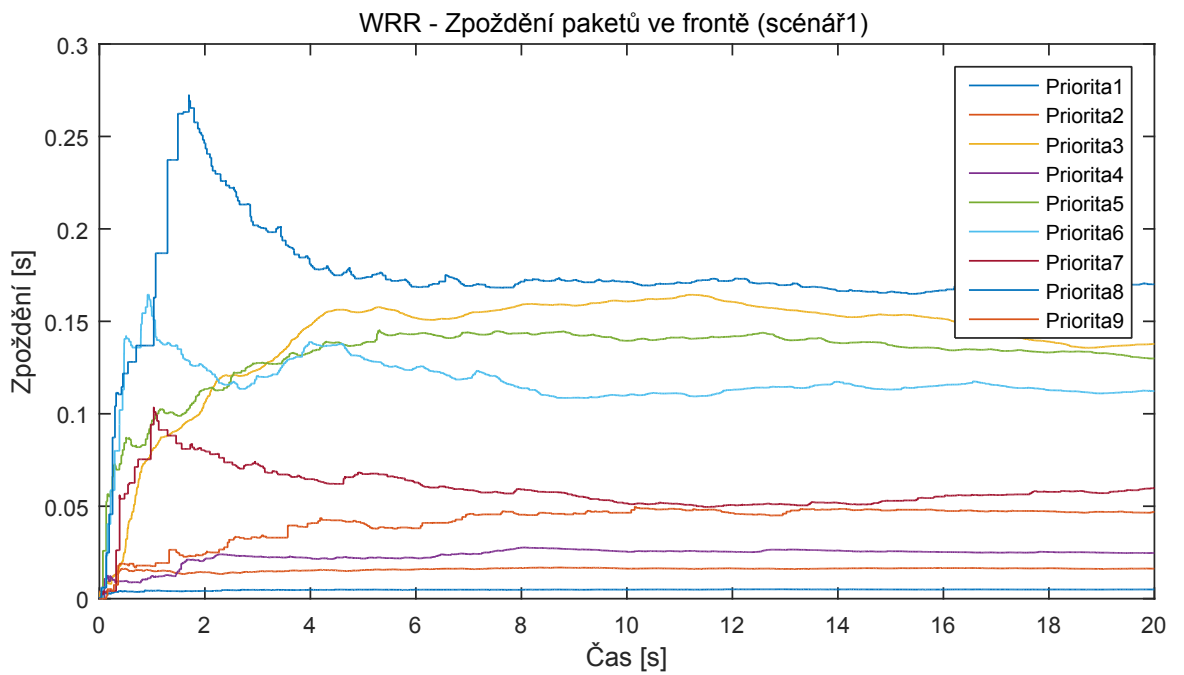
Obr. A.17: SPQ - Ztrátovost paketů na čase - scénář 3



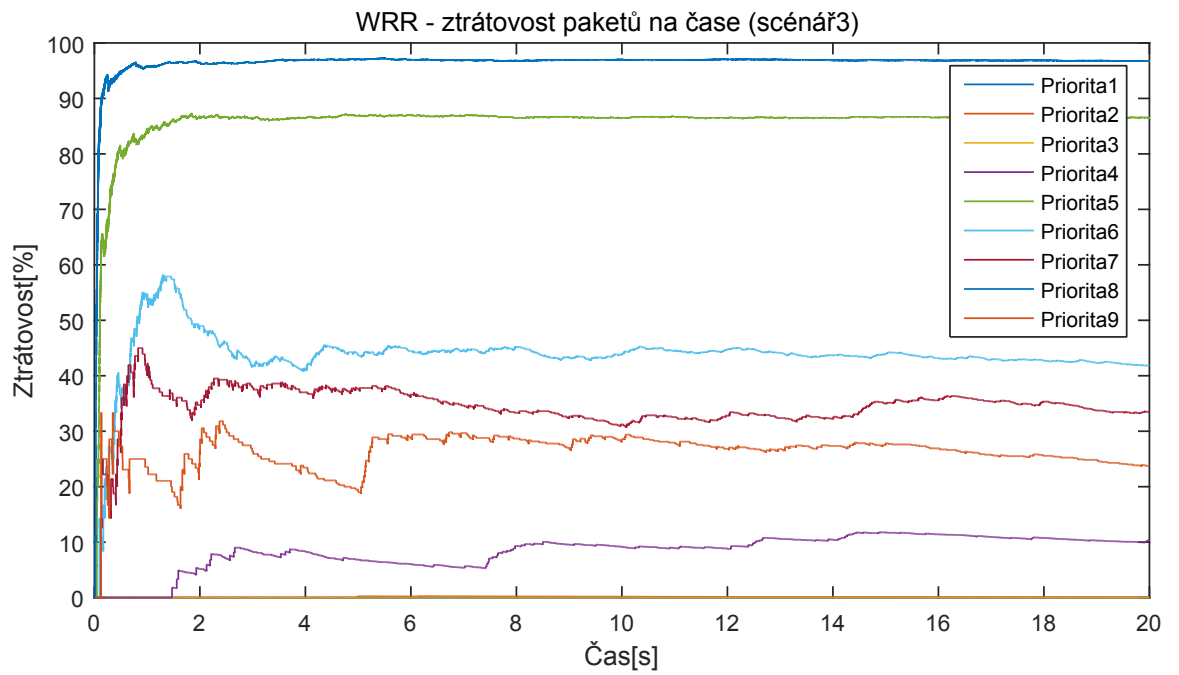
Obr. A.18: SPQ - Zpoždění paketů ve frontě - scénář 3



