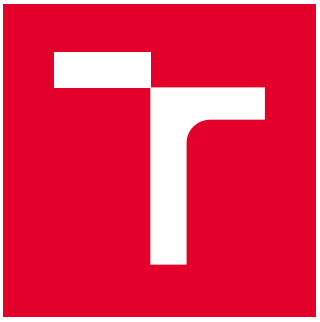


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY
DEPARTMENT OF CONTROL AND INSTRUMENTATION

NÁVRH LABORATORNÍCH ÚLOH V PROSTŘEDÍ RIVERBED MODELER
DESIGN OF LABORATORY TASKS IN RIVERBED MODELER

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. Stanislav Lojek**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Lukáš Langhammer, Ph.D**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**
Ústav telekomunikací

Student: Bc. Stanislav Lojek ***ID:*** 164898
Ročník: 2 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Návrh laboratorních úloh v prostředí Riverbed Modeler

POKYNY PRO VYPRACOVÁNÍ:

Náplň diplomové práce spočívá v prostudování problematiky komunikačních protokolů a následný návrh a vypracování tří nových laboratorních úloh v simulačním prostředí Riverbed Modeler Academic Edition. Laboratorní úlohy budou zahrnovat kompletní návody vhodné pro studenty včetně výchozího scénáře i doplňujících úkolů a kontrolních otázek. Časová náročnost každé úlohy musí být přibližně dvě hodiny. Při návrhu úloh se zaměřte na okruhy: porovnání transportních protokolů TCP, UDP, SCTP, rozbor aplikačních protokolů (FTP, HTTP, SMTP, DNS, popřípadě další), porovnání technologií ATM a Frame Relay, srovnání IPv4 a IPv6.

DOPORUČENÁ LITERATURA:

[1] Network Simulation Experiments Manual: Dokumentace k Riverbed Modeler, 2015. Dostupné online <https://booksite.elsevier.com/9780123850591/manual.php>, citováno 6.9.2017.

[2] JEŘÁBEK, J. Komunikační technologie. Skriptum FEKT Vysoké učení technické v Brně, 2016. s. 1-172.

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 21.5.2018

Vedoucí práce: Ing. Lukáš Langhammer, Ph.D.
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato diplomová práce se zabývá návrhem laboratorní úlohy do předmětu Komunikační technologie. Tento předmět je určený pro obor Teleinformatika v bakalářském studiu a měl by studentům poskytnout základní seznámení se síťovými protokoly a technologiemi. Z tohoto důvodu se práce zabývá základními transportními protokoly, protokoly pro přenos na páteřních sítích a protokoly síťové vrstvy.

Úvodní část slouží k seznámení se s návrhovým prostředím OPNET, přesněji s jeho bezplatnou verzí Riverbed Modeler Academic Edition 17.5, ve kterém je proveden návrh laboratorních úloh. Druhá část práce je věnována teoretickým znalostem k jednotlivým úlohám. První laboratorní úloha je zaměřena na rozdíly TCP (*Transmission Control Protocol*) a UDP (*User Datagram Protocol*) protokolů. Druhá laboratorní úloha je zaměřena na technologie pro přenos převážně na páteřních sítích WAN *(Wide Area Network)* a to ATM *(Asynchronous Transfer Mode)* a Frame Relay. Poslední navržená úloha se zabývá dvěma nejznámějšími internetovými protokoly, které se využívají pro komunikaci. Jedná se o IPv4 *(Internet Protocol version 4)* a IPv6 *(Internet Protocol version 6)* protokoly.

K této úloze byl vypracován návod pro studenty a v závěru každé z úloh jsou doplňující úkoly a otázky, které vedou k zamyšlení se nad danou problematikou.

KLÍČOVÁ SLOVA

Riverbed Modeler Academic Edition 17.5, laboratorní úloha, technologie, protokoly, TCP, UDP, ATM, Frame Relay, IPv4, IPv6.

ABSTRACT

This semestral thesis deals with the creation of a laboratory exercise for the course of Communication Technology. This course is designed for Teleinformatics in bachelor's degree program and should provide students with basic knowledge of network protocols and technologies. For this reason, the thesis deals with basic transport protocols, protocols for transmission via backbone networks and network layer protocols.

The introductory part introduces the OPNET design environment, more precisely its free version of Riverbed Modeler Academic Edition 17.5, where the design of the laboratory exercises is done. The second part is a necessary theory to the exercises. The lab exercise is focused on differences in TCP *(Transmission Control Protocol)* and UDP *(User Datagram Protocol)* protocols. The second laboratory exercises is focused on technologies for transmission mainly on WAN *(Wide Area Network)* namely ATM (Asynchronous Transfer Mode) and Frame Relay. The last proposed exercises deals with the two most prominent Internet protocols that are used for communications IPv4 *(Internet Protocol version 4)* and IPv6 *(Internet Protocol version 6)* protocols.

Guides for students have been created for this task, and at the end of each part, the complementary tasks and questions are given to students in oder to test the gained knowledge of the discussed issues.

KEYWORDS

Riverbed Modeler Academic Edition, laboratory exercise, technology, protocols, TCP, UDP, ATM, Frame Relay, IPv4, IPv6.

LOJEK, S. *Návrh laboratorních úloh v prostředí Riverbed Modeler*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2018. 108 s. Vedoucí diplomová práce Ing. Lukáš Langhammer, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svojí diplomové práci na téma Návrh laboratorních úloh v prostředí Riverbed Modeler jsem vypracoval samostatně pod vedením vedoucího diplomové práce, s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a jsou uvedeny v seznamu literatury na konci práce.

Jako autor uvedeného diplomového projektu dále prohlašuji, že v souvislosti s vytvořením tohoto diplomového projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních nebo majetkových, jsem si plně vědom následků porušení ustanovení § 11 a následujících 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.

V Brně dne ....

(podpis autora)

PODĚKOVÁNÍ

Rád bych věnoval poděkování Ing. Lukáši Langhammerovi, Ph.D. za odborné vedení, trpělivost, ochotu a nezištnou spolupráci při vypracování této diplomové práce.

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Technicka 12, CZ-61600 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

Tématem diplomové práce je návrh a realizace laboratorní úlohy do laboratorních cvičení v předmětu Komunikační technologie. Tento předmět se řadí v bakalářském studiu do povinných předmětů oboru Teleinformatika (TLI) a má za cíl poskytnout studentům orientaci v oblasti základních druhů komunikačních sítí a struktur určených pro přenos dat, hovorů a informací. Pro návrh laboratorní úlohy je použit simulační program Riverbed Modeler Academic Edition 17.5 od společnosti OPNET Technologies, Inc.

Úvodní část slouží k seznámení se s návrhovým prostředím OPNET, přesněji s jeho bezplatnou verzí Riverbed Modeler Academic Edition 17.5 a to včetně popsání všech jeho jednotlivých důležitých částí. Druhá část se věnuje teorii vztahující se k laboratorním úlohám, jež se zabývají transportními protokoly, přenosovými technologiemi a internetovými protokoly *(IP)*. Konkrétně se jedná o popis transportních protokolů TCP (*Transmission Control Protocol*) a UDP (*User Datagram Protocol*), technologií pro přenos převážně na páteřních sítích WAN *(Wide Area Network)* což jsou ATM *(Asynchronous Transfer Mode)* a Frame Relay a dále komunikační protokoly IPv4 *(Internet Protocol version 4)* a IPv6 *(Internet Protocol version 6)*.

Celkově v této práci jsou navrženy a vypracovány tři laboratorní úlohy. První úloha je zaměřena na rozdíly TCP a UDP protokolů, jako je například jejich rozdílná délka. Ve druhé laboratorní úloze se pracuje s technologií ATM, kde je například možné srovnat jednotlivé třídy služeb pro různé aplikace. V neposlední řadě zde dojde ke srovnání technologií ATM a Frame Relay. Poslední navržená úloha se zabývá konfigurací IP adres dvou internetových protokolů IPv4 a IPv6. Následně jsou tyto dva protokoly mezi sebou porovnány z hlediska odezvy pro různě velké sítě. Mimo jiné zde bude vidět odlišnost fragmentace v IPv4 a IPv6 protokolu.

Úkoly jsou koncipovány tak, aby se v první části student seznámil s danou problematikou příslušného úkolu a na konci každého úkolu jsou studentovi zadány samostatné otázky a úkoly, které by měl zvládnout vypracovat s pomocí nově nabytých znalostí. Otázky vedou k zamyšlení se nad danou problematikou. K laboratorní úloze jsou vypracována vzorová řešení a zdrojové soubory s vytvořenými scénáři v simulačním prostředí Riverbed Modeler Academic Edition 17.5.

1 SIMULAČNÍ PROSTŘEDÍ RIVERBED MODELER

Společnost OPNET Technology Inc. se zabývá vývojem síťových aplikací, jejich optimalizací a v neposlední řadě také vývojem simulačních programů, jako například SteelHead, SteelCentral, SteelFusion, SteelScript, SteelConnect a mnohé další. V rámci vzdělávání vytvořila firma Riverbed pro studenty a uživatele simulační aplikaci Riverbed Modeler Academic Edition 17.5, jež nahrazuje program IT Guru Academic Edition, který obsahuje nástroje pro všechny fáze studia včetně návrhů modelů sítí, jejich simulací, sběrů dat a jejich následné analýzy [1]. Tento program tedy umožňuje studentům lepší pochopení základních pojmů a principů jednotlivých síťových protokolů při vytváření sítě a tím i efektivní řešení a řízení infrastruktur reálných sítích. Velmi výhodné také je, že si tento produkt mohou studenti nainstalovat i doma, kde si stačí dle návodu v Riverbedu vygenerovat licenci a je možné ho používat.

1.1 Úvod do programu Riverbed Modeler

Jak již bylo zmíněno výše, tento program je silným nástrojem nejen pro návrh a následnou simulaci sítě, ale hlavně pro její analýzu. Je zde možné podrobně zkoumat jak síťové technologie a protokoly, tak i jejich mechanizmy. Riverbed poskytuje virtuální síťové prostředí, které modeluje reálné chování vytvořené sítě, včetně přepínačů, směrovačů, serverů, klientů a i velkého množství síťových protokolů. V tomto virtuálním síťovém prostředí je možné si nasimulovat i vlastní síť ze skutečného světa a případně zde diagnostikovat její problémy, efektivně ověřovat změny ještě před jejich skutečnou implementací a plánovat budoucí scénáře včetně jejich růstu nebo analyzovat případné selhání [2]. Pomocí Riverbed Modeleru lze identifikovat příčinu problémů výkonosti aplikací typu end-to-end a vyřešit je efektivně z hlediska nákladů. Také je možné díky tomu lépe porozumět a pochopit, jaké dopady mají určité změny v síti, ať už na prvcích v topologii nebo dalších možných nastaveních. Toto je možní díky tomu, že Riverbed má velké množství různých statistik. Tyto statistiky je možné si pro jednotlivé scénáře navolit podle toho, co je třeba pro vytvořenou síť zkoumat [1].

Tyto statistiky zde můžeme rozdělit na:

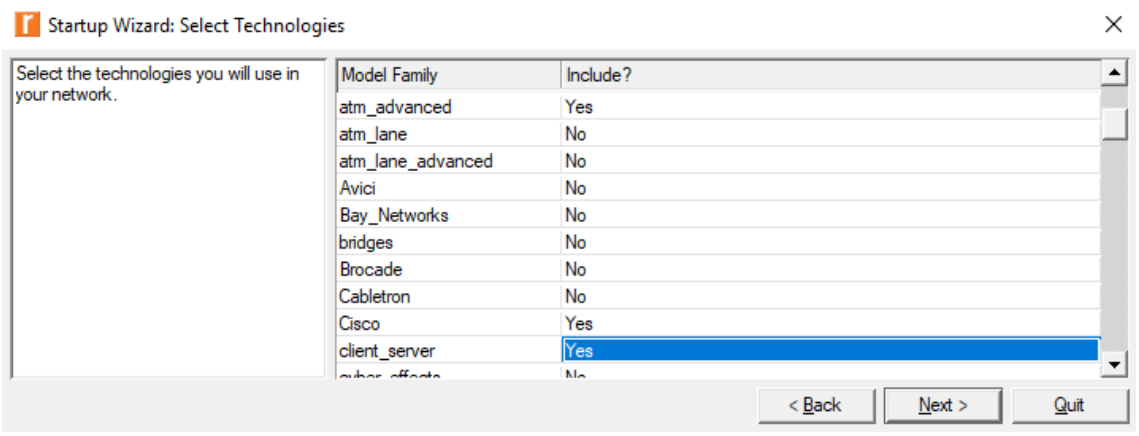
- Objektové statistiky, které popisují chování procesu v rámci konkrétního objektu popisovaného systému. V závislosti na tom, zda je daným objektem uzel (*Node*), požadavek (*Demand*) nebo linka (*Link*), se jednotlivé statistiky nazývají buď statistikou uzlu, statistikou požadavku nebo statistikou odkazu.
- Globální statistiky, které popisují chování konkrétního protokolu v celém simulovaném systému. Globální statistika je sdílena všemi objekty v simulaci a všechny tyto objekty přispívají k celkové hodnotě statistiky.

1.2 Založení nového projektu

Po spuštění Riverbedu je možné otevřít již vytvořený soubor se scénáři kliknutím v hlavním okně programu na **File > Open** nebo pomocí klávesové zkratky **Ctrl-O** nebo založit nový projekt pomocí **File > New.** Následně se otevře nové okno, kde se vybere možnost **Project**. Jako další se zadává *Jméno projektu* a *Jméno scénáře*.

Okna, která se budou zobrazovat dále, budou záviset na výběru, který byl proveden. Možnosti **Výběr měřítka** na mapě, kde je na výběr: *Svět, Podnik, Kampus, Kancelář* nebo si jde vybrat konkrétní místo na mapě. Možnosti *Podnik, Kampus* a *Kancelář* umožňují zadat rozměry simulované oblasti pomocí šířky a výšky obdélníkové plochy přes okno *Specifikace velikosti.*

V okně **Výběr technologie Obr. 1.1** se vybírají skupiny modelů, které budou použity, klepnutím na políčko *Zahrnout* je možné jej změnit z Ne na Ano a tím tento model aktivovat pro tento projekt. **Rodina modelů** je sbírka modelů, které patří do koherentního souboru technologií, jako je *internet_toolbox, ethernet, Cisco, atm, frame_relay* atd. Když se vybere jedna nebo více skupin modelů, objekty v těchto rodinách se stanou součástí výchozího modelu - *rodiny*, která se zobrazí při otevření **Palety objektů**. Nezávisle na výběru technologií (nebo dokonce i v případě, že se nevyberou vůbec žádné technologie), je v **Paletě objektů** vždy k dispozici celá sada objektů ze všech technologií. Následuje už jen okno s kontrolou všech nastavení pro tento projekt a poté už jen zbývá daný projekt vytvořit kliknutím na tlačítko **Dokončit**.



Obr. 1.1: Výběr technologie pro scénář.

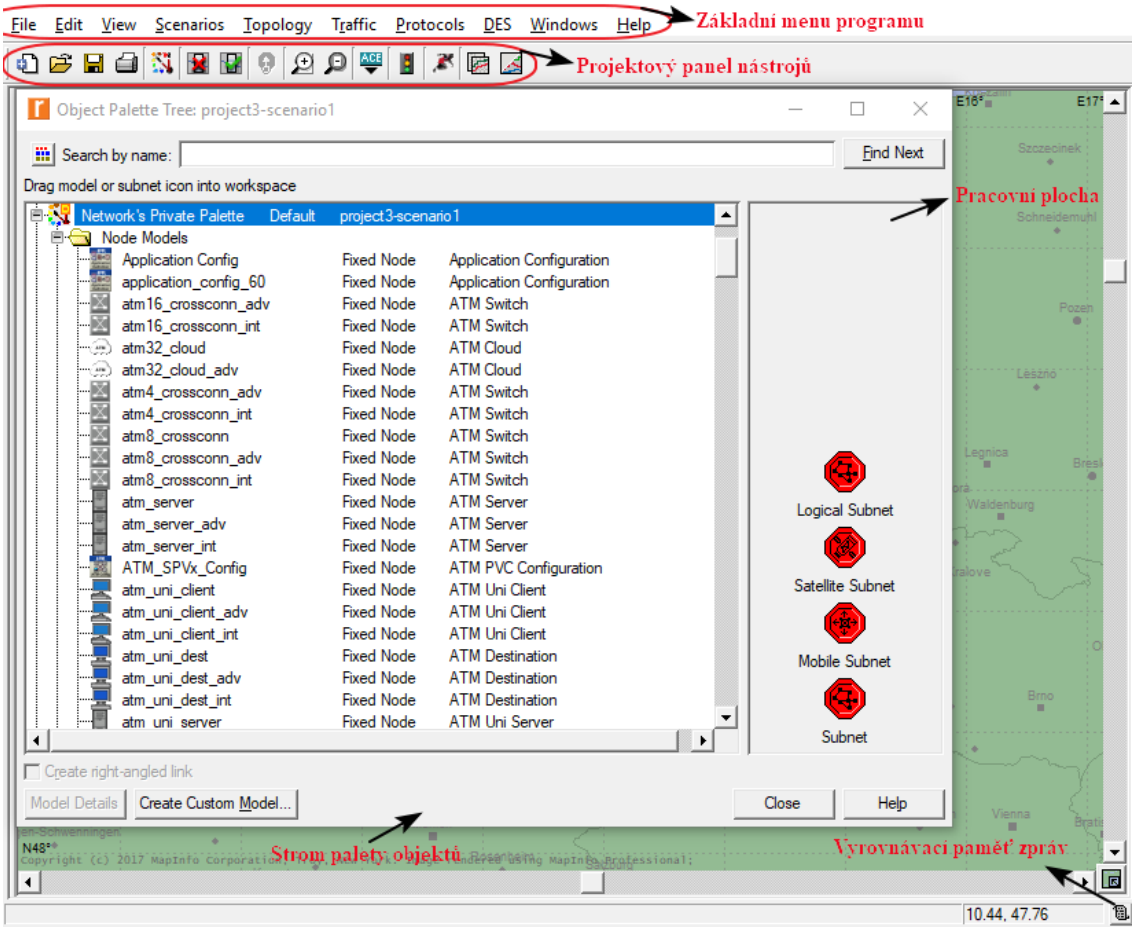
1.2.1 Paleta objektů a Editor projektu

Po vytvoření projektu se otevře okno **Paleta objektů** a **Editor projektu**. Nyní je možné vybrat objekty z Palety objektů a umístit je do pracovního prostoru projektu v Editoru projektu a také vytvářet a konfigurovat síť, která má být použita viz **Obr. 1.2.**

Paleta objektů organizuje dostupné modely do několika kategorií:

- **Paleta uzlových modelů** obsahuje dostupné modely uzlů komunikačních zařízení, jako jsou rozbočovače, přepínače, směrovače, brány, pracovní stanice a servery.

- **Linkové modely** obsahují modely linek, jako je 1000BASE-T Ethernet link, T1 duplexní spojení a 16 Mbps Token Ring.
- **Paleta Modely cest** obsahuje modely pro specifikaci síťových cest, které podporují takové technologie, jako je Encryptor internetového protokolu vysokorychlostního zabezpečení (HAIPE), Multiprotocol Label Switching (MPLS) a Veřejná přepínaná telefonní síť (PSTN).
- **Modely palety požadavku** obsahují modely pro určení provozních toků a připojení, jako je tok hlasové komunikace protokolu IP, hlasový přenos sítě PSTN a zabezpečení protokolu IP.
- **Modely palety bezdrátových domén** obsahují modely pro reprezentaci takových bezdrátových domén, jako mobility, sparse, grid a full grid.
- **Paleta sdílených objektů** obsahuje sadu uzlů, linek, cest, požadavků a doménových modelů seskupených podle společných vlastností. Například kolekce 3Com obsahuje modely přístrojů vyráběných společností 3Com Corporation, paleta aplikací obsahuje modely potřebné pro specifikaci a nasazení aplikací a skupina internet_toolbox obsahuje uzly, odkazy a užitné modely běžně používané pro modelování Internetu.



Obr. 1.2: Paleta objektů a Editor projektu v programu Riverbed.

1.2.2 Projektový panel nástrojů

Panel nástrojů, který je vidět na **Obr. 1.3**, zahrnuje nejdůležitější a nejčastěji používané funkce programu Rirverbed. Jsou to například *založení, otevření, uložení* nebo *tisk projektu*. Dále zde můžeme vyvolat *panel objektů*, vyřadit nebo obnovit vybrané objekty. K důležitým tlačítkům patří *přechod do nadřazení sítě*, přiblížení nebo oddálení pracovní plochy, případně spuštění a *nastavení parametrů simulace* a v neposlední řadě také *zobrazení výsledných statistik* pro daný scénář. Samotný panel nástrojů si lze přizpůsobit pomocí všudy přítomného horního menu **Obr. 1.2** (**Windows > Configure Toolbar**). Každá z těchto ikon nebo komponent na pracovní ploše má kontextovou nápověd, kterou je možné si zobrazit, když uživatel najede kurzorem myši na kteroukoliv z těchto ikon nebo komponent na ploše.

Obr. 1.3: Panel nástrojů.

1.2.3 Důležité atributy objektů

Objekt, který je přidaný z palety objektů, lze editovat po kliknutí pravým tlačítkem myši na nově přidaný objekt a výběrem položky Edit Attributes.

Zde jsou uvedeny některé atributy, které jsou důležité nebo se nejčastěji editují a nastavují při vytváření nového scénáře na přidaných objektech viz **Obr. 1.4.**

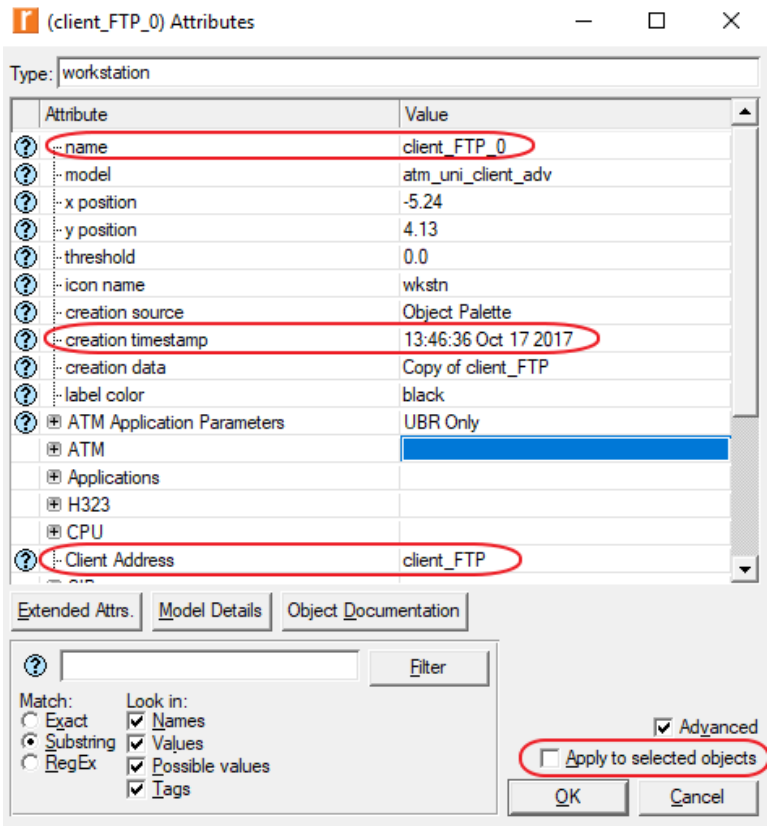
Jméno je výhodné změnit, a to hlavně z důvodu lepší orientace v zapojení na pracovní ploše. *Název modelu* z palety objektů. *Souřadnice pozice* na pracovní ploše. *Časová značka*, kdy byl objekt vytvořen a přidán na pracovní plochu. *Adresa klienta*, která musí být jedinečná pro každý uzel.

Parametry aplikace ATM zahrnují kombinaci kategorií služeb provozu a parametrů QoS, kde budou žádosti porovnány s podporovanými parametry kategorie, provozu a QoS na každém uzlu ATM. Pokud žádost nemůže být podporována, bude hovor odmítnut.

Záložka *ATM* slouží pro parametry používané pro konfiguraci ATM. Obsahuje například adresu, která určuje adresu ATM uzlu nebo konfiguraci fronty určující individuální mapování mezi frontami výstupních portů a podporovaným QoS.

V záložce *Aplikace* se nastavuje například mapování mezi symbolickými názvy cílů zadaných v objektech, dále je zde možné nastavit specifikaci multicastu, velikost segmentu, podporované profily, podporované služby a také transportní protokoly.

V neposlední řadě je možné provádět hromadné změny atributů pro všechny vybrané objekty zatrhnutím pole *Použití na vybrané objekty* (Apply to selected objects).



