

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra telekomunikační techniky

**Simulace metod pro zpracování paketů směrovači v
prostředí OMNeT**

květen 2014

**Bakalant:
Vedoucí práce:**

**Lukáš Novotný
Ing. Petr Hampl, Ph.D.**

Čestné prohlášení

Prohlašuji, že jsem zadanou bakalářskou práci zpracoval sám s přispěním vedoucího práce a používal jsem pouze literaturu v práci uvedenou. Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry.

Datum: 25. 5. 2012

.....
podpis bakalanta

Anotace:

Tato bakalářská práce se zabývá metodami pro zpracování paketů na směrovači, zvláště pak metodu Priority Queueing. Pro tuto metodu byly sestaveny dva simulační modely v prostředí OMNeT++. Zmíněnému vývojovému prostředí je také věnována část bakalářské práce. Samotné simulace byly provedeny pro několik scénářů, představující různé zatížení sítě.

Klíčová slova: Priority Queueing, OMNeT++, INET Framework

Summary:

This final project deals with methods for processing packets at the router, particular method of Priority Queueing. For this method were compiled two simulation models in OMNeT++ environment. Part of this project is also paid to this actual environment mentioned above. The actual simulations were performed for several scenarios representing various network load.

Index Terms: Priority Queueing, OMNeT++, INET Framework

Obsah

1.Úvod.....	4
2.Metody uplatňující se v případě přetížení sítě.....	5
2.1.Klasifikace datových toků.....	5
2.2.Jednotlivé metody.....	6
3.Popis a sestavení simulačních modelů.....	8
3.1.Model systému s prioritami.....	8
3.2.Sestavení modelu PQ v základním prostředí OMNeT++.....	10
3.3.Sestavení modelu PQ v INET Framework.....	13
4.Simulace.....	17
4.1.Model obsluhového systému M/M/N/R bez priorit.....	17
4.2.Simulační modely.....	19
4.2.1.Scénář č. 1 pro model PriorityQueueing.....	20
4.2.2.Scénář č. 1 pro model PQinet.....	22
4.2.3.Porovnání jednotlivých scénářů.....	23
5.Závěr.....	29
6.Reference.....	30

Seznam zkratek a symbolů

AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
BE	Best Effort
CBWFQ	Class-Based Weighted Fair Queueing
CDT	Congestive Discard Threshold
CLP	Cell Loss Priority
CoS	Class of Service
CQ	Custom Queueing
DE	Discard Eligible
DSCP	Differentiated Services Code Point
ECN	Explicit Congestion Notification
EF	Expedited Forwarding
FIFO	First-In-First-Out
ISL	Inter-Switch Link
LIFO	Last-In-First-Out
LLQ	Low-Latency Queueing
MAC	Media Access Control
MPLS	Multiprotocol Label Switching
MPLS EXP	Multiprotocol Label Switching experimental bits
MQC	Modular Quality of Service Command line
MTU	Maximum Transmission Unit
NBAR	Network Based Application Recognition
NED	Network Description
PHB	Per-Hop Behavioral
PQ	Priority Queueing
RFC	Request for Comments
RM-OSI	Reference Model Open Systems Interconnection
RR	Round-Robin
RTP	Real-Time Protocol
TCP	Transmission Control Protocol
ToS	Type of Service
TTL	Time To Live
UDP	User Data Protocol
WFQ	Weighted Fair Queueing
WRED	Weighted Random Early Detection
WRR	Weighted Round-Robin

A	nabídka
L	velikost paketu
μ	intenzita obsluhy
t_{os}	doba obsluhy
v_p	přenosová rychlost odchozího kanálu
λ	intenzita příchozích paketů
t_{sim}	doba simulačního úseku
C_p	počet příchozích paketů datového toku
C_y	počet odeslaných paketů datového toku
C_z	počet ztracených paketů datového toku
C_w	počet paketů, které skutečně čekaly

B	pravděpodobnost ztráty paketu v datovém toku
$E[W]$	střední hodnota doby čekání paketu datového toku vztažená k paketům, které byly systémem odeslány
$E[W_W]$	střední hodnota doby čekání paketu datového toku vztažená k paketům, které v systému skutečně čekaly
$E[W_0]$	střední hodnota doby čekání paketu datového toku vztažená k paketům, které byly systému nabízeny
T_w	celková doba čekání paketů datového toku
P_j	pravděpodobnost j -tého stavu systému
P_0	pravděpodobnost nultého stavu systému
N	počet obsluhových linek systému
R	počet stavů systému
$P[W > 0]$	pravděpodobnost čekání paketu v paměti

Seznam tabulek

Tab. 1: Vstupní parametry pro srovnávací model.....	18
Tab. 2: Počty paketů, výsledky z analytického a simulačního modelu.....	19
Tab. 3: Doby čekání, výsledky z analytického a simulačního modelu.....	19
Tab. 4: Vstupní parametry I pro Scénář č. 1.....	20
Tab. 5: Vstupní parametry II pro Scénář č.1.....	20
Tab. 6: Hodnoty počtů paketů získané z modelu PriorityQueueing pro scénář č. 1.....	21
Tab. 7: Hodnoty dob čekání získané z modelu PriorityQueueing pro scénář č. 1.....	21
Tab. 8: Hodnoty dob čekání získané z modelu PQinet pro scénář č. 1.....	22
Tab. 9: Hodnoty dob čekání získané z modelu PQinet pro scénář č. 1.....	23

Seznam obrázků

Obr. 3.1: Priority Queueing.....	9
Obr. 3.2: PriorityQueueing topologie.....	11
Obr. 3.3: Úprava zdrojového kódu modulu Server.....	12
Obr. 3.4: Zdrojový kód topologie PriorityQueueing.....	13
Obr. 3.5: PQinet topologie.....	14
Obr. 3.6: PQqueue topologie.....	17
Obr. 4.1: Ztracené, přijaté, a odeslané pakety 1. datového toku v závislosti na nabídce.....	24
Obr. 4.2: Ztracené, přijaté, a odeslané pakety 2. datového toku v závislosti na nabídce.....	24
Obr. 4.3: Ztracené, přijaté, a odeslané pakety 3. datového toku v závislosti na nabídce.....	25
Obr. 4.4: Ztracené, přijaté, a odeslané pakety 4. datového toku v závislosti na nabídce.....	25
Obr. 4.5: Doby čekání 1. datového toku.....	26
Obr. 4.6: Doby čekání 2. datového toku.....	27
Obr. 4.7: Doby čekání 3. datového toku.....	27
Obr. 4.8: Doby čekání 4. datového toku.....	28

1. Úvod

Tato práce se zabývá simulacemi metod pro zpracování paketů na směrovači. Následující kapitoly pojednávají jak o klasifikaci a označení paketů datových toků pro zpracování těmito metodami, tak o metodách samotných. Z těchto je pak vybrána metoda Priority Queueing, která je následně podrobována bližšímu zkoumání. Je zde zahrnut analytický model pro tuto metodu z hlediska obsluhových systémů. Následně je model Priority Queueing podroben různým simulačním scénářům. Simulační modely byly sestaveny v prostředí OMNeT++. Modely jsou dva. První využívá základních modulů prostředí a má blíže právě k obsluhovým systémům. Druhý využívá rozšíření INET Framework, což je soubor modulů, vytvořených speciálně pro simulace paketově orientovaných sítí. Menší část této práce je také věnována zacházení s prostředím OMNeT++. Je zde uvedeno jeho uplatnění, základní struktura a dále poskytuje náhled na sestavení simulačních modelů z modulů, které OMNeT++ poskytuje. Pro použité moduly jsou zde popsány jejich parametry a způsob propojení. Ukázka zdrojového kódu popisuje syntaxi jazyka NED, jenž byl speciálně vyvinut pro toto simulační prostředí. Krátce je zde také zmíněna konfigurace simulací. Větší část práce je věnována statistickému zpracování výsledků získaných ze simulačních modelů pro stanovené scénáře. Tyto scénáře mají představovat zatížení sítě, které se postupně zvyšuje. Za tohoto trendu se výsledky mění a jsou porovnávány pomocí příslušných grafů. Cílem této práce bylo především získat základní zkušenost s komplexním simulačním nástrojem OMNeT++ a zpracování dané problematiky v již zmíněném simulačním prostředí.

2. Metody uplatňující se v případě přetížení sítě

V této kapitole jsou uvedeny konkrétní metody řazení paketů na síťových prvcích, které se snaží vyhnout ztrátám přenášené informace. Patří k nim systémy First-In-First-Out (FIFO), Priority Queueing (PQ), Round-Robin Queueing (RR), Weighted Round-Robin Queueing (WRR), dále pak Weighted Fair Queueing (WFQ), Class-Based Weighted Fair Queueing (CBWFQ) a Low-Latency Queueing (LLQ) [1].

2.1. Klasifikace datových toků

Aby mohla být na datový provoz uplatněna některá z výše uvedených metod, musí být nejprve klasifikován a označen. Jednotlivým paketům určitého datového toku je tedy přidělen identifikátor. Klasifikace a značkování se provádí na okrajových prvcích Diffserv (DS) domény. Uvnitř sítě už se rozhoduje pouze podle těchto značek, což snižuje celkové zpoždění přenosu. Klasifikace se provádí pomocí access listu, NBAR nástroje pro třídění dat nebo jiných kritérií popisující datový provoz. Mezi tato kritéria patří:

- příchozí nebo odchozí rozhraní,
- CoS hodnota v záhlaví ISL nebo 802.1Q/P rámci,
- zdrojová nebo cílová IP adresa,
- IP precedence nebo hodnota pole DSCP v záhlaví paketu,
- hodnota pole EXP v MPLS záhlaví nebo typ aplikace.

NBAR (*Network Based Application Recognition*) je nástroj, který klasifikuje datový provoz na prvních čtyřech vrstvách RM-OSI, kdy zkoumá obsah segmentu (TCP nebo UDP). Nevýhodou této metody klasifikace je její hardwarová náročnost. Označování paketů se může provádět jak na spojové, tak síťové vrstvě RM-OSI. Označování na třetí vrstvě má tu výhodu, že záhlaví paketu se při přenosu sítě nemění (až na TTL, checksum a pole informující o fragmentaci) a to umožňuje zachovat označení paketu po celou dobu přenosu. Značkování si však nemohou přečíst prvky pracující na spojové vrstvě (jakými jsou např. přepínače). Tuto nevýhodu nemá označování na druhé vrstvě. Při průchodu síťovým prvkem pracujícím na třetí vrstvě RM-OSI (jakým je například směrovač) je však původní záhlaví druhé vrstvy nahrazeno novým. Níže jsou uvedeny druhy značkování:

- CoS (*Class of Service*) pole v záhlaví ethernetového rámce s rozšířením. Čtyřbitové 802.1Q/P záhlaví, je vloženo za zdrojovou MAC adresu. Toto záhlaví obsahuje tříbitové pole (*PRI*) známé jak CoS a umožňuje nastavit osm různých hodnot.
- DE (*Discard Eligible*) pole v záhlaví Frame Relay a CLP (*Cell Loss Priority*) v záhlaví ATM buňky. Obě metody představují jeden bit v záhlaví rámce nebo buňky. Nabývají dvou hodnot a to 0 pro přenos rámce a 1 pro zahození rámce v případě přehlcení sítě. Tyto metody nejsou kompatibilní CoS.
- MPLS (*Multiprotocol Label Switching*) EXP (*experimental bits*) pole. MPLS pakety představují IP pakety s čtyřbitovou MPLS hlavičkou. Technologie MPLS je používána v páteřních sítích pro rychlé směrování paketů. Záhlaví MPLS obsahuje tříbitové pole EXP určené k označení paketu. U této metody existuje možnost vzájemného mapování značení v rámci technologií CoS a IP precedence.
- DSCP (*Differentiated Services Code Point*) a IP precedence. Záhlaví protokolu IP obsahuje osmibitové pole ToS (*Type of Service*). Tři z těchto bitů náleží hodnotě IP

precedence. Čím vyšší je tato hodnota, tím vyšší má paket šanci na odeslání. RFC 2474 z roku 1998 mění význam bitů pole ToS a zavádí pole DiffServ (*Differentiated Services*). Z tohoto pole je šest nejvýznamnějších bitů (most significant bits) nazýváno DSCP. Právě tyto bity jsou použity k označení paketu. Zbývající dva nejméně významné bity (*least significant bits*) pole DiffServ jsou použity pro ovládání provozu a jsou nazývány ECN (*Explicit Congestion Notification*). DSCP je zpětně kompatibilní s IP Precedence. Současné hodnoty DSCP zahrnují čtyři definice PHB (*per-hop behavior*):

- *Class Selector PHB* zajišťuje kompatibilitu s IP Precedence tím, že nastaví tři nejméně významné bity DSCP na nuly a tři nejvýznamnější bity korespondují s hodnotou IP Precedence. Když je tedy paket poslán ze zařízení podporující DSCP na zařízení podporující IP Precedence, jsou vyhodnoceny pouze tři nejdůležitější bity.
- *Default PHB* s nastavením tří nejvýznamnějších bitů na nuly používá službu *best effort* (BE). Pokud není hodnota DSCP namapována na směrovači, je paket přiřazen do *default PHB*.
- *Assured forwarding PHB* (AF) nastavuje tři nejvýznamnější bity DSCP na hodnoty 001, 010, 011 nebo 100 (také známy jako AF1, AF2, AF3 a AF4). AF PHB zaručuje paketům datového toku určitou pravděpodobnost na odeslání. Standardně zavádí čtyři třídy datového provozu (AFxy). Každá třída má vyhrazenou přenosovou rychlost (kapacitu). Pokud intenzita datového provozu překročí přidělenou kapacitu dané třídy, jsou jeho pakety zahazovány. Třídám AF1, AF2, AF3, AF4 mohou být přiděleny různé přenosové rychlosti. Druhý digit „y“ AF PHB specifikuje pravděpodobnost zahazení paketu. Větší číslo indikuje větší pravděpodobnost zahazení.
- *Expedited forwarding PHB* (EF) vždy nastaví tři nejdůležitější bity DSCP na 101. Celé DSCP pole pak vypadá takto: 101110. EF PHB se používá pro datový tok, u kterého potřebujeme nízkou hodnotu zpoždění.

2.2. Jednotlivé metody

FIFO (First-In-First-Out) je základní metoda řazení paketů pro většinu rozhraní. Nevyžaduje žádnou manuální konfiguraci. Pakety se zařadí do zásobníku v pořadí, ve kterém přišly a kde jakýkoli identifikátor nehraje žádnou roli. V důsledku použití této metody může být vyhrazená přenosová rychlost vyčerpána a méně agresivní provoz (např. hlas) může být zahazen nebo nepříjemně zdržen. Z tohoto důvodu je metoda FIFO nevhodná pro rozhraní s nízkými přenosovými rychlostmi.