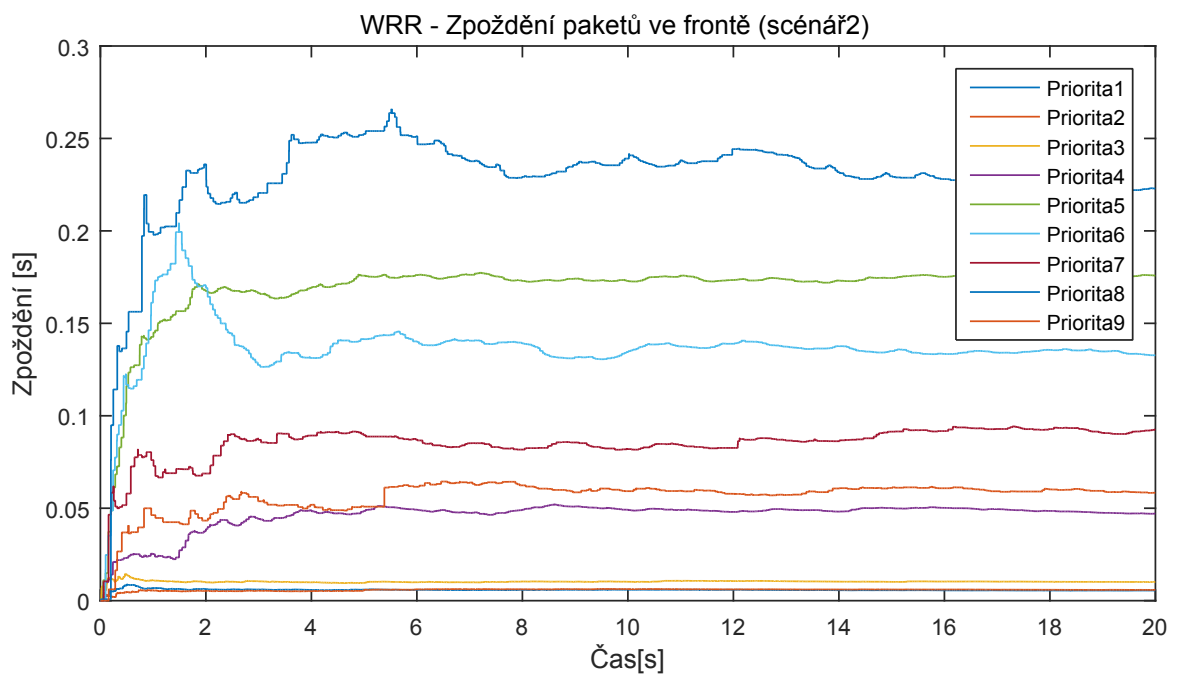
Obr. A.19: WRR - Ztrátovost paketů na čase - scénář 1



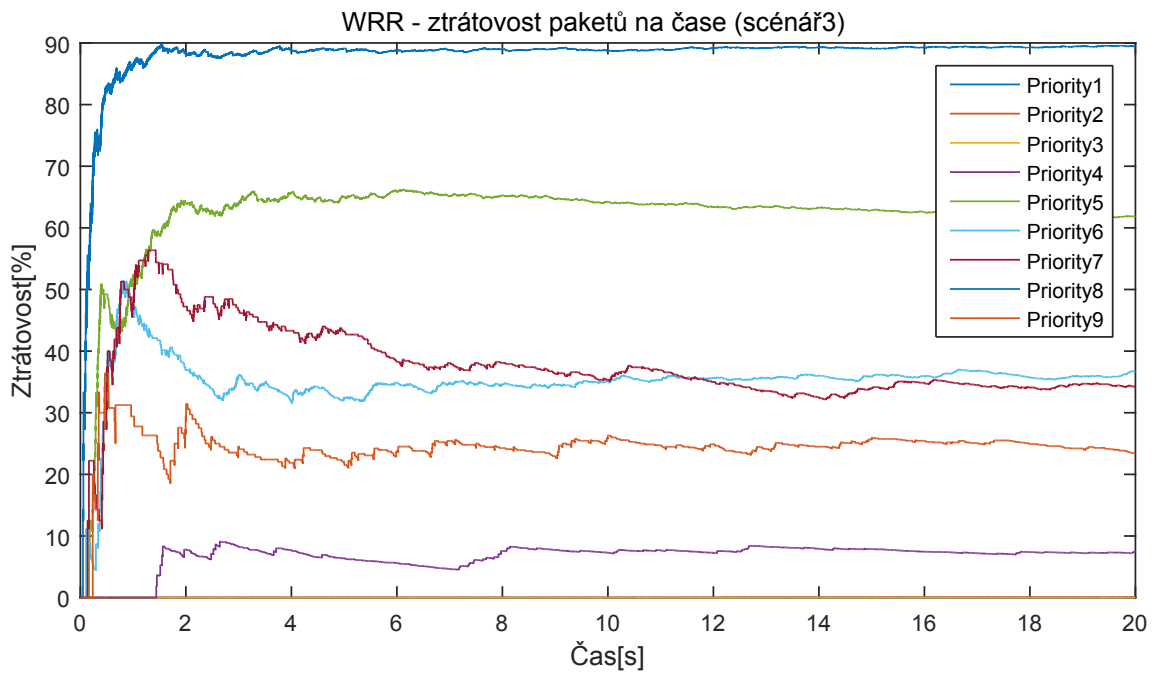
Obr. A.20: WRR - Zpoždění paketů ve frontě - scénář 1