Obr. 1.4: Ukázka s atributy objektu stanice.

2 TRANSPORTNÍ PROTOKOLY

TCP a UDP jsou jednoduché základní protokoly v transportní vrstvě. [3] Tyto dva protokoly se mezi sebou liší. Zatímco TCP je spojově orientovaný a spolehlivý protokol, jež se stará o doručování ve správném pořadí, UDP je jeho pravým opakem, který nedává žádné záruky na doručení datagramu, a proto je označován jako nespolehlivý a nespojovaný protokol a není v něm tedy zaručeno, že přenášené datagramy nebudou doručeny vícekrát a v jiném pořadí než byly vyslány. [5]

2.1 TCP – Transmission Control Protokol

U TCP se datové jednotce říká segment, který vždy vyžaduje potvrzení, a protože je protokol TCP spojově orientovaný a potvrzovaný, zaručuje tedy, že pakety doručí ve správném pořadí a pokud dojde k jejich ztrátě během přenosu, tak se tyto pakety pošlou znovu. Záhlaví segmentu TCP má pevnou část o velikosti 20 bajtů a volitelnou část s variabilní délkou. Před samotným přenosem se jako první navazuje spojení, které se udržuje po celou dobu komunikace až do jeho ukončení. V praxi se tento protokol nachází prakticky ve všech sítových zařízeních a používají ho například služby FTP – *(File Transfer Protocol),* HTTP – (*Hyper Text Transfer Protocol),* SMTP – *(Simple Mail Transfer Protocol)* a další. [3], [5]

2.1.1 Segment TCP

Na **Obr. 2.1** jsou vidět položky záhlaví, které je mnohem rozsáhlejší než je tomu u UDP a obsahuje tyto části:

- **Zdrojový port** – 16 bitová hodnota, je to hodnota zdrojového portu na straně odesílatele.
- **Cílový port** – 16 bitová hodnota, jedná se o hodnotu indikující cílový port na straně příjemce.
- **Pořadové číslo odesílaného bajtu** – 32 bitová hodnota, pole obsahuje pořadové číslo prvního odeslaného bajtu. Pořadové číslo je náhodně zvolená hodnota.
- **Pořadové číslo potvrzovaného bajtu** – 32 bitová hodnota, zde je uvedena hodnota dalšího očekávaného bajtu.
- **Délka záhlaví** – 4 bitová hodnota, určuje celkovou délku záhlaví v násobcích 32 bitů.
- **Pole příznakových bit** – 6 bitová hodnota, mohou být mezi sebou různě kombinovány. Jejich význam pro nastavení na hodnotu „1“ je:

  -**URG** – segment obsahuje naléhavá data.

  -**ACK** – daný segment slouží i jako potvrzovací dříve přijatých dat.

  -**PSH** – udává, že data mají být po přijetí ihned bez čekání předána aplikaci.

  -**RST** – slouží k odmítnutí spojení.

  -**SYN** – využívané při navazování spojení, když začala nová sekvence číslování.

-FIN –při ukončení spojení.

- **Délka okna** – 16 bitová hodnota, specifikuje, kolik dat je přijímač schopen přijmout bez jejich potvrzení. Hodnota se mění dle potřeby.
- **Kontrolní součet** – 16 bitová hodnota, vypočítává se z celkového záhlaví TCP a z části záhlaví IP protokolu.
- **Ukazatel naléhavých dat** – 16 bitová hodnota, ukazuje na první datový oktet, který následuje po naléhavých datech a je vyplněn pouze v případě, že je u URG nastavena hodnota „1“.
- **Volitelné a doplňující položky záhlaví** – pole může zůstat nevyplněno a jeho délku lze odvodit z celkové délky záhlaví.

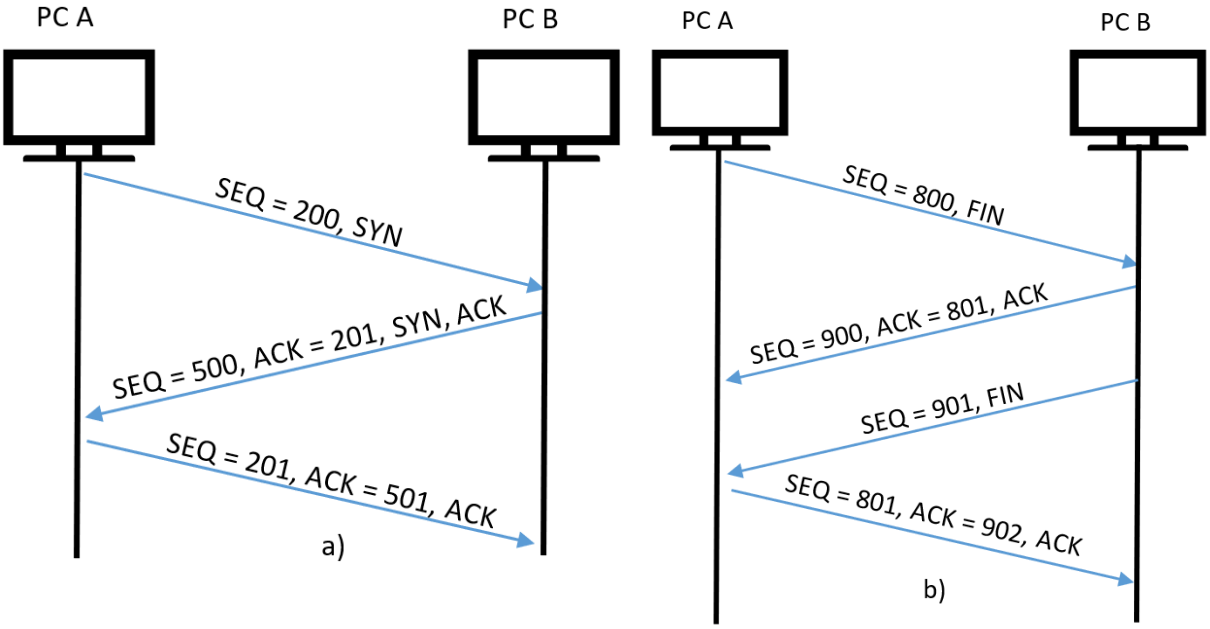
<table>
<tr><td colspan="8">Bity 0-15</td><td>16-31</td></tr>
<tr><td colspan="8">Zdrojový port</td><td>Cílový port</td></tr>
<tr><td colspan="9">Pořadové číslo odesílaného bajtu</td></tr>
<tr><td colspan="9">Pořadové číslo potvrzovaného bajtu</td></tr>
<tr><td>Délka záhlaví</td><td>Rezerva</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Délka okna</td></tr>
<tr><td colspan="8">Kontrolní součet</td><td>Ukazatel naléhavých dat</td></tr>
<tr><td colspan="9">Volitelné položky záhlaví</td></tr>
<tr><td colspan="9">Data aplikace</td></tr>
</table>

Obr. 2.1: TCP segment. [5]

2.1.2 Navazování a ukončení spojení u protokolu TCP

TCP protokol vytváří při navázání spojení dvou komunikujících stran virtuální okruh a po přenosu dat následně toto spojení i ukončí. Samotnému navázání spojení se říká **tree-way handshake**, které je vidět na **Obr. 2.2** a), kde *PC A* navazuje spojení a vysílá segment s příznakovým bitem SYN a tím dojde k požadavku na synchronizaci číslování přenášených bajtů a toto číslo si nastaví do pole SEQ. *PC B* potvrdí přijetí segmentu s příznakovým bitem ACK a také vyšle svoji synchronizaci SYN. *PC A* zde náhodně zvolil hodnotu číslování 200, a proto musí dostat potvrzení ACK od *PC B* s dalším očekávaným bajtem s hodnotou 201. Po přijetí druhého segmentu ví *PC A*, že je druhá strana dostupná a připravena komunikovat, a proto odesílá poslední třetí segment, který dokončí navazování spojení. V tomto segmentu se opět posílá příznakový bit ACK a do potvrzovaného bajtu dá číslo 501 (číslo vygenerované stanicí A bylo 500), kterým potvrdí náhodně vygenerované číslo od *PC B.* Je zde samozřejmě i SEQ, které pokaždé vzroste minimálně o hodnotu 1.

Ukončení spojení je znázorněno na **Obr. 2.2** b). které probíhá podobným způsobem jako jeho navázání. Standardně probíhá ve čtyřech korcích a využívá se zde příznaků FIN, který slouží pro ukončení spojení od odesilatele a ACK.[5]



Obr. 2.2: a) navázání spojení TCP, b) ukončení spojení TCP. [5]

2.2 UDP – User Datagram Protocol

Použití protokolu UDP je vhodné tam, kde je vyžadováno co možná nejmenší zpoždění jako například VoIP *(Voice over IP),* kde není kritické, že dojde ke ztrátě datagramů, protože v tomto případě přenáší datagram malou část slova a tím pádem při jeho ztrátě není narušena celková srozumitelnost přenášených dat. Proto je zde nevhodné tyto datagramy posílat znovu, protože by vzrostlo zpoždění a zpomalilo by to komunikaci. Proto pokud je důležité pořadí datagramů, jako je tomu právě u VoIP, musí to být řešeno na aplikační úrovni. Protokol se také hodí pro přenos krátkých zpráv typu otázka-odpověď v systémech typu DNS – (*Domain Name Systém),* kde pro komunikaci mohou stačit dva datagramy. Naproti tomu u protokolu TCP jich je potřeba minimálně devět jednotek. Dále se UDP využívá například pro DHCP - *Dynamic Host Configuration Protocol*, RIP - *Routing Information Protocol*, SNMP - *Simple Network Management Protocol*. [3] [5]

2.2.1 Datagram UDP

Záhlaví UDP je proti TCP maximálně zjednodušeno a obsahuje pouze čtyři šestnáctibitová pole, která mají tedy dohromady 8 bajtů, jak je vidět z **Obr. 2.3**. Jsou to stejně jak u TCP zdrojový a cílový port. Zdrojový port je volitelný a může být vynechán. Pokud odesilatel nevyžaduje odpověď, musí být zdrojový port nastaven na nulu. Následuje celková délka datagramu v bajtech a to včetně záhlaví. Jako poslední je zde kontrolní součet, tvořený jak z hlavičky, tak z dat aplikace.

Bity 0-15	16-31
Zdrojový port	Cílový port
Celková délka	Kontrolní součet
Data aplikace	

Obr. 2.3: Záhlaví UDP. [3]

3 PROTOKOLY PRO PŘENOS NA PÁTEŘNÍCH SÍTÍCH

WAN sítě spojují rozsáhlou oblast s dosahem stovek až tisíců kilometrů a mohou tak propojovat i celé kontinenty. Bývají budovány na pronajatých linkách, které mohou být velmi drahé. Mezi používané metody patří přepojování paketů a přepojování okruhů. Tato kapitola se věnuje protokolům ATM – *(Asynchronous Transfer Mode)* a Frame Relay, které jsou v dnešní době využívány. [3] [6]

3.1 ATM - Asynchronous Transfer Mode

Tato technologie využívá asynchronní přenosový režim s přepojováním paketů na spojové vrstvě. ATM buňky mají pevně stanovenou délku 53 bajtů. Každá buňka se skládá z 5 bajtů hlavičky a 48 bajtů informační části, viz **Obr. 3.1**. Tato délka ATM buňky vznikla jako kompromis pro začlenění různých druhů služeb do jedné sítě jako jsou telefonní hovory a přenos dat. [6]

Obr. 3.1: Formát ATM buňky. [3]

3.1.1 Jednotlivé třídy provozu

Jak již bylo zmíněno, ATM technologie vznikla jak pro hlasové služby, tak zároveň i pro data. Hlas i data mají své specifické nároky a požadavky na přenosovou síť a z tohoto důvodu jsou v ATM adaptační vrstvě přesně určeny ATM třídy služeb, jež definují parametry, které musí být dodrženy u konkrétního datového toku. Označují se písmeny A, B, C, D, kde třída A a B slouží pro synchronizované spojení a třídy C a D se používají pro nesynchronizované spojení. [7]

Constant Bit Rate (CBR) – třída CBR se označuje písmenem A. Protože využívá konstantní bitovou rychlost, používá se pro služby, které vyžadují nízké zpoždění a kolísání u přenosové rychlosti. Třída CBR je typicky používána například pro přenos nekomprimovaného hlasu.

Variable Bit Rate (VBR) – jedná se o třídu služeb s variabilní přenosovou rychlostí, která je také označována písmenem B. V tomto případě technologie ATM vychází vstříc přenosům v reálném čase, které nevyžadují konstantní přenosovou rychlost. Jedná se především o komprimované video, obrazové a zvukové přenosy, kde je proměnlivá přenosová rychlost, avšak ostatní požadavky, jako jsou zpoždění a kolísání zpoždění, zůstávají stejné.

Available Bit Rate (ABR) – třída ABR zaručuje minimální předem dohodnutou přenosovou rychlost. Pokud bude mít linka dostatečnou přenosovou kapacitu, může na požádání navýšit přenosovou rychlost. Označuje se písmenem C a používá se všude tam, kde nejsou nároky na zpoždění a kolísání zpoždění. Třída je vhodná pro přenos dat.

Unspecified Bit Rate (UBR) – oproti třídě ABR nemá UBR předem dohodnutou minimální přenosovou rychlost, a protože patří do poslední skupiny s písmenem D, jedná se o nesynchronizované spojení. Třída UBR může využít jen takovou kapacitu přenosové linky, která je momentálně k dispozici a pokud bude přenosová linka vytížena jinými službami s vyšší prioritou, je možné, že se služby třídy UBR nedostanou vůbec na řadu a nebudou tak obslouženy. Třída UBR bývá také označována jako best effort. [7]

3.1.2 Protokoly adaptační vrstvy

V doporučení ITU-T I.363 jsou definovány protokoly používané na ATM adaptační vrstvě, které jsou použity pro kvalitu služeb ATM a označují se jako AAL (*ATM Adaptation Layer).* AAL shromažďuje vzorky řečového signálu a provádí segmentaci datových jednotek vyšších vrstev do ATM buněk. Případně provádí konverzi mezi pevnými délkami ATM buněk se zprávami, které jsou užívány ve vyšších vrstvách. [7]

- **AAL1 –** používá pro třídu CBR a podporuje synchronní spojově orientovaný provoz
- **AAL2 –** používá pro třídu VBR, s podporou synchronního spojově orientovaného provozu
- **AAL3 – AAL5 –** používá se pro ABR a UBR s podporou spojově i nespojově orientovaného datového provozu

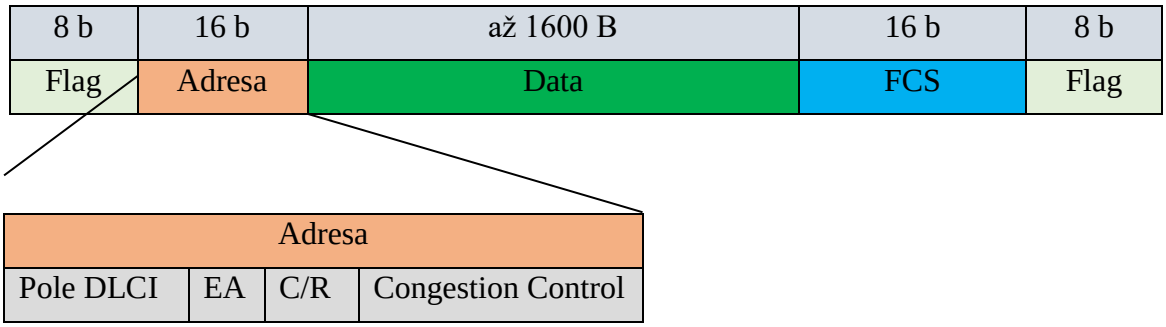
3.1.3 ATM síťová vrstva

ATM síťová vrstva se stará o doručení ATM buněk od odesilatele k příjemci. Pro doručení ATM buňky je zapotřebí dopředu sestavit virtuální spojení mezi odesilatelem a příjemcem. Virtuální spojení může být buď VCC - *(Virtual Channel Connection)* nebo VPC - *(Virtual Path Connection)*. Tato spojení, která se nacházejí mezi dvěma body a virtuální cestou, lze zapsat identifikátory virtuálních kanálů VCI – *(Virtual Channel Identifier)* a VPI – *(Virtual Path Identifier).* Čísla VCI a VPI jsou uložena v 5 bajtovém záhlaví ATM buňky, kde VCI identifikuje buňku virtuálního spojení a VPI určí kanál se stejnou cestou a tím urychluje spojení v ATM uzlech. [8]

3.2 Frame Relay

Frame Relay je založena na komutaci paketů ve WAN sítích. Stejně tak jako ATM, byl i Frame Relay původně navržen pro používání v ISDN sítích, ale postupem času se stal univerzálním protokolem. [3] Využívá spojově orientovaný přenos a před zahájením přenosu dat musí být vytvořen virtuální okruh VC – *(Virtual Circuit).* Představitelem virtuálního okruhu je číselný identifikátor DLCI - *(Data Link Connection Identifier),* který předává data sítí. VC mohou být dvojího druhu: PVC - *(Permanent Virtual Circuit)* nebo SVC - *(Switched Virtual Circuit).* PVC jsou permanentně zřízeny v dané WAN síti mezi síťovými zařízeními. Naproti tomu se SVC liší v tom, že komutovaný virtuální spoj se musí sestavit před každým datovým

přenosem. Frame Relay umožňuje přenos s proměnnou délkou rámce a disponuje menším záhlavím, tím se stává efektivní i při menších přenosových rychlostech. Formát rámce je znázorněn na **Obr. 3.2**.



Obr. 3.2: Formát rámce Frame Relay. [3]

Flag nebo také signalizace či návěští má 8 bitů. Slouží k rozpoznání začátku a konce rámce. Má vždy stejnou sekvenci binárních čísel 01111110.

Adresa může mít až 16 bitů a obsahuje:

- pole DLCI je 10 bitový identifikátor virtuálního okruhu, kterým je rámec přenášen.
- EA – *(Extended Address)* 2 bitový, rozšiřuje formát DLCI.
- C/R – *(Command response bit)* 1 bitový, není nepoužíván.
- Congestion Control – 3 bitový a oznamuje zahlcení Frame Relay sítě a to včetně uvedení typu zahlcení.

Data – nacházejí se zde uživatelem přenášená data a nemá pevně stanovenou délku. Pole je standardně dlouhé až 1600 bajtů.

FCS – *(Frame Check Sequence)* 16 bitový CRC kód pro zabezpečení a kontrolu rámce. [9] [3]

4 SROVNÁNÍ PROTOKOLŮ IPV4 A IPV6

Internet Protocol – *(IP)* je hlavním komunikačním protokolem, který pracuje na síťové vrstvě. IP má za úkol přenášet pakety od zdroje k cíli a to pouze na základě IP adres v hlavičkách paketů. Za tímto účelem IP definuje struktury paketů, jež zapouzdřují data, která mají být doručena. Rovněž definuje metody adresování, které se používají k označování datagramu se zdrojovou a cílovou informací. První z hlavních verzí IP byl Internet Protocol verze 4 – *(IPv4)*, který je dodnes dominantním protokolem na celém internetu. Jeho nástupcem je Internet Protocol verze 6 – *(IPv6)*. [3] [11]

4.1 IPv4 – Internet Protocol Version 4

IPv4 vznikla v roce 1981 a podrobně je popsána v dokumentu RFC 791. Jedná se o paketově orientovaný protokol, který je používán v sítích s přepojováním paketů, jako je například Ethernet. Funguje způsobem best effort nebo jinými slovy nezaručuje doručení ani zachování pořadí paketů nebo jejich případnou duplicitu při průchodu sítí. Díky IP adrese, která je globálním unikátním identifikátorem, je možné jednoznačně určit konkrétní stanice v rámci Internetu a směrovat data, která se přenášejí sítí k jejich cíli. [11]

4.1.1 IP adresy

Pro tento protokol jsou IP adresy 32-bitové a adresní prostor je na 2^{32} adres. Nejčastěji jsou psány tečkovou desítkovou notací, která se skládá ze čtyř oktetů adresy a každá část se pak vyjádří jako celé desítkové číslo oddělené tečkou. IP adresa je tvořena číselnou adresou sítě a číselnou adresou hostitelského počítače. IP adresy je možné dělit na třídy označené písmeny podle počtu bitů na sítě velké *A*, střední *B* a malé *C*, viz **Tab. 4.1**. Nicméně toto rozdělení se časem ukázalo jako neefektivní. Z tohoto důvodu se začala používat technika podsíťování, díky které je možné hierarchicky rozdělit IP sítě na další sítě. Základ podsíťování je, že IP adresa je 32 bitové číslo, kterou prefix rozděluje na výše zmíněnou číselnou adresu sítě a číselnou adresu hostitelského počítače.

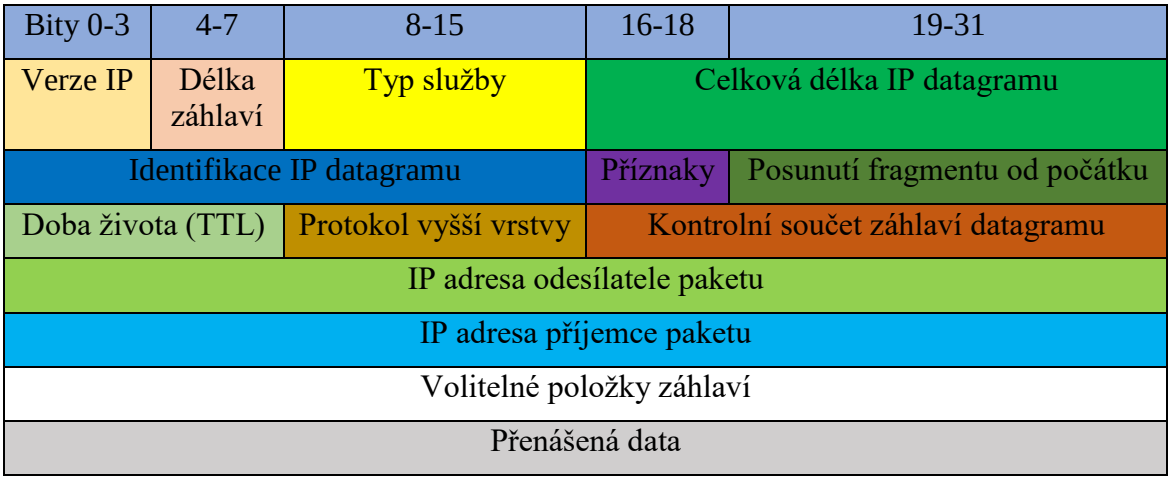
Tab. 4.1 Třídy IP adres.

Třída	Rozsah prvního oktetu adresy	Maska sítě	Počet možných sítí	Počet možných hostů na jednu síť
A	0 – 127	255.0.0.0	128	16 777 214
B	128 – 191	255.255.0.0	16 383	65 534
C	192 – 223	255.255.255.0	2 097 150	254
D	224 – 239	Multicastové adresy		
E	240 – 255	Experimentální adresy		

Maska sítě je tvořena nahrazením všech bitů vyhrazených pro adresu sítě binárními jedničkami. Z tohoto vzniklého čísla se po převodu na desítkové číslo stává maska sítě, kterou je možné zapsat i jako takzvanou délku prefixu, která v masce vyjadřuje počet jedniček a píše se s lomítkem za IP adresou. [3] [11]

4.1.2 Formát IPv4 datagramu

IPv4 datagram je možné rozdělit na sekci se záhlavím a na datovou část. V sekci se záhlavím se nacházejí údaje pro jeho přepravu. Kromě TTL, který vyjadřuje životnost, zůstává datagram po celou dobu přenosu od zdroje k cíli ve stejném formátu nezměněn. Struktura IPv4 datagramu je znázorněna na **Obr. 4.1**.



Obr. 4.1: Záhlaví IPv4 datagramu. [3]

Popis jednotlivých polí:

- **Verze** – 4 bitová hodnota, obsahuje verzi IP protokolu, pro IPv4 je to hodnota 4.
- **Délka záhlaví** – 4 bitová hodnota. Minimální délka záhlaví je 20 bajtů, naopak maximální délka je 60 bajtů. Délka záhlaví může být proměnlivá, ale vždy musí být v násobcích 32 bitů.
- **Typ služby** – 8 bitová hodnota, která se v dnešní době využívá pro služby s definovanou kvalitou služby neboli QoS.
- **Celková délka IP datagramu** – 16 bitová hodnota. Maximální délka datagramu je 65536 bajtů.
- **Identifikace IP datagramu** – 16 bitová hodnota, má stejnou hodnotu jako původní datagram, díky tomu je možné poznat, které fragmenty patří k sobě.
- **Příznaky** – 3 bitová hodnota, pokud je zde nataven DF-bit *(don't fragment),* datagram nebude fragmentován. V případě MF-bit *(more fragments)* byl datagram fragmentován a bude následovat další část fragmentu.
- **Posunutí fragmentu od začátku** – 13 bitová hodnota, udává posun datové části fragmentu od části původního datagramu.