PQ (Priority Queueing) má k dispozici čtyři klasifikační třídy (s vysokou, střední, normální a nízkou prioritou). Pakety jsou rozřazeny do jednotlivých tříd podle priority. PQ plánovač pak vybírá pakety k odbavení následujícím způsobem. Vyšší třída má vždy prioritu nad tou nižší. Pokud jsou v paměti třídy s vysokou prioritou stále pakety, třídy s nižší prioritou nemají šanci na odbavení svých paketů. Stejný princip platí i pro vztahy mezi ostatními třídami. Nevýhodou režimu PQ je možnost tzv. Vzniku monopolizace rozhraní. To znamená, že pakety toků s nižšími prioritami jsou zahazovány v důsledku využití veškeré přenosové kapacity toku tříd s vyšší prioritou.

RR (Round-Robin) je podobný PQ. Má několik tříd, do kterých jsou zařazovány pakety. RR plánovač odbaví jeden paket z první třídy, pak jeden z druhé, atd. Poté začne opět

od první a zopakuje celý proces. Žádná třída nemá prioritu, všechny jsou rovnocenné. Pokud jsou pakety všech tříd stejně velké, přenosová rychlost přidělená odchozímu rozhraní je rovnoměrně rozdělena mezi jednotlivé třídy. Jakmile má ale jedna třída pakety větší než ostatní, uzurpuje si větší přenosovou rychlost odchozího rozhraní. V Systému RR sice nenastane možnost neodbavení paketů z nějaké třídy, ale neexistuje zde možnost prioritizace provozu.

WRR (Weighted Round-Robin) je modifikovaná verze RR. Umožňuje jednotlivým třídám přiřadit „váhu“. Každá třída tedy obdrží určitou část přenosové kapacity rozhraní, která se nemusí nezbytně rovnat částem přiděleným ostatním třídám. Speciálním případem WRR je CQ (Custom Queueing). V CQ je jednotlivým třídám přidělen počet bytů, jenž musí být odbaveny dříve, než plánovač přejde k třídě další.

WFQ (*Weighted Fair Queueing*)¹ mezi jehož základní charakteristiky patří rozdělení datového provozu do datových toků, rovnoměrné alokování přenosové rychlosti aktivním tokům, rychlejší odbavování méně náročných toků (low-volume) a přiřazení větší přenosové rychlosti tokům s vyšší prioritou. Klasifikace a plánování probíhá ve WFQ následovně. Příchozí pakety jsou klasifikovány do dynamicky alokovaných tříd, a každé třídě je přiřazena paměť FIFO. Datové toky jsou identifikovány na základě zdrojové IP adresy, cílové IP adresy, čísla protokolu, pole ToS, čísla zdrojového TCP/UDP portu nebo cílového TCP/UDP portu. Podle některého z výše uvedených kritérií je paket zařazen do zásobníku příslušného dynamicky alokované třídy. Pokud dorazí paket, který odpovídá již existujícímu datovému toku, je zařazen do jeho zásobníku (třídy). Jakmile přijde paket, jenž neodpovídá žádnému existujícímu toku, je tento datový tok vytvořen a je mu přiřazen zásobník. Naopak neaktivním datovým tokům jsou zásobníky odebrány. WFQ tedy dynamicky vytváří a maže zásobníky jednotlivých datových toků. Maximální počet zásobníků je závislý na implementaci výrobce. Pokud počet aktivních toků překročí toto číslo, je více datovým tokům přidělen jeden zásobník. Vzhledem k tomu, že tento datový provoz obsahuje zpravidla tisíce datových toků, je tato situace běžným stavem. Metodě WFQ je možné nastavit maximální celkový počet paketů přítomných v systému. Jedná se o sumu veškeré paměti zabrané pakety přítomnými v systému WFQ. Pokud paket dorazí v době, kdy je tento limit naplněn, je zahozen. Tomu se říká *WFQ aggressive dropping*. Existuje jedna výjimka. V případě, že má být paket přiřazen datovému toku s prázdným zásobníkem, vyhne se zahození. Dalším pojmem, který WFQ zavádí, je CDT (*Congestive Discard Threshold*). Parametrem CDT je možné nastavit maximální počet paketů v každé paměti. Pokud není maximální celkový počet paketů přítomných v systému zaplněn, ale limitu CDT bylo dosaženo, je příchozí paket zahozen. Této metodě se říká *WFQ early dropping* a také má jednu výjimku. Pokud má paket v jiném zásobníku vyšší sekvenční číslo než ten příchozí, je zahozen ten s vyšší prioritou. Systém uvažuje tak, že zahozený paket patří agresivnějšímu datovému toku. WFQ má tedy přísnější pravidla zahazování pro agresivnější datové toky. Mezi přednosti WFQ tedy patří: jednoduchá konfigurace, nevyžaduje explicitní klasifikaci, garantuje propustnost každému datovému toku, zahazuje pakety z nejagresivnějších toků a poskytuje rychlejší odbavení paketům z neagresivních toků. Na druhou stranu disponuje tento systém také nevýhodami: klasifikace a plánování nejsou konfigurovatelné a modifikovatelné, WFQ je podporované pouze na pomalých linkách (2,048 Mbit/s a méně), negarantuje přenosovou rychlost a zpoždění datovým tokům, několika datovým tokům může být přiřazena společná paměť.

CBWFQ (*Class-Based Weighted Fair Queueing*) vychází ze systému WFQ, ale na rozdíl od něj umožňuje uživateli definovat vlastní třídu, přiřadit jí zásobník a garantovanou přenosovou rychlost. Třída může použít i větší přenosovou rychlost, pokud je dostupná.

¹ je výchozím systémem na sériových rozhraních Cisco směrovačů o rychlosti 2,048 Mbit/s (E1) a nižší.

CBWFQ nezvýhodňuje interaktivní aplikace s nároky na nízké zpoždění (neexistuje striktně prioritní třída) jakou je například VoIP. Systém CBWFQ může vytvořit až 64 pamětí (FIFO), jeden pro každou třídu. Každý zásobník má kromě garantované přenosové rychlosti také maximální limit paketů, které mohou v daném zásobníku čekat. Pokud je limit překročen, dojde k zahazování od konce (tail drop). Mezi přednostmi CBWFQ patří: dovoluje uživateli definovat vlastní třídy pomocí *MQC class map (modular command line interface)*, alokaci přenosové rychlosti pro každou třídu založenou na uživatelem definovaných pravidlech. K jeho nevýhodám můžeme jmenovat: nemožnost nastavení zásobníku vhodného pro interaktivní aplikace jako VoIP, neumí garantovat malé zpoždění provozu.

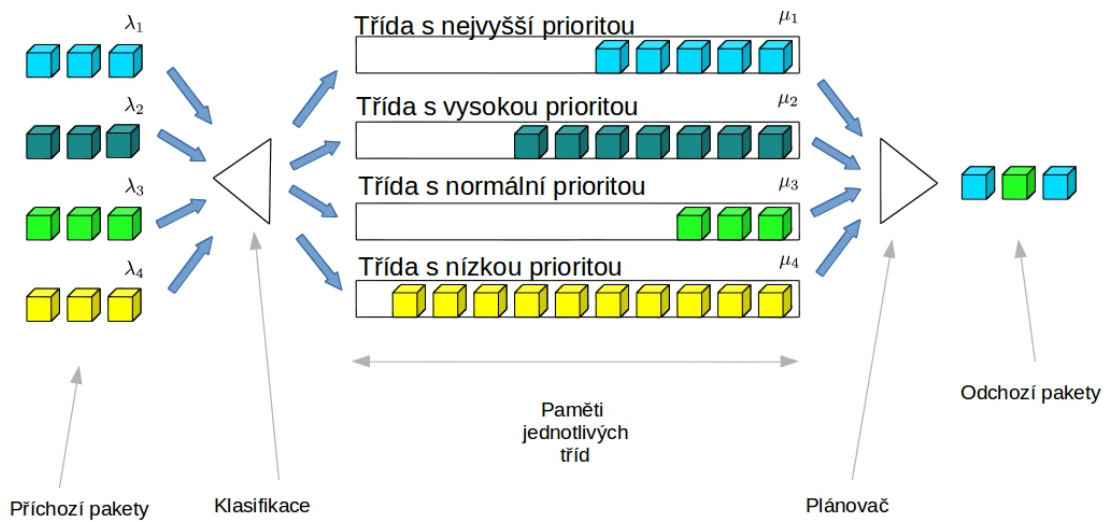
LLQ (Low-Latency Queueing) umí na rozdíl od WFQ a CBWFQ garantovat vedle přenosové rychlosti pro danou třídu také malé zpoždění provozu vybraných aplikací jako je VoIP. Toto je možné, protože LLQ zavádí prioritní třídu, jež má přednost před ostatními. Na rozdíl od PQ si ale tato prioritní třída nemůže uzurpovat celou přenosovou rychlost odchozího rozhraní. Prioritní třída má v modelu LLQ rezervovanou část z celkové kapacity rozhraní a nemůže přenášet víc dat, než tato kapacita dovolí. Pokud dorazí více paketů, než je prioritní třída schopná odbavit, jsou tyto zařazeny do zásobníku. Pokud je zásobník plný, jsou pakety zahazovány. Díky těmto omezením mají (na rozdíl od PQ) i další třídy vždy možnost odbavení svých paketů. Do paměti prioritní třídy je možné umístit více datových tříd. Je možné jich mít několik a zpravidla se používá systém LLQ, který do paměti prioritní třídy umísťuje dvě. Jedna přenáší hlas a druhá obraz. Prioritní třída v LLQ nemusí (opět na rozdíl od PQ) akceptovat pouze provoz standardizovaný protokolem RTP (*Real-time Transport Protocol*). Uživatel si může sám definovat jaký datový provoz této třídě přiřadí. V případě, že do systému přijde datový provoz, který není explicitně klasifikován, je přiřazen do výchozí třídy (*default class*). Přednosti systému LLQ jsou následující: LLQ zahrnuje výhody CBWFQ včetně možnosti vytvoření uživatelem definovaných tříd a přiřazení a garanci jejich přenosové rychlosti a PQ, kdy je umožňuje existenci prioritní třídy. Na každou třídu kromě prioritních může být aplikován systém WRED (*Weighted Random Early Detection*), což je metoda pro předcházení zahlcení sítě.

3. Popis a sestavení simulačních modelů

Tato práce se dále zabývá zpracováním metody PQ. Následující kapitola popisuje model systému s prioritami a sestavení jeho modelů v simulačním prostředí OMNeT++. Dále zde bude uveden způsob komunikace mezi moduly a výčet a funkce jejich parametrů. OMNeT++ je nástroj určený pro provádění simulací systémů s diskrétními událostmi. Modely jsou sestaveny jak v základním balíku prostředí OMNeT++, tak i v jeho rozšíření INET Framework. Toto rozšíření se, na rozdíl od základního balíku, zabývá výhradně simulacemi paketových sítí. Pro každý model je založen vlastní projekt, který obsahuje soubory s topologií sítě, popř. složených modelů s příponou *.ned*, konfigurační soubor *omnetpp.ini* a další, které simulaci doplňují nebo jsou v nich uvedeny výstupy simulace, ale pro funkčnost simulace samotné zásadní nejsou. Soubory se simulačními modely jsou umístěny na příloženém CD, stručný popis adresářové struktury s klíčovými soubory je uveden v příloze A

3.1. Model systému s prioritami

Do modelů je třeba zadat množství parametrů, ty se vypočítají dle následujících vztahů. Jako vzor je použit systém $M/M/N/R$ s prioritami [8]. První M reprezentuje exponenciální rozložení intervalů mezi příchody, druhé M reprezentuje exponenciálně rozloženou dobu obsluhy, N je počet odchozích linek a R je kapacita paměti jednotlivých tříd. Na obrázku níže se nachází schématický náčrt zpracovávaného modelu PQ.



Obr. 3.1: Priority Queueing

Doba obsluhy je čas, po který bude odchozí vedení přenášet paket do sítě. Jakmile tato doba uplyne, může systém přejít k odbavení dalšího paketu. Střední hodnota doby obsluhy t_{os_i} paketu i -tého vstupního toku se vypočítá podle rovnice:

$$t_{os_i} = \frac{1}{\mu_i} \quad (3.1)$$

kde μ_i je intenzita obsluhy i -tého vstupního toku.

Intenzita obsluhy i -tého vstupního toku je definována jako průměrný počet požadavků, které je kanál schopen obsloužit za jednotku času. Tato veličina se vypočítá ze vzorce:

$$\mu_i = \frac{v_p}{L_i}, \quad (3.2)$$

kde v_p je přenosová rychlost odchozího kanálu a hodnota L_i představuje velikost paketu v i -tém vstupním datovém toku.

Intervaly mezi příchody jednotlivých paketů i -tého vstupního toku t_{p_i} jsou exponenciálně rozloženy se střední hodnotou:

$$t_{p_i} = \frac{1}{\lambda_i}, \quad (3.3)$$

kde λ_i je intenzita příchozích paketů i -tého vstupního toku.

Nabízené zatížení je definováno jako je průměrný počet paketů přicházejících do systému během střední doby obsazení t_{os_i} :

$$A_i = \frac{\lambda_i}{\mu_i} = \lambda t_{os_i}, \quad (3.4)$$

Průměrný počet paketů C_{p_i} i -tého datového toku, které byly systému nabízeny během délky simulačního úseku t_{sim} :

$$C_{p_i} = \lambda_i t_{sim}, \quad (3.5)$$

Průměrný počet paketů C_{y_i} i -tého datového toku, které byly systémem odeslány během doby pozorování simulace t_{sim} :

$$C_{y_i} = \lambda_i (1 - B_i) t_{sim}, \quad (3.6)$$

kde B_i je pravděpodobnost ztráty paketu v i -té třídě.

Průměrný počet paketů C_{z_i} i -tého datového toku, které byly systémem zahozeny během doby pozorování simulace t_{sim} :

$$C_{z_i} = \lambda_i B_i t_{sim} \quad (3.7)$$

Pravděpodobnost ztráty paketu B_i v i -té třídě:

$$B_i = \frac{C_{z_i}}{C_{p_i}}, \quad (3.8)$$

Existují tři definice střední hodnoty doby čekání. Střední hodnota doby čekání paketů i -tého datového toku vztažená k paketům, které systémem odeslány $E[W_i]$:

$$E[W_i] = \frac{T_{w_i}}{C_{y_i}}, \quad (3.9)$$

Střední hodnota doby čekání paketů i -tého datového toku vztažená k paketům, které skutečně čekaly $E[W_{W_i}]$:

$$E[W_{W_i}] = \frac{T_{w_i}}{C_{w_i}}, \quad (3.10)$$

kde C_{w_i} [-] je počet paketů i -tého datového toku, které skutečně čekaly.