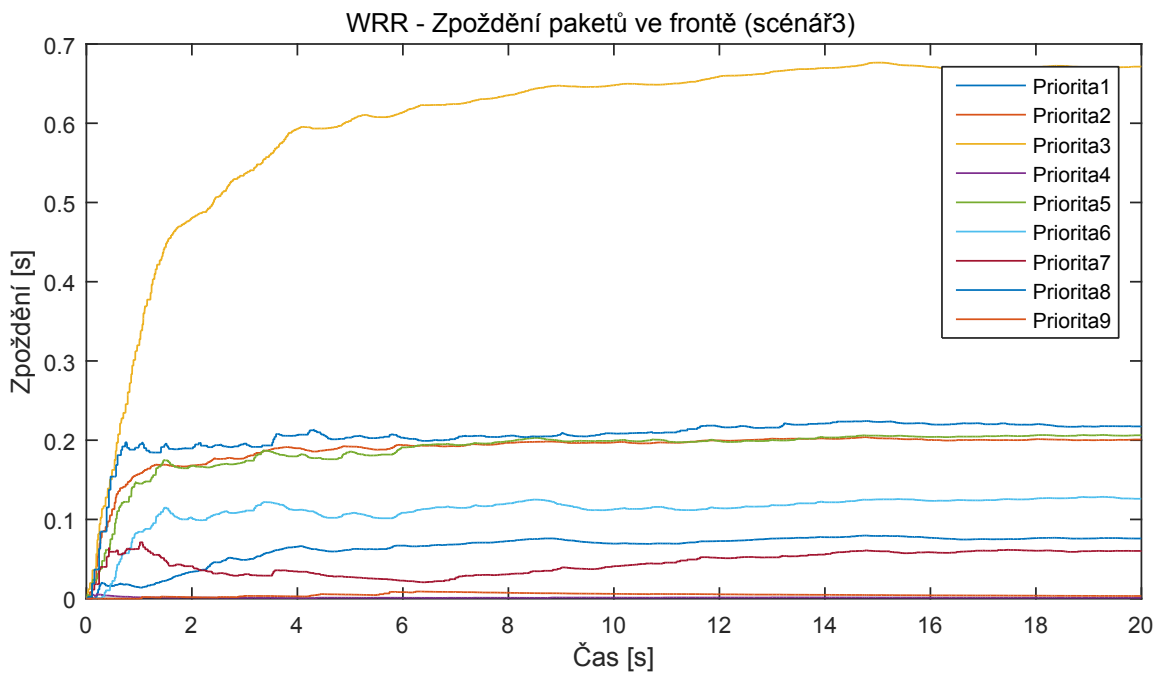
Obr. A.21: WRR - Ztrátovost paketů na čase - scénář 2



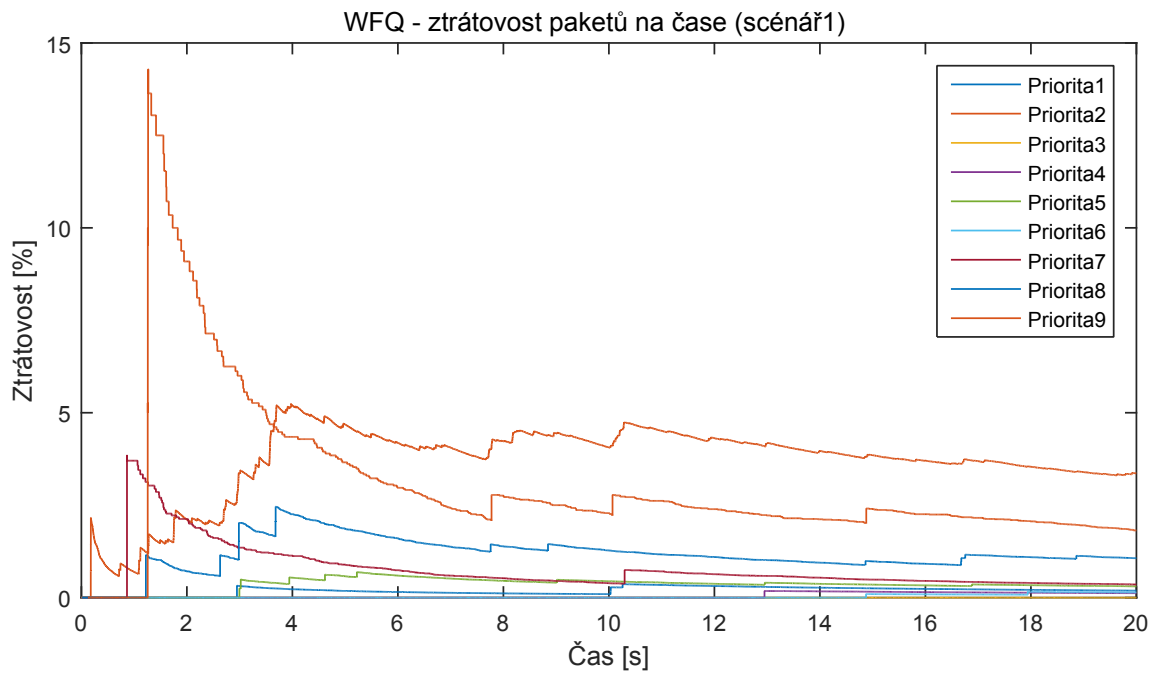
Obr. A.22: WRR - Zpoždění paketů ve frontě - scénář 2