- **Doba života datagramu** – 8 bitová hodnota, jedná se o počet skoků, který může datagram udělat k cíli než bude zahozen.
- **Protokol vyšší vrstvy** – 8 bitová hodnota, udává číselnou hodnotu protokolu z vyšší vrstvy, který bude dále s datagramem pracovat.
- **Kontrolní součet záhlaví datagramu** – 16 bitová hodnota, pokud kontrolní součet nebude validní, dojde k zahození paketu.
- **IP adresa odesilatele/příjemce paketu** – 32 bitová hodnota, IPv4 adresy odesilatele a příjemce.
- **Volitelné položky záhlaví** – až 40 bajtová hodnota, nemusí být využita.
- **Přenášená data** – maximální možná délka je 65536 včetně záhlaví.

4.1.3 Fragmentace IP datagramů

Protože je velikost paketu rozdílná v různých sítích, viz **Tab. 4.2**, musí být k dispozici mechanizmy, které rozdělí příliš velké datagramy na menší části označované také jako fragmenty. Fragmenty se mohou k příjemci šířit různými cestami. Z tohoto důvodu se fragmenty sestavují až u příjemce. Fragmentované datagramy je možné dále fragmentovat a v IPv4 může fragmentovat jak odesílající uzel, tak i kterýkoliv směrovač na cestě. Jak již bylo zmíněno výše, maximální hodnota datagramu je 65536 bajtů a je možné ji také označit jako MTU – *(Maximum Transmission Unit)*. Může nastat situace, kdy fragmentace není možná, je nastaven příznakový bit na DF *(don't fragment)*. V takovém případě je na směrovači datagram zahozen a odesilatel je informován o chybě a nemožnosti fragmentace pomocí ICMP zprávy.

Tab. 4.2: Linkové protokoly a jejich MTU.

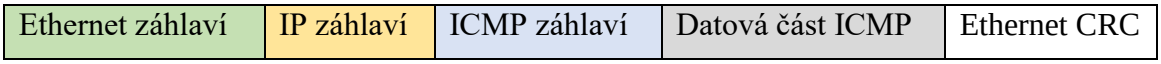
Linkový protokol	MTU [bajty]
ATM	48
Ethernet 802.3 SNAP	1492
Ethernet II	1500
Frame Relay	1600
FDDI	4478

4.1.4 ICMP – Internet Control Message Protocol

Byl vyvinut jako doplňující protokol k protokolu IP, který postrádal hlášení o chybách a nestandardních situacích. Nepřenáší uživatelská data. Stal se součástí sady TCP/IP protokolů a přenáší se přímo v IP datagramech, viz **Obr. 4.2**. Je implementován i na směrovačích, které generují ICMP zprávy nejčastěji. Pokud dojde k nestandardnímu stavu, je odesilatel informován jednou z ICMP zpráv, jako například:

- **Vypršel čas** – *(Time Exceeded)*
- **Nedosažitelný cíl** – *(Destination Unreachable)*

- **Hrozí zahlcení** – *(Source Quench)*
- **Přesměrování** – *(Redirect)*
- **Testování dostupnosti** – *(Echo Request/Reply)*



Obr. 4.2: ICMP paket přenášený v síti Ethernet. [3]

4.2 IPv6 – Internet Protocol Version 6

IPv6 vznikl jako odpověď na blížící se vyčerpání adresního prostoru IPv4, nicméně řeší i další její nedostatky, jako stále narůstající velikost směrovacích tabulek nebo neustále se zvyšující nároky na přenosovou rychlost, zejména kvůli přenosu multimediálních dat. S IPv4 protokolem došlo k velkému rozšíření techniky NAT, díky které není možná skutečná komunikace modelu end-to-end. Protokol IPv6 vznikl jako celá sada protokolů, do které se řadí DHCPv6, ICMPv6, mobilita stanic, multicast a další. IPv4 a IPv6 mezi sebou bohužel nejsou kompatibilní, proto muselo být navrženo několik mechanizmů, které umožňují jejich konverzi, jako například jednoduché tunelování protokolů, Dual stack, DSlite nebo překlad protokolů IPv6 a IPv4 *(NAT64)*. [3] [10]

4.2.1 Výhody a nevýhody IPv6

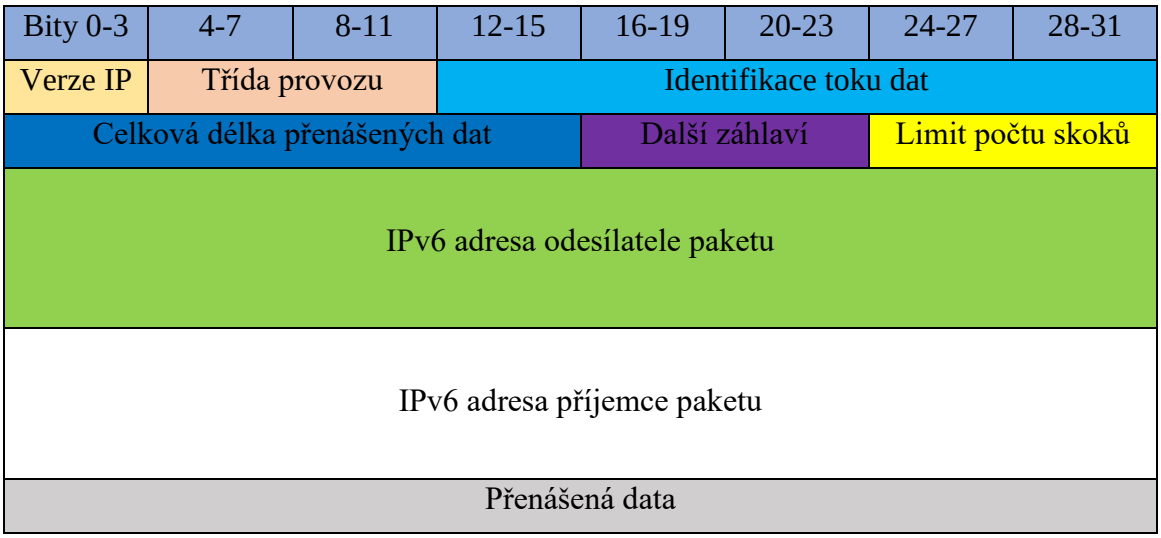
Mezi hlavní výhody IPv6 patří rozšíření adresního prostoru z 2^{32} na 2^{128} adres, který by měl být navržen tak, aby byl již navždy dostatečný. Došlo k zjednodušení formátu záhlaví, ve kterém je nyní méně povinných položek. Zredukování velikosti směrovacích tabulek na směrovačích, což umožní podporu hierarchického směrování. Díky tomu, že se neprovádí fragmentace paketů v průběhu cesty a došlo k odstranění položky kontrolního součtu *(CRC)*, dojde ke zrychlení směrování ve směrovacích tabulkách a tím k nižšímu zpoždění. Jsou zde předpřipravené mechanizmy pro zajištění kvality *(QoS)*. IPv6 umožňuje větší délku paketů takzvaných *„Jumbogramů“* o velikosti až 4 GB/jednotka, což může snížit režii, a naopak zvýšit průchodnost přenosové trasy. V protokolu Adresy IPv6 lze rozdělit na individuální *(unicast)*, skupinové *(multicast)* a výběrové *(anycast)*.

I když je rozšíření adresního prostoru v IPv6 velkou výhodu, může být z hlediska administrátora takto velký adresní prostor i nevýhodou v případě zjišťování přítomnosti nebo nepřítomnosti IP adres pomocí pingu. IPv6 adresy nejsou tak dobře zapamatovatelné jako tomu je v případě IPv4 adres. Druhou velkou nevýhodou je, že IPv6 není zcela dořešena z hlediska bezpečnosti, respektive to, co bylo vyřešeno v IPv4, nemusí být vyřešeno pro IPv6. Poslední velkou nevýhodou je, že v IPv4 není možný okamžitý přechod na IPv6. [3] [11]

4.2.2 IPv6 datagram

Oproti IPv4 datagramu se IPv6 datagram maximálně zjednodušil. Jsou zde vynechány položky fragmentace, kontrolního součtu, rozšiřujících voleb a délky

záhlaví. Fragmentace se v IPv6 neprovádí tak často a primárně se využívají délky s 1500 B, minimální velikost fragmentu je 1280 B. IPv6 datagram dostal pevnou délku hlavičky 40 bajtů, z nichž 32 B tvoří adresy odesilatele a příjemce a obsahuje již jen ty nejdůležitější položky, i když se délka ve srovnání s IPv4 prodloužila na čtyřnásobek, viz **Obr. 4.3**.



Obr. 4.3: Záhlaví IPv6 datagramu. [3]

Popis jednotlivých polí:

- **Verze** – 4 bitová hodnota, stejně jako u IPv4 zajišťuje kompatibilitu s ostatními aplikacemi, které budou dále využívat jednotlivá pole datagramu.
- **Třída provozu** – 8 bitová hodnota, pole pro zajištění kvality služeb.
- **Identifikace toku dat** – 20 bitová hodnota, protože mají spolu související pakety nakonfigurovanou stejnou hodnotu, může je směrovač odeslat stejnou cestou a tím zjednodušit a urychlit směrování. Tato funkce však zatím není standardně implementována.
- **Celková délka přenášených dat** – 16 bitová hodnota, velikost dat je bez základního záhlaví a jeho maximální délka může dosahovat až 64 kB.
- **Další záhlaví** – 8 bitová hodnota, jedná se o rozšiřující informace o fragmentaci nebo zde může být odkaz na přítomnost protokolu z vyšší vrstvy, například protokolů TCP a UDP.
- **Limit počtu skoků** – 8 bitová hodnota, koresponduje s TTL u IPv4, jejímž cílem je zabránění zacyklení paketů.
- **IP adresa odesilatele a příjemce paketu** – 128 bitová hodnota jak pro odesilatele, tak i pro příjemce.

4.2.3 ICMPv6 – Internet Control Message Protocol Version 6

Jedná se o novou verzi protokolu ICMP pro podporu protokolu IPv6, bez něhož by samotná IPv6 nemohla fungovat. Stejně jako v předchozí verzi slouží k nahlašování

chybových stavů při přenosu paketů. ICMPv6 je víceúrovňový protokol, který disponuje novými funkcemi. Je například možné objevovat sousedy, získávat základní informace o uzlu. Má implementovanou podporu správy multicástových skupin, podporuje zajištění mobility a překlad adres. Zprávy ICMPv6 jsou přenášeny v rozšířené hlavičce uvnitř IPv6 datagramů. Formát zprávy ICMPv6 je znázorněn na **Obr. 4.4**. [3] [11]

Bity 0-7	8-15	16-31
Typ	Kód	Kontrolní součet
Tělo zprávy		

Obr. 4.4: Formát zprávy ICMPv6. [3]

4.2.4 Hlavní výhody IPv6 oproti IPv4

- Rozšíření adresního prostoru
- Zjednodušené záhlaví, které má méně povinných položek
- Je zde snaha o zredukování směrovacích tabulek a tím spjatý menší počet větších sítí
- Zrychlení směrování, kde se v IPv6 nepočítá CRC a nedochází k fragmentaci
- Přímá komunikace end-to-end bez použití NAT
- Nové podpůrné protokoly
- Předpřipravené mechanizmy pro zajištění kvality služeb

5 SROVNÁNÍ TCP A UDP PROTOKLŮ

5.1 Úvod k laboratorní úloze

TCP a UDP jsou důležité protokoly transportní vrstvy. U TCP se datové jednotce říká segment, který vždy vyžaduje potvrzení, a protože je protokol TCP spojově orientovaný a potvrzovaný, zaručuje tedy, že pakety doručí ve správném pořadí, a pokud dojde k jejich ztrátě během přenosu, tak se tyto pakety pošlou znovu. Záhlaví segmentu TCP má pevnou část o velikosti 20 bajtů a volitelnou část s variabilní délkou. Před samotným přenosem se jako první navazuje spojení, které se udržuje po celou dobu komunikace až do jeho ukončení. V praxi se tento protokol nachází prakticky ve všech síťových zařízeních a používají ho například služby FTP, HTTP, SMTP a další. Použití protokolu UDP je vhodné tam, kde je vyžadováno co možná nejmenší zpoždění, jako například VoIP, kde není kritické, že dojde ke ztrátě datagramů, protože v tomto případě přenáší datagram malou část slova a tím pádem při jeho ztrátě není narušena celková srozumitelnost přenášených dat. Proto je zde nevhodné tyto datagramy posílat znovu, protože by vzrostlo zpoždění a zpomalilo by to komunikaci. Proto pokud je důležité pořadí datagramů, jako je tomu právě u VoIP, musí to být řešeno na aplikační úrovni. Protokol se také hodí pro přenos krátkých zpráv typu otázka-odpověď v systémech typu DNS, kde pro komunikaci mohou stačit dva datagramy. Naproti tomu u protokolu TCP jich je potřeba minimálně devět jednotek. Dále se UDP využívá například pro DHCP, RIP.

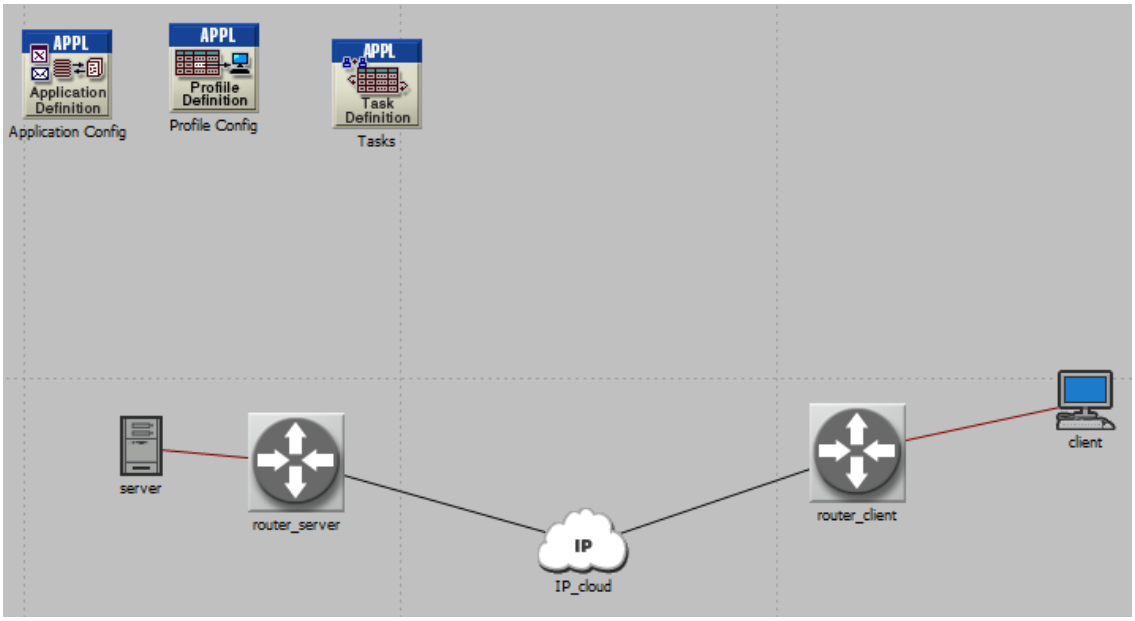
Laboratorní úloha se bude zabývat porovnáváním základních protokolů transportní vrstvy. Při práci s protokoly TCP a UDP budete seznámeni se základními principy a ověříte si zde své teoretické poznatky v praxi. V dalších scénářích si vyzkoušíte, jak se chová TCP a UDP při zahazování paketů. Podíváte se na potvrzování přenášených segmentů a na chování při FTP a Voice při výpadku na lince. Poslední úkol je zaměřen na postupné navyšování provozu, kde budete sledovat chování FTP aplikace a Videokonference.

5.2 Úkol 1 – základní srovnání TCP a UDP

V prvním úkolu si stáhnete a nahrajete do Riverbed Modeleru vypracovaný projekt s hotovými scénáři pro TCP a UDP. Zde se podíváte na tyto základní statistiky a porovnáte je mezi sebou.

5.2.1 Pracovní postup

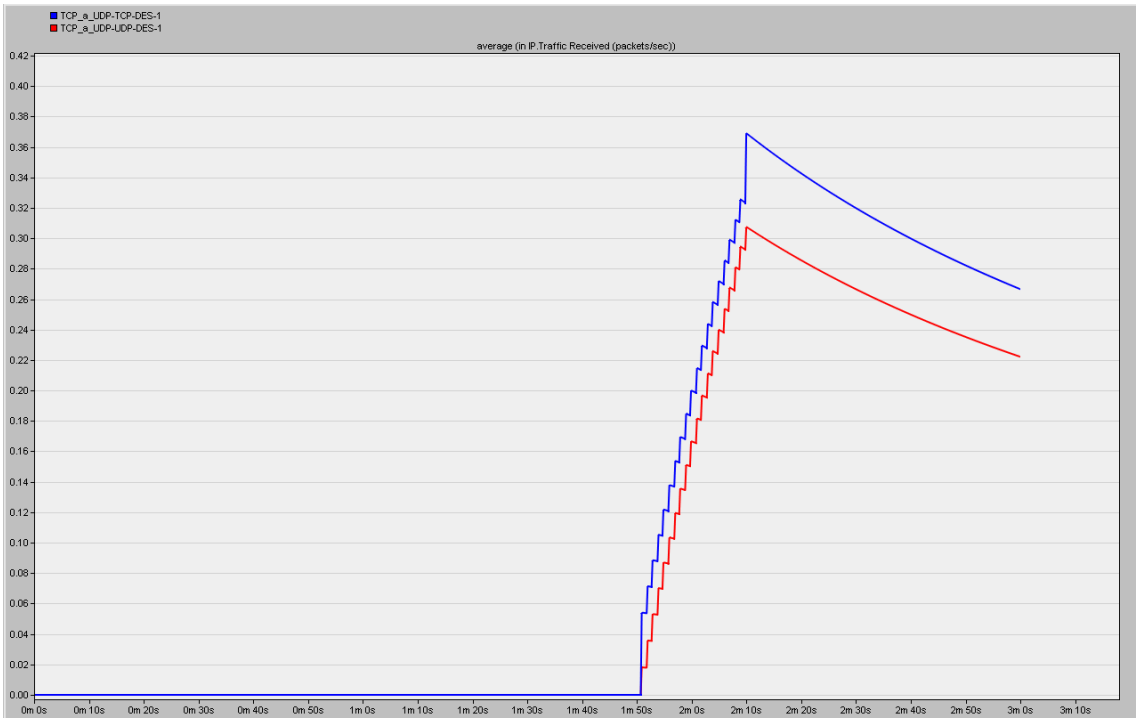
1. Stáhněte vypracovaný projekt *„TCP_a_UDP“* a uložte jej například na **Plochu**.
2. Spusťte program Riverbed Modeler.
3. V menu programu vyberte **File/Open…(**Ctrl + O) a otevřete projekt *„TCP_a_UDP“*. Zobrazí se okno s připravenou topologií, viz **Obr. 5.1**.



Obr. 5.1: Ukázka výchozí sítě pro projekt „*TCP_a_UDP*“.

4. Po načtení projektu jsou zde vytvořeny dva scénáře TCP a UDP, jejichž statistiky si zobrazíte kliknutím pravým tlačítkem myši na plochu a z nabídky vyberte *View Results.* Výsledné statistiky si zobrazte pro oba scénáře změnou hodnoty položky *Currnet Scenario* na *Current Project* a projděte si jednotlivé statistiky. Pro oba scénáře je nastaven stejný objem generovaných dat.
5. Zobrazte statistiky *Global Statistics/Custom Application/Traffic Sent (bytes/sec)*[1] a dole pod grafem změňte hodnotu ze *Stacked Statistics* na *Overlaid Statistic*, poté si stejným způsobem zobrazte i *Traffic Received (bytes/sec).*
6. Zobrazte si statistiku *Object Statistics/Campus Network/Client/IP/Traffic Received (packets/sec)* opět pro scénáře TCP a UDP do jednoho grafu *(Overlaid Statistic)* a místo zobrazení *as is* zvolte průměrování statistik *average*. Měli byste vidět totožný obrázek jako na **Obr. 5.2**.
7. Uvědomte si základní rozdíly mezi TCP a UDP. Obě statistiky zdůvodněte.

[1] Podrobnější statistiku lze vždy zobrazit kliknutím na tlačítko **Show** zde lze pomocí kurzoru myši vybírat zajímavé oblasti k bližšímu zkoumání. **Obr. 5.2**



Obr. 5.2: Detail statistik TCP a UDP pro přijatý provoz v paketech/s.

5.3 Úkol 2 – zahazování paketů pro TCP

V tomto úkolu vytvoříte šest scénářů pro zahazovaní paketů na *IP_cloudu* a to pro 1, 2, 3, 4, 5, 10 % a budete tak sledovat, jak se protokol TCP zachová.

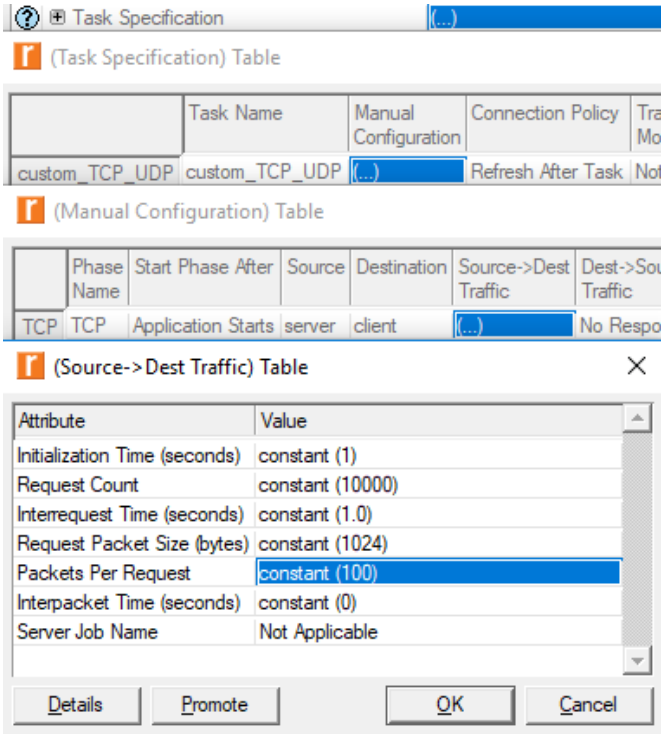
5.3.1 Postup

1. Současný projekt si uložte (Ctrl + S).
2. Přepněte se do scénáře „*TCP_bez_zahazovani“ (Scenarios/Switch To Scenario).*
3. Nyní budete editovat komponentu Tasks **(Obr. 5.3)** se zdrojovými daty, která budou stejná pro všechny další duplikované scénáře se zahazováním. Klikněte pravým tlačítkem myši na *Tasks* a zvolte *(Edit Attributes)*. V nově otevřeném okně najděte položku *Task Specification* a rozklikněte jeho nabídku a zvolte *Edit.* V této tabulce zvolte *Manual Configuration* zde *Source->Dest Traffic.* V tomto okně se budou editovat položky počet požadavků (*Request Count)* na *„10000“* a počet paketů za jeden požadavek *(Packets Per Request)* na *„100“*.
4. Pro následující scénáře je nutné vybrat vhodné statistiky ke sledování, proto klikněte pravým tlačítkem myši na plochu a vyberte *Choose Individual DES Statistics*. V nově otevřeném okně přidejte statistiky:

   a) Node Statistics: *IP/Traffic Dropped (packets/sec)*

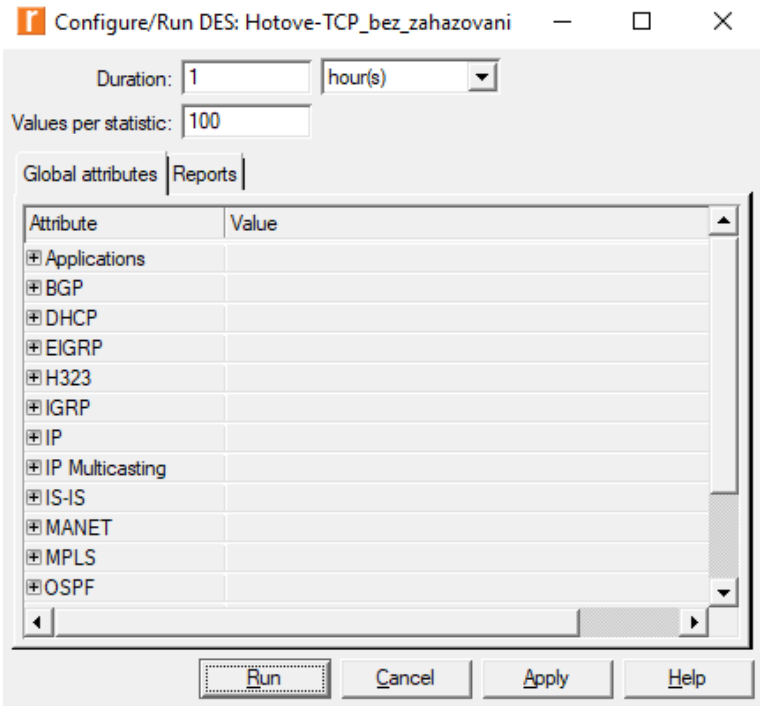
   IP/ Traffic Sent (packets/sec)

   TCP/Retransmission Count



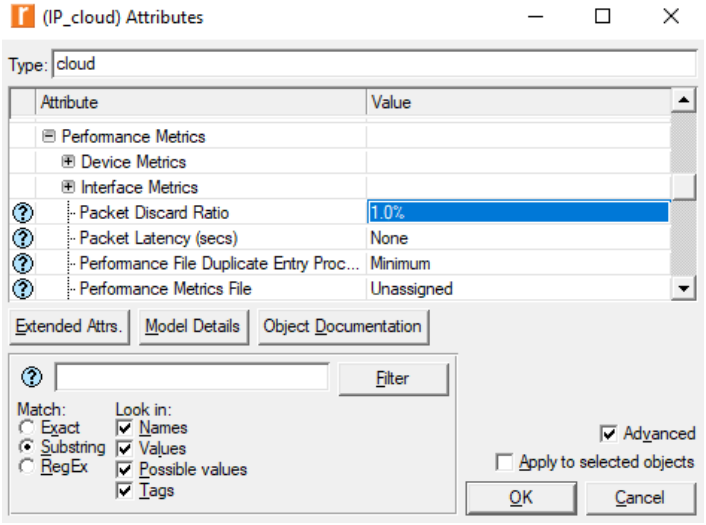
Obr. 5.3: Definice síťového provozu v komponentě Tasks.