Střední hodnota doby čekání paketů i -tého datového toku vztažená ke všem paketům, které byly systému nabízeny $E[W_{0_i}]$:

$$E[W_{0_i}] = \frac{T_{w_i}}{C_{p_i}}, \quad (3.11)$$

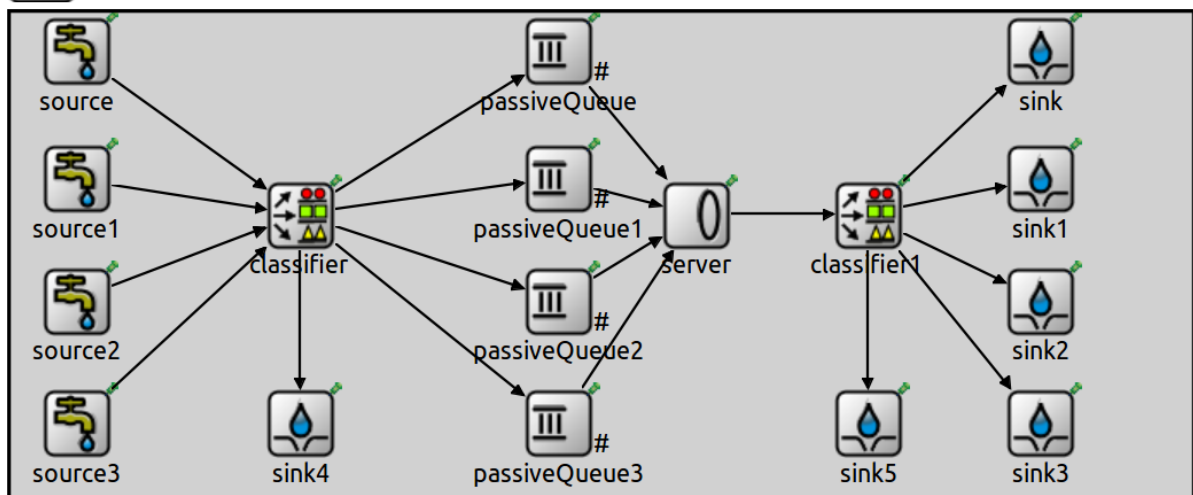
Celková doba čekání všech paketů i -té třídy T_{w_i} se vypočítá jako:

$$T_{w_i} = \sum_{j=1}^{C_{w_i}} t_{w_{ij}}, \quad (3.12)$$

kde $t_{w_{ij}}$ je doba čekání j -tého paketu i -té třídy.

3.2. Sestavení modelu PQ v základním prostředí OMNeT++

Projekt s tímto modelem je nazván *PriorityQueueing*, který využívá moduly z vestavěného projektu *Queueinglib* [3]. Nachází se v něm soubor *PriorityQueueing.ned*, ve kterém je definována topologie simulace. Soubory s příponou *.ned* nabízejí otevření ve dvou perspektivách a to *Design* anebo *Source*. Náhled na model z perspektivy *Design*, který je grafický se nachází níže (viz. Obr. 3.2). Skládá se ze čtyř modulů *Source*, jež generují datový provoz pro každou prioritní třídu. Moduly *Classifier* klasifikují datový provoz a rozřazují ho do jednotlivých tříd. Moduly *PassiveQueue* představují zásobníky jednotlivých prioritních tříd. Modul *Server* je plánovač, který určuje jaká třída bude využívat odchozí rozhraní. Moduly *Sink* slouží jako cílová destinace pro datový provoz.



Obr. 3.2: PriorityQueueing topologie

Modul *Source* je v modelu PQ využíván jako generátor paketů. Generuje zprávy (*jobs*), které procházejí systémem. Mezi jeho parametry patří počet vygenerovaných zpráv (*numJobs*), čas začátku vysílání (*startTime*), čas ukončení vysílání (*stopTime*), časový interval mezi generovanými zprávami (*interArrivalTime*), jméno zprávy (*jobName*), typ zprávy (*jobType*) a priorita zprávy (*jobPriority*). Model PQ využívá parametry *numJobs*, *interArrivalTime* a *jobType*. Výchozí hodnota začátku vysílání je totiž hodnota parametru *interArrivalTime* a čas ukončení vysílání hodnotou *-1s*. Záporná hodnota v jazyce NED znamená nekonečno. Vysílání se tedy ukončí, když modul odešle specifikovaný počet zpráv. Parametr *jobType* určuje kritérium klasifikace zprávy. Modul má pouze jednu bránu a to odchozí. Moduly *Source* jsou v modelu PQ právě čtyři, každý pro jednu prioritní třídu. Všechny moduly *Source* jsou napojeny na modul *Classifier*.

Modul *Classifier* je v modelu PQ využíván pro klasifikaci zpráv. Ty prochází modulem podle pravidla: index vstupní brány je stejný, jako index výstupní. Má jediný parametr a tím je kritérium klasifikace (*dispatchField*). Má dvě hodnoty a to *type* a *priority*. Jelikož v modulu *Source* je pro klasifikaci použit parametr *jobType*, nemusíme parametr *dispatchField* modifikovat. *Type* je výchozí volba. Vstupní brány tohoto modulu jsou napojeny na moduly *Source* a výchozí na moduly *PassiveQueue*. Má ještě jednu bránu a tou je *rest*. Ta je užita v případě, že na *Classifier* přijde zpráva, kterou modul neumí klasifikovat (stává se při větším počtu vstupů než výstupů). Tato brána je napojena na modul *Sink*.

Modul *PassiveQueue* je v modelu PQ využíván jako paměť. Pakety zde čekají, dokud nejsou odbaveny nebo zahozeny. To se děje v případě příchodu zprávy do fronty, která je maximálně zaplněná. Má tři nastavitelné parametry. Těmi jsou kapacita paměti (*capacity*), režim fronty *fifo* (*fifo*) a způsob posílání zpráv (*sendingAlgorithm*). Výchozí hodnota parametru *fifo* je *true*. To znamená, že se modul bude chovat jako FIFO (*first-in-first-out*) a nikoli jako LIFO (*last-in-first-out*). Způsob posílání zpráv má také na výběr několik voleb. Tou výchozí je *priority*, kterou budeme využívat i v modelu PQ. Dále jsou zde parametry *@signal*, které říkají, že se má zaznamenat nějaká událost a *@statistic*, jenž přiřazují těmto událostem atributy (např. název, jednotky,...). Moduly *PassiveQueue* jsou, jako v případě modulů *Source*, právě čtyři, každý pro jednu prioritní třídu. Příchozí brány jsou napojeny na modul *Classifier*, odchozí na modul *Server*.

Modul *Server* je v modelu PQ využíván pro výběr a přeposílání zpráv z modulů *PassiveQueue* do modulu *Classifier*. Zde jsou důležité parametry způsob výběru zpráv (*fetchingAlgorithm*), který je opět pomocí prefixu *default* nastaven na volbu priority a doba obsluhy (*serviceTime*). Doba obsluhy je čas, který je třeba pro odbavení jedné zprávy. Tento modul disponuje vstupní bránou, na kterou může být napojeno více modulů *PassiveQueue* a jednou výstupní bránou na napojenou na druhý modul *Classifier*. Původní modul *Server* pro simulování modelu PQ nevyhovoval, neboť poskytoval možnost konfigurace jen jedné doby obsluhy pro všechny druhy paketů. Jeho zdrojový kód musel tedy být upraven. Byla do něj přidána sekce, umožňující přiřadit dobu obsluhy jednotlivým druhům paketů.

```

simtime_t serviceTime;
Job *job = check_and_cast<Job *>(msg);
int inGateIndex = -1;
inGateIndex = job->getKind();
EV << "inGateIndex = " << inGateIndex << endl;
if (inGateIndex==0) {serviceTime = par("serviceTime0"); }
else if (inGateIndex==1) {serviceTime = par("serviceTime1");}
else if (inGateIndex==2) {serviceTime = par("serviceTime2");}
else if (inGateIndex==3) {serviceTime = par("serviceTime3");}
else {serviceTime = par("serviceTime0");}

```

Obr. 3.3: Úprava zdrojového kódu modulu *Server*

Modul *Sink* je v modelu PQ využíván pro zpracovávání statistik a zachytávání zpráv, které prošly systémem. Má jeden nastavitelný parametr a tím je, zda-li si došlou zprávu ponechat nebo zahodit. Výchozí hodnotou je *false*, čili zahodit. Mezi jeho brány patří pouze pole vstupních bran napojené na modul *Classifier*.

Náhled na modul z perspektivy *Source*, který je textový se nachází na Obr. 3.4. Tato perspektiva popisuje topologii detailněji, je však nutné rozumět syntaxi jazyka NED. Začátek zdrojového kódu ukazuje import modulů, které budou potřeba k sestavení modelu. Import se neprovádí, pokud se zdrojové soubory (s příponou *.cc*, *.ned*, popř. *.h*) nacházejí uvnitř projektu, jenž je používá. Za klíčovým slovem *network* následuje jméno modelu (v tomto případě *PriorityQueueing*) a ve složených závorkách jeho topologie. Za klíčovým slovem *parameters* jsou uvedeny parametry modelu. V tomto případě je tato sekce ochuzena, jelikož parametry použité v modelu jsou definovány v importovaných modulech, které jsou uvedeny výše. Je zde pouze parametr *@display*, jenž definuje podobu a pozici ikony objevující se v perspektivě *Design* (viz. Obr. 3.2 vlevo nahoře). V sekci, kterou uvozuje další klíčové slovo *types* jsou definovány typy použitých přenosových kanálů. Tato sekce je opět prázdná, protože model *PriorityQueueing* používá pouze nativní *IdealChannel*, jenž byl importován na začátku zdrojového kódu. Za klíčovým slovem *submodules* následují vlastnosti a parametry modulů. Nejprve je uvedeno jméno modulu v rámci modelu a pak jeho typ. Ve složených závorkách pak následují parametry modulu. V tomto případě je opět jediným parametrem pouze podoba a pozice modulu v perspektivě *Design* (viz. Obr. 3.2). Ve výše uvedeném náhledu na perspektivu *Source* je tato sekce zkrácena, neboť syntax pro zbývající moduly vypadá podobně. Poslední sekce, která je za klíčovým slovem *connections*, definuje propojení jednotlivých modulů následujícím způsobem. Nejprve je uvedeno jméno modulu, ze kterého spojení vychází, a za tečkou jeho brána (*gate*). Ta může být příchozí (*in*) nebo odchozí (*out*). Pomocí šipek je uveden typ spojení. Jednosměrná šipka představuje jednosměrný spoj. Mimo ní existuje i obousměrný, který se definuje pomocí obousměrných šipek. Za typem spoje je uvedeno jméno cílového modulu a za tečkou jeho brána. Operátor „++“ inkrementuje počet bran, pokud je brána polem.


```

import ned.IdealChannel;
import org.omnetpp.queueing.Classifier;
import org.omnetpp.queueing.PassiveQueue;
import org.omnetpp.queueing.Server;
import org.omnetpp.queueing.Source;
import org.omnetpp.queueing.Sink;

network PriorityQueueing
{
    parameters:

        @display("bgb=687,282");
    types:
    submodules:
        source: Source {
            @display("p=39,23");
        }
    connections:
        source.out --> classifier.in++;
        classifier.out++ --> passiveQueue.in++;
        classifier.out++ --> passiveQueue1.in++;
        classifier.out++ --> passiveQueue2.in++;
        classifier.out++ --> passiveQueue3.in++;
        source1.out --> classifier.in++;
        source2.out --> classifier.in++;
        source3.out --> classifier.in++;
        classifier1.rest --> sink5.in++;
        classifier.rest --> sink4.in++;
        classifier1.out++ --> sink.in++;
        classifier1.out++ --> sink1.in++;
        classifier1.out++ --> sink2.in++;
        classifier1.out++ --> sink3.in++;
        server.out --> classifier1.in++;
        passiveQueue.out++ --> server.in++;
        passiveQueue1.out++ --> server.in++;
        passiveQueue2.out++ --> server.in++;
        passiveQueue3.out++ --> server.in++;
}

```

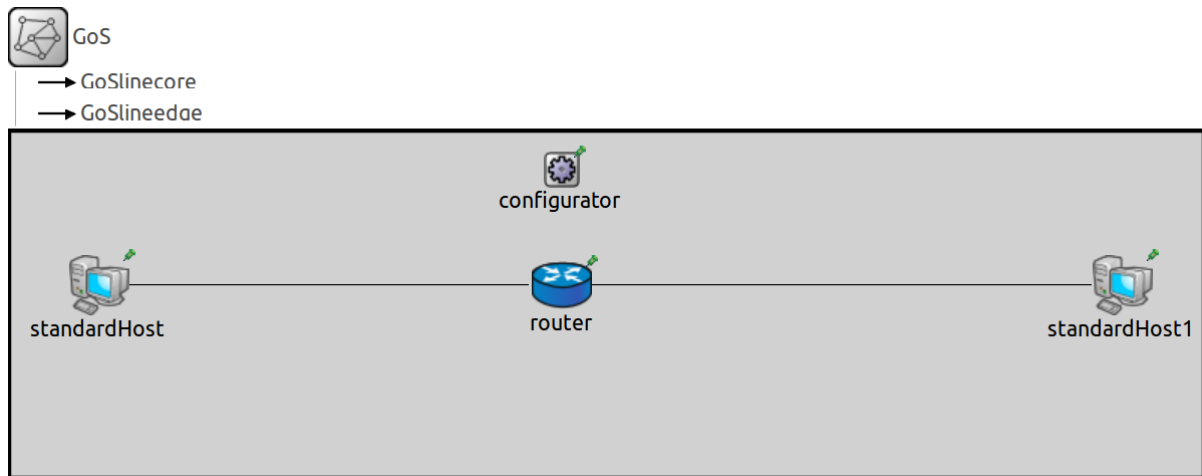
Obr. 3.4: Zdrojový kód topologie *PriorityQueueing*

Samotné hodnoty parametrů mohou být uvedeny přímo v samotných souborech s příponou *.ned* patřících jednotlivým modulům. Tento způsob se nepoužívá. V souborech s příponou *.ned* je nanejvýš uvedena hodnota parametru s klíčovým slovem *default*, které značí deklaraci výchozí hodnoty. Tuto je ale možno přepsat v konfiguračním souboru. Hodnoty parametrů jsou tedy modelům přiřazovány především v konfiguračním souboru *omnetpp.ini*. Jméno souboru není pevně dáno, je však, až na výjimky, pravidlem pojmenovat konfigurační soubory názvem uvedeným výše. Přiřazování parametrů v konfiguračním souboru má tu výhodu, že jsou v něm nastavitelné hodnoty parametrů všech modulů, kterými daný model disponuje.

3.3. Sestavení modelu PQ v INET Framework

Pro tento model je vytvořen projekt z názvem *PQinet*, který využívá moduly projektu *inet* [5] Uvnitř tohoto projektu se nachází soubor *PQinet.ned*. Jako v předchozím modelu (popisovaném v kapitole 3.2) obsahuje i tento soubor topologii simulace. Náhled na modul z perspektivy *Design* se nachází na Obr. 3.5. Na rozdíl od předchozího modelu sestavovaném v základním balíku vývojového prostředí OMNeT++, kde jsou použity výhradně jednoduché

moduly, je model z INET sestaven ze složených modulů [6]. Přijímací a vysílací stanice jsou tvořeny moduly *StandardHost* a směrovač pak modulem *Router*. Tuto trojici doplňuje ještě modul *IPv4NetworkConfigurator* s názvem *configurator*, jenž počítá a ukládá IP adresy jednotlivým síťovým rozhraním.