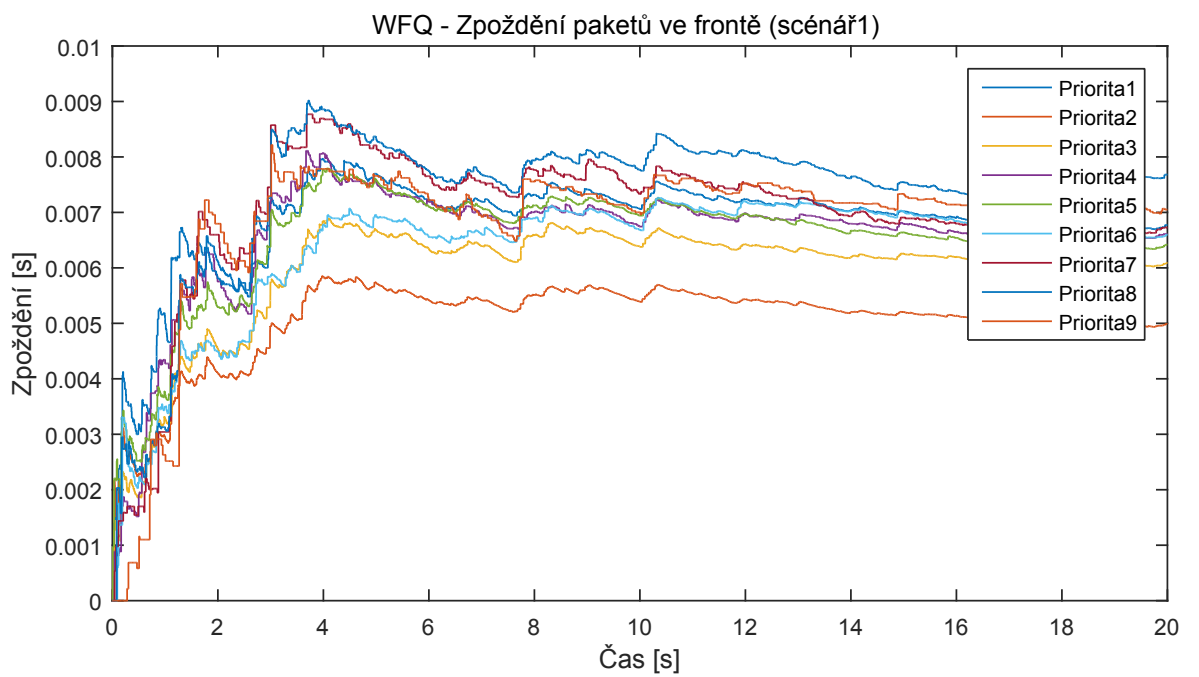
Obr. A.23: WRR - Ztrátovost paketů na čase - scénář 3



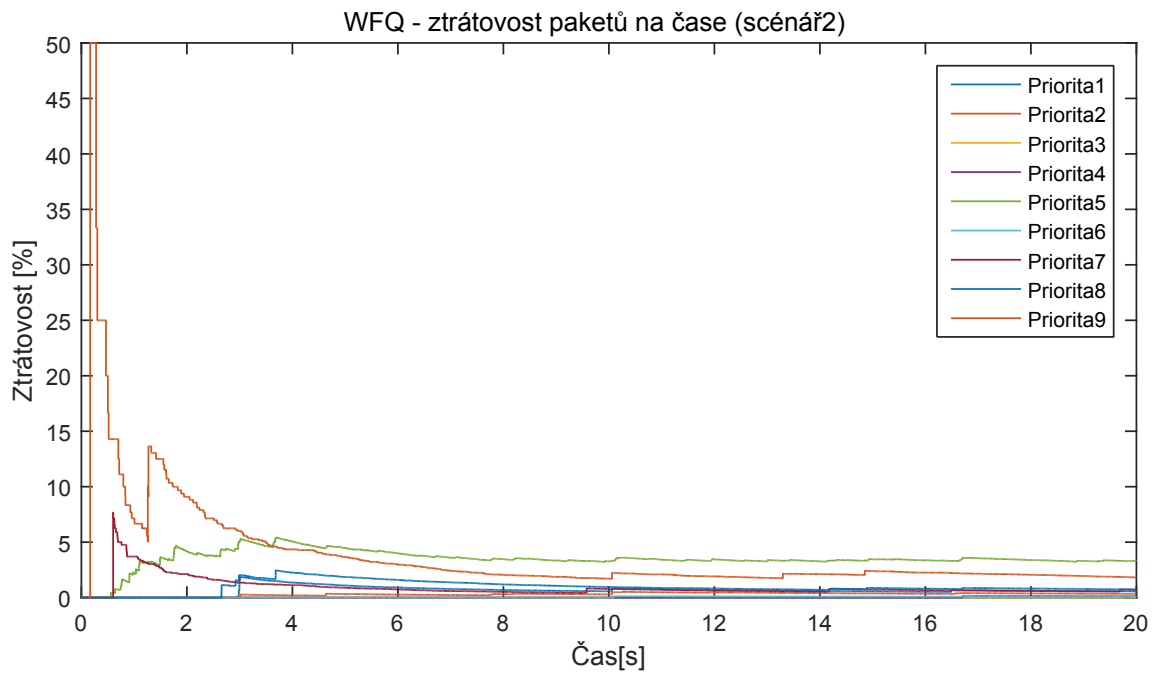
Obr. A.24: WRR - Zpoždění paketů ve frontě - scénář 3