5. Tento scénář dejte odsimulovat ručně kliknutím na ikonu *Configure/Run Discrete Event Simulation (DES).* V okně simulace nastavte čas simulace na 1 hodinu a počet hodnot na statistiku 100. Poté klikněte na tlačítko **Run**.**(Obr. 5.4)**



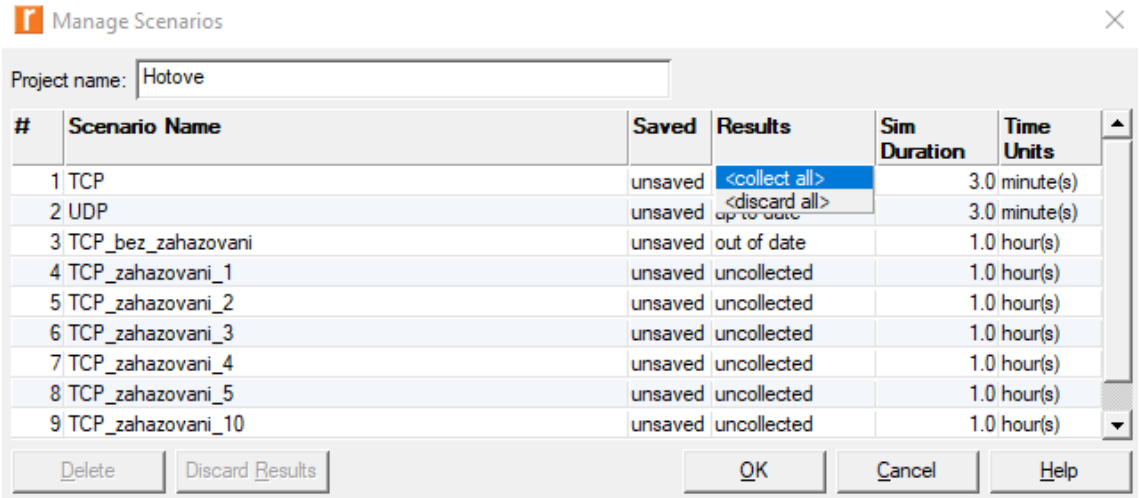
Obr. 5.4: Nastavení parametrů simulace.

6. Po konfiguraci a odsimulování scénáře *„TCP_bez_zahazovani“* ho uložte (Ctrl +S). Duplikujte známým způsobem *Scenarios/Duplicate Scenario* a vhodně si ho pojmenujte, například *„TCP_zahazovani_1“* V tomto scénáři budete nastavovat zahazování na komponentě *IP_cloud* a s 1% zahazování paketů. Vstupte do komponenty *IP_cloud* pomocí *(Edit Attributes)* a rozklikněte kolonku *Performace Metrics* a v poli *Packet Discard Ratio* vyberte hodnotu 1%, viz **Obr. 5.5**.



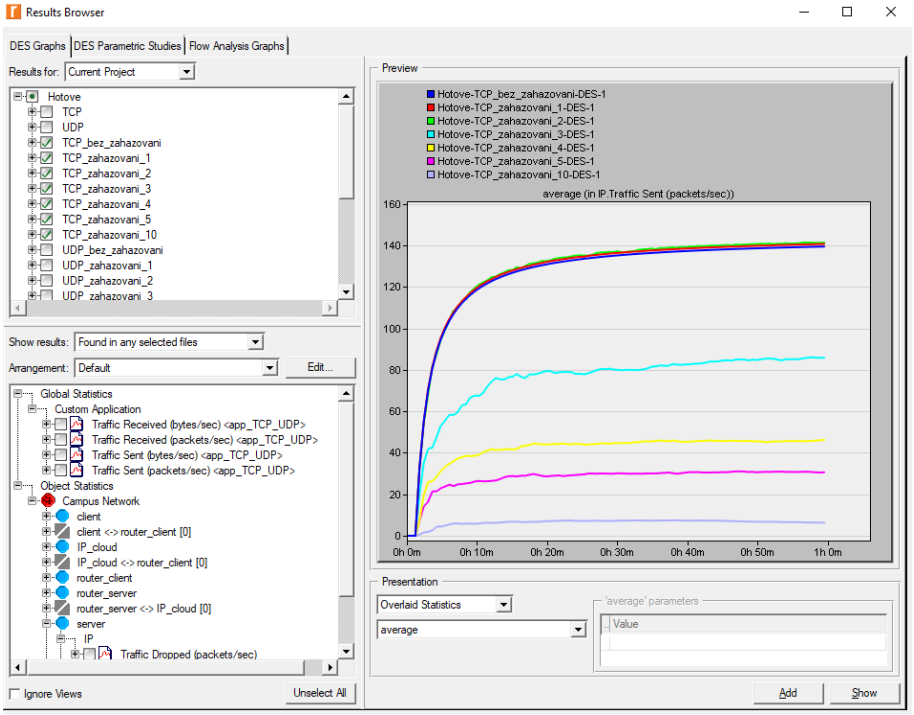
Obr. 5.5: Konfigurace pro zahazování paketů.

7. Vytvořte dalších **pět scénářů** pomocí duplikování se zahazováním paketů pro 2, 3, 4, 5, 10% a to stejným způsobem, jaký je popsán v předchozím bodě.
8. Po vytvoření scénářů a nastavení zahazování spusťte simulaci pro všechny scénáře pomocí položky menu *Scenarios/Manage Scenarios* (**Obr. 5.6**), délku simulací *(Sim Duration)* nastavte na 1 hodinu. Klikněte na položku *Results* a vyberte *<collect all>*. Poté klikněte na tlačítko *OK* pro spuštění simulace.



Obr. 5.6: Okno Manage Scenarios s vytvořenými scénáři.

Po dokončení simulací si zobrazte výsledky pro statistiku *Objekt Statistics/Campus Network/server/IP/Traffic Sent (packets/sec)*. Vaše statistiky by se měly shodovat s **Obr. 5.7**.



Obr. 5.7: Vyslaný provoz TCP ze serveru pro různé hodnoty zahazování paketů za sekundu.

5.3.2 Doplňující otázky a úkoly

1) Zobrazte statistiky pro propustnost linky a to pro oba směry (tedy od serveru -> a k serveru <-) *Objekt Statistics/Campus Network/server<-> router_server/throughput (bit/sec)* -> a *throughput (bit/sec)* <-. Pro umístění více scénářů do jednoho grafu zvolte pod obrázkem v sekci *Presentation* místo *Current Project* zobrazení *Overlaid Statistics*. Na grafy je možné se dívat i za pomoci průměrování hodnot *avarage*, které v některých případech umožňují názornější zobrazení.

2) Zobrazte si výsledky simulace pro *Object Statistics/Campus Network/server/TCP/Retransmission Count*. **Retransmission Count** vyjadřuje počet opakovaných přenosů TCP v tomto uzlu. Tuto statistiku zobrazte pro všechny scénáře se zahazováním paketů a všechny scénáře zobrazte do jednoho *(Overlaid Statistics)* a vyberte průměrování hodnot *(average)*. Vysvětlete, proč je počet opakovaných přenosů pro jedno procento o tolik nižší oproti scénáři s dvouprocentním zahazováním.

3) Zobrazte si výsledky simulace pro *Object Statistics/Campus Network/IP_cloud/IP/Traffic Dropped (packets/sec)*. Tuto statistiku zobrazte pro všechny scénáře se zahazováním paketů a všechny scénáře zobrazte do jednoho *(Overlaid Statistics)* a vyberte průměrování hodnot *(average)*.

5.4 Úkol 3 – zahazování paketů pro UDP

Stejným způsobem jako v úkolu 2 budete vytvářet zahazování paketů s tím rozdílem, že tentokrát se bude zahazování provádět s protokolem UDP pro stejné hodnoty 1, 2, 3, 4, 5, 10%

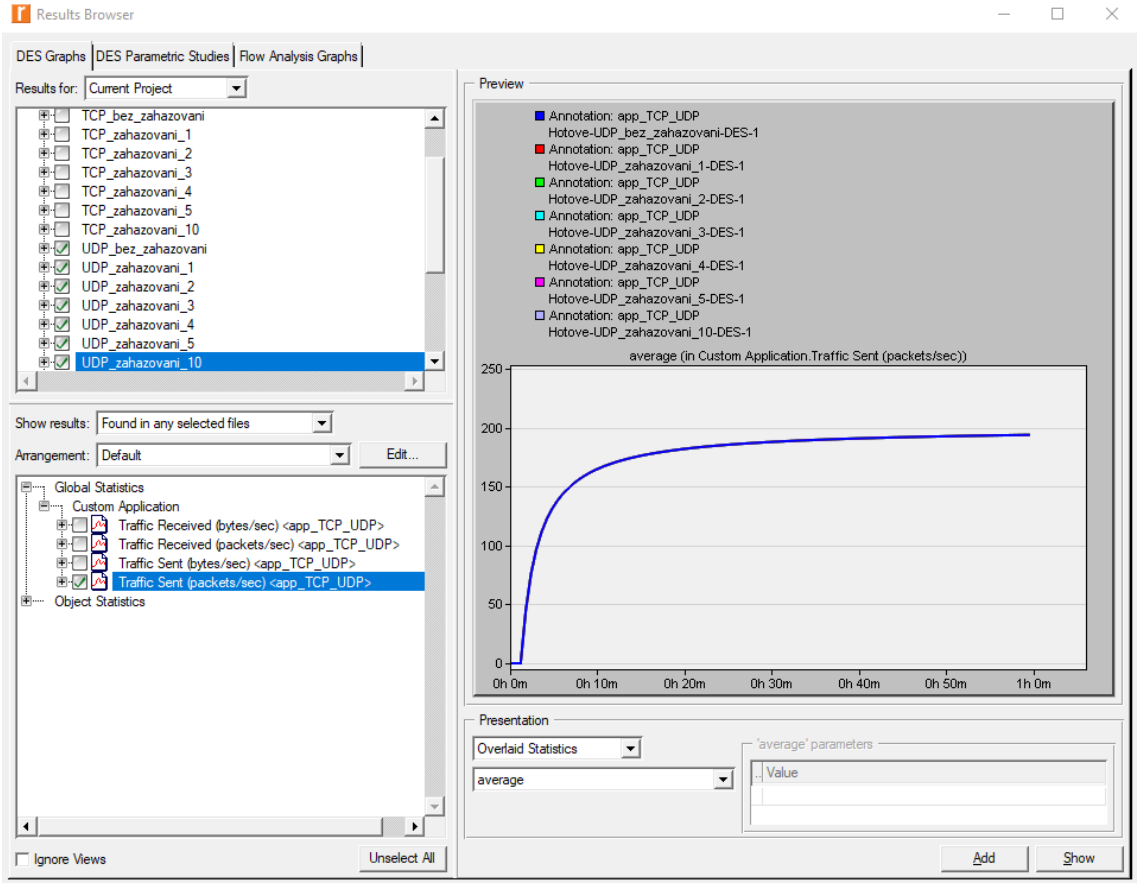
5.4.1 Postup

1. Současný projekt si uložte (Ctrl + S).
2. Přepněte se do scénáře (Scenarios/Switch To Scenario). „*UDP_bez_zahazovani“*.
3. Editace komponenty Tasks pro UDP je stejná jako v případě TCP, viz **Obr. 5.3**. Na *Tasks* zvolte *(Edit Attributes).* V nově otevřeném okně najděte položku *Task Specification*, rozklikněte jeho nabídku a zvolte *Edit.* V této tabulce zvolte *Manual Configuration* zde *Source->Dest Traffic.* V tomto okně se budou editovat položky počet požadavků (*Request Count)* na *„10000“* a počet paketů za jeden požadavek *(Packets Per Request)* na *„100“*.
4. Sledované statistiky - klikněte pravým tlačítkem myši na plochu a vyberte *Choose Individual DES Statistics*. V nově otevřeném okně přidejte statistiky:

   a) Node Statistics: *IP/Traffic Dropped (packets/sec)*

5. Scénář „*UDP_bez_zahazovani“* dejte opět odsimulovat ručně kliknutím na ikonu *Configure/Run Discrete Event Simulation (DES).* V okně simulace nastavte čas simulace na 1 hodinu a počet hodnot na statistiku 100. Poté klikněte na tlačítko **Run**.
6. Vytvořte **šest scénářů** pomocí duplikování se zahazováním paketů pro 1, 2, 3, 4, 5, 10%, pro každý scénář nastavte na *IP_cloudu* v položce *Packet Discard Ratio* procentuální zahazování paketů stejným způsobem, jak tomu bylo u TCP, viz **Obr. 5.5**.
7. Po vytvoření scénářů a nastavení zahazování, spusťte simulaci pro všechny scénáře pomocí položky menu *Scenarios/Manage Scenarios,* délku simulací *(Sim Duration)* nastavte na 1 hodinu. Klikněte na položku *Results* a vyberte <*collect all*>. Poté klikněte na tlačítko *OK* pro spuštění simulace.

Po dokončení simulací si zobrazte výsledky pro statistiku *Global Statistics/Custom Application/Traffic Sent (packets/sec).* Vaše statistiky by se měly shodovat s **Obr. 5.8**.



Obr. 5.8: Vyslaný provoz UDP pro různé hodnoty zahazování paketů za sekundu.

5.4.2 Doplňující otázky a úkoly

1) Zobrazte statistiky pro propustnost linky a to pro oba směry (tedy od serveru -> a k serveru <-) *Objekt Statistics/Campus Network/server<-> router_server/throughput (bit/sec) ->* a *throughput (bit/sec) <-*. Pro umístění více scénářů do jednoho grafu zvolte pod obrázkem v sekci *Presentation* místo *Current Project* zobrazení *Overlaid Statistics*. Na grafy je možné se dívat i za pomoci průměrování hodnot *avarage*, které v některých případech umožňují názornější zobrazení.

2) Zobrazte si výsledky simulace pro *Object Statistics/Campus Network/IP_cloud/IP/Traffic Dropped (packets/sec)*. Tuto statistiku zobrazte pro všechny scénáře se zahazováním paketů a všechny scénáře zobrazte do jednoho *(Overlaid Statistics)* a vyberte průměrování hodnot *(average).*

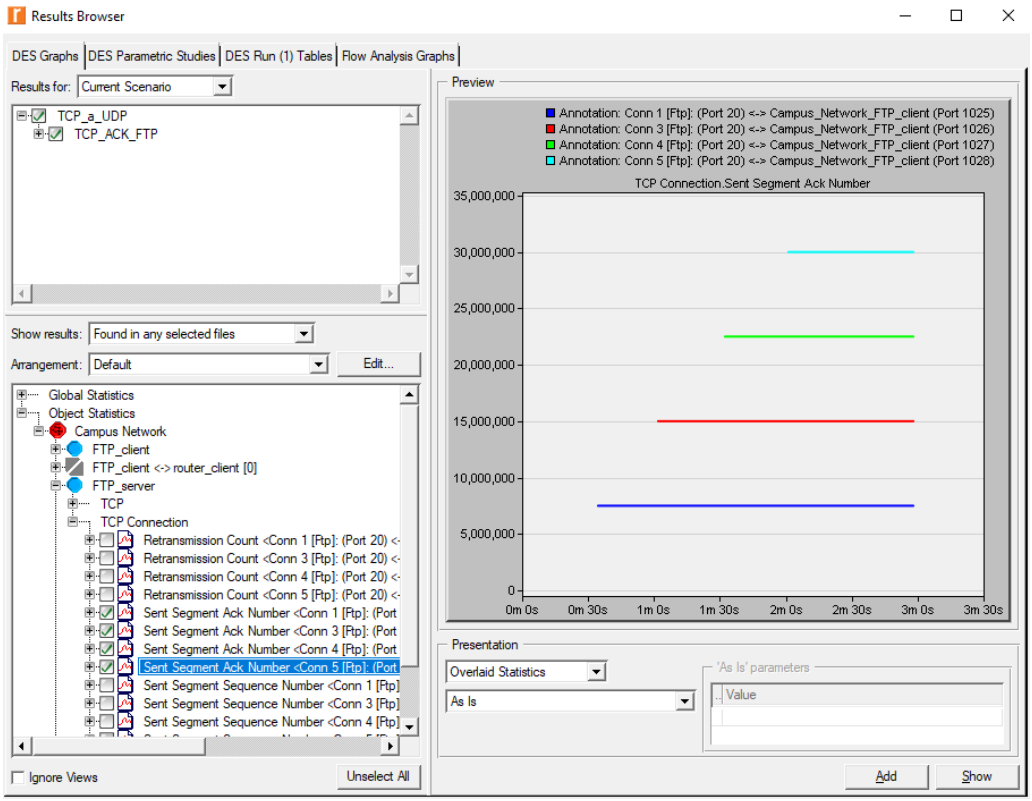
3) Na závěr si zobrazte statistiky pro propustnost směrem od serveru *Objekt Statistics/Campus Network/server<-> router_server/throughput (bit/sec)<-* pro scénáře *„TCP_bez_zahahazovani“* a *„UDP_bez_zahahazovani“* a vysvětlete, co se podle zobrazených statistik na TCP přenáší oproti UDP.

5.5 Úkol 4 – Potvrzování - ACK (Acknowledgment) v TCP

V úkolu 4 se podíváte na principy přenosu v TCP protokolu. Nahrajete si již předvytořený projekt s vytvořeným scénářem s FTP. Poté si zobrazíte statistiky přenášených segmentů, v nichž si budete moci prohlédnout potvrzování přenášených segmentů.

5.5.1 Postup

1. Současný projekt si uložte (Ctrl + S).
2. V menu programu vyberte **File/Open…**(Ctrl + O) a otevřete projekt *„TCP_ACK“*.
3. Po načtení si ve scénáři „*TCP_ACK_FTP*“ zobrazte statistiku pro *Sent Segment Sequence Number*, což je pořadové číslo odeslaného segmentu.
4. Již známým způsobem si pomocí *View Results* zobrazte potvrzovací číslo odeslaného segmentu v *Object Statistics/Campus Network/FTP_server/TCP Connection/Sent Segment ACK Number*. Tyto statistiky nastavte tak, aby se zobrazovaly všechny najednou pomocí *Overlaid Statistics*, viz **Obr. 5.9**. Na tomto grafu se pak podívejte na čísla portů na obou stranách přenosu. Z výsledné simulace je zřejmé, že pro FTP se vždy používá na straně serveru **port 20**, který je pro FTP vyhrazený. Naopak na straně klienta se náhodně zvolí pro každou aplikaci jiný port, aby bylo jednoznačně určeno, pro kterou aplikaci je daný segment určen.



Obr. 5.9: Potvrzovací číslo odeslaného segmentu - Sent Segment ACK Number.

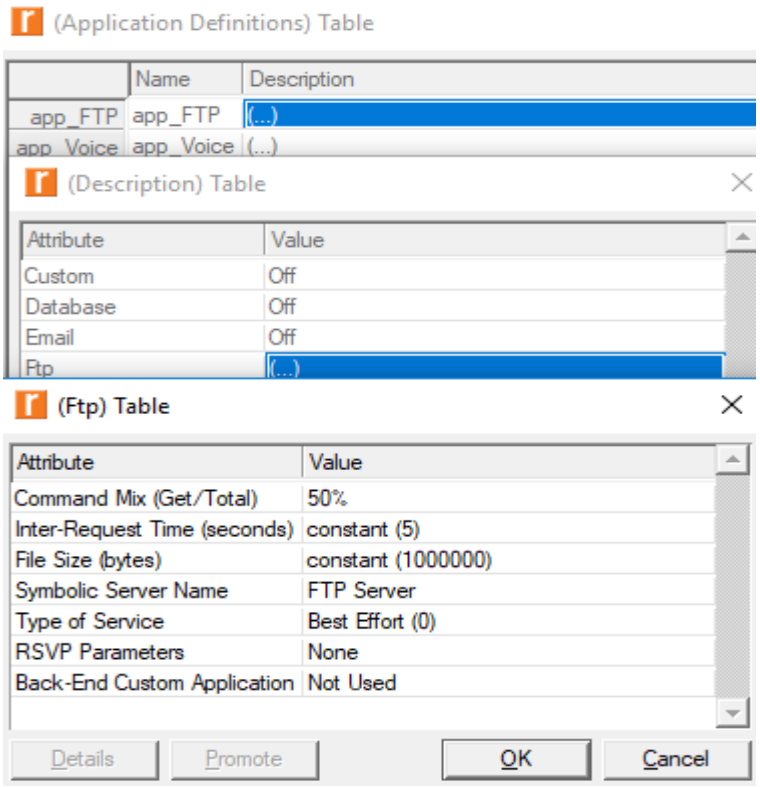
5. V grafu *Sent Segment ACK Number* je vidět i to, že segmenty, které byly přijaty mimo pořadí, se nepotvrzují. Z tohoto důvodu je v tomto případě pořadové číslo dva vynecháno. Protože se vždy potvrzuje poslední správně přijatý segment, můžete si všimnout, že pořadové číslo roste a po potvrzení jednoho se čeká na další segment v pořadí. V simulaci je také možné vidět, v jakém čase byly segmenty odeslány.

5.6 Úkol 5 – Výpadek na lince s FTP a Voice

V úkolu číslo 5 se podíváte, jak zareagují aplikace FTP a Voice, když na lince vznikne výpadek spojení. V tomto úkolu je nutné si uvědomit, že FTP pracuje s protokolem TCP a naopak Voice s protokolem UDP. I v tomto úkolu budete mít k dispozici již předvytvořený scénář *„FTP_VOICE_bez_vypadku“*. Výpadek na lince provedete komponentou *Failure Recovery*.

5.6.1 Postup

1. Předchozí scénář si uložte (Ctrl + S) a přepněte se do scénáře *„FTP_VOICE_bez_vypadku“ (Scenarios/Switch To Scenario).*
2. Nyní je potřeba v tomto scénáři nastavit vyšší přenos pro FTP. Klikněte v komponentě *Application config (Edit Attributes)* a rozklikněte položku *Application Definitions*, kde naleznete *app_FTP*. Zde zvolte *Edit...* a v položce *Ftp* znovu klikněte na *Edit...* a nastavte parametry podle **Obr. 5.10**.



Obr. 5.10: Nastavení parametrů aplikace FTP.