Obr. 3.5: *PQinet* topologie

Modul *StandardHost* je složený modul, který může generovat a přijímat datový provoz. Generátorem je v modelu *PQinet* modul *standardHost*, přijímačem pak modul *standardHost1*. Složený modul *standardHost* se skládá z řady jednoduchých modulů. Tyto představují transportní, síťovou a spojovou vrstvu RM-OSI modelu. Čtyři vyšší vrstvy tvoří moduly se jmény *tcpApp*, *udpApp*, *sctpApp*, *tcp*, *udp*, *sctp* a *pingApp*. Modul *tcpApp* je typu *ITCPApp*, což je rozhraní pro modul, kterému se musí v konfiguračním souboru přiřadit počet modulů a jejich typ. Typ modulu přiřadíme z adresáře */inet/src/applications/tcpapp*. Modulů generující datový provoz se v tomto adresáři nachází celá řada. Protokol TCP není v tomto modelu využíván, proto zůstane toto rozhraní nedefinované. Modul *udpApp* je typu *IUDPApp*. Jako v předchozím případě se jedná o rozhraní pro modul se stejnými vlastnostmi jako modul *tcpApp*. Typy modulů najdeme nyní v adresáři */inet/src/applications/udpapp*. Pro model *PQinet* jsou z pohledu řešené problematiky vhodné moduly *UDPBasicApp* a *UDPSink*. *UDPBasicApp* je základní modul generující a přijímající datový provoz na základě protokolu UDP. Mezi jeho parametry patří:

- *localPort*, což je místní port daného udp modulu. Jedná se o jedinečný identifikátor každého *UDPBasicApp* modulu. Jeho výchozí hodnotou je „-1“, což znamená, že modul má použít pseudonáhodně zvolený port.
- *destPort* je port modulu z transportní vrstvy, který má generovaný datový provoz zachytit.
- *destAddresses*, který určuje cíl datového provozu. Může být vyjádřen seznamem ip adres nebo přímo jménem cílového modulu. Cílový modul musí být v tomto případě viditelný v topologii sítě (např. modul *standardHost1*). Výchozí hodnotou je prázdný řetězec, což znamená vyřazení modulu z činnosti.
- *messageLength*, který definuje délku zprávy (segmentu). Tento parametr se udává v bytech.
- *sendInterval*, jehož hodnota udává intervaly, ve kterých se posílají zprávy. Udává se v sekundách.
- *startTime*, jenž specifikuje čas v sekundách, ve kterém se odešle první zpráva. Výchozí hodnotou je hodnota parametru *sendInterval*.

- *timeToLive* má stejnou funkci jako pole TTL v záhlaví paketu. Jedná se o parametr, který určuje kolik hopů mezi síťovými prvky může daná zpráva provést. V modelu *PQinet* je tento parametr irelevantní, proto mu ponechává jeho výchozí hodnotu, kterou je „-1“. Tato hodnota pro paket znamená nekonečnou dobu života.
- *typeOfService*, který symbolizuje pole *ToS* v záhlaví paketu. Charakterizuje tedy typ služby. I tento parametr je pro model *PQinet* irelevantní. Výchozí hodnota „-1“ udává, že model nemá tento parametr brát v potaz.
- *multicastInterface*, jehož hodnotou je název odchozího rozhraní, které generuje multicastový provoz. Ani tento parametr model *PQinet* nepoužívá. Funkce, jež parametr definuje, je vyřazena jeho výchozí hodnotou a tím je prázdný řetězec.
- *receiveBroadcast* udává, jestli má modul přijímat broadcastový datový provoz. Hodnotami tohoto parametru jsou *true* nebo *false*. Model *PQinet* tento parametr ke své funkci nepotřebuje a proto mu ponechává výchozí hodnotu „false“, jež tuto funkci deaktivuje.
- *joinLocalMulticastGroups* nabývá hodnoty *true* nebo *false* a udává, jestli má daný modul přijímat datový provoz z multicastových skupin nastavených na lokálním rozhraní. Model *PQinet* si vystačí s výchozí hodnotou „false“, neboť tuto funkci nepoužívá.

Modul *UDPSink* je modul pro přijímání datového provozu a má parametr *localPort*, což je místní port daného udp modulu. Jedná se o jedinečný identifikátor každého *UDPSink* modulu. Jeho výchozí hodnotou je „-1“, což znamená, že modul má použít dočasný port.

Modul *sctpApp* je typu *ISCTPApp*, což je opět rozhraní pro modul, kterému se musí v konfiguračním souboru přiřadit počet modulů a jejich typ. Typ modulu přiřadíme z adresáře */inet/src/applications/sctpapp*. Protokol SCTP není v tomto modelu využíván, proto zůstane toto rozhraní stejně jako v případě TCP nedefinované.

Modul *pingApp* je typu *IpingApp*. Toto rozhraní, které má stejné vlastnosti jako tři předchozí, není v modelu *PQinet* používáné. Typy modulů, jež by byly možné mu přiřadit, se nachází v adresáři */inet/src/applications/pingapp*.

Jelikož rozhraní pro moduly *tcpApp*, *udpApp*, *sctpApp* a *pingApp*, jsou pole (tzn. může jich být víc, než jeden), je třeba každému prvku pole přiřadit identifikátor v podobě portu (viz. parametr *localPort*). Přichodící datový provoz je podle těchto portů rozřazován jednotlivým prvkům, v příslušných modulech *tcp*, *udp* a *sctp*, jedinou výjimkou je rozhraní *pingApp*.

Síťovou vrstvu RM-OSI v modulu *StandardHost* reprezentuje složený modul *NetworkLayer*. Tento modul se skládá z modulů *IPv4 (ip)*², *ICMP (icmp)*, *ErrorHandling (errorHandling)*, *ARP (arp)* a *IPv4NodeConfigurator (configurator)* a rozhraní pro modul *IIGMP (igmp)*. Jednoduchý modul *IPv4* implementuje protokol IP verze čtyři. Tento model má několik parametrů, jejich manuální konfiguraci však model *PQinet* nevyžaduje a vyhovuje mu jejich výchozí nastavení. Výchozí hodnotou je „0“.

Modul *ICMP* je v modulu *NetworkLayer* obsažen z důvodu implementace *Internet Controll Message Protocol*. Modul *ErrorHandling* vyřizuje notifikace chyb, které dorazí z ostatních modulů implementujících nějaký protokol. Tento modul pošle zprávu o chybě, ale neřeší ji.

Modul *ARP* implementuje *Address Resolution Protocol*, který získává MAC adresy

² Před závorkou je uveden typ modulu a v závorce jeho jméno.

síťových prvků z jejich IP adres. V modelu PQinet jsou hodnoty parametrů modulu ARP ryze výchozí.

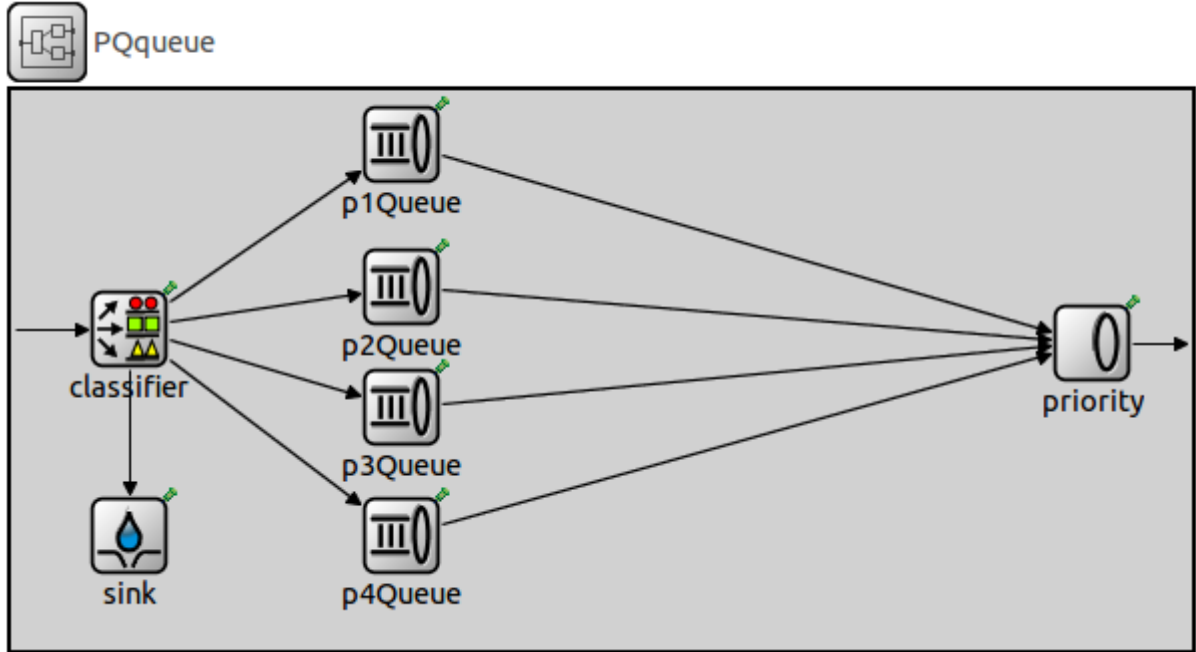
Modul *IPv4NodeConfigurator* konfiguruje IP adresy, směrovací tabulky, atd., které mu předává modul *IPv4NetworkConfigurator*.

Rozhraní pro modul IIGMP. Internet Group Management Protocol rozšiřuje možnosti IP protokolu o multicastové skupiny.

Spojovou vrstvu tvoří rozhraní modulů *eth*, *ppp*, *wlan* a *ext*. Těmto rozhraním se v konfiguračním modulu musí opět přiřadit počet, ale už ne typ. Ten je daný přímo jménem rozhraní (např. *ppp* je *PPPInterface*). Model *PQinet* využívá rozhraní *ppp*.

Zbytek jsou podpůrné moduly. Modul *mobility* zde zastupuje rozhraní pro mobilní moduly (např. *MobileHost*). *NotificationBoard* je modul, díky kterému se mohou ostatní moduly navzájem informovat o událostech, jakými jsou změny v směrovací tabulce, statusy rozhraní, atd.

Složený modul *router* má podobnou stavbu jako modul *standardHost*. Rozdíl mezi nimi je absence modulů představujících čtyři vyšší vrstvy RM-OSI. Právě na tomto modulu se implementuje metoda PQ. Na vstupním rozhraní je zajištěna klasifikace a značkování paketů. Klasifikaci provádí modul *MultiFieldClassifier*. Tento modul má parametr s názvem *filtres*, jehož hodnotou je řetězec s názvem filtru ve formátu xml (v tomto případě *PQinet_filtres.xml*), podle něhož modul klasifikuje datový provoz. Modul *MultiFieldClassifier* je přímo napojen na modul *DSCPMarker*, jenž značuje příchozí pakety, dle hodnot uvedených v parametru *dscps*. Hodnotou tohoto parametru je řetězec značek, které jsou přiřazovány jednotlivým paketům. Přiřazování probíhá tak, že značka uvedená v řetězci jako první je přiřazena paketům, jenž přišly první vstupní branou, značka uvedená v řetězci jako druhá je přiřazena paketům, jenž přišly druhou vstupní branou atd. Takto pakety projdou modulem *router*, kde se na jeho výchozím rozhraní zařadí do některé z front uvedených na obrázku 3.6. Tento systém front je sestaven ve složeném modulu s názvem *PQqueue* a koresponduje s jednoduchými moduly představující systém front v modelu *PriorityQueueing* (viz. Obr. 3.2). V modulu *PQqueue* se nachází modul *BehaviorAggregateClassifier*, který klasifikuje datový provoz podle značky DSCP s prioritou uvedenou v parametru *dscps*. Čtyři moduly *DropTailQueue* s parametrem *framecapacity*, který určuje jejich velikost paměti, představuje jednotlivé prioritní třídy. Modul *Priortity* určuje pořadí odesílání paketů jednotlivých datových toků, přičemž přednost mají pakety jejich prioritní třída je na modul *Priority* napojena branou s nižším indexem. Do modulu *Sink* se odesílá nezařazený datový provoz (ten se v modelu *PQinet* negeneruje).



Obr. 3.6: PQqueue topologie

4. Simulace

Ve dvou následujících podkapitolách jsou výše uvedené simulační modely podrobovány různým scénářům a jsou zde zpracovány z nich získané statistické výsledky. Nejprve je provedeno srovnání simulačního a analytického modelu a pak následují scénáře představující různé zatížení sítě.

4.1. Model obsluhového systému M/M/N/R bez priorit

Následující vztahy popisují analytické řešení systému $M/M/N/R$ bez priorit [7], jenž byl užit pro ověření správné činnosti simulačního modelu. Pro konkrétní srovnání pak musí být v simulátoru generován tok pro každou třídu. Tento model využívá jednu obsluhovou linku.

Stacionární pravděpodobnosti stavů P_j jsou dány:

$$P_j = \begin{cases} \frac{A^j}{j!} P_0, & 0 \leq j < N. \\ \frac{A^j}{N!} \frac{1}{N^{j-N}} P_0, & N \leq j. \end{cases}, \quad (4.1)$$

kde j je stav systému, A je nabídka a P_0 je pravděpodobnost stavu, že systém je prázdný.

Pravděpodobnost P_0 :

$$P_0^{-1} = \sum_{j=0}^{N-1} \frac{A^j}{j!} + \frac{A^N}{N!} \sum_{j=N}^{N+R} \left(\frac{A}{N}\right)^{j-N}, \quad (4.2)$$

kde A je nabídka N je počet obsluhových linek a R je počet míst na čekání v paměti třídy.

Pravděpodobnost čekání paketu v paměti $P(W > 0)$:

$$P(W > 0) = \sum_{j=N}^{N+R-1} P_j \quad (4.3)$$

Střední hodnota počtu čekajících paketů $E[X_f]$:

$$E[X_f] = \sum_{j=N}^{N+R} (j - N)P_j \quad (4.4)$$

Pravděpodobnost ztráty B :

$$B = P_{N+R} \quad (4.5)$$

Střední hodnoty dob čekání $E[W]$, $E[W_w]$, $E[W_0]$:

$$E[W] = \frac{E[X_f]}{(1 - B)\lambda} \quad (4.6)$$

$$E[W_w] = \frac{E[W]}{P(W > 0)} \quad (4.7)$$

$$E[W_0] = E[W](1 - B) \quad (4.8)$$

Vztahy pro výpočet paketů, které byly systému nabízeny, byly jím odeslány nebo zahozeny jsou uvedeny v kapitole 3.1 (viz. 3.5, 3.6, 3.7).

Následuje porovnání výsledků z matematických vztahů popisujících systém $M/M/N/R$ bez priorit s výsledky získanými z prostředí OMNeT++. Pro zpracování matematického modelu dodal vedoucí práce skript v programu MATLAB.