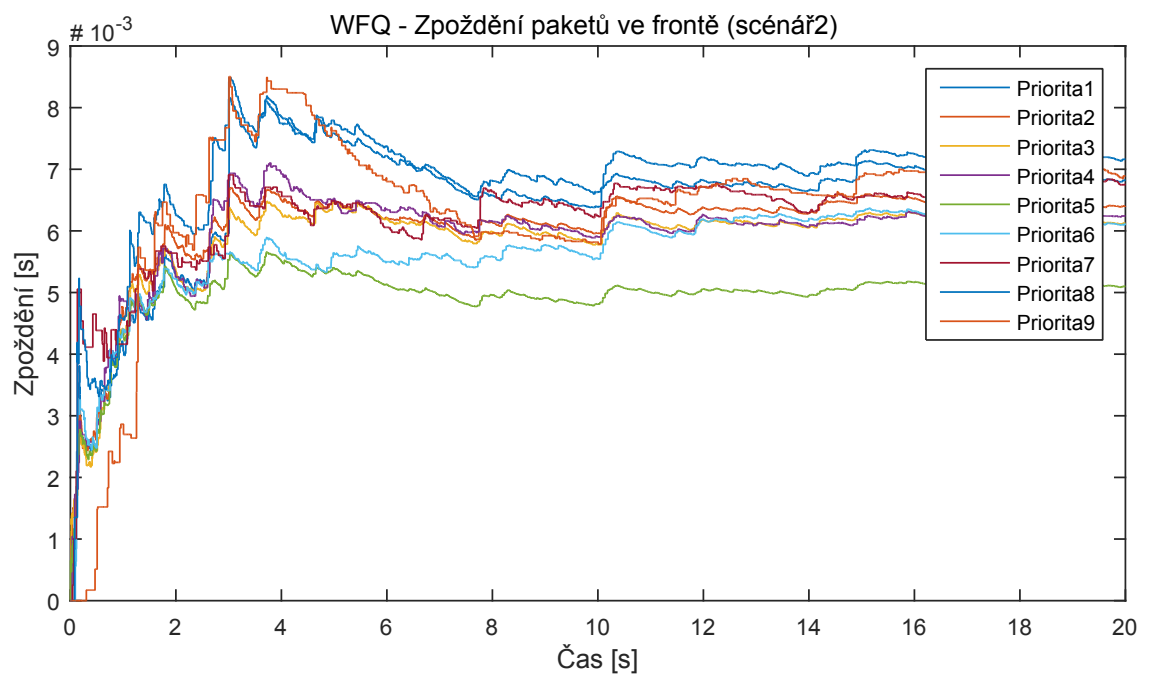
Obr. A.25: WFQ - Ztrátovost paketů na čase - scénář 1



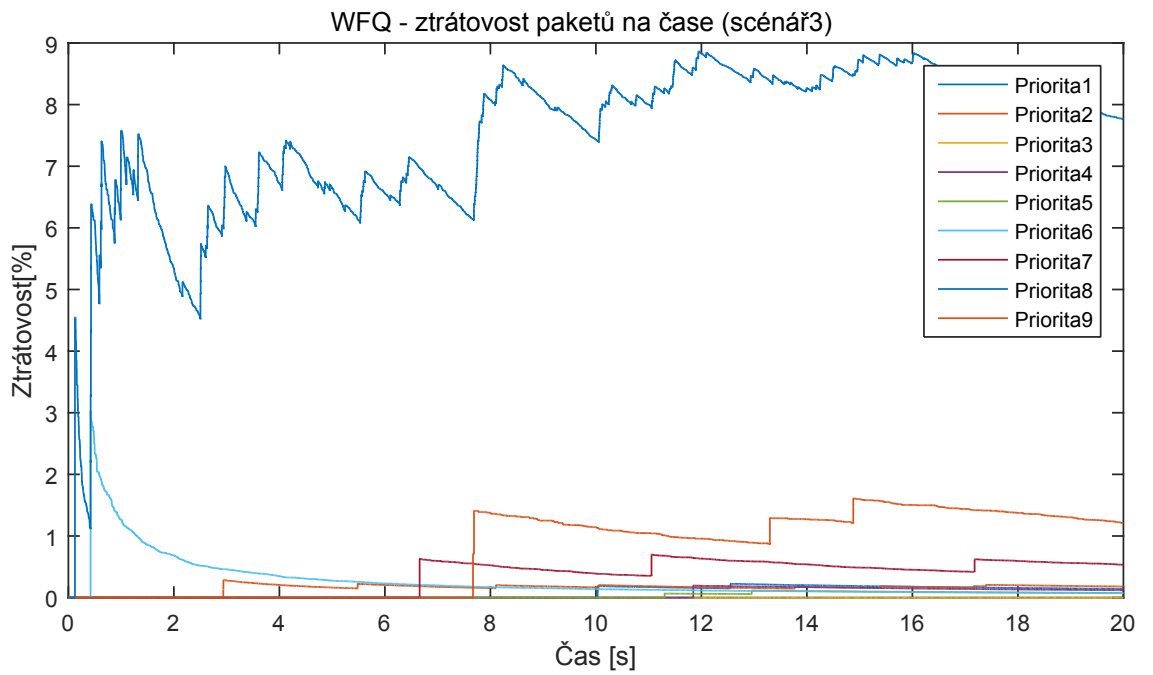
Obr. A.26: WFQ - Zpoždění paketů ve frontě - scénář 1