3. Sledované statistiky - klikněte pravým tlačítkem myši na plochu a vyberte *Choose Individual DES Statistics*. V nově otevřeném okně přidejte statistiky:

a) Global Statistics: *FTP/Traffic Received (bytes/sec)*
FTP/Traffic Sent (bytes/sec)
Voice/ *Traffic Received (bytes/sec)*
Voice/ *Traffic Sent (bytes/sec)*

4. Scénář dejte odsimulovat ručně kliknutím na ikonu *Configure/Run Discrete Event Simulation (DES)*. V okně simulace nastavte čas simulace na 10 minut a počet hodnot na statistiku 100.
5. Scénář uložte (Ctrl +S) a duplikujte *Scenarios/Duplicate Scenario*. Nový scénář nazvěte například *„FTP_VOICE_vypadek“*.
6. V nově vytvořeném scénáři přidejte komponentu *Failure Recovery* kliknutím v panelu na *Object Palette*. 
7. Komponentu *Failure Recovery* si pojmenujte například *vypadek/obnova (Edit Attributes* položka *name)*. Pomocí této komponenty realizujete výpadek a obnovení provozu na lince. Najděte atribut *Link Failure/Recovery Specification* a rozkliknutím nabídky *(...)* zvolte *Edit...*, nastavení provedete podle **Obr. 5.11**.

Link Failure/Recovery Specification (...)

(Link Failure/Recovery Specification) Table

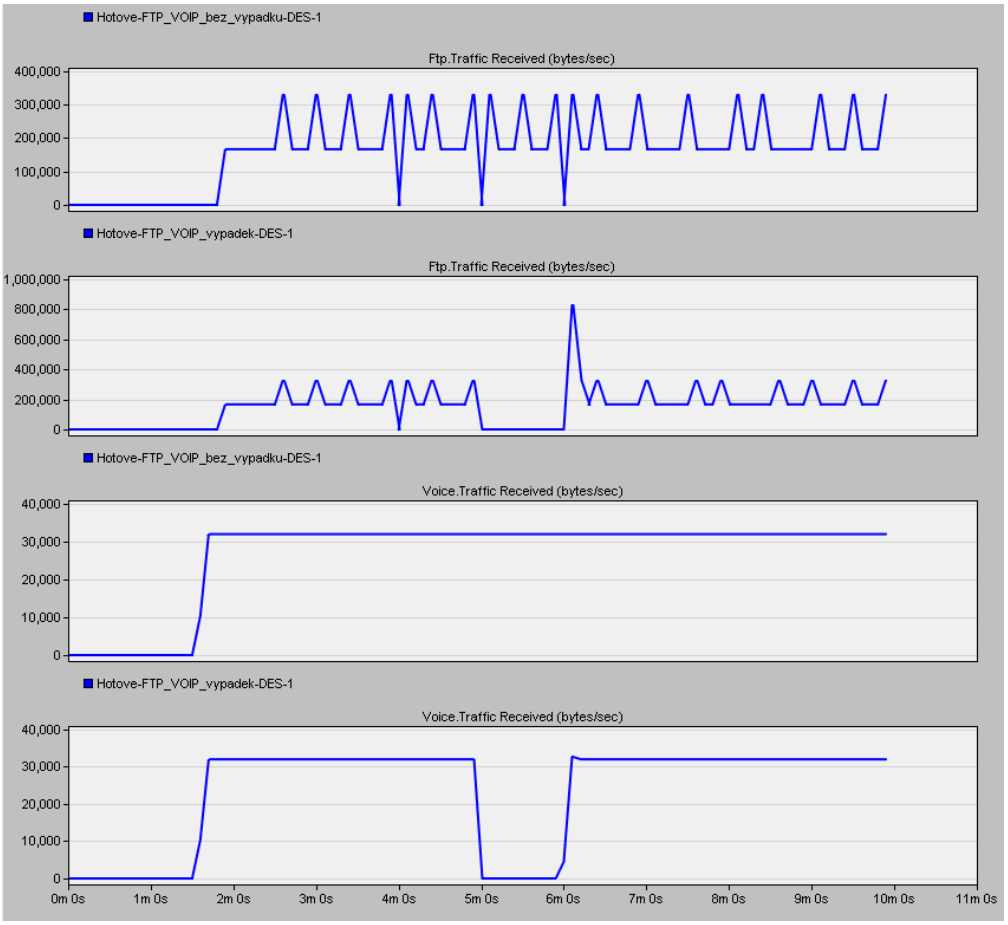
	Name	Time (seconds)	Status
Campus Network.router_server <-> IP_cloud	Campus Network.router_server <-> IP_cloud	300	Fail
Campus Network.router_server <-> IP_cloud	Campus Network.router_server <-> IP_cloud	350	Recover

Obr. 5.11: Komponeta Failure Recovery.

8. Po dokončení konfigurace spusťte simulaci *Configure/Run Discrete Event Simulation (DES)*. Nastavení simulace ponechte stejné jako v případě scénáře *„FTP_VOICE_bez_vypadku“*.

5.6.2 Doplňující otázky a úkoly

1) Zobrazte statistiky pro oba scénáře *Global Statistics/FTP/Traffic Received (bytes/sec)* a Voice/ *Traffic Received (bytes/sec)*, měli byste dostat podobné statistiky jako jsou na **Obr. 5.12**. Tyto statistiky srovnejte s *FTP/Traffic Sent (bytes/sec)* a Voice/ *Traffic Sent (bytes/sec)* a vysvětlete rozdíly mezi FTP a Voice aplikacemi.



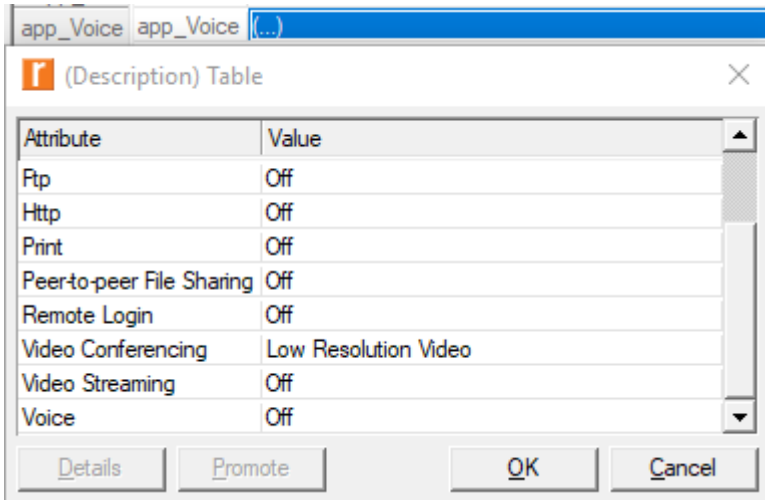
Obr. 5.12: Ukázka FTP a Voice statistik bez výpadku a pro výpadek.

5.7 Úkol 6 – FTP a Video se zvyšujícím se provozem

V úkolu 6 budete sledovat, jak bude reagovat FTP a Video, když se bude postupně navyšovat provoz a tím i zátěž na linkách. Navyšování provozu budete provádět v *Application Configu* pro videokonferenci pro několik scénářů.

5.7.1 Postup

1. Předchozí scénáře si uložte (Ctrl + S). a přepněte se do scénáře *„FTP_VOICE_bez_vypadku“ (Scenarios/Switch To Scenario).* Tento scénář duplikujte *Scenarios/Duplicate Scenario* a nový scénář nazvěte *„FTP_Videokonference_bez_navysovani“.*
2. Nyní je třeba pro tento scénář nastavit videokonferenci, ve které se bude postupně zvyšovat provoz v ostatních scénářích. Klikněte v komponentě *Application config (Edit Attributes)* a v *Application Definitions* zvolte *(...)*. Zde naleznete *app_Voice*, zvolte *Edit...*, v atributu *Voice* zvolte hodnotu *off* a naopak v atributu *Video Conferencing* nastavte hodnotu *Low Resolution Video*, viz **Obr. 5.13**.

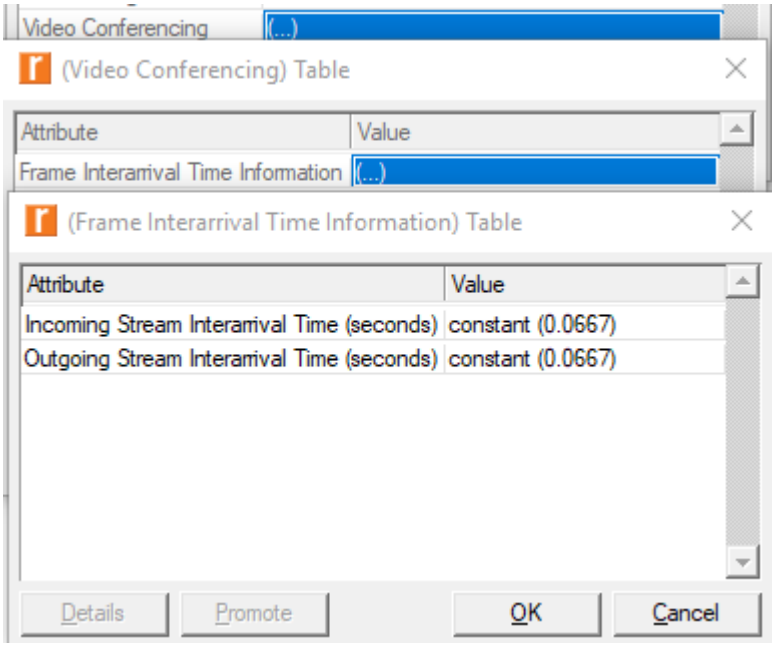


Obr. 5.13: Nastavení parametrů pro Videokonferenci.

3. Sledované statistiky - klikněte pravým tlačítkem myši na plochu a vyberte *Choose Individual DES Statistics*. V nově otevřeném okně přidejte statistiky:

a) Global Statistics: *FTP/Traffic Received (bytes/sec)*
FTP/Traffic Sent (bytes/sec)
Video Conferencing/ *Traffic Received (bytes/sec)*
Video Conferencing / *Traffic Sent (bytes/sec)*

4. Nyní scénář *„FTP_Videokonference_bez_navysovani“* duplikujte a nový scénář nazvěte například *„FTP_Videokonference_navysovani_provozu_1“*. V *Aplication Configu* přejdete do *Application Definitions* a rozklikněte *(...)*. Zde naleznete *app_Voice*, zvolte *Edit...* a vstupte do atributu *Video Conferencing*. Nyní budete měnit hodnotu *Frame Interarrival Time (seconds)*, viz **Obr. 5.14**.
5. Stejným postupem vytvořte další 4 scénáře, kde budete měnit hodnoty *Incoming a Outgoing Stream Interarrival Time (seconds)* pro hodnoty **constant (0,05)**, **constant (0,03)**, **constant (0,029)** a **constant (0,025)**. Vždy nastavujte oba atributy na stejnou hodnotu.



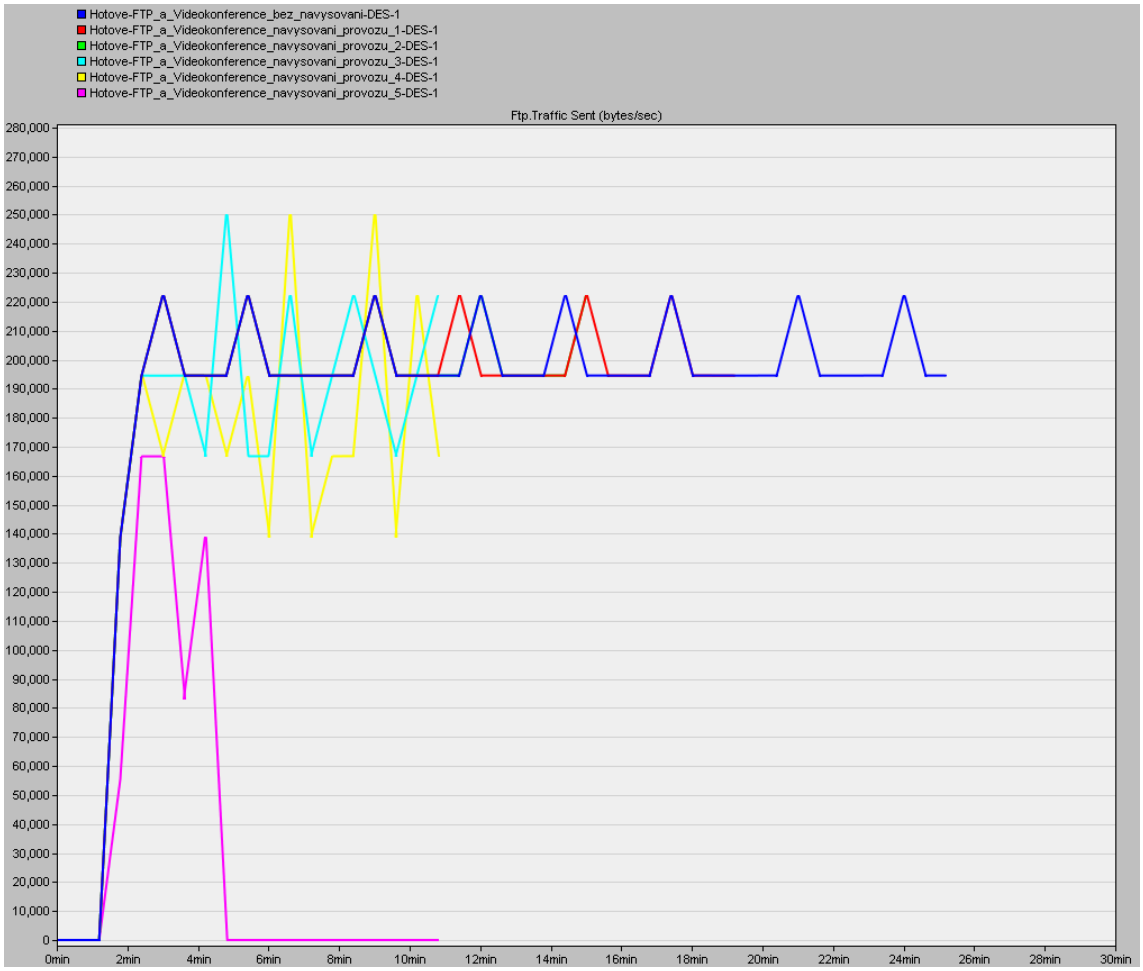
Obr. 5.14: Nastavení hodnoty Frame Interarrival Time (seconds) ve videokonferenci.

6. Scénáře můžete dát odsimulovat jednotlivě nebo všechny najednou pomocí menu *Scenarios/Manage Scenarios*, délku simulací *(Sim Duration)* nastavte na 1 hodinu. Klikněte na položku *Results* a vyberte <*collect all*>.

5.7.2 Doplňující otázky a úkoly

1) Zobrazte statistiky *Global Statistics/FTP/Traffic Sent (bytes/sec)*, měli byste dostat podobné statistiky jako jsou na **Obr. 5.15**. Tyto statistiky porovnejte se statistikami *Video Conferencing/ Traffic Sent (bytes/sec)*.

2) Zobrazte statistiky *Global Statistics/FTP/Traffic Received (bytes/sec)* a porovnejte se statistikami *Video Conferencing/ Traffic Received (bytes/sec)*. Zdůvodněte, proč provoz FTP klesá ve srovnání s Videokonferencí.



Obr. 5.15: Ukázka statistiky Traffic Sent (bytes/sec) pro FTP při navyšování provozu.

3) Zdůvodněte, proč je např. pro FTP vhodnější použít protokol TCP a pro Voice protokol UDP.

4) Vyjmenujte tři aplikace používající protokol TCP a dvě aplikace používající protokol UDP.

6 SROVNÁNÍ TECHNOLOGIÍ ATM A FRAME RELAY

6.1 Úvod k laboratorní úloze

Laboratorní úloha se zabývá dvěma používanými přenosovými technologiemi pro sítě typu WAN – *(Wide Area Network).* Díky těmto technologiím je možné pokrýt rozsáhlé geografické území a přenášet jimi data na vzdálenosti stovek i tisíců kilometrů. ATM – *(Asynchronous Transfer Mode)* a Frame Relay se využívají například pro přenos informací v reálném čase, jež vyžadují vysokou datovou rychlost s malým zpožděním a s efektivním přenosem dat. Před zahájením přenosu dat se sestavují virtuální okruhy. Frame Relay používá pro přenos přepojování paketů. Pakety jsou přenášeny mezi dvěma účastníky pomocí trvalého virtuálního okruhu. Frame Realy je efektivnější oproti ATM na nižších rychlostech. V ATM se používá přepojování buněk, které mají pevnou délku 53 bajtů, z čehož je 5B záhlaví a 48B datová část. V záhlaví ATM buněk se nacházejí identifikátory virtuálních spojení VPI a VCI, které se používají pro přepínání buněk. Díky hodnotě VPI a VCI může ATM přepínač určit výstupní port, na který má být daná buňka dále předána.

6.2 Úkol 1 – Porovnání tříd CBR a UBR v ATM pro konferenci

První úkol je zaměřen na kvalitu služeb v ATM. Konkrétně na CBR a UBR, které budete mezi sebou porovnávat a následně ve scénářích navyšovat provoz. Třídy služeb se od sebe liší, protože aplikace mají různé nároky na parametry sítě. Na tyto odlišnosti se tedy podíváte v této úloze, kde se nacházejí hlasové a datové aplikace. V předvytvořeném projektu se nacházejí dva scénáře. První scénář má nastavenu třídu služeb na UBR a adaptační vrstvu na AAL5 pro všechny aplikace. Druhý scénář má pro konferenci nastaveno CBR s AAL1. V prvním kroku se tedy seznámíte s vlastnostmi ATM protokolu. Použití jednotlivých tříd a AAL, používaných v této úloze, si můžete připomenout v **Tab. 6.1** a **Tab. 6.2**.

Tab. 6.1: Použité třídy služeb.

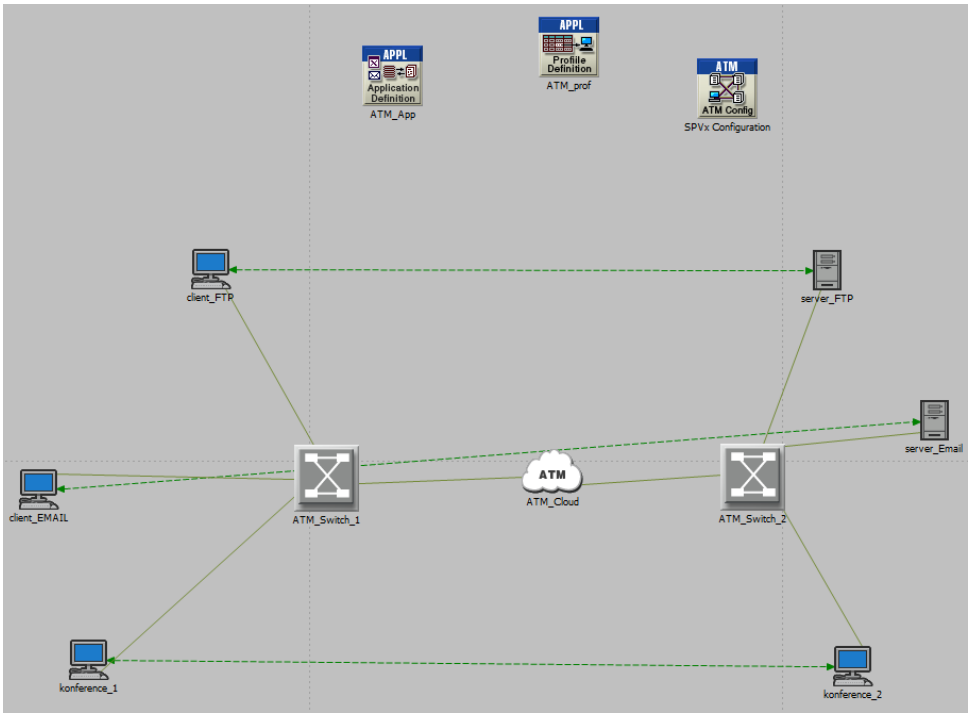
Třídy služeb	Použití
CBR	Pro přenos nekomprimovaného zvuku a obrazu v reálném čase.
ABR	Pro aplikace nenáročné na zpoždění a kolísání zpoždění. Vhodné pro přenos dat a souborů.
UBR	Negarantuje minimální přenosovou rychlost ani doručení, tzv. best effort.

Tab. 6.2: Přehled adaptačních vrstev AAL.

Adaptační vrstva AAL	Použití
AAL1	Hlasové aplikace a podpora CBR.
AAL2	Pro komprimované přenosy obrazu a zvuku.
AAL5	Používá se pro přenos dat.

6.2.1 Postup

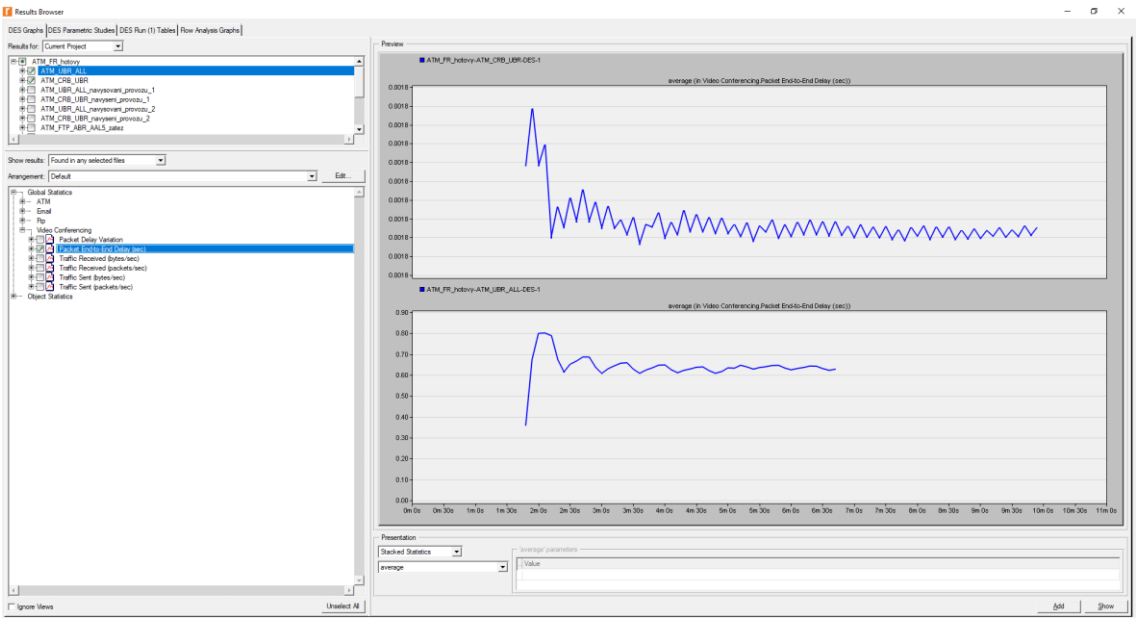
1. Stáhněte vypracovaný projekt *„ATM_FR_predvypracovany“* a uložte jej například na **Plochu**.
2. Spusťte program Riverbed Modeler.
3. V menu programu vyberte File Open…(Ctrl + O) a otevřete projekt *„ATM_FR_predvypracovany“*. Zobrazí se okno s připravenou topologií, viz **Obr. 6.1.**



Obr. 6.1: Ukázka výchozí sítě pro projekt *„ATM_FR_predvypracovany“*.

4. V projektu se nacházejí dva scénáře *„ATM_UBR_ALL“* a *„ATM_CRB_UBR“*. Ve scénáři *„ATM_UBR_ALL“* jsou nastaveny všechny aplikace (FTP, Email a konference) na UBR s adaptační vrstvou AAL5. Ve druhém scénáři s názvem *„ATM_CRB_UBR“* je na konferenci nastaveno CBR s AAL1 a na FTP a Emailu je UBR s AAL5. Jejich statistiky si zobrazte kliknutím pravým tlačítkem myši na plochu a z nabídky vyberte *View Results*. Výsledné statistiky si zobrazte pro oba scénáře změnou hodnoty položky *Currnet Scenario* na *Current Project* a projděte si jednotlivé statistiky.

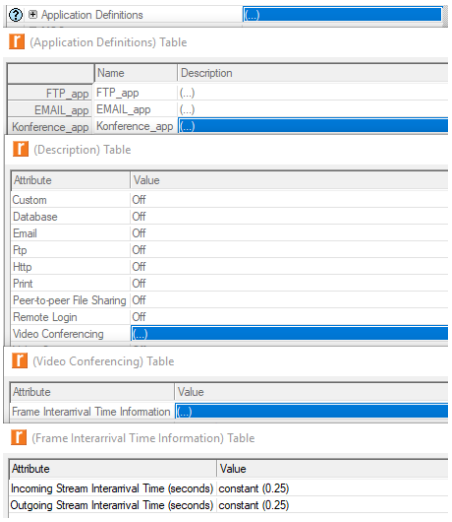
5. Především si prohlédněte statistiky *Global Satistic/Video Conferencing/Packet End-to-End Delay(sec)* a *Packet Delay Variation*. Měli byste vidět stejnou statistiku pro Packet End-to-End Delay(sec) jako na **Obr. 6.2**.