Tab. 1: Vstupní parametry pro srovnávací model

symbol	parametr	hodnota
N	Počet odchozích linek	1
R	Kapacita paměti třídy	10
v_p	Přenosová rychlost odchozí linky	128 kbit/s
t_{sim}	Doba simulace	1800 s
L	Velikost paketu datového toku	145 byte
A	Nabídka pro datový tok	1,2 erl
μ	Intenzita obsluhy datového toku	110,344 s ⁻¹
λ	Intenzita příchozích paketů datového toku	131,578 s ⁻¹
t_{os}	Doba obsluhy paketu datového toku	0,0091 s
t_p	Doba mezipříchodového intervalu paketu datového toku	0,0076 s

Velikost paketu datového toku reprezentuje službu *VoIP*, a byla mu přidělena nabídka odpovídající překročení přenosové kapacity odchozí linky o 20%. Velikost přenosové rychlosti odchozí linky byla vybrána pro jeden ze starších standardů technologie ADSL. Ostatní parametry se za pomoci těchto zvolených hodnot dopočítají ze vztahu uvedených v kapitole 3.1 a to následovně: intenzita obsluhy se určí ze vztahu 3.2, střední hodnota doby obsluhy ze vztahu 3.1, intenzita příchozích paketů datových toků ze vztahu 3.4 a střední hodnota mezipříchodových intervalů jednotlivých datových toků ze vztahu 3.3.

3 Indexy 1 až 4 u jednotlivých parametrů rozlišují jednotlivé datové toky od nejvyšší priority sestupně

Dále je uvedeno porovnání výsledků z matematického a simulačního modelu. Odchytky jsou uvedeny jako rozdíl hodnot analytického a simulačního modelu.

Tab. 2: Počty paketů, výsledky z analytického a simulačního modelu

počet paketů [-]	analytický model	simulační model	absolutní odchylka	relativní odchylka [%]
C_{p_1}	238340	236842	1498	0,63
C_{z_1}	44742	44147	595	1,33
C_{y_1}	193600	192694	906	0,47
C_{w_1}	186188	186598	410	0,2
$E_{X_{f_1}}$	6,5412	6,5201	0,0211	0,32

Tab. 3: Doby čekání, výsledky z analytického a simulačního modelu

doby čekání [s]	analytický model	simulační model	absolutní odchylka	relativní odchylka [%]
$E[W_1]$	0,0608	0,0609	0,0001	0,16
$E[W_{W_1}]$	0,0628	0,0628	0	0
$E[W_{0_1}]$	0,0494	0,0495	0,0001	0,2

4.2. Simulační modely

V této kapitole jsou popsány výsledky, z jednotlivých scénářů, které byly odsimulovány v obou modelech prostředí OMNeT++ (viz. kapitoly 3.2 a 3.3). Pro první scénář, simulovaný v obou modelech jsou uvedeny hodnoty vstupních parametrů a výsledné statistické údaje. Výsledky z ostatních simulačních scénářů jsou uvedeny v Příloze B. Simulace se stejnými vstupními parametry byla provedena stokrát. Z těchto běhů byly následovně spočítány střední hodnoty, směrodatné odchytky, 95% interval spolehlivosti a zaznamenány maximální a minimální hodnoty [4]. Každý scénář představuje jiné zatížení sítě (vždy se zvedá nabídka v každém toku o 0,05 erl) a je odsimulován v modelu *PriorityQueueing*, za nimi následuje jejich porovnání v podobě grafů. V obou modelech je simulace provedena pro systém $M/M/1/10$.

4.2.1. Scénář č. 1 pro model PriorityQueueing

Tab. 4: Vstupní parametry I pro Scénář č. 1.

symbol	parametr	hodnota
N	Počet odchozích linek	1
R	Kapacita paměti jednotlivých tříd	10
v_p	Přenosová rychlost odchozí linky	128 kbit/s
t_{sim}	Doba simulace	1800 s
L_1	Velikost paketu datového toku s nejvyšší prioritou	145 byte
L_2	Velikost paketu datového toku s vysokou prioritou	160 byte
L_3	Velikost paketu datového toku s normální prioritou	126 byte
L_4	Velikost paketu datového toku s nízkou prioritou	190 byte
A_1	Nabídka pro datový tok s nejvyšší prioritou	0,24375 erl
A_2	Nabídka pro datový tok s vysokou prioritou	0,15625 erl
A_3	Nabídka pro datový tok s normální prioritou	0,15625 erl
A_4	Nabídka pro datový tok s nízkou prioritou	0,3125 erl

Velikost přenosové rychlosti odchozí linky byla vybrána pro jeden ze starších standardů technologie *ADSL*. Velikost paketů jednotlivých datových toků reprezentuje službu *VoIP* ve třídě s nejvyšší prioritou, interaktivní spojení s herním serverem ve třídě s vysokou prioritou, spojení se vzdáleným serverem protokolem *SSH* ve třídě s normální prioritou a upload souboru protokolem *FTP*. Výchozí hodnoty nabídek každého datového toku byly vypočítány z rychlostí jednotlivých toků, tak že jejich velikost podělíme přenosovou rychlostí rozhraní:

$$A_i = \frac{v_i}{v_p}, \quad (4.9)$$

kde A_i [erl] je nabídka i -tého datového toku, v_i je [bit/s] rychlost i -tého datového toku a v_p [bit/s] je přenosová rychlost odchozího rozhraní.

V následujícím scénáři bylo 32 kbit/s zvoleno pro třídu s nejvyšší prioritou 20 kbit/s pro třídy s vysokou a normální prioritou a 40 kbit/s pro třídu s nízkou prioritou.

Tab. 5: Vstupní parametry II pro Scénář č.1.

symbol	hodnota [s ⁻¹]	symbol	hodnota [s]	symbol	hodnota [s ⁻¹]	symbol	hodnota [s]
μ_1	110,3448	t_{os1}	0,0091	λ_1	26,8966	t_{p1}	0,0362
μ_2	100,0000	t_{os2}	0,0100	λ_2	15,6250	t_{p2}	0,0640
μ_3	126,9841	t_{os3}	0,0079	λ_3	19,8413	t_{p3}	0,0504
μ_4	84,2105	t_{os4}	0,0119	λ_4	26,3158	t_{p4}	0,0380

Dále následuje zpracování výstupních hodnot do statistických údajů.

Počty paketů jednotlivých datových toků, které byly systému nabízeny (viz. 3.5), byly jím zahozeny (viz. 3.7), odeslány (viz. 3.6) a které skutečně v systému skutečně čekaly.

Tab. 6: Hodnoty počtů paketů získané z modelu PriorityQueueing pro scénář č. 1

symbol	střední hodnota [-]	minimální hodnota [-]	maximální hodnota [-]	směrodatná odchylka [-]
C_{p_1}	48410,16 ± 43,72	47915	49053	220,35
C_{p_2}	28114,39 ± 31,61	27743	28500	159,29
C_{p_3}	35695,17 ± 25,67	35257	36085	179,78
C_{p_4}	47354,12 ± 44,42	46696	47913	223,88
C_{z_1}	0,05 ± 0,05	0	2	0,26
C_{z_2}	0,2 ± 0,13	0	5	0,66
C_{z_3}	19,12 ± 1,44	5	41	7,27
C_{z_4}	2002,34 ± 25,72	1720	2343	129,63
C_{y_1}	48410,11 ± 43,72	47914	49053	220,34
C_{y_2}	28114,19 ± 31,62	27743	28499	159,40
C_{y_3}	35676,05 ± 35,73	35246	36065	180,10
C_{y_4}	45351,78 ± 42,93	44805	45852	216,39
C_{w_1}	41499,79 ± 52,29	40681	42100	263,55
C_{w_2}	24095,88 ± 34,04	23686	24392	171,56
C_{w_3}	30590,01 ± 42,17	29958	31097	212,55
C_{w_4}	38591,14 ± 47,87	37975	39189	241,28
X_{f_1}	0,305672 ± 0,000707	0,295339	0,313989	0,003566
X_{f_2}	0,296482 ± 0,000825	0,285929	0,303761	0,004161
X_{f_3}	0,640407 ± 0,002044	0,611136	0,658191	0,010305
X_{f_4}	2,607779 ± 0,009954	2,486896	2,731502	0,050168

Doby čekání vztahované k paketům, které byly systémem odeslány (viz. 3.9), které v systému skutečně čekali (viz. 3.10) a které byly systému nabízeny (viz. 3.11), jednotlivých datových toků.

Tab. 7: Hodnoty dob čekání získané z modelu PriorityQueueing pro scénář č. 1

symbol	střední hodnota [s]	minimální hodnota [s]	maximální hodnota [s]	směrodatná odchylka [s]
W_1	0,011365 ± 0,000022	0,110945	0,011598	0,000111
W_2	0,018982 ± 0,000048	0,018370	0,019822	0,000243
W_3	0,032310 ± 0,000095	0,031180	0,033202	0,000481
W_4	0,103502 ± 0,000402	0,098805	0,108955	0,002029

W_{W_1}	$0,013258 \pm 0,000018$	0,013041	0,013488	0,000111
W_{W_2}	$0,022147 \pm 0,000049$	0,021540	0,022496	0,000248
W_{W_3}	$0,037682 \pm 0,000098$	0,036543	0,038654	0,000494
W_{W_4}	$0,121631 \pm 0,000418$	0,116821	0,127182	0,002110
W_{0_1}	$0,011365 \pm 0,000024$	0,011094	0,011598	0,000111
W_{0_2}	$0,018981 \pm 0,000048$	0,018370	0,019518	0,000243
W_{0_3}	$0,032293 \pm 0,000095$	0,031167	0,033188	0,000480
W_{0_4}	$0,099122 \pm 0,000343$	0,095121	0,103577	0,001728

4.2.2. Scénář č. 1 pro model PQinet

Tato část kapitoly je věnována statistickým výsledkům z modelu *PQinet*. Tato simulace bude spuštěna se vstupními parametry scénáře č.1 a to z důvodu porovnání výsledků s modelem *PriorityQueueing*.

Tab. 8: Hodnoty dob čekání získané z modelu *PQinet* pro scénář č. 1

symbol	střední hodnota [-]	minimální hodnota [-]	maximální hodnota [-]	směrodatná odchylka [-]
C_{p_1}	$48399,92 \pm 167,02$	47844	48800	209,89
C_{p_2}	$28115,77 \pm 33,14$	27756	28464	167,02
C_{p_3}	$35727,66 \pm 37,34$	35200	36142	188,21
C_{p_4}	$47337,01 \pm 45,64$	46755	47945	230,04
C_{z_1}	0 ± 0	0	0	0
C_{z_2}	$0,03 \pm 0,04$	0	2	0,22
C_{z_3}	$6,19 \pm 0,79$	0	20	3,99
C_{z_4}	$1248,92 \pm 17,96$	1042	1435	90,51
C_{y_1}	$48399,21 \pm 41,65$	47844	48797	209,91
C_{y_2}	$28115,28 \pm 33,13$	27755	28464	166,99
C_{y_3}	$35720,61 \pm 37,31$	35197	36139	188,04
C_{y_4}	$46085,54 \pm 45,04$	45513	46593	227,01
X_{f_1}	$0,245527 \pm 0,000509$	0,239047	0,252077	0,002569
X_{f_2}	$0,237463 \pm 0,000795$	0,228673	0,249424	0,004008
X_{f_3}	$0,511385 \pm 0,001706$	0,489700	0,529536	0,008600
X_{f_4}	$2,278346 \pm 0,008932$	2,183137	2,395531	0,045019

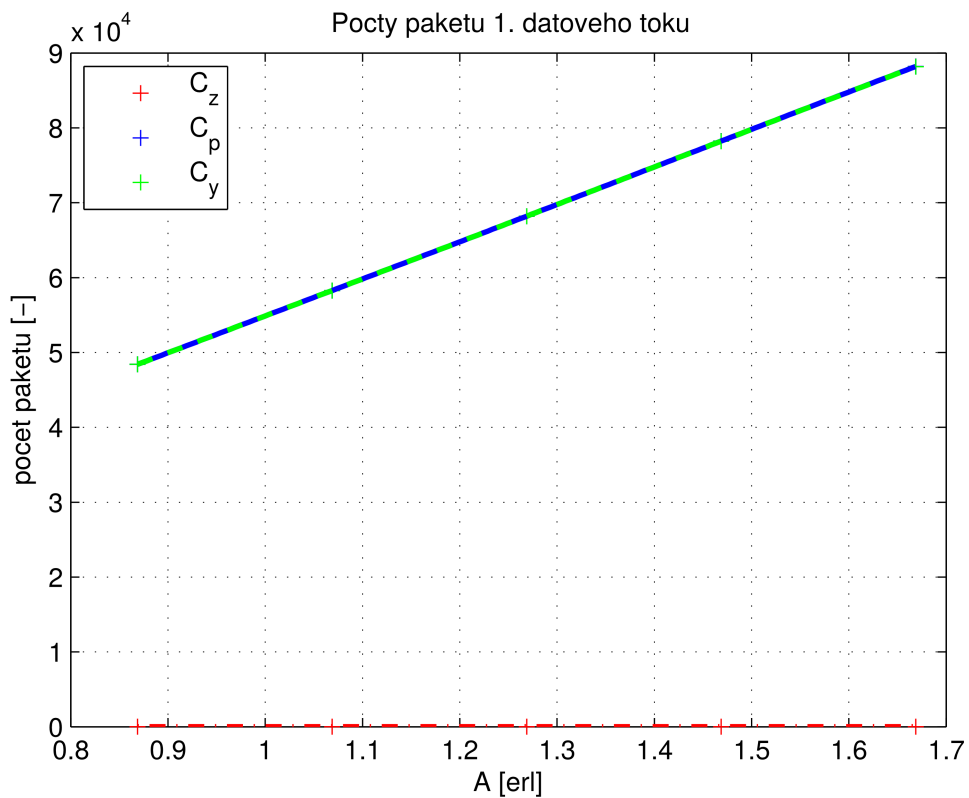
Tab. 9: Hodnoty dob čekání získané z modelu PQinet pro scénář č. 1

symbol	střední hodnota [s]	minimální hodnota [s]	maximální hodnota [s]	směrodatná odchylka [s]
W_1	$0,009131 \pm 0,000015$	0,008946	0,009344	0,000078
W_2	$0,015202 \pm 0,000043$	0,014659	0,015817	0,000220
W_3	$0,025768 \pm 0,000078$	0,024856	0,026585	0,000395
W_4	$0,088983 \pm 0,000354$	0,085068	0,093779	0,001786
W_{0_1}	$0,009131 \pm 0,000015$	0,008946	0,009344	0,000078
W_{0_2}	$0,015202 \pm 0,000043$	0,014659	0,015817	0,000220
W_{0_3}	$0,025763 \pm 0,000078$	0,024852	0,026582	0,000394
W_{0_4}	$0,086629 \pm 0,000322$	0,083001	0,091038	0,001627