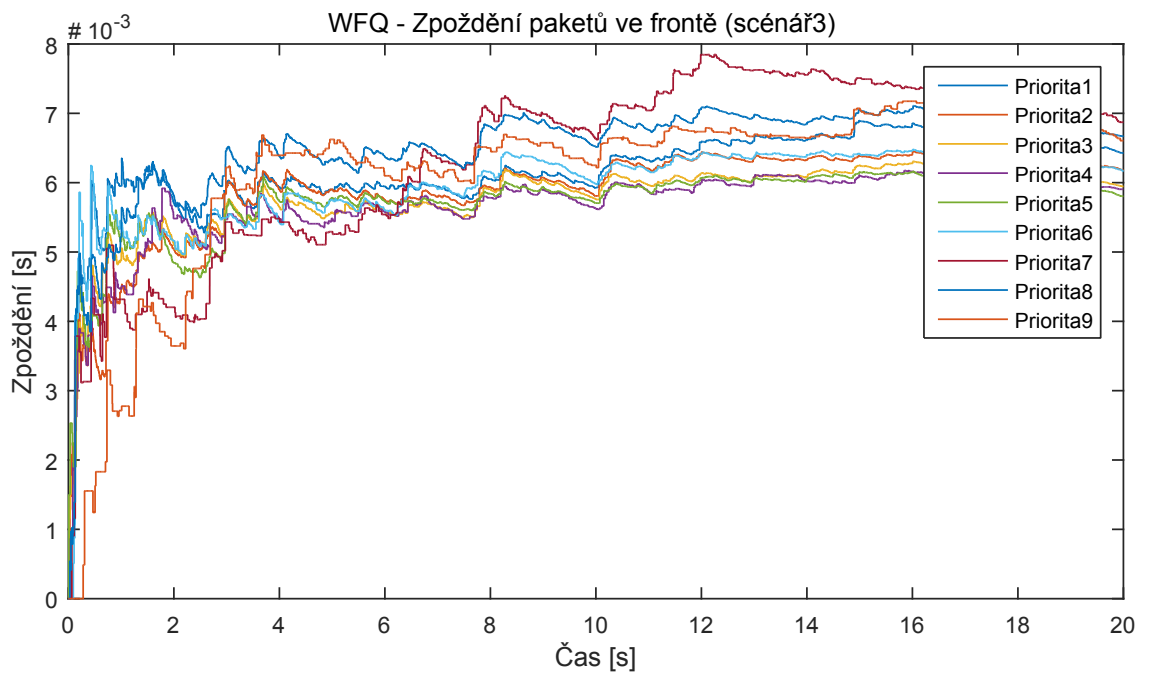
Obr. A.27: WFQ - Ztrátovost paketů na čase - scénář 2



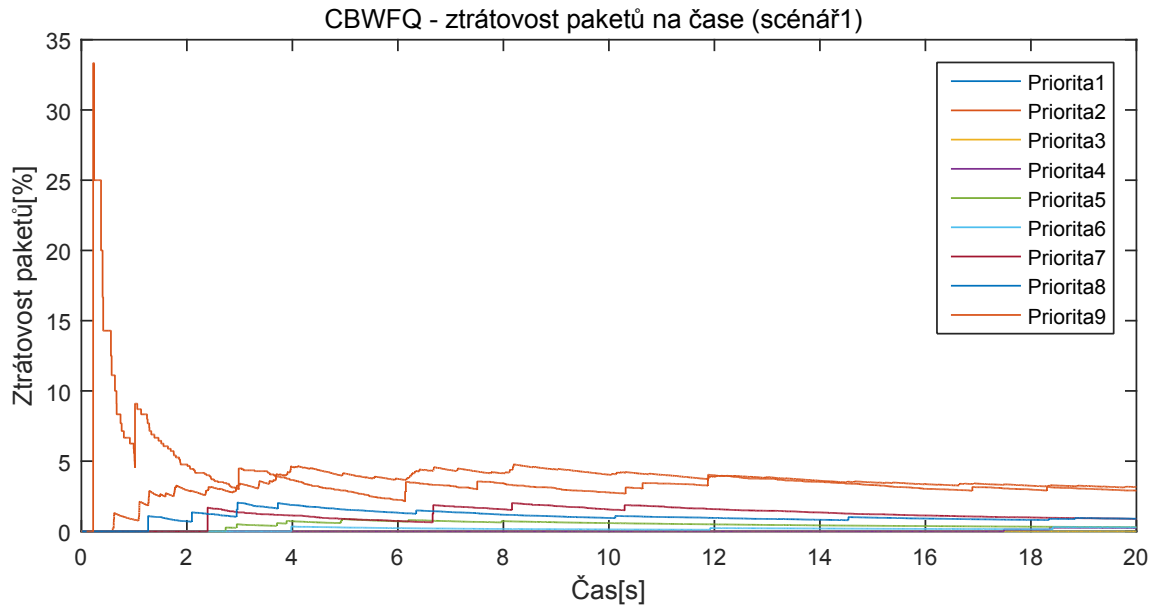
Obr. A.28: WFQ - Zpoždění paketů ve frontě - scénář 2