Obr. 6.2: Srovnání CBR a UBR pro konferenci v Packet End-to-End Delay (sec).

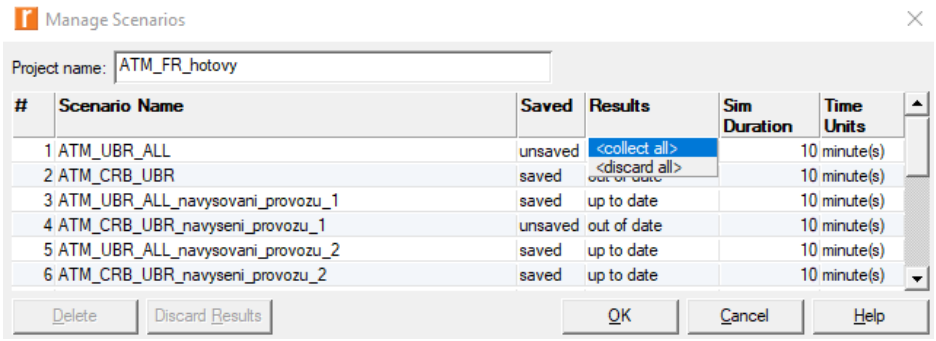
6. Nyní se přepněte do scénáře *„ATM_UBR_ALL“*, který duplikujte *Scenarios/Duplicate Scenario* a vhodně si ho pojmenujte, například *„ATM_UBR_ALL_navysovani_provozu_1“*.

7. V tomto scénáři budete nastavovat vyšší provoz na konferenci. Vstupte do komponenty *ATM_App* pomocí *(Edit Attributes)*. V otevřeném okně najděte položku *Application Definitions*, rozklikněte tuto nabídku a zvolte *Edit*. V tabulce vstupte do aplikace *Konference_app/Video Conferencing/Frame Interarrival Time Information*. Zde navyšte *Incoming* a *Outgoing Stream Interarrival Time Information (seconds)* na **constant (0.25)**, viz **Obr. 6.3**.



Obr. 6.3: Nastavení parametrů konference.

8. Známým způsobem *Scenarios/Duplicate Scenario* duplikujte i scénář *„ATM_CRB_UBR“*, pojmenujte si ho například *„ATM_CRB_UBR_navyseni_provozu_1“* a *ATM_app* nastavte stejně jako v bodě **7**.
9. Nyní ještě jednou duplikujte oba scénáře *„ATM_UBR_ALL“* a *„ATM_CRB_UBR“* s tím rozdílem, že nyní na těchto nových scénářích nastavíte *Incoming* a *Outgoing Stream Interarrival Time Information (seconds)* na **constant (0.1)**.
10. Po vytvoření všech čtyř scénářů s navyšováním provozu spusťte simulaci pro všechny scénáře pomocí položky menu *Scenarios/Manage Scenarios* **(Obr. 6.4)**, délku simulací (*Sim Duration*) ponechte na 10 minut. Klikněte na položku *Results* a vyberte *<collect all>*. Poté klikněte na tlačítko *OK* pro spuštění simulace.
11. Při simulacích si můžete všimnout, že ne všechny scénáře se budou simulovat. Na všech scénářích ponechte nastavených 10 minut. Některé z nich mohou skončit dříve. Je to způsobeno tím, že Riverbed není v plné verzi a má své limity, zde je konkrétně limit na maximálně 50 milionů eventů na jednu simulaci.



Obr. 6.4: Ukázka hromadného od simulování scénářů.

6.2.2 Doplňující otázky a úkoly

1) Zobrazte si statistiky Download Response Time (sec) nejprve pro první tři scénáře s CBR a poté i pro zbylé tři scénáře s nastaveným UBR s FTP a Emailem. Pro *Global Statistic/Email/Download Response Time (sec)* a *Global Statistic/FTP/Download Response Time (sec)*. Zdůvodněte rozdíly v FTP a Emailu.
2) Podívejte se na výsledky scénářů s CBR pro *Global Statistic/ Video Conferencing/ Traffic Recieved (bytes/sec)* a *Global Statistic/ Video Conferencing/ Traffic Sent (bytes/sec)*. Pro umístění více scénářů do jednoho grafu zvolte pod grafem v sekci *Presentation* místo *Stacked Statistics* zobrazení *Overlaid Statistics*. Na grafy je možné se dívat i za pomoci průměrování hodnot *avarage*, které v některých případech umožňují názornější zobrazení, v tomto případě ponechte výchozí zobrazení *As Is*.
3) Nyní se podívejte na statistiky pro zbylé tři scénáře s UBR, nejprve pro *Global Statistic/ Video Conferencing/ Traffic Sent (bytes/sec)*. Jako v přechozím případě dejte všechny tři UBR scénáře do jednoho grafu pomocí *Overlaid Statistics* A následně se podívejte i na výsledky *Global Statistic/ Video Conferencing/ Traffic Recieved (bytes/sec)*. Zamyslete se, proč je odeslaný provoz jiný než přijatý.
4) Zdůvodněte, proč jsou výsledky UBR jiné než v případě CBR.

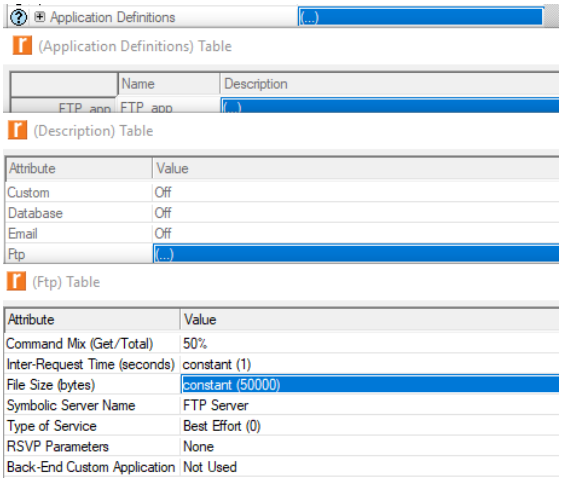
5) Ve scénáři *„ATM_CBR_UBR“* je nastaveno automatické vygenerování souboru VCI a VPI na směrovačích. Tento soubor se jmenuje *ATM_FR_predvypracovany-ATM_CRB_UBR-DES-1-vc_routes.GDF*. Naleznete jej ve staženém projektu a můžete si ho zobrazit pomocí poznámkového bloku. Na tento soubor se budeme moci podívat i pro všechny duplikované scénáře. Mimo jiné je možné v souboru vidět použitou třídu služeb s adaptační vrstvou AAL.

6.3 Úkol 2 – Srovnání tříd ABR a UBR v ATM pro FTP

V tomto úkolu vytvoříte dva scénáře, kde budete porovnávat, jaký vliv mají třídy ABR a UBR na FTP.

6.3.1 Postup

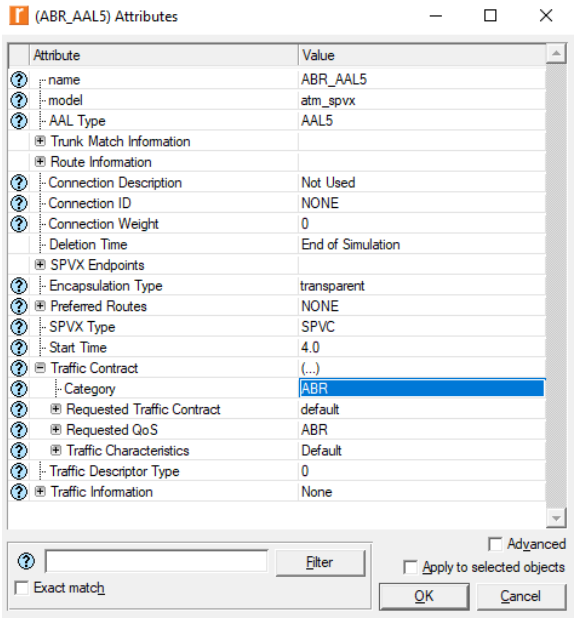
1. Současný projekt si uložte (Ctrl + S).
2. Přepněte se do scénáře s největším nastaveným provozem na konferenci s CBR a AAL1 *„ATM_CRB_UBR_navyseni_provozu2“*, pomocí *Scenarios/Switch To Scenario* tento scénář duplikujte *Scenarios/Duplicate Scenario* a vhodně pojmenujte, například *„ATM_FTP_UBR_AAL5_zatez“*
3. Nyní je třeba přenastavit *ATM_app* s FTP. Vstupte do *ATM_App* pomocí *(Edit Attributes)*. V otevřeném okně najděte položku *Application Definitions*, rozklikněte tuto nabídku a zvolte *Edit*. V tabulce vstupte do aplikace *FTP_app/FTP* a nastavte parametry *Inter-Request Time (seconds)* na hodnotu **1** a *File Size (bytes)* na hodnotu **50000**, viz **Obr. 6.5**.



Obr. 6.5: Konfigurace parametrů FTP aplikace.

4. Scénář *„ATM_FTP_UBR_AAL5_zatez“* opět duplikujte *Scenarios/Duplicate Scenario* a pojmenujte například *„ATM_FTP_ABR_AAL5_zatez“*
5. V tomto scénáři budete muset správně nastavit třídu služeb na SPVX. Vyberte tedy tento zeleně čerchovaný spoj mezi prvky *client_FTP* a *server_FTP* a známým způsobem editujte *Name*, nastavte na *ABR_AAL5*. Obdobně nastavte položku *AAL type* na hodnotu **AAL5**. Po rozkliknutí *Traffic Contract* nastavte položky *Category*

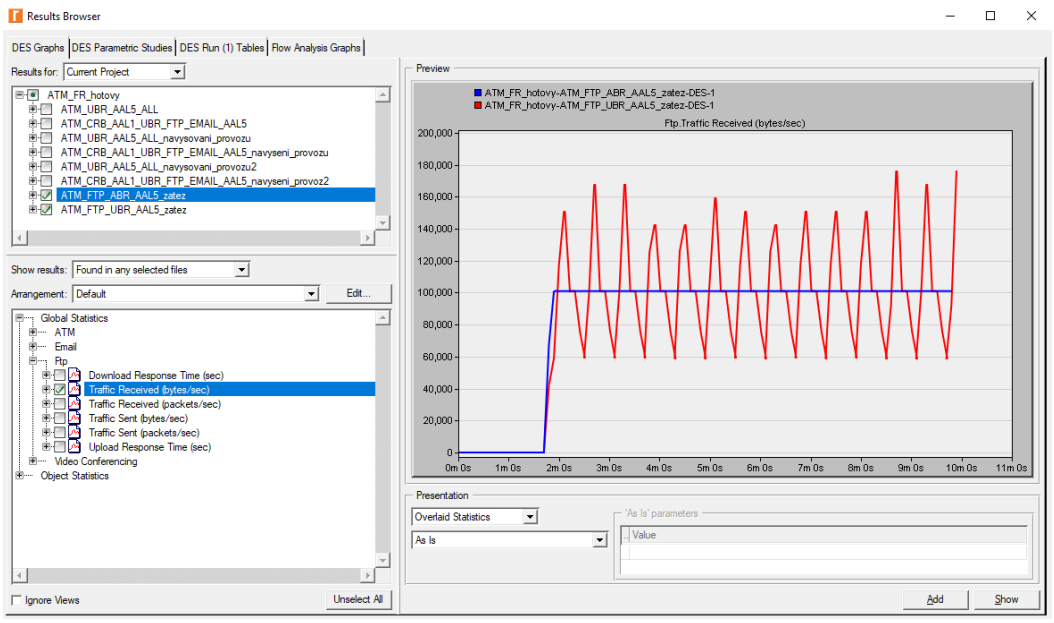
a *Requested Traffic Contract* na **ABR**, viz **Obr. 6.6**.



Obr. 6.6: Nastavení virtuálního okruhu na ABR.

6. Po vytvoření scénářů spusťte hromadnou simulaci pomocí položky menu *Scenarios/Manage Scenarios*, délku simulací (*Sim Duration*) ponechte na 10 minut. Klikněte na položku *Results* a vyberte <*collect all*>. Poté klikněte na tlačítko *OK* pro spuštění simulace.

Po dokončení simulací si zobrazte výsledky pro statistiku *Global Statistics/FTP/Traffic Recieved (bytes/sec)*. Vaše statistiky by se měly shodovat s **Obr. 6.7**.



Obr. 6.7: Ukázka přijatého provozu pro FTP s ABR a UBR v bytes/sec.

6.3.2 Doplňující otázky a úkoly

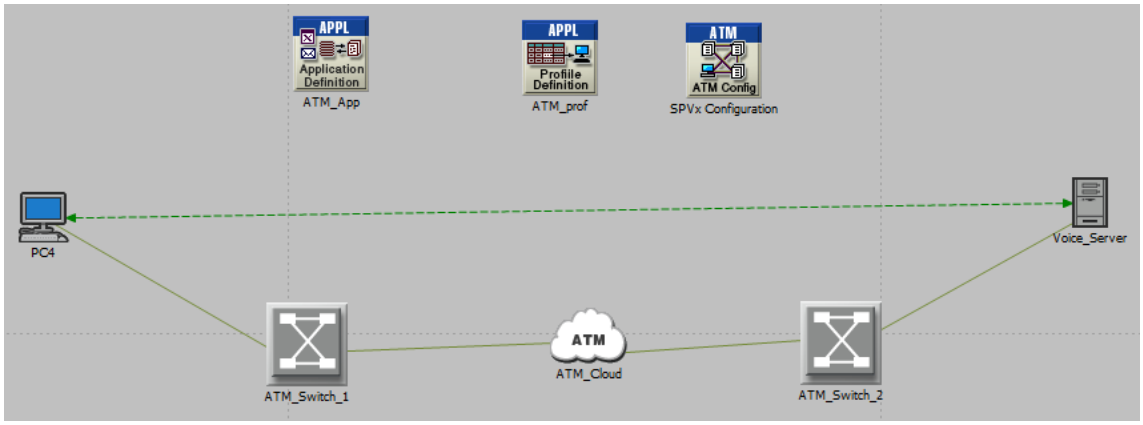
1) Zobrazte si výsledky s oběma scénáři s FTP pro *Global Statistics/FTP/Download Response Time(sec)* a zamyslete se nad výsledky.

6.4 Úkol 3 – Srovnání AAL s hlasovou aplikací

V následujícím úkolu se podíváte na chování dvou adaptačních vrstev AAL pro Voice aplikaci, konkrétně se bude jednat o třídu CBR s AAL1 a ABR s AAL5 v těchto předem připravených scénářích.

6.4.1 Postup

1. Současný projekt si uložte (Ctrl + S).
2. Přepněte se do jednoho z těchto předpřipravených scénářů *„ATM_Voice_CBR_AAL1“*, *„ATM_Voice_ABR_AAL5“*. Zobrazí se okno s připravenou topologií, viz **Obr. 6.8**. Pomocí *View Results* si zobrazte následující výsledky.



Obr. 6.8: Ukázka výchozí sítě pro následující scénáře.

6.4.2 Doplňující otázky a úkoly

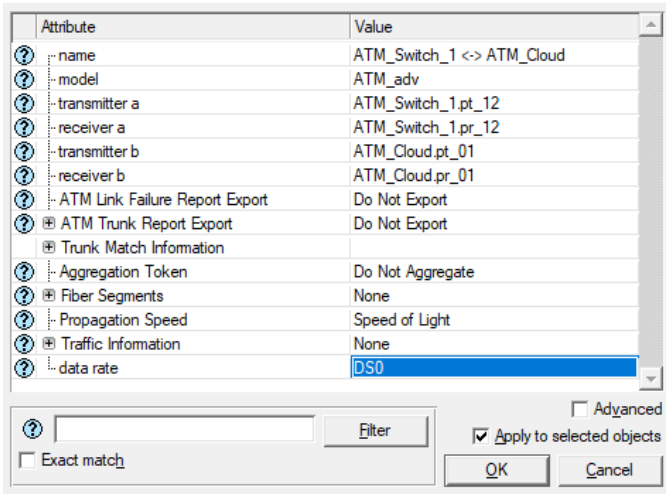
1) Následující statistiky zobrazte pro oba scénáře do jednoho grafu (*Overlaid Statistics*) s pomocí průměrování *(average)*. Prohlédněte si výsledný graf pro *Global Statistics/Voice/MOS Value*. Vysvětlete, co je MOS.
2) Diskutujte statistiky pro *Global Statistics/Voice/Packet End-to-End-(sec)* a *Packet Delay Variation*.

6.5 Úkol 4 – Srovnání Frame Relay a ATM

V tomto úkolu se podíváte na základní rozdíly mezi technologiemi Frame Relay a ATM. Opět máte k dispozici dva předpřipravené scénáře, první z prvky ATM a druhý z prvky pro Frame Relay. Na obou sítích je nakonfigurována hlasová aplikace s přenosovými linkami E1.

6.5.1 Postup

1. Současný projekt si uložte (Ctrl + S).
2. Přepněte se do předpřipraveného scénáře *„Srovnani_ATM_FR_Voice“* a ten duplikujte *Scenarios/Duplicate Scenario* a vhodně pojmenujte například *„ATM_efektivita“*
3. V nově vytvořeném scénáři pomocí tlačítka Ctrl označte všechny čtyři ATM spoje a vstupte do nastavení parametrů jednoho z nich *(Edit Attributes)*. V nově otevřeném okně najděte položku *data rate*, kterou změňte na hodnotu *DS0*. Poté dole na pravé straně nad tlačítkem OK zaškrtněte kolonku *Apply to selected objects* a výběr potvrďte, viz **Obr. 6.9**.



Obr. 6.9: Ukázka nastavení všech linek na přenosovou rychlost DS0.

4. Nyní duplikujte i druhý scénář *„Srovnani_FR_ATM_Voice“*, kde linky nastavte stejně dle postupu v bodě 3.
5. Po vytvoření scénářů spusťte hromadnou simulaci pomocí položky menu *Scenarios/Manage Scenarios*, délku simulací (*Sim Duration*) ponechte na 10 minut. Klikněte na položku *Results* a vyberte *<collect all>*. Poté klikněte na tlačítko *OK* pro spuštění simulace.

6.5.2 Doplňující otázky a úkoly

1) Zobrazte si statistiky *Global Statistics/ATM/Cell Delay(sec) a Global Statistics/Frame Relay/Delay(sec)* pro scénáře *„Srovnani_ATM_FR_Voice“* a *„Srovnani_FR_ATM_Voice“*, oba scénáře dejte do jednoho grafu (*Overlaid Statistics*).
2) Zobrazte si statistiky *Global Statistics/ATM/Cell Delay Variation a Global Statistics/Frame Relay/Delay Variance* pro scénáře *„Srovnani_ATM_FR_Voice“* a *„Srovnani_FR_ATM_Voice“*, oba scénáře dejte do jednoho grafu (*Overlaid Statistics*).
3) Porovnejte propustnost linek ATM a Frame Relay pro statistiku *Object*

Statistics/Campus Network/ATM_Switch_1 ↔ ATM_Cloud[0]/point-to-point/throughput (bites/sec) → a *Object Statistics/Campus Network/FR_1 ↔ FR_Cloud[0]/point-to-point/throughput (bites/sec)* ->. Vysvětlete, z jakého důvodu přenese ATM větší objem dat.

4) Ve scénářích *„FR_efektivita“* a *„ATM_efektivita“* se podívejte na statistiky pro *Global Statistics/Voice/Traffic Sent (bytes/sec)* a *Global Statistics*/Voice/*Traffic Received (bytes/sec)*. Obě statistiky dejte do jednoho grafu. Jaká technologie je efektivnější pro rychlost linky DS0 a proč?

7 PRÁCE S PROTOKOLY IPV4 A IPV6

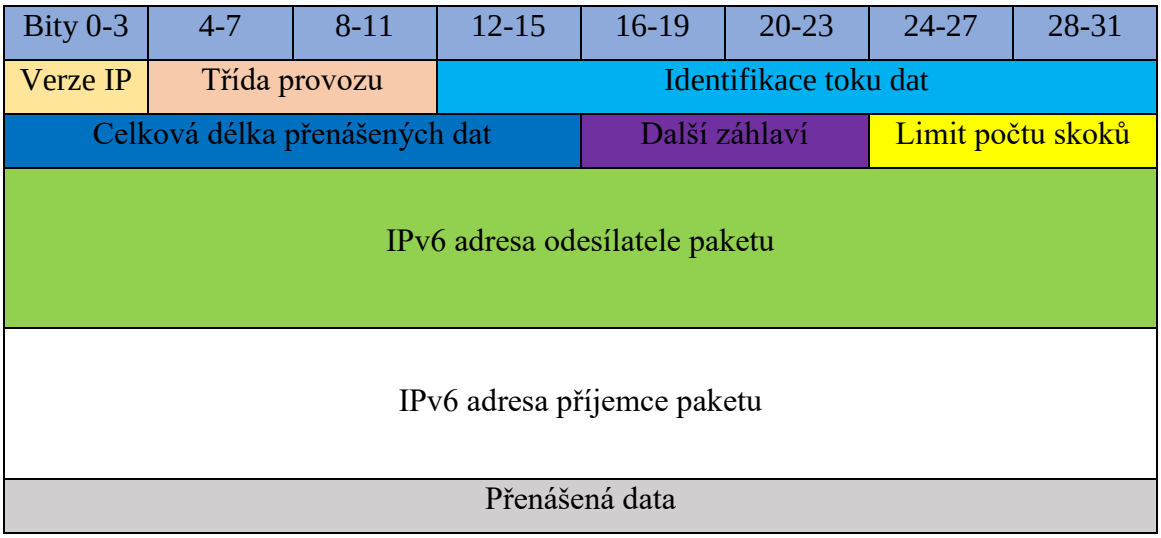
7.1 Úvod k laboratorní úloze

Na síťové vrstvě se nachází Internet Protocol – *(IP)*, který přenáší pakety od zdroje k cíli na základě IP adres v hlavičce paketu. Dosud nejrozšířenějším je Internet Protocol verze 4 neboli IPv4. Tento paketově orientovaný protokol se používá v sítích s přepojováním paketů. IP adresy jsou 32-bitové a velikost adresního prostoru je 2^{32} adres. IP adresa je tvořena číselnou adresou sítě a číselnou adresou hostitelského počítače. Z formátu IPv4 datagramu, který je znázorněn na **Obr. 7.1**, je pro tuto úlohu zajímavá hlavně **Délka záhlaví**, což je 4 bitová hodnota, kde minimální délka záhlaví je **20 bajtů**, naopak maximální délka je 60 bajtů. Délka záhlaví může být proměnlivá, ale vždy musí být v násobcích 32 bitů. Dále pole s názvem **Příznaky**, což je 3 bitová hodnota. Pokud je zde nataven DF-bit *(don't fragment)*, datagram nebude fragmentován. V případě MF-bit *(more fragments)* byl datagram fragmentován a bude následovat další část fragmentu.

<table>
<tr><td>Bity 0-3</td><td>4-7</td><td>8-15</td><td>16-18</td><td>19-31</td></tr>
<tr><td>Verze IP</td><td>Délka záhlaví</td><td>Typ služby</td><td colspan="2">Celková délka IP datagramu</td></tr>
<tr><td colspan="3">Identifikace IP datagramu</td><td>Příznaky</td><td>Posunutí fragmentu od počátku</td></tr>
<tr><td colspan="2">Doba života (TTL)</td><td>Protokol vyšší vrstvy</td><td colspan="2">Kontrolní součet záhlaví datagramu</td></tr>
<tr><td colspan="5">IP adresa odesílatele paketu</td></tr>
<tr><td colspan="5">IP adresa příjemce paketu</td></tr>
<tr><td colspan="5">Volitelné položky záhlaví</td></tr>
<tr><td colspan="5">Přenášená data</td></tr>
</table>

Obr. 7.1: Formát IPv4 datagramu.

Počátkem devadesátých let 20. století začalo být zřejmé, že adresní prostor IPv4 není dostačující. S IPv4 protokolem došlo k velkému rozšíření techniky NAT, díky které není možná skutečná **komunikace modelu end-to-end**. Z tohoto důvodu vnikl protokol IPv6, který řeší i nedostatky jako stále narůstající velikost směrovacích tabulek nebo neustále se zvyšující nároky na přenosovou rychlost. Při porovnání formátu IPv6 datagramu, viz **Obr. 7.2,** s IPv4 datagramem, je zřejmé, že se IPv6 datagram maximálně zjednodušil. Jsou zde vynechány položky fragmentace, kontrolního součtu, rozšiřujících voleb a délky záhlaví. IPv6 datagram dostal pevnou délku hlavičky **40 bajtů**, z nichž 32 B tvoří adresy odesilatele a příjemce a obsahuje již jen ty nejdůležitější položky. Ve formátu IPv6 datagramu je z hlediska této úlohy důležitá položka **další záhlaví**, což je 8 bitová hodnota, která mimo jiné udává i rozšiřující informace o **fragmentaci**. V tomto poli může být odkaz na přítomnost protokolu z vyšší vrstvy, například protokolů TCP a UDP.