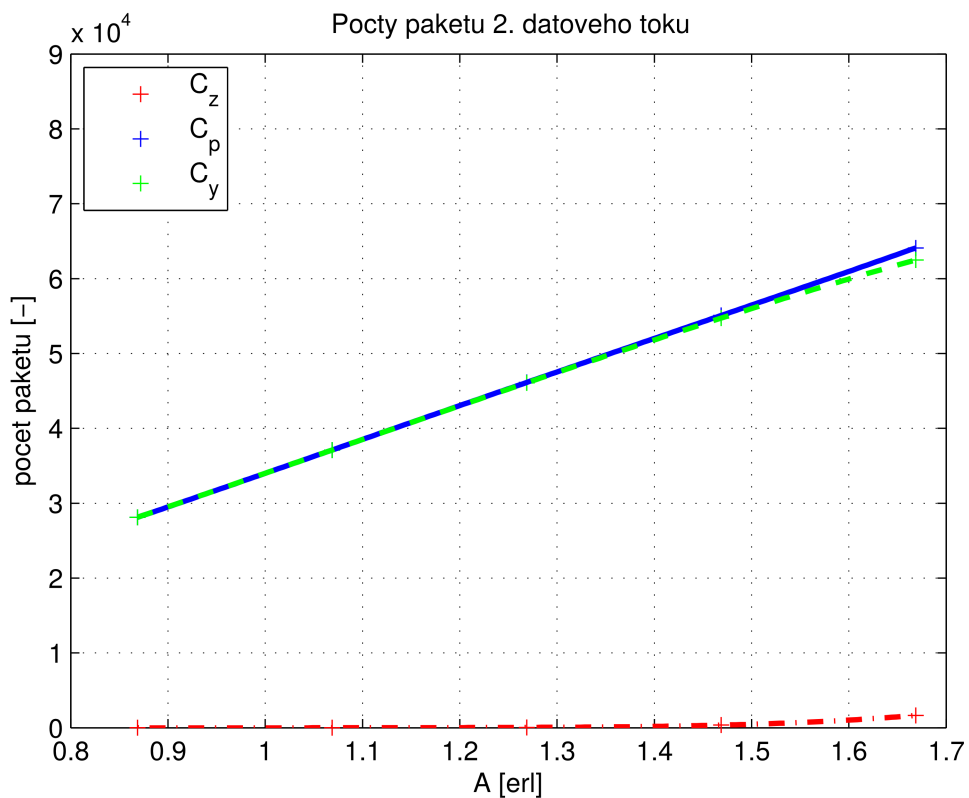
4.2.3. Porovnání jednotlivých scénářů

V následující části kapitoly je vyobrazeno grafické znázornění pozorovaných výsledků ze scénářů 1 až 5. Na vodorovné ose je vždy vyobrazen součet nabídek jednotlivých datových toků, na ose svislé pak počet paketů nebo čas.

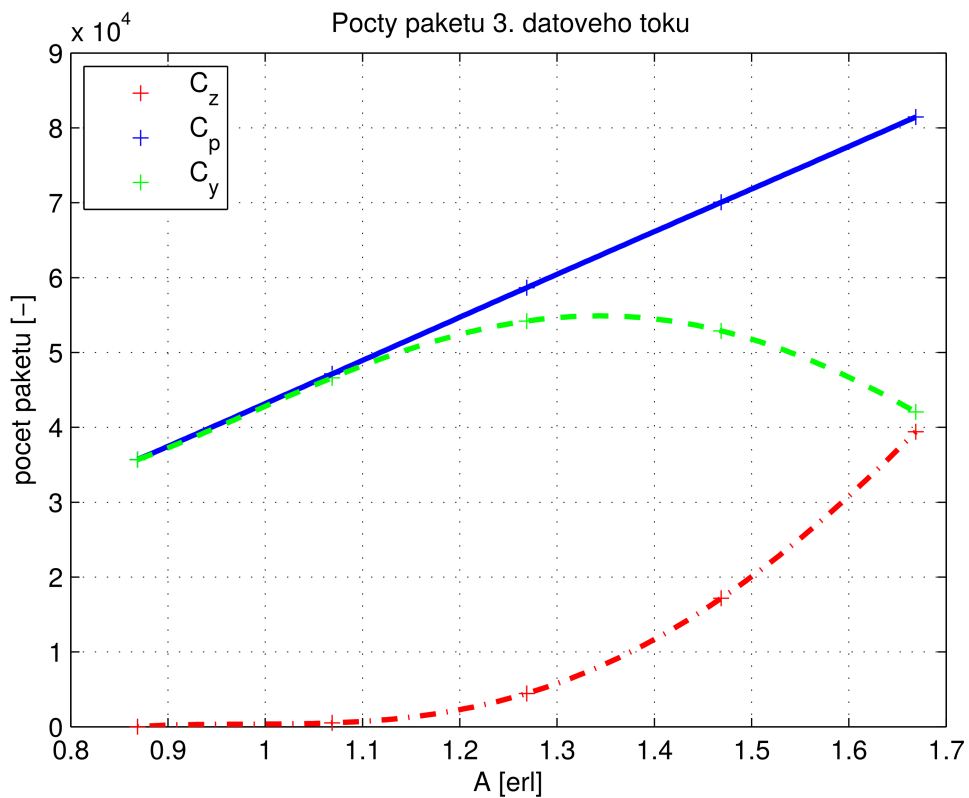
Je vidět, že třída s nejvyšší prioritou vykazuje minimální ztrátovost paketů a to i při nabídce větší než jedna, tedy v případě, že je přenosová kapacita odchozí linky překročena. Ztráta ve třídě s nejvyšší prioritou je při nabízených zatíženích téměř nulová (viz. Obr. 4.1). Počty paketů, které byly systému nabízeny a byly jím odeslány, se tedy téměř rovnají. U tříd s nižšími prioritami už je ztráta znatelnější. Třída s vysokou prioritou vykazuje slabě viditelnou ztrátu při nabídce překračující hodnotu 1,4 erl (viz. Obr. 4.2). U třídy s normální prioritou je už ztráta paketů při vyšší nabídce znatelnější a při nabídnutém zatížení cca 1,7 erl se křivky počtů ztracených a odeslaných paketů protínají (viz. Obr. 4.3). Markantní ztrátovost paketů vykazuje třída s nejnižší prioritou. Křivky ztracených a odeslaných paketů se zde protínají už při nabízeném zatížení 1,2 erl a při nabídce blížící se k hodnotě 1,7 erl se už počet paketů zahozených systémem blíží počtu paketů, které byly systému nabízeny (viz. Obr. 4.4).



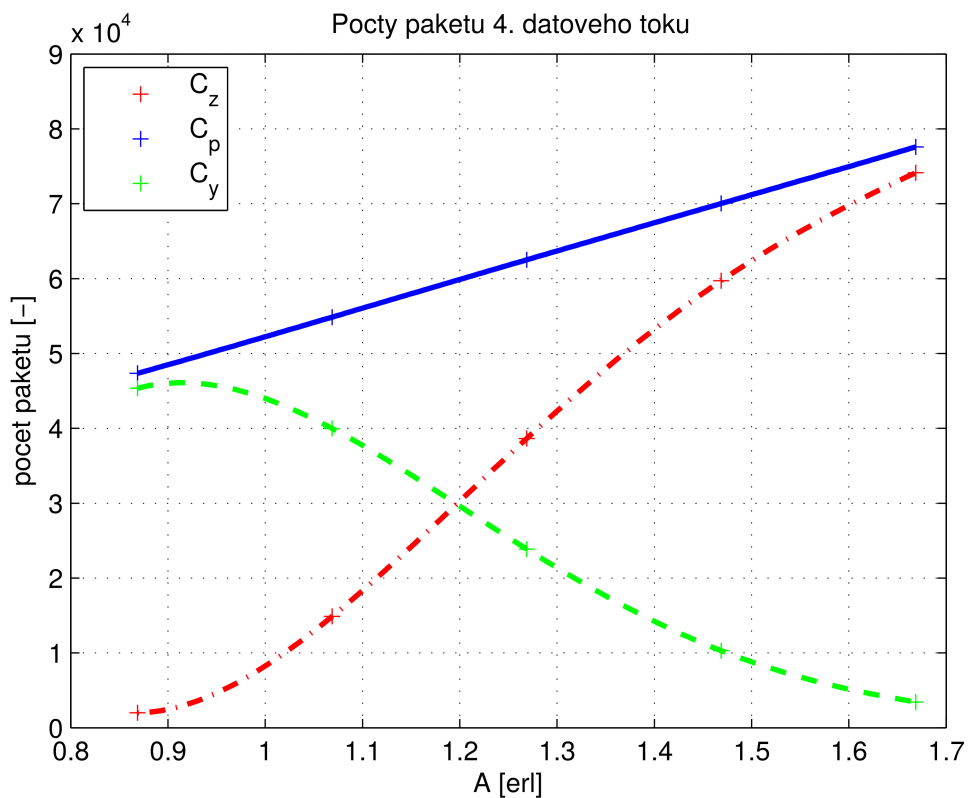
Obr. 4.1: Ztracené, přijaté, a odeslané pakety 1. datového toku v závislosti na nabídce



Obr. 4.2: Ztracené, přijaté, a odeslané pakety 2. datového toku v závislosti na nabídce

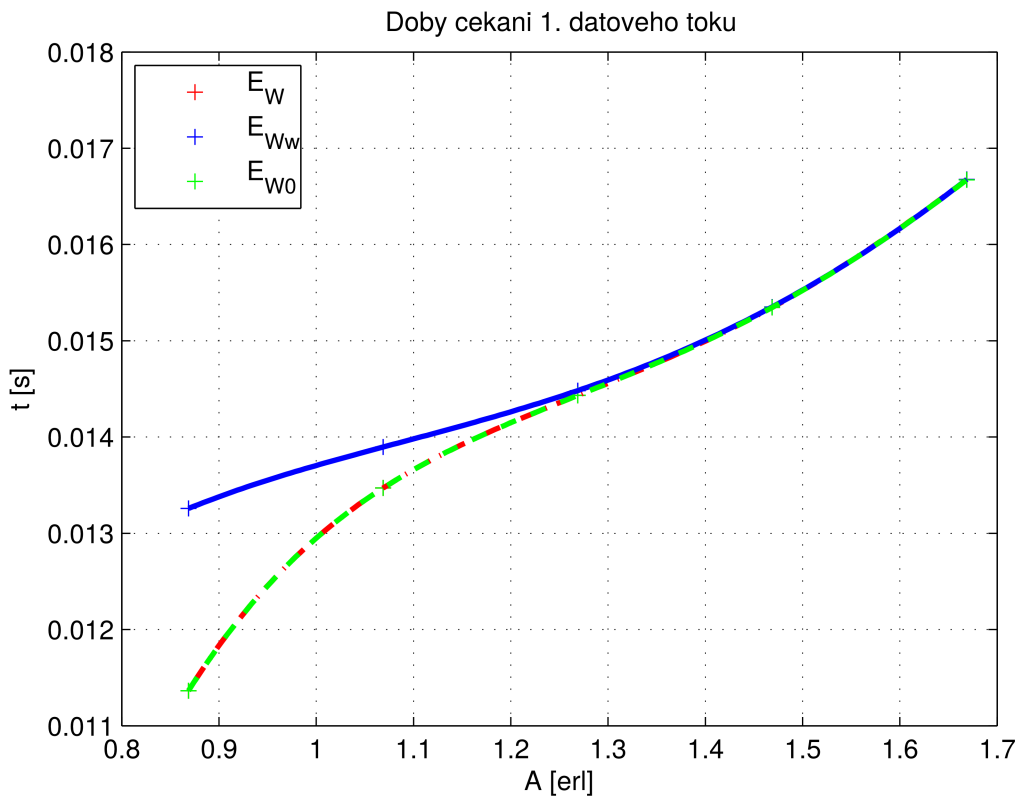


Obr. 4.3: Ztracené, přijaté, a odeslané pakety 3. datového toku v závislosti na nabídce

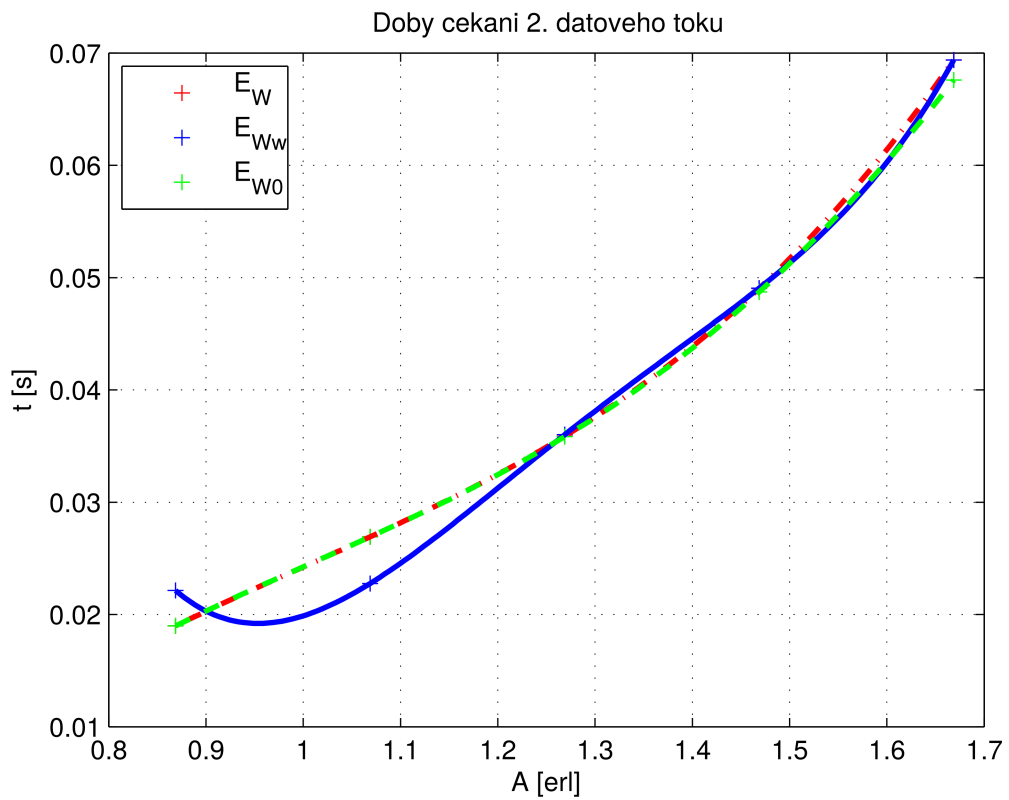


Obr. 4.4: Ztracené, přijaté, a odeslané pakety 4. datového toku v závislosti na nabídce