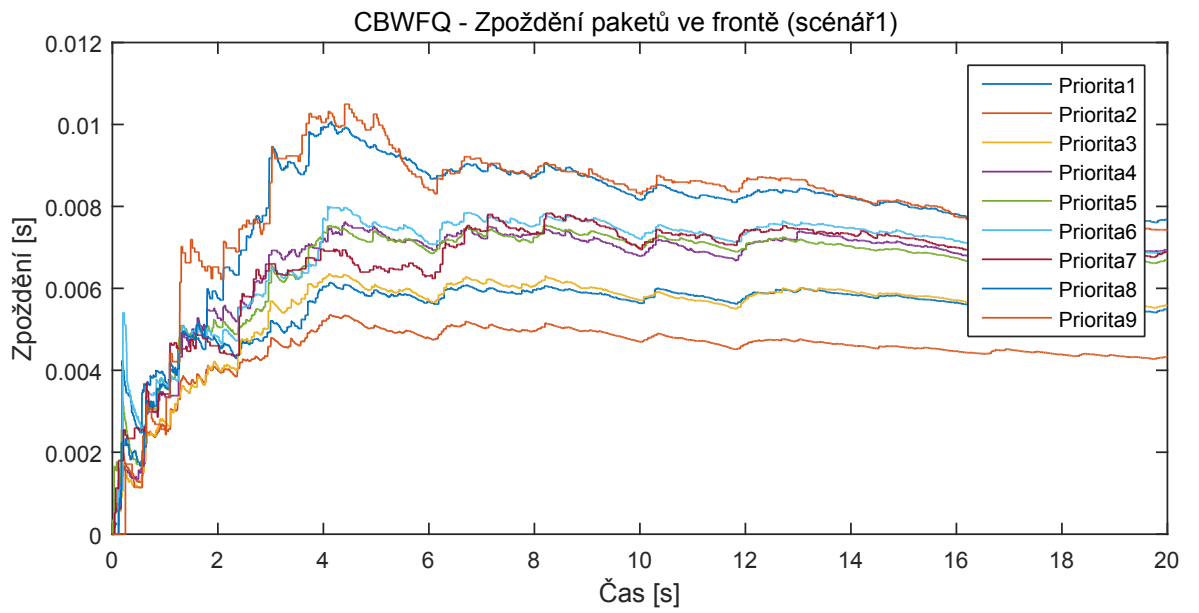
Obr. A.29: WFQ - Ztrátovost paketů na čase - scénář 3



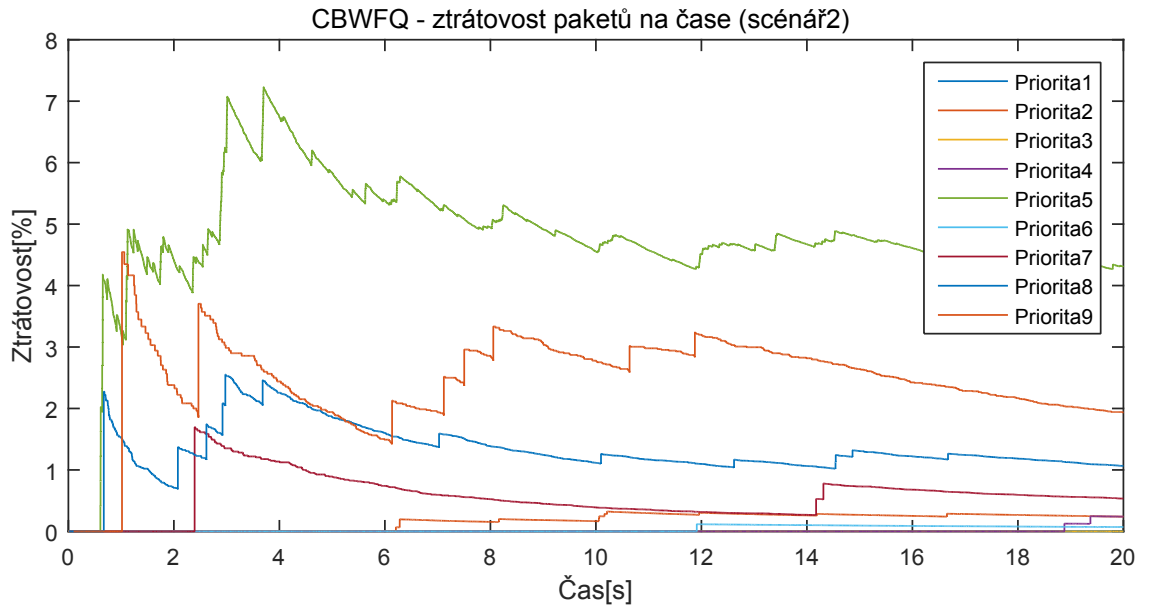
Obr. A.30: WFQ - Zpoždění paketů ve frontě - scénář 3



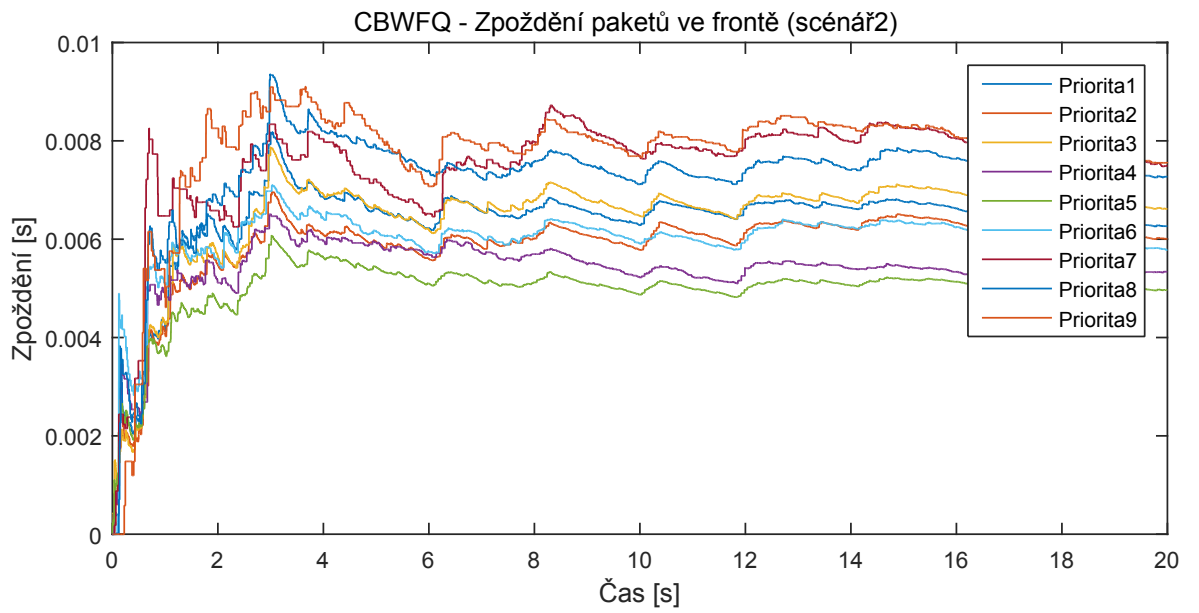
Obr. A.31: CBWFQ - Ztrátovost paketů na čase - scénář 1