Obr. 7.2: Formát IPv6 datagramu.

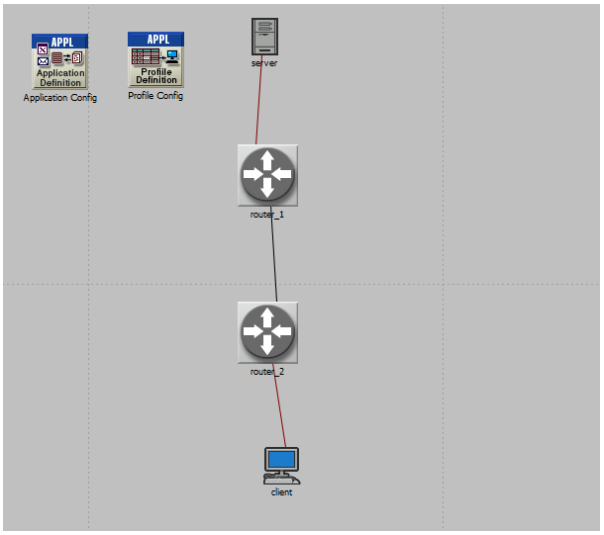
V poslední části se tato úloha zabývá fragmentací IP datagramů, a to jak pro IPv4, tak pro IPv6 protokol. Ačkoliv má fragmentace pro oba protokoly stejné rysy, jsou zde výrazné odlišnosti. Protože je velikost paketu rozdílná v různých sítích, musí být k dispozici mechanizmy, které rozdělí příliš velké datagramy na menší části označované také jako fragmenty. Fragmenty se mohou k příjemci šířit různými cestami, z tohoto důvodu se **fragmenty sestavují až u příjemce**. Fragmentované datagramy je možné dále fragmentovat a v IPv4 může fragmentovat jak **odesílající uzel**, tak i kterýkoliv **směrovač** na cestě. Maximální hodnota datagramu je 65536 bajtů a je možné ji také označit jako MTU – *(Maximum Transmission Unit)*. Může nastat situace, kdy fragmentace není možná, je nastaven příznakový bit na DF *(don't fragment)*, v takovém případě je na směrovači datagram zahozen a odesilatel je informován o chybě a nemožnosti fragmentace pomocí ICMP zprávy. V Riverbedu je minimální velikost MTU **20 bajtů**. Fragmentace se v IPv6 neprovádí tak často a primárně se využívají defaultní délky s 1500 bajty, minimální velikost fragmentu je **1280 bajtů**. Fragmentaci u IPv6 může provést **pouze odesilatel**. Pokud bude na lince nastavena příliš malá velikost MTU, bude datagram zahozen a odesilateli se zašle ICPMv6 chybová zpráva.

7.2 Úkol 1 – Konfigurace protokolů IPv4 a IPv6

V prvním úkolu si po stažení předvypracovaného projektu vyzkoušíte konfiguraci IPv4 a IPv6 pro malou síť. Následně si vygenerujete směrovací tabulky a oba protokoly mezi sebou porovnáte.

7.2.1 Postup

1. Stáhněte projekt *„IPv4_IPv6_predvytvorene"*, který si uložte na Plochu.
2. Spusťte program Riverbed Modeler.
3. V menu programu vyberte **File/Open…(**Ctrl + O) a otevřete projekt *„IPv4_IPv6_predvytvorene"*. Zobrazí se okno s připravenou topologií, viz **Obr. 7.3**.



Obr. 7.3: Ukázka výchozí sítě projekt *„IPv4_IPv6_predvytvorene“*.

4. V projektu je vytvořeno několik scénářů, se kterými budete postupně pracovat. Nyní se přepněte do scénáře s názvem *„IPv4“ (Scenarios/Switch To Scenario)*. Jako aplikace je zde nastaveno HTTP.
5. Ve scénáři se nacházejí čtyři prvky, které budete konfigurovat *(2x router, client a server)*. Pro správnou komunikaci v této síti je nutné přiřadit všem čtyřem prvkům jedinečnou adresu v rámci celé sítě.
6. IPv4 adresu si na tyto prvky můžete zvolit dle svého uvážení. Nicméně je možné spočítat rozsah IP adres, jež mohou být v sítí použity, pokud znáte číslo sítě a masku nacházející se v této síti.
7. Příklad pro určení rozsahu IP adres, které bude možné použít pro počítače, viz **Tab. 7.**1, když je známa například IP adresa 192.0.0.2/28. Maximálně lze použít 14 IP adres pro síť 192.0.0.0/28. Z toho 192.0.0.1 je první použitelná adresa, která se binárně zvětší o jedničku, aby byla větší než adresa sítě. Poslední možná IP adresa je 192.0.0.14. Broadcast bude 192.0.0.15, což binárně značí poslední čtyři jedničky na konci. Maximální počet podsítí bude 16 (2^4 - 2), jedna adresa je využita právě pro broadcast a jedna pro adresu sítě.

Tab. 7.1: Učení rozsahu IP adres.

IP adresa binárně:	**11000000.00000000.00000000.00000010**	**192.0.0.2**
Maska binárně:	11111111.11111111.11111111.11110000 **(28 bitů)**	255.255.255.240
Inverzní (Wildcard):	00000000.00000000.00000000.00001111	0.0.0.15
Adresa sítě:	11000000.00000000.00000000.00000000 **(Třída C)**	192.0.0.0/28
Broadcast:	11000000.00000000.00000000.00001111	192.0.0.15
První IP:	11000000.00000000.00000000.00000001	192.0.0.1

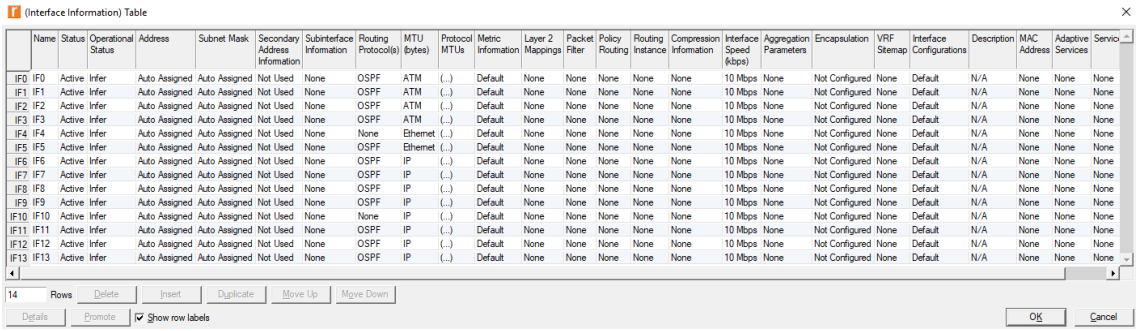
Poslední IP:	11000000.00000000.00000000.00001110	192.0.0.14

8. Pomocí logického součinu lze ověřit, zda daná IP adresa patří do dané sítě. Například ověření, zda IP adresa 192.0.0.9 patří do sítě 192.0.0.0/28.

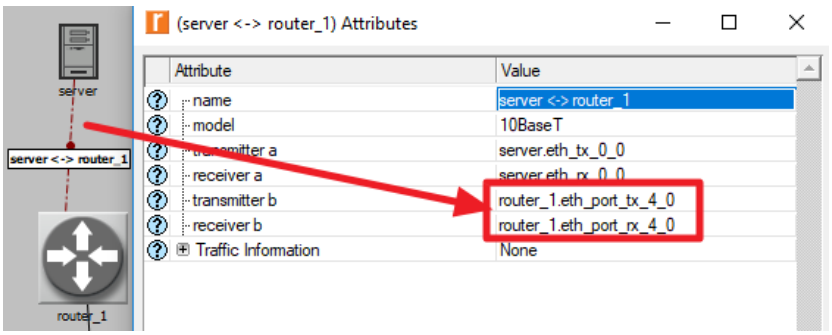
Tab. 7.2: Příslušnost IP adresy k síti.

Maska binárně:	**11111111.11111111.11111111.11110000**	**255.255.255.240**
IP adresa binárně:	11000000.00000000.00000000.00001001	192.0.0.9
Logický součin:	11000000.00000000.00000000.00000000	192.0.0.0 - výsledek

9. Nyní provedete konfiguraci postupně na všech čtyřech prvcích. Například pro *router_1* zobrazíte jeho atributy pomocí pravého tlačítka, kliknete na směrovač a vyberete *Edit Attributes*. Dále na položku *IP → IP Routing Parameters → Interface Information*. Seznam nerozklikávejte, ale klikněte na „ ... “, čímž se otevře nové okno stejné jako na **Obr. 7.4**. Ještě předtím než se pustíte do konfigurace IP adres, budete potřebovat vědět, k jakému portu je daná linka přiřazena. To zjistíte tak, že si zobrazíte atributy linky, kterou budete chtít konfigurovat. Na **Obr. 7.5** můžete vidět, že linka ze *serveru* do *routeru_1* je připojena do portu **IF4**. V *Interface Inforamation* nastavíte pro *router_1* u **IF4** položku *Adress* a položku *Subnet Mask* na vámi zvolenou IP adresu a masku sítě.

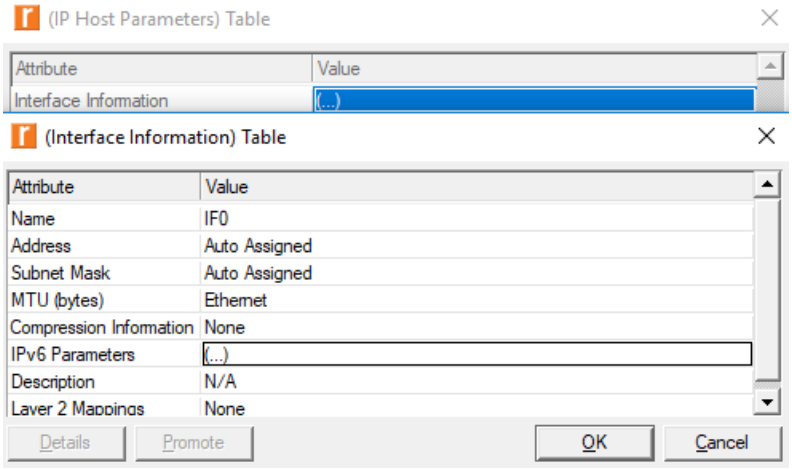


Obr. 7.4: Ukázka okna Interface Information routery v IPv4.



Obr. 7.5: Atributy linky server ↔ router_1.

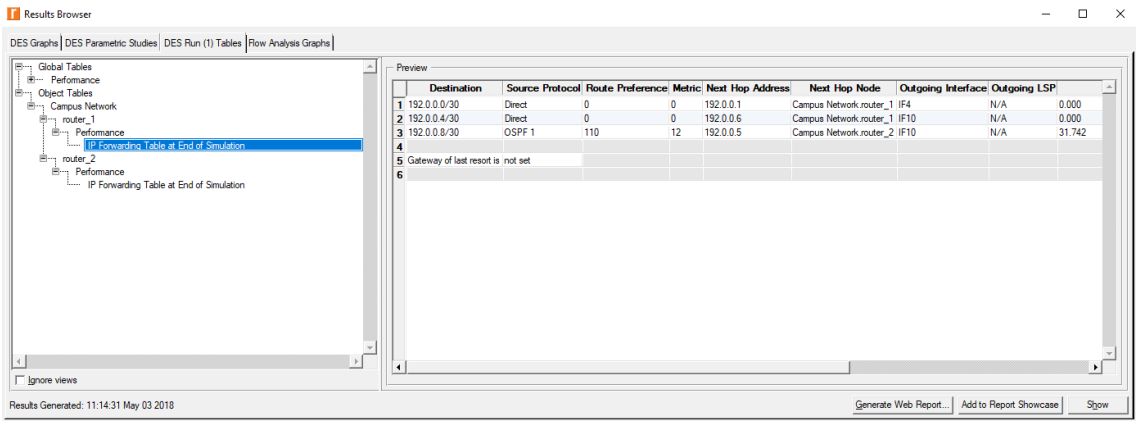
10. IP adresa pro server a clienta se konfiguruje podobně jako pro routery, ale je zde rozdílná cesta. Po editaci atributů serveru nebo clienta klikněte na položku *IP → IP Host Parameters → Interface Information*, seznam nerozklikávejte, ale klikněte na „ ... “, čímž se otevře nové okno stejné jako na **Obr. 7.6**. Zde opět vyplníte vámi vybranou IP adresu a masku sítě.



Obr. 7.6: Konfigurace pro clienta a server v IPv4.

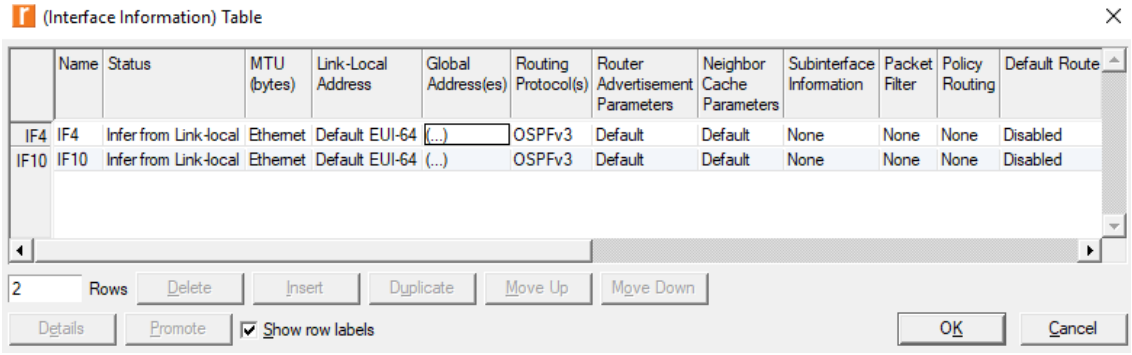
11. Nyní musíte nastavit IP adresy i na zbylých prvcích (celkem budete doplňovat **šest** IP adres) stejným postupem jako v bodech 9 a 10.

12. Nyní, když jsou všechny IP adresy nastaveny, můžete si zobrazit směrovací tabulku. Označte všechny čtyři prvky a na hlavní liště zvolte *Protocols → IP → Routing → Export Routing Tables → Selected nodes → OK*. Projekt uložte a spusťte simulaci na liště, zvolte ikonu *Configure/Run DES*, délku simulace ponechte na 10 minut. Pro zobrazení směrovacích tabulek klikněte na *View Results → DES Run () Tables*. Po rozkliknutí položky *Object Tables* si můžete prohlédnout jednotlivé směrovací tabulky. Směrovací tabulky mohou vypadat jako na **Obr. 7.7**. Ve směrovacích tabulkách si všimněte, že sítě, které jste konfigurovali, jsou připojené přímo s metrikou 0 a zbytek je naučen od druhého směrovače díky směrovacímu protokolu OSPF.

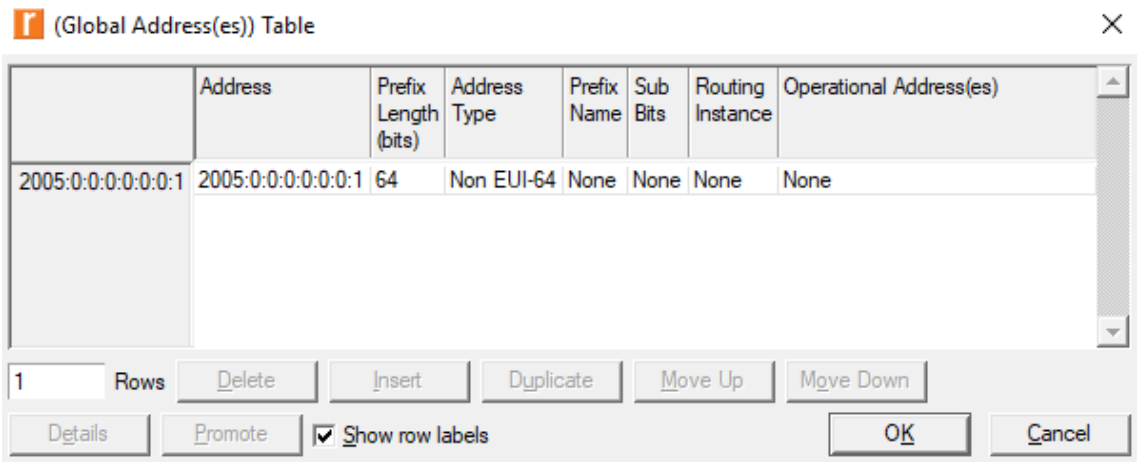


Obr. 7.7: Ukázka směrovací tabulky pro router_1.

13. Nyní se přepněte do scénáře s názvem *„IPv6“ (Scenarios/Switch To Scenario)*. Zde budete podobně jako v předchozím případě konfigurovat protokol IPv6 opět na všech čtyřech prvcích. Například pro *router_1* zobrazíte jeho atributy pomocí pravého tlačítka, kliknete na směrovač a vyberete *Edit Attributes*. Dále na položku *IP → IPv6 Parameters → Interface Information* seznam nerozklikávejte, ale klikněte na *„…“*. V tabulce *Interface Information* budete nastavovat na všech prvcích *Link-Local Adress* na *Default EUI-64* a *Routing Protocol(s)* na *OSPFv3*, viz **Obr. 7.8**. Poté vstupte do tabulky Global Address(es), kde vyplníte adresu a délku prefixu podle **Obr. 7.9**.



Obr. 7.8: Tabulka Interface Information pro IPv6.



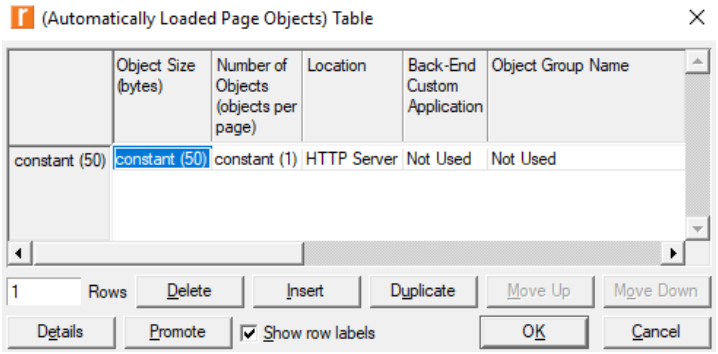
Obr. 7.9: Nastavení IPv6 v tabulce Global Adress(es).

14. IPv6 adresa pro server a clienta se konfiguruje podobně jako pro routery, ale je zde opět rozdílná cesta. Pro editaci atributů serveru nebo clienta klikněte na položku *IP → IP Host Parameters → Interface Information → IPv6 Parameters → Global Adress(es)*, seznam nerozklikávejte, ale klikněte na *„…“*. V tabulce IPv6 Parametres nastavíte jak pro clienta, tak pro server položku *Link-Local Adress* na *Default EUI-64* a v tabulce Global Adress(es)) IPv6 adresu.
15. Stejným postupem jako v předchozích dvou bodech nakonfigurujte i zbytek prvků v síti dle tabulky **Tab. 7.3**. Poté opět vygenerujte směrovací tabulku a opět ručně spusťte simulaci (délka simulace bude stejná jako v bodě **12**, tedy 10 minut). Na směrovací tabulku se po dokončení simulace podívejte.

Tab. 7.3: Konfigurace IPv6.

Prvek	Port	IPv6 adresa
server	IF0	2005:0:0:0:0:0:0:2
router_1	IF4	2005:0:0:0:0:0:0:1
	IF10	2005:0:0:1:0:0:0:2
router_2	IF10	2005:0:0:1:0:0:0:1
	IF4	2005:0:0:2:0:0:0:1
client	IF0	2005:0:0:2:0:0:0:2

16. Nyní duplikujte scénář *„IPv4“* (*Scenarios/Duplicate Scenario)* a nazvěte ho *„IPv4_mensi_provoz“*. V tomto scénáři na Application Configu snížíte provoz aplikace HTTP. Editujte atributy komponenty *Application Conf* a poté rozklikejte *Application Definitions* → *app_HTTP* → *(...)* → *Page Properties* → *Automatically Loaded Page Objects.* V této tabulce odstraníte položku **Short Video** tím, že nastavíte počet řádků **Rows** na **1**. Položku v *Object Size (bytes)* nastavte na **constant (50)**, viz **Obr. 7.10**.



Obr. 7.10: Nastavení nízkého provozu pro aplikaci HTTP.

17. Obdobně duplikujete scénář *„IPv6“*, pojmenujete jej *„IPv6_mensi_provoz“* a proveďte stejné nastavení pro komponentu *Application Conf* jako v přechozím bodě. Oba nově vytvořené scénáře ručně známým způsobem odsimulujte.

7.2.2 Doplňující otázky a úkoly

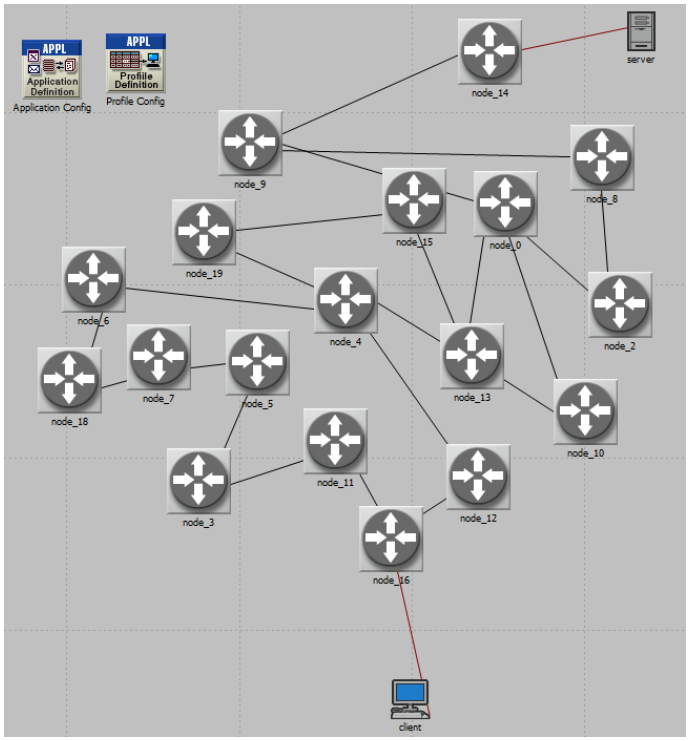
1) Porovnejte mezi sebou propustnost linek pro scénáře *„IPv4“* a *„IPv6“* pro statistiku *Object Statistics/Campus/router_1* ↔ *router_2/point-to-point throughput (bites/sec)* pro oba směry ↔. Scénáře si zobrazte v jednom grafu pomocí *Overlaid Statistics*.
2) Stejně tak mezi sebou porovnejte scénáře *„IPv4_mensi_provoz“* a *„IPv6_mensi_provoz“*, opět pro *Object Statistics/Campus/router_1* ↔ *router_2/point-to-point throughput (bites/sec)* →.
3) Zdůvodněte, proč IPv6 přenáší větší objem dat než IPv4.

7.3 Úkol 2 – Porovnání IPv4 a IPv6 v závislosti na velikosti sítě

V tomto úkolu si porovnáte IPv4 a IPv6 s předchozí malou sítí a sítí rozsáhlejší v závislosti na jejich odezvě.

7.3.1 Postup

1. Předchozí scénáře si uložte (Ctrl + S) a přepněte se do scénáře *„IPv4_vetsi_sit“ (Scenarios/Switch To Scenario).* Zobrazí se okno s připravenou topologií, viz **Obr. 7.**11 Myší označte celou síť a opět vygenerujte směrovací tabulku *Protocols → IP → Routing → Export Routing Tables → Selected nodes → OK*. Stejně tak vygenerujte směrovací tabulku i pro scénář *„IPv6_vetsi_sit“*. Tyto dva scénáře jsou již nakonfigurovány a zamyslíte se jen nad výsledky.



Obr. 7.11: Ukázka topologie s rozsáhlejší sítí v IPv4.

2. Pomocí *Scenarios/Manage Scenarios* na hlavní liště spusťte hromadnou simulaci pro všechny scénáře kliknutím na *Results* a vybráním *<collect all>*, délka simulace bude 10 minut, viz **Obr. 7.12**. Po odsimulování si můžete opět prohlédnout směrovací tabulky pro nové scénáře.

Obr. 7.12: Hromadná simulace scénářů.