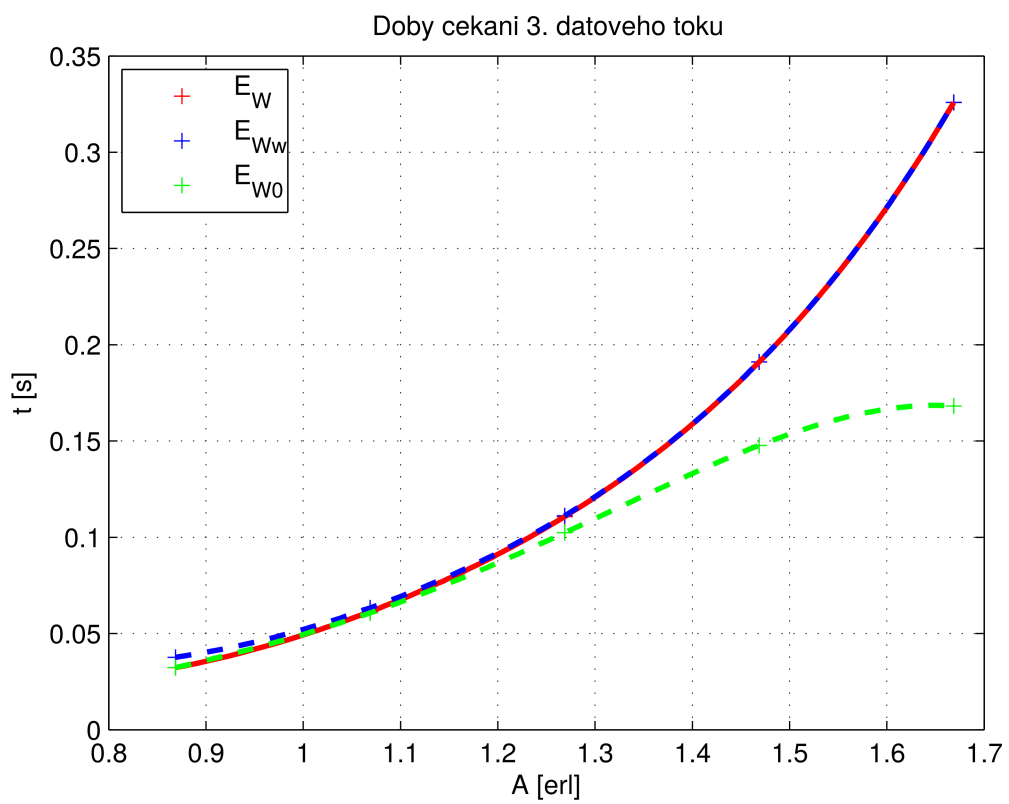
Závislost simulačního modelu metody PQ na nabídce je pozorovatelná také z hlediska dob čekání. Ve třídě s nejvyšší prioritou se doba čekání vztažená k paketům, které byly systému nabízeny a k paketům, které byly systémem odeslány, až na zanedbatelné odchylky rovnají, což odpovídá skutečnosti, že počet příchozích a odeslaných paketů se vzhledem ke ztrátovosti blíží k nule rovná. Při nabídce přibližně 1,3 erl začnou všechny tři doby splývat a což zapříčiňuje zvětšující se počet paketů, které skutečně čekaly se zvyšující se nabídkou (viz. Obr. 4.5). Ve třídě s vysokou prioritou mají všechny doby čekání tendenci splývat. Doby čekání vztažené k příchozím a odeslaným paketům splývají ze stejného důvodu, jako u třídy s nejvyšší prioritou, neboť ztrátovost se stále drží blízko nuly. Vzhledem k nižší prioritě této třídy v systému musejí ale téměř všechny pakety čekat (viz. Obr. 4.6). Jako v předchozím případě musejí i téměř všechny odeslané pakety z třídy s normální prioritou čekat, doba čekání vztažená ke všem paketům však od nabídky 1,2 erl roste pomaleji, než ostatní dvě. Důvodem je vzrůstající počet zahozených paketů (viz. Obr. 4.7). U třídy s nízkou prioritou vidíme, že doby čekání vztažené k odeslaným a čekajícím paketům rostou výrazně rychle (viz. Obr. 4.8). Důvodem je velké zatížení a ztrátovost.



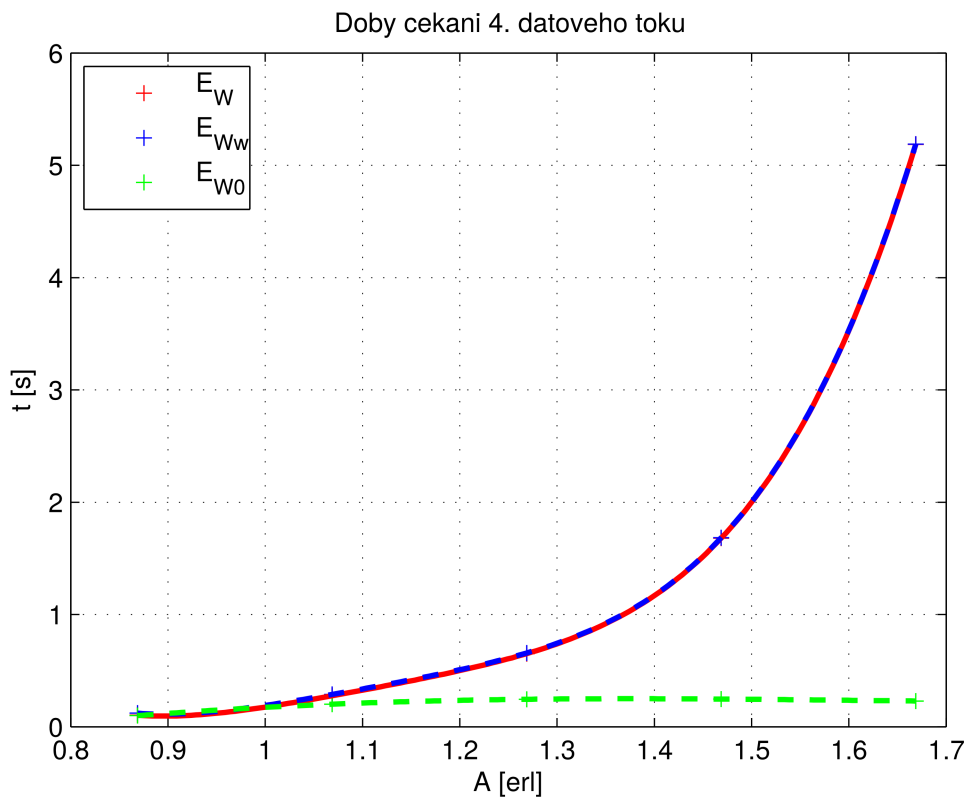
Obr. 4.5: Doby čekání 1. datového toku



Obr. 4.6: Doby čekání 2. datového toku



Obr. 4.7: Doby čekání 3. datového toku



Obr. 4.8: Doby čekání 4. datového toku

5. Závěr

Z hlediska počtu paketů se simulační model chová dle předpokladu, čím větší zatížení, tím větší ztrátovost paketů. Jak je vidět z výsledků simulací, třída s nejvyšší prioritou se i při vyšším zatížení snaží udržet co nejmenší ztrátovost paketů a to na úkor tříd ostatních. Zatímco počet ztracených paketů se v prvním datovém toku při zvětšující se nabídce mění v řádu desítek paketů, ve čtvrtém datovém toku se tyto počty pohybují v řádu několika desítek tisíc paketů. Z hlediska dob čekání se systém chová podobně. Doby čekání v prvním datovém toku jsou podstatně kratší než v tocích ostatních. Extrémním případem je čtvrtý datový tok, kde se průměrné hodnoty dob čekání pohybují v řádu sekund. Odlišně se chová doba čekání vztahovaná k paketům, které byly systému nabízeny. Tento výstupní parametr je totiž málo závislý na zatížení systému.

Simulační model sestavený v INET Framework a spuštěný pro scénář vykazuje značné odchylky od modelu sestaveném v základním prostředí OMNeT++. Z tohoto důvodu nejsou modely srovnávány pro další scénáře. Důvodem těchto nesrovnalostí je s největší pravděpodobností exponenciální rozložení velikosti paketů. V simulačním modelu *PQinet* se toto rozložení aplikuje při generování paketu na vyšších síťových vrstvách RM-OSI a k této hodnotě jsou následně přičítána záhlaví nižších vrstev o pevné délce. Najít analytické řešení tohoto typu systému se mi nepodařilo. Pokud by délka paketu byla konstantní, model v INET Framework by vzhledem k modelu v základním prostředí OMNeT++ při deterministické době obsluhy odpovídal realitě.

Oba dva modely lze přepnout do režimu RR. V současnosti ale nemám dostatečné matematické zázemí pro jeho analytické řešení.

Díky této práci jsem se naučil pracovat s komplexním simulačním nástrojem OMNeT++ a prohloubil znalosti o metodách zpracování paketů na směrovačích. Jako předmět k zpracování navazující na tuto práci se nabízí sestavení simulačních modelů dalších zmíněných metod v prostředí OMNeT++.

6. Reference

- [1] RANJBAR, Amir S. *CCNP ONT Official Exam Certification Guide*. 800 East 96th Street Indianapolis, IN 46240 USA: Cisco Press, 2007. ISBN 978-1-58720-176-9.
- [2] ODOM, Wendell a Michael J. CAVANAUGH. *Cisco DQOS Exam Certification Guide*. 201 West 103rd Street Indianapolis, IN 46290 USA: Cisco Press, 2003. ISBN 1-58720-058-9.
- [3] VARGA, András. *OMNeT++ User Manual: Version 4.4.1* [online]. 2014, 432 s. [cit. 2014-05-14]. Dostupné z: <http://www.omnetpp.org/doc/omnetpp/Manual.pdf>
- [4] VARGA, András. *OMNeT++ User Guide: Version 4.4.1* [online]. 2014, 123 s. [cit. 2014-05-12]. Dostupné z: <http://www.omnetpp.org/doc/omnetpp/UserGuide.pdf>
- [5] *INET Framework for OMNeT++ Manual*. [online]. 2012, 157 s. [cit. 2014-05-14]. Dostupné z: <http://inet.omnetpp.org/doc/INET/inet-manual-draft.pdf>
- [6] *INET Framework for OMNeT++/OMNEST*. [online]. 2012, [cit. 2014-05-14]. Dostupné z: <http://inet.omnetpp.org/doc/INET/neddoc/index.html>
- [7] NOVÁK, J, KRÍŽOVSKÝ, F. *Spojovací systémy I : Přednášky*. 1st edition. Praha: ČVUT, 1983. 313 p.
- [8] HAMPL, Petr. *Kendallová klasifikace obsluhových systémů*. Acces Server [online]. 2005, [cit. 2014-05-11]. Dostupné z: <http://access.feld.cvut.cz/view.php?Cisloclanku=2005111601>

Seznam příloh

Příloha A: Simulační modely v prostředí OMNeT++ (uvedeno na přiloženém CD)

- A1: Model v základním prostředí OMNeT++
- A2: Model v INET Framework

Příloha B: Numerické výsledky simulací modelu *PriorityQueueing* pro scénáře č. 2 až č. 6

- B1: Scénář č. 2
- B2: Scénář č. 3
- B3: Scénář č. 4
- B4: Scénář č. 5

Příloha A

Příloha A1

Simulační model *PriorityQueueing* je uložen v stejnojmenném adresáři a obsahuje soubory:

- *omnetpp.ini*: konfigurační soubor prostředí OMNeT++
- *PriorityQueueing.ned*: soubor obsahující topologii modelu
- *PriorityQueueing_run_script*: spouštěcí soubor simulace
- *PriorityQueueing.anf*: soubor pro výpočet statistických výsledků

Příloha A2

Simulační model PQinet je uložen ve stejnojmenném adresáři a obsahuje soubory:

- *omnetpp.ini*: konfigurační soubor prostředí OMNeT++
- *PQinet.ned*: soubor obsahující topologii modelu
- *PQinet_run_script*: spouštěcí soubor simulace
- *PQ.anf*: soubor pro výpočet statistických výsledků
- *PQ_filtres.xml*: soubor s filtrem přiřazující DSCP hodnotu do záhlaví paketů

Příloha B

Příloha B1

Tab. 10: Vstupní parametry I pro Scénář č. 2

značka	parametr	hodnota
N	Počet odchozích linek	1
R	Kapacita paměti jednotlivých tříd	10
v_p	Přenosová rychlost odchozí linky	128 kbit/s
t_{sim}	Doba simulace	1800 s
L_1	Velikost paketu datového toku s nejvyšší prioritou	145 byte
L_2	Velikost paketu datového toku s vysokou prioritou	160 byte
L_3	Velikost paketu datového toku s normální prioritou	126 byte
L_4	Velikost paketu datového toku s nízkou prioritou	190 byte
A_1	Nabídka pro datový tok s nejvyšší prioritou	0,29375 erl
A_2	Nabídka pro datový tok s vysokou prioritou	0,20625 erl
A_3	Nabídka pro datový tok s normální prioritou	0,20625 erl
A_4	Nabídka pro datový tok s nízkou prioritou	0,3625 erl

Tab. 11: Vstupní parametry II pro scénář č. 3

symbol	hodnota [s ⁻¹]	symbol	hodnota [s]	symbol	hodnota [s ⁻¹]	symbol	hodnota [s]
μ_1	110,3448	t_{os1}	0,0091	λ_1	32,4138	t_{p1}	0,0309
μ_2	100,0000	t_{os2}	0,0100	λ_2	20,6250	t_{p2}	0,0485
μ_3	126,9841	t_{os3}	0,0079	λ_3	26,1905	t_{p3}	0,0382
μ_4	84,2105	t_{os4}	0,0119	λ_4	30,5263	t_{p4}	0,0328

Tab. 12: Hodnoty počtů paketů získané z modelu PriorityQueueing pro scénář č. 2

symbol	střední hodnota [-]	minimální hodnota [-]	maximální hodnota [-]	směrodatná odchylka [-]
C_{p1}	58273,27 ± 55,26	57650	58955	278,54
C_{p2}	37103,90 ± 37,34	36762	37583	188,20
C_{p3}	47129,84 ± 45,05	46606	47732	227,09
C_{p4}	54853,76 ± 46,44	54084	55376	229,04
C_{z1}	0,28 ± 0,14	0	4	0,73
C_{z2}	4,16 ± 0,62	0	14	3,17
C_{z3}	515,20 ± 11,54	358	643	58,18
C_{z4}	14870,27 ± 78,54	14001	16124	395,83

C_{y_1}	$58272,99 \pm 55,26$	57648	58955	278,52
C_{y_2}	$37099,74 \pm 37,41$	36756	37583	188,56
C_{y_3}	$46614,64 \pm 42,37$	46106	47223	213,57
C_{y_4}	$39983,49 \pm 67,64$	38997	40886	340,91
C_{w_1}	$56487,95 \pm 62,91$	55857	57271	317,08
C_{w_2}	$35963,73 \pm 40,71$	35568	36485	205,53
C_{w_3}	$45172,72 \pm 48,74$	44552	45796	245,65
C_{w_4}	$38302,91 \pm 61,86$	37404	39230	311,80
X_{f_1}	$0,436095 \pm 0,000940$	0,423901	0,447097	0,004738
X_{f_2}	$0,554709 \pm 0,001736$	0,536412	0,580265	0,008752
X_{f_3}	$1,589799 \pm 0,005559$	1,500657	1,658312	0,028019
X_{f_4}	$6,129734 \pm 0,014217$	5,987158	6,324789	0,071653