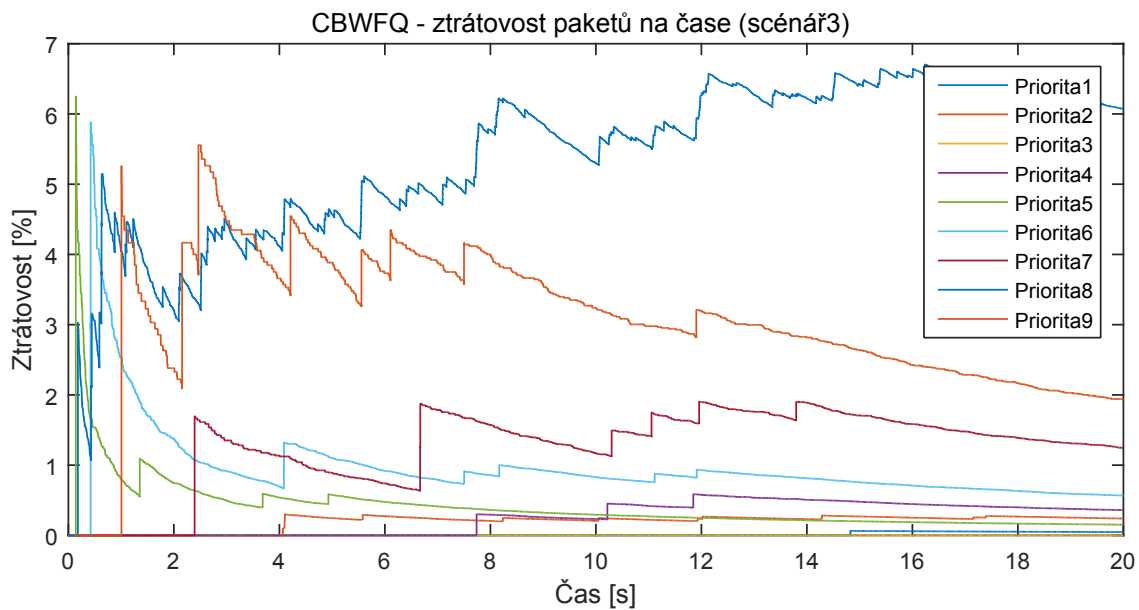
Obr. A.32: CBWFQ - Zpoždění paketů ve frontě - scénář 1



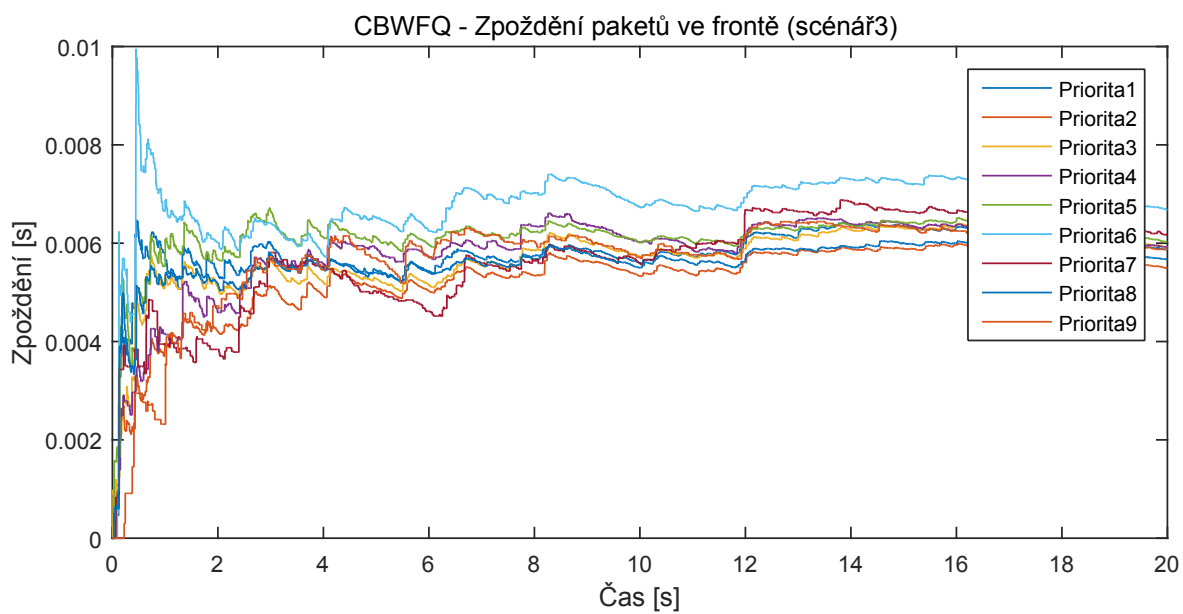
Obr. A.33: CBWFQ - Ztrátovost paketů na čase - scénář 2



Obr. A.34: CBWFQ - Zpoždění paketů ve frontě - scénář 2



Obr. A.35: CBWFQ - Ztrátovost paketů na čase - scénář 3



Obr. A.36: CBWFQ - Zpoždění paketů ve frontě - scénář 3