Tab. 13: Doby čekání vztahené k paketům, které systémem prošly pro scénář č. 2

symbol	střední hodnota [s]	minimální hodnota [s]	maximální hodnota [s]	směrodatná odchylka [s]
W_1	$0,013470 \pm 0,000021$	0,013227	0,013732	0,000110
W_2	$0,026913 \pm 0,000081$	0,026029	0,028035	0,000410
W_3	$0,061387 \pm 0,000198$	0,057955	0,063706	0,000999
W_4	$0,275993 \pm 0,001029$	0,264155	0,291963	0,005187
W_{W_1}	$0,013896 \pm 0,000017$	0,013655	0,014112	0,000105
W_{W_2}	$0,027763 \pm 0,000081$	0,026910	0,028883	0,000409
W_{W_3}	$0,063347 \pm 0,000197$	0,059881	0,065580	0,000995
W_{W_4}	$0,288096 \pm 0,001011$	0,275306	0,304395	0,005097
W_{0_1}	$0,013470 \pm 0,000021$	0,013227	0,013732	0,000110
W_{0_2}	$0,026910 \pm 0,000081$	0,026028	0,028030	0,000409
W_{0_3}	$0,060716 \pm 0,000187$	0,057481	0,063001	0,000944
W_{0_4}	$0,201145 \pm 0,000431$	0,196376	0,206558	0,002172

Příloha B2

Tab. 14: Vstupní parametry I pro scénář č. 3

symbol	parametr	hodnota
N	Počet odchozích linek	1
R	Kapacita paměti jednotlivých tříd	10
v_p	Přenosová rychlost odchozí linky	128 kbit/s
t_{sim}	Doba simulace	1800 s
L_1	Velikost paketu datového toku s nejvyšší prioritou	145 byte
L_2	Velikost paketu datového toku s vysokou prioritou	160 byte
L_3	Velikost paketu datového toku s normální prioritou	126 byte
L_4	Velikost paketu datového toku s nízkou prioritou	190 byte
A_1	Nabídka pro datový tok s nejvyšší prioritou	0,34375 erl
A_2	Nabídka pro datový tok s vysokou prioritou	0,25625 erl
A_3	Nabídka pro datový tok s normální prioritou	0,25625 erl
A_4	Nabídka pro datový tok s nízkou prioritou	0,4125 erl

Tab. 15: Vstupní parametry II pro scénář č. 3

symbol	hodnota [s ⁻¹]	symbol	hodnota [s]	symbol	hodnota [s ⁻¹]	symbol	hodnota [s]
μ_1	110,3448	t_{os1}	0,0091	λ_1	37,9310	t_{p1}	0,0264
μ_2	100,0000	t_{os2}	0,0100	λ_2	25,6250	t_{p2}	0,0390
μ_3	126,9841	t_{os3}	0,0079	λ_3	32,5397	t_{p3}	0,0307
μ_4	84,2105	t_{os4}	0,0119	λ_4	34,7368	t_{p4}	0,0288

Tab. 16: Hodnoty počtů paketů získané z modelu PriorityQueueing pro scénář č. 3

symbol	střední hodnota [-]	minimální hodnota [-]	maximální hodnota [-]	směrodatná odchylka [-]
C_{p1}	68201,37 ± 57,47	67422	69088	289,63
C_{p2}	46131,12 ± 46,76	45535	46643	235,66
C_{p3}	58651,22 ± 56,15	57933	59341	283,02
C_{p4}	62504,43 ± 45,83	61939	63131	230,97
C_{z1}	1,32 ± 0,30	0	8	1,54
C_{z2}	51,91 ± 2,78	18	87	14,05
C_{z3}	4456,60 ± 45,24	3893	4978	228,01
C_{z4}	38629,51 ± 87,41	37712	39535	440,55
C_{y1}	68200,50 ± 57,50	67418	69086	289,79

C_{y_2}	46079,21 ± 46,03	45470	46603	232,02
C_{y_3}	54194,62 ± 48,51	53687	54806	244,49
C_{y_4}	23874,92 ± 78,81	61717	62938	239,02
C_{w_1}	67964,04 ± 58,69	67180	68862	295,82
C_{w_2}	45919,02 ± 46,39	45283	46438	233,82
C_{w_3}	53991,69 ± 49,17	53490	54621	247,85
C_{w_4}	23657,75 ± 76,92	22749	24709	387,70
X_{f_1}	0,546834 ± 0,000918	0,534359	0,557024	0,004630
X_{f_2}	0,918599 ± 0,003008	0,884396	0,957798	0,015160
X_{f_3}	3,334643 ± 0,009844	3,209321	3,436096	0,049613
X_{f_4}	8,637415 ± 0,007689	8,547362	8,723666	0,038752

Tab. 17: Hodnoty dob čekání získané z modelu PriorityQueueing pro scénář č. 3

symbol	střední hodnota [s]	minimální hodnota [s]	maximální hodnota [s]	směrodatná odchylka [s]
W_1	0,014432 ± 0,000020	0,014190	0,014669	0,000100
W_2	0,035882 ± 0,000102	0,034788	0,037192	0,000514
W_3	0,110758 ± 0,000359	0,106391	0,114791	0,001810
W_4	0,651425 ± 0,002641	0,618788	0,684765	0,013314
W_{W_1}	0,014482 ± 0,000016	0,014242	0,014718	0,000101
W_{W_2}	0,036007 ± 0,000102	0,034910	0,037318	0,000514
W_{W_3}	0,111174 ± 0,000359	0,106834	0,115225	0,001813
W_{W_4}	0,657398 ± 0,002623	0,625427	0,690185	0,013219
W_{0_1}	0,014432 ± 0,000019	0,014190	0,014669	0,000100
W_{0_2}	0,035842 ± 0,000101	0,034774	0,034150	0,000509
W_{0_3}	0,102337 ± 0,000262	0,099273	0,105171	0,001323
W_{0_4}	0,245208 ± 0,000259	0,245208	0,251641	0,001308

Příloha B3

Tab. 18: Vstupní parametry I pro Scénář č. 4

symbol	parametr	hodnota
N	Počet odchozích linek	1
R	Kapacita paměti jednotlivých tříd	10
v_p	Přenosová rychlost odchozí linky	128 kbit/s
t_{sim}	Doba simulace	1800 s
L_1	Velikost paketu datového toku s nejvyšší prioritou	145 byte
L_2	Velikost paketu datového toku s vysokou prioritou	160 byte
L_3	Velikost paketu datového toku s normální prioritou	126 byte
L_4	Velikost paketu datového toku s nízkou prioritou	190 byte
A_1	Nabídka pro datový tok s nejvyšší prioritou	0,39375 erl
A_2	Nabídka pro datový tok s vysokou prioritou	0,30625 erl
A_3	Nabídka pro datový tok s normální prioritou	0,30625 erl
A_4	Nabídka pro datový tok s nízkou prioritou	0,4625 erl

Tab. 19: Vstupní parametry II pro scénář č. 4

symbol	hodnota [s ⁻¹]	symbol	hodnota [s]	symbol	hodnota [s ⁻¹]	symbol	hodnota [s]
μ_1	110,3448	t_{os1}	0,0091	λ_1	43,4483	t_{p1}	0,0230
μ_2	100,0000	t_{os2}	0,0100	λ_2	30,6250	t_{p2}	0,0327
μ_3	126,9841	t_{os3}	0,0079	λ_3	38,8889	t_{p3}	0,0257
μ_4	84,2105	t_{os4}	0,0119	λ_4	38,9474	t_{p4}	0,0257

Tab. 20: Hodnoty počtů paketů získané z modelu PriorityQueueing pro scénář č. 4

symbol	střední hodnota [-]	minimální hodnota [-]	maximální hodnota [-]	směrodatná odchylka [-]
C_{p1}	78230,19 ± 62,97	77577	78935	317,38
C_{p2}	55068,69 ± 47,44	54499	55756	239,12
C_{p3}	70051,40 ± 46,86	69504	70686	236,19
C_{p4}	70027,94 ± 47,80	69450	70646	240,94
C_{z1}	4,88 ± 0,64	0	12	3,24
C_{z2}	358 ± 9,12	238	483	46,00
C_{z3}	17159,81 ± 73,63	16323	17922	371,10
C_{z4}	59722,36 ± 75,74	58791	60486	381,75
C_{y1}	78225,31 ± 62,99	77572	78933	317,49

C_{y_2}	$54710,69 \pm 44,38$	54172	55348	223,71
C_{y_3}	$52891,59 \pm 75,39$	51988	53641	379,97
C_{y_4}	$10305,58 \pm 56,77$	9696	11031	286,11
C_{w_1}	$78200,62 \pm 63,07$	77534	78910	317,87
C_{w_2}	$54693,06 \pm 44,35$	54157	55328	223,55
C_{w_3}	$52870,23 \pm 75,43$	51962	53613	380,17
C_{w_4}	$10238,84 \pm 56,48$	9678	11003	284,68
X_{f_1}	$0,666979 \pm 0,001318$	0,652373	0,686282	0,006646
X_{f_2}	$1,490801 \pm 0,005396$	1,430988	1,554397	0,027196
X_{f_3}	$5,613092 \pm 0,010327$	5,480481	5,741387	0,052049
X_{f_1}	$0,666979 \pm 0,001318$	0,652373	0,686282	0,006646

Tab. 21: Hodnoty dob čekání získané z modelu PriorityQueueing pro scénář č. 4

symbol	střední hodnota [s]	minimální hodnota [s]	maximální hodnota [s]	směrodatná odchylka [s]
W_1	$0,015347 \pm 0,000023$	0,015101	0,015706	0,000120
W_2	$0,049046 \pm 0,000162$	0,047076	0,050957	0,000818
W_3	$0,191041 \pm 0,000572$	0,184187	0,197942	0,002886
W_4	$1,679353 \pm 0,009667$	1,563578	1,789530	0,048719
W_{W_1}	$0,015352 \pm 0,000019$	0,015104	0,015708	0,000120
W_{W_2}	$0,049062 \pm 0,000162$	0,047085	0,050969	0,000818
W_{W_3}	$0,191118 \pm 0,000572$	0,184263	0,198015	0,002886
W_{W_4}	$1,682895 \pm 0,009660$	1,567418	1,792677	0,048684
W_{0_1}	$0,015346 \pm 0,000023$	0,015100	0,015704	0,000120
W_{0_2}	$0,048727 \pm 0,000156$	0,046871	0,050567	0,000788
W_{0_3}	$0,147744 \pm 0,000268$	0,140489	0,147744	0,001351
W_{0_4}	$0,246945 \pm 0,000177$	0,244144	0,249202	0,000893

Příloha B4

Tab. 22: Vstupní parametry I pro Scénář č.5

symbol	parametr	hodnota
N	Počet odchozích linek	1
R	Kapacita paměti jednotlivých tříd	10
v_p	Přenosová rychlost odchozí linky	128 kbit/s
t_{sim}	Doba simulace	1800 s
L_1	Velikost paketu datového toku s nejvyšší prioritou	145 byte
L_2	Velikost paketu datového toku s vysokou prioritou	160 byte
L_3	Velikost paketu datového toku s normální prioritou	126 byte
L_4	Velikost paketu datového toku s nízkou prioritou	190 byte
A_1	Nabídka pro datový tok s nejvyšší prioritou	0,44375 erl
A_2	Nabídka pro datový tok s vysokou prioritou	0,35625 erl
A_3	Nabídka pro datový tok s normální prioritou	0,35625 erl
A_4	Nabídka pro datový tok s nízkou prioritou	0,5125 erl

Tab. 23: Vstupní parametry II pro scénář č. 5

symbol	hodnota [s ⁻¹]	symbol	hodnota [s]	symbol	hodnota [s ⁻¹]	symbol	hodnota [s]
μ_1	110,3448	t_{os1}	0,0091	λ_1	48,9655	t_{p1}	0,0204
μ_2	100,0000	t_{os2}	0,0100	λ_2	35,6250	t_{p2}	0,0281
μ_3	126,9841	t_{os3}	0,0079	λ_3	45,2381	t_{p3}	0,0221
μ_4	84,2105	t_{os4}	0,0119	λ_4	43,1579	t_{p4}	0,0232

Tab. 24: Hodnoty počtů paketů získané z modelu PriorityQueueing pro scénář č. 5

symbol	střední hodnota [-]	minimální hodnota [-]	maximální hodnota [-]	směrodatná odchylka [-]
C_{p1}	88199,13 ± 60,73	87542	88891	306,07
C_{p2}	64096,32 ± 50,88	63538	64864	256,44
C_{p3}	81453,35 ± 54,96	80692	82078	277,02
C_{p4}	77587,56 ± 56,37	76981	78555	284,12
C_{z1}	15,56 ± 1,34	3	35	6,76
C_{z2}	1658,27 ± 20,50	1424	1910	103,34
C_{z3}	39408,27 ± 93,27	38404	40553	470,09
C_{z4}	74141,01 ± 62,58	73399	75026	315,43

C_{y_1}	$88183,57 \pm 60,69$	87526	88885	305,88
C_{y_2}	$62438,05 \pm 49,11$	61788	63081	247,52
C_{y_3}	$42045,08 \pm 92,50$	40887	43272	466,22
C_{y_4}	$3445,86 \pm 29,99$	3048	3826	151,16
C_{w_1}	$88181,01 \pm 60,77$	87522	88882	306,27
C_{w_2}	$61786,99 \pm 49,15$	61786	63079	247,74
C_{w_3}	$42042,67 \pm 92,44$	40887	43267	465,88
C_{w_4}	$3443 \pm 29,96$	3045	3824	150,99
X_{f_1}	$0,816964 \pm 0,001702$	0,800456	0,832477	0,008579
X_{f_2}	$2,406632 \pm 0,006720$	2,333235	2,496690	0,033871
X_{f_3}	$7,611711 \pm 0,008427$	7,525456	7,718256	0,042473
X_{f_4}	$9,901027 \pm 0,001280$	9,886376	9,916881	0,006454

Tab. 25: Hodnoty dob čekání získané z modelu PriorityQueueing pro scénář č. 5

symbol	střední hodnota [s]	minimální hodnota [s]	maximální hodnota [s]	směrodatná odchylka [s]
W_1	$0,016675 \pm 0,000028$	0,016402	0,016943	0,000145
W_2	$0,069380 \pm 0,000193$	0,067152	0,071651	0,000976
W_3	$0,325930 \pm 0,001027$	0,313414	0,337657	0,005180
W_4	$5,185423 \pm 0,045910$	4,651637	5,859334	0,231378
W_{W_1}	$0,016676 \pm 0,000024$	0,016403	0,016943	0,000145
W_{W_2}	$0,069382 \pm 0,000193$	0,067154	0,071651	0,000976
W_{W_3}	$0,325949 \pm 0,001027$	0,313443	0,337649	0,005178
W_{W_4}	$5,189084 \pm 0,045916$	4,652857	5,863194	0,231410
W_{0_1}	$0,016672 \pm 0,000028$	0,016401	0,016940	0,000145
W_{0_2}	$0,067584 \pm 0,000172$	0,065668	0,069554	0,000870
W_{0_3}	$0,168213 \pm 0,000208$	0,165459	0,170374	0,001052
W_{0_4}	$0,229855 \pm 0,000185$	0,227128	0,231698	0,000933