7.3.2 Doplňující otázky a úkoly

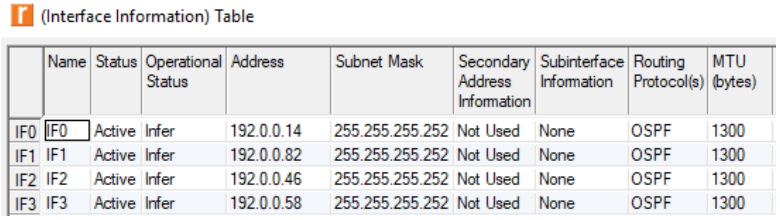
1) Zobrazte si statistiky pro scénáře „IPv4“ a „IPv6“ pro *Global Statistics/HTTP/Object Response Time (seconds)* a *Global Statistics/HTTP/Page Response Time (seconds)*.
2) Obdobně jako v předchozím bodě si pro scénáře *„IPv4_vetsi_sit“* a *„IPv6_vetsi_sit“* zobrazte statistiky *Global Statistics/HTTP/Object Response Time (seconds)* a *Global Statistics/HTTP/Page Response Time (seconds)*.
3) Z jakého důvodu vychází v prvním bodě IPv6 z hlediska odezvy hůře a naopak lépe v bodě druhém?

7.4 Úkol 3 – Fragmentace

Poslední úkol je zaměřen na práci s MTU – *(Maximum Transmission Unit)*. V následujících scénářích se podíváte, jak se projeví fragmentace paketů s použitím protokolů IPv4 a IPv6.

7.4.1 Postup

1. Předchozí scénáře si uložte (Ctrl + S). Nyní budete pracovat s předvypracovanými scénáři *„IPv4_MTU“* a *„IPv6_MTU“*. V těchto scénářích je nastavena jako aplikace video konference a **jednotka MTU je zde defaultně na 1500 bajtů**. Z těchto dvou scénářů budete vycházet a budou sloužit pro porovnání s ostatními nově vytvořenými scénáři.
2. Nejprve se přepněte se do scénáře *„IPv4_MTU“ (Scenarios/Switch To Scenario)*, ten duplikujte *(Scenarios/Duplicate Scenario)* a pojmenujte ho například *„IPv4_fragmentace_na_ceste“*. V tomto nově vzniklém scénáři budete nastavovat na *node_4* velikost MTU na 1300 bajtů a zbytek ponechejte beze změny.
3. Pro *router_4* zobrazíte jeho atributy pomocí pravého tlačítka, kliknete na směrovač a vyberete *Edit Attributes*. Dále na položku *IP → IP Routing Parameters → Interface Information*, seznam nerozklikávejte, ale klikněte na *„…“*, čímž se otevře nové okno. Poté budete editovat položku *MTU (bytes)*, kde ručně zadáte hodnotu 1300 pro všechny čtyři vyplněné IP adresy, viz **Obr. 7.13**.



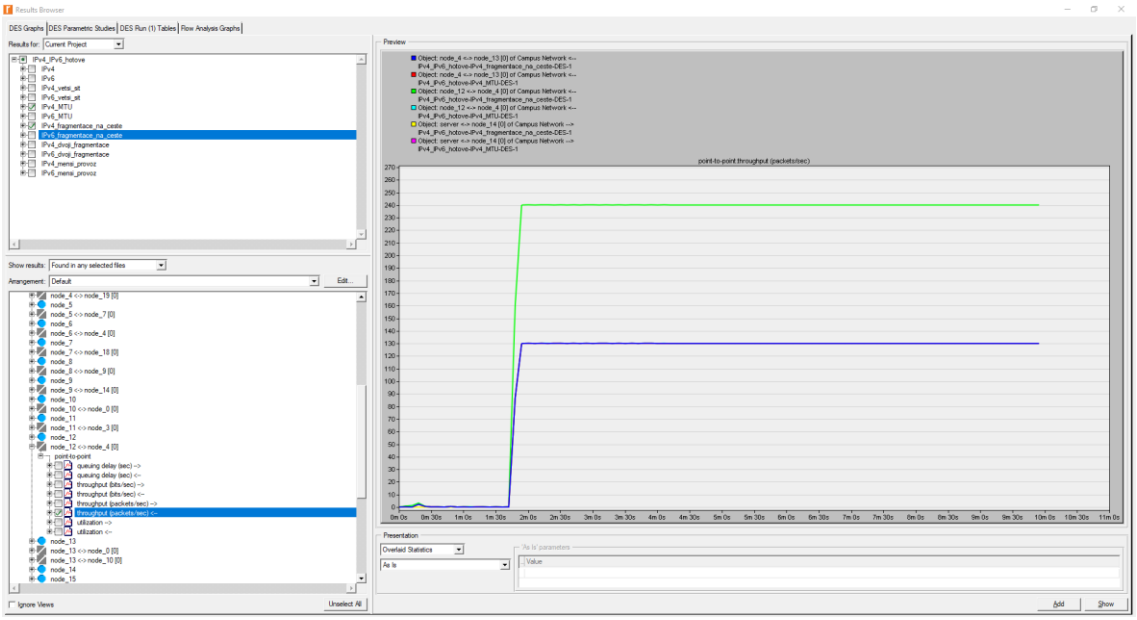
Obr. 7.13: Nastavení hodnoty MTU.

4. Přepněte se do scénáře *„IP6_MTU“*, který rovněž duplikujte. Nové jméno scénáře zadejte *„IPv6_fragmentace_na_ceste“*, nastavíte jej stejně jako předchozí scénář s tím rozdílem, že zde budete nastavovat *MTU (bytes)* v *IP → IPv6 Parameters → Interface Information*. Opět pro *node_4*, jež bude mít hodnotu *MTU 1300 bajtů*. Tuto hodnotu musíte stejně jako v předchozím bodě nakonfigurovat na **všechny připojené linky**.
5. Scénář *„IPv4_fragmentace_na_ceste“* opět duplikujte a pojmenujte „IPv4_dvoji_fragmentace“. Nyní budete nastavovat MTU na směrovači *node_9* na hodnotu 1400 bajtů stejným postupem jako **v třetím bodě**. To stejné musíte opět nastavit i pro IPv6 (MTU pro IPv6 budete konfigurovat podobným způsobem jako **ve čtvrtém bodě**), kde musíte opět duplikovat scénář *„IPv6_fragmentace_na_ceste“* a zadat nový název *„IPv6_dvoji_fragmentace“*. Po vytvoření a konfiguraci musíte mít na obou nově vzniklých scénářích nastavenu jednotku MTU na směrovači *node_9* na **1400** bajtů a na směrovači *node_4* hodnotu **1300** bajtů. Je třeba si dát pozor na to, abyste konfigurovali **správné IF porty**.
6. Známým způsobem spusťte hromadnou simulaci pro všechny scénáře *(Scenarios/Manage Scenarios)*, délka simulace bude opět nastavena na 10 minut.

7.4.2 Doplňující otázky a úkoly

1) Zobrazte statistiky *(pomocí View Results)* zároveň pro scénáře *„IPv4_MTU“* a *„IPv4_fragmentace_na_ceste“* (stačí v levém horním rohu dát místo *Current Scenario → Current Project* a zaškrtnout příslušný scénář):
   - *Object Statistics/Campus Network/server ↔ node_14 [0]/point-to-point/throughput (packets/sec)* → (směr od serveru k node_14)
   - *Object Statistics/Campus Network/node_4 ↔ node_13 [0]/point-to-point/throughput (packets/sec)* ← (směr od node_13 k node_4)
   - *Object Statistics/Campus Network/node_12 ↔ node_4 [0]/point-to-point/throughput (packets/sec)* ← (směr od node_4 k node _12)

Všechny tři linky si potupně zobrazte, výsledné pakety na lince si zapište a porovnejte je mezi sebou s ostatními linkami, viz **Obr. 7.14**.



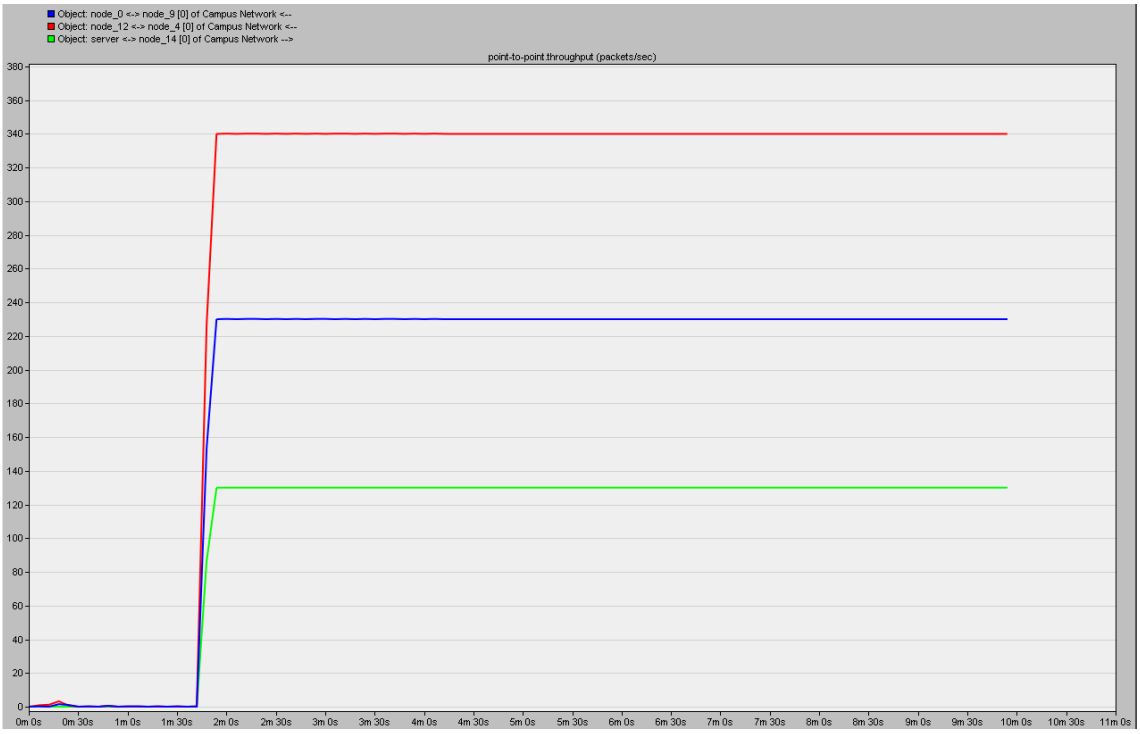
Obr. 7.14: Ukázka statistik pro scénáře *„IPv4_MTU“* a *„IPv4_fragmentace_na_ceste“*.

2) Stejné statistiky jako v prvním bodě si zobrazte i pro scénáře *„IPv6_MTU“* a *„IPv6_fragmentace_na_ceste“*. Případně si můžete zobrazit scénáře s IPv4 a IPv6 dohromady. Zamyslete se nad chováním IPv6 protokolu.

3) Pro scénář *„IPv4_dvoji_fragmentace“* zobrazte statistiky:

   - *Object Statistics/Campus Network/server ↔ node_14 [0]/point-to-point/throughput (packets/sec)* → (směr od serveru k node_14)
   - *Object Statistics/Campus Network/node_0 ↔ node_9 [0]/point-to-point/throughput (packets/sec)* ← (směr od node_9 k node_0)
   - *Object Statistics/Campus Network/node_12 ↔ node_4 [0]/point-to-point/throughput (packets/sec)* ← (směr od node_4 k node_12)

Všechny statistiky dejte do jednoho grafu v sekci *Presentation* vybráním *Overlaid Statistics*, výsledek by měl korespondovat s **Obr. 7.15**.



Obr. 7.15 Ukázka statistik pro *„IPv4_fragmentace_na_ceste“*.

4) Stejným způsobem jako ve třetím bodě zobrazte i scénář *„IPv6_dvoji_fragmentace“*. Výsledky pro oba scénáře mezi sebou porovnejte a zdůvodněte, proč je statistika s IPv6 neměnná.

5) Pro scénáře *„IPv4_dvoji_fragmentace“* a *„IPv6_dvoji_fragmentace“* zobrazte *Object Statistics/Campus Network/node_4/IP/Processing Delay (sec)*. Opět dejte obě statistiky do jednoho grafu *(Overlaid Statistics)*. Processing Delay vyjadřuje zpoždění od doby, kdy paket dorazí na IP vrstvu, až do jeho zpracování.

6) Jako poslední si zobrazte pro scénáře *„IPv4_dvoji_fragmentace“* a *„IPv6_dvoji_fragmentace“* statistiku *Global Statistics/Video Conferencing/Packet End-to-End Delay (sec)*. Statistiku můžete porovnat se scénáři *„IPv4_MTU“* a *„IPv6_MTU“*.

8 ZÁVĚR

Cílem diplomové práce bylo navrhnout laboratorní úlohy zabývající se transportními protokoly, přenosovými technologiemi a internetovými protokoly *(IP)* v simulačním prostředí od firmy OPNET. K návrhu laboratorní úlohy byl použit program Riverbed Modeler Academic Edition 17.5. Výhodou tohoto programu je jeho bezplatné používání pro vzdělávání, oproti plné placené verzi má omezenou funkčnost, což bylo v některých případech limitující, ale zároveň je tato verze dostačující pro studijní účely. Úvodní část slouží k seznámení se se simulačním prostředím Riverbed Modeler Academic Edition 17.5 a to včetně popsání všech jeho jednotlivých důležitých částí. Druhá část se věnuje teorii, v níž jsou zahrnuty protokoly TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), IPv4 *(Internet Protocol version 4),* IPv6 *(Internet Protocol version 6)* a technologie pro přenos převážně na páteřních sítích WAN *(Wide Area Network),* konkrétně ATM *(Asynchronous Transfer Mode)* a Frame Relay. Tyto teoretické poznatky byly využity při návrhu laboratorních úloh pro předmět Komunikační technologie.

První navržená laboratorní úloha byla zaměřena na protokoly transportní vrstvy TCP a UDP. První úkol byl zaměřen na rozdíl mezi TCP a UDP protokolem, především pak na to, jak ovlivňuje spojový charakter protokolu TCP dobu přenosu. Druhý a třetí úkol byl zaměřen na různá procentuální zahazování paketů, kde u TCP protokolu bylo vidět, že dochází k opětovnému posílání ztracených paketů, naopak protokol UDP takovéto mechanizmy nemá. Ve čtvrtém úkolu bylo možné pozorovat přenášené segmenty v FTP aplikaci. Bylo zde možné vidět, že se pro FTP server používá port 20, který je pro FTP vyhrazený. Naopak na straně klienta se náhodně zvolí pro každou aplikaci jiný port, aby bylo jednoznačně určeno, pro kterou aplikaci je daný segment určen. Pro úkol s výpadkem na lince bylo možné v grafech pozorovat, že se FTP aplikace bude snažit po navázání spojení odeslat ztracená data během výpadku, naopak hlasová aplikace, která používá protokol UDP, těmito mechanizmy nedisponuje. V posledním úkolu byl postupně navyšován provoz. Zatímco UDP bude data do sítě posílat, aniž by se starala o to, zda síť takovýto provoz zvládne, TCP zjistí, že se v síti provoz zahazuje a bude postupně snižovat provoz. Druhá laboratorní úloha, jež byla zaměřena na WAN technologie ATM a Frame Relay, se v prvním úkolu zabývala porovnáním tříd služeb CBR a UBR v ATM s aplikací video konference a příslušnými adaptačními vrstvami AAL. V této úloze byly například prakticky ověřeny teoretické znalosti, že pro hlasové služby, které vyžadují malé zpoždění a kolísání zpoždění s pevným časováním, je mnohem lepší třída CBR s adaptační vrstvou AAL1. V druhém úkolu byly vidět rozdíly tříd ABR a UBR pro FTP aplikaci. Naopak třetí úkol se zabýval adaptačními vrstvami AAL pro hlasovou aplikaci. Jsou zde porovnány třídy CRB s nastavenou adaptační vrstvou AAL1 a ABR s AAL5. Poslední úkol se zabýval srovnáním technologií Frame Relay a ATM s nakonfigurovanou hlasovou aplikací a přenosovými linkami E1 a jejich následnou změnou na pomalejší DS0 linku. V poslední laboratorní úloze se pracuje s protokoly IPv4 a IPv6, u kterých se v prvním úkolu konfigurují IP adresy, v malé síti s klientem, serverem a dvěma směrovači. V druhém úkolu je porovnáno zpoždění v HTTP aplikaci pro velkou síť. Je možné pozorovat, že díky zrychlenému směrování u IPv6 protokolu, který nepočítá CRC

a nedochází zde k fragmentaci, bude zpoždění pro velkou síť s mnoha směrovači nižší. V posledním úkolu bude možné pozorovat rozdíly ve fragmentaci IP datagramů pro IPv4 a IPv6 protokol. Zatímco v IPv4 může fragmentovat jak odesílající uzel, tak i kterýkoliv směrovač na cestě, v IPv6 může provést fragmentaci pouze odesilatel.

Úkoly byly sestaveny tak, aby se v první části student seznámil s danou problematikou úkolu a na konci každého úkolu byly studentovi zadány samostatné otázky a úkoly, jež by měl zvládnout vypracovat s pomocí nově nabytých znalostí. Otázky i úkoly vedou k zamyšlení se nad danou problematikou a k jejímu pochopení. Všechny tři laboratorní úlohy byly koncipovány tak, aby časová náročnost každé z nich byla přibližně dvě hodiny. K laboratorní úloze bylo vypracováno vzorové řešení a zdrojové soubory s vytvořenými scénáři.

LITERATURA

[1] ABOELELA, Imad. *Network simulátor experiment manual*. 4th ed. Oxford: Elsevier Science [distributor], c2008.

[2] SETHI, Adarshpal S. a Vasil Y. HNATYSHIN. *The practical OPNET user guide for computer network simulation*. Boca Raton, FL: CRC Press, c2013. ISBN 9781439812051.

[3] JEŘÁBEK, J. *Komunikační technologie*, Skriptum FEKT VUT v Brně, 175 stran, 2017. ISBN 9788021447134.

[4] PARZIALE, Lydia. *TCP/IP tutorial and technical overview*. 8th ed. United States: IBM International Technical Support Organization, c2006. ISBN 0738494682.

[5] *TCP, Transmission Control Protocol* [online]. 1998, 2012 [cit. 2017-11-02]. Dostupné z: http://www.networksorcery.com/enp/protocol/tcp.htm

[6] NOVOTNÝ, V. Architektura sítí, Skriptum FEKT VUT v Brně, 2011. 152 s.

[7] PETERKA, Jiří. Počítačová encyklopedie: ATM - Asynchronous Transfer Mode. CHIPweek [online]. 15.9.1998, [cit. 2018-01-30]. Dostupné z URL: http://www.earchiv.cz/a98/a838k180.php3

[8] SLAVÍČEK, K. ATM. Zpravodaj ÚVT MU. ISSN 1212-0901, 1996, roč. VI, č. 4, s. 14-17. [cit. 2018-01-25]. Dostupné z URL: http://webserver.ics.muni.cz/bulletin/articles/68.html

[9] STALLINGS, William. *DATA AND COMPUTER COMMUNICATIONS*. Eighth Edition. New Jersey 07458: Upper Saddle River, 2007. ISBN 0-13-243310-9.

[10] SATRAPA, Pavel. *IPv6: internetový protokol verze 6*. 3., aktualiz. a dopl. vyd. Praha: CZ.NIC, c2011. CZ.NIC. ISBN 978-80-904248-4-5.

[11] DOSTÁLEK, Libor a Alena KABELOVÁ. *Velký průvodce protokoly TCP/IP a systémem DNS.* 2. aktualiz. vyd. Praha: Computer Press, 2000. Komunikace & sítě. ISBN 80-7226-323-4.

SEZNAM ZKRATEK

DNS – Domain Name System

DHCP – Dynamic Host Configuration Protocol

RIP – Routing Information Protocol

SNMP – Simple Network Management Protocol

FTP – File Transfer Protocol

HTTP – Hyper Text Transfer Protocol

SMTP – Simple Mail Transfer Protocol

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

WAN – Wide Area Network

ATM – Asynchronous Transfer Mode

CBR – Constant Bit Rate

VBR – Variable Bit Rate

ABR – Available Bit Rate

UBR – Unspecified Bit Rate

AAL – ATM Adaptation Layer

VCC – Virtual Channel Connection

VPC – Virtual Path Connection

VCI – Virtual Channel Identifier

VPI – Virtual Path Identifier

VC – Virtual Circuit

DLCI – Data Link Connection Identifier

PVC – Permanent Virtual Circuit

SVC – Switched Virtual Circuit

EA – Extended Address

C/R – Command response bit

FCS – Frame Check Sequence

IP – Internet Protocol

IPv4 – Internet Protocol version 4

IPv6 – Internet Protocol version 6

DF-bit – Don’t fragment

MF-bit – More fragments

MTU – Maximum Transmission Unit

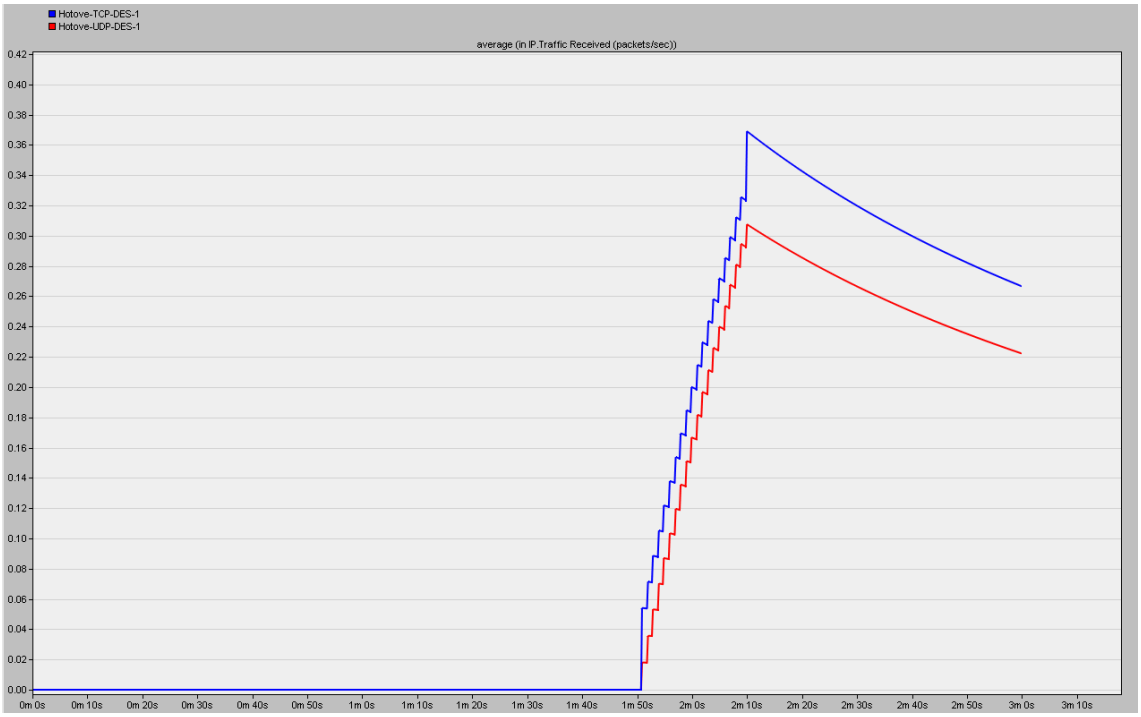
ICMPv6 – Internet Control Message Protocol Version 6

A ŘEŠENÍ OTÁZEK A ÚKOLŮ PRO SROVNÁNÍ TCP A UDP PROTOKLŮ

Tato laboratorní úloha je zaměřena na srovnání dvou základních protokolů TCP a UDP v první části. V druhé části se zaměřuje na porovnání TCP protokolu s určitým procentem zahozených paketů. Pro úkol s výpadkem na lince bylo možné v grafech pozorovat, že se FTP aplikace bude snažit po navázání spojení odeslat ztracená data během výpadku, naopak hlasová aplikace, která používá protokol UDP, těmito mechanizmy nedisponuje. V posledním úkolu byl ověřen fakt, že hlas a obraz má přednost před daty a to pomocí FTP, Videokonference a postupného zvyšování provozu pro Videokonferenci.

A.1 Úkol 1

Tato úloha byla již předvytvořena a jsou zde dva scénáře, jeden pro TCP a druhý pro UDP. Každý z těchto protokolů je vhodný pro jiné aplikace. V tomto úkolu bude řešeno seznámení s výsledky simulací pro tyto dva transportní protokoly. Z výsledků je patrné, že pro přenos protokolu TCP bude přenášen vyšší objem dat než pro protokol UDP. Tento fakt je způsoben tím, že protokol TCP na rozdíl od UDP musí navázat a potvrzovat spojení a má větší záhlaví, jak je vidět na **Obr. A.1.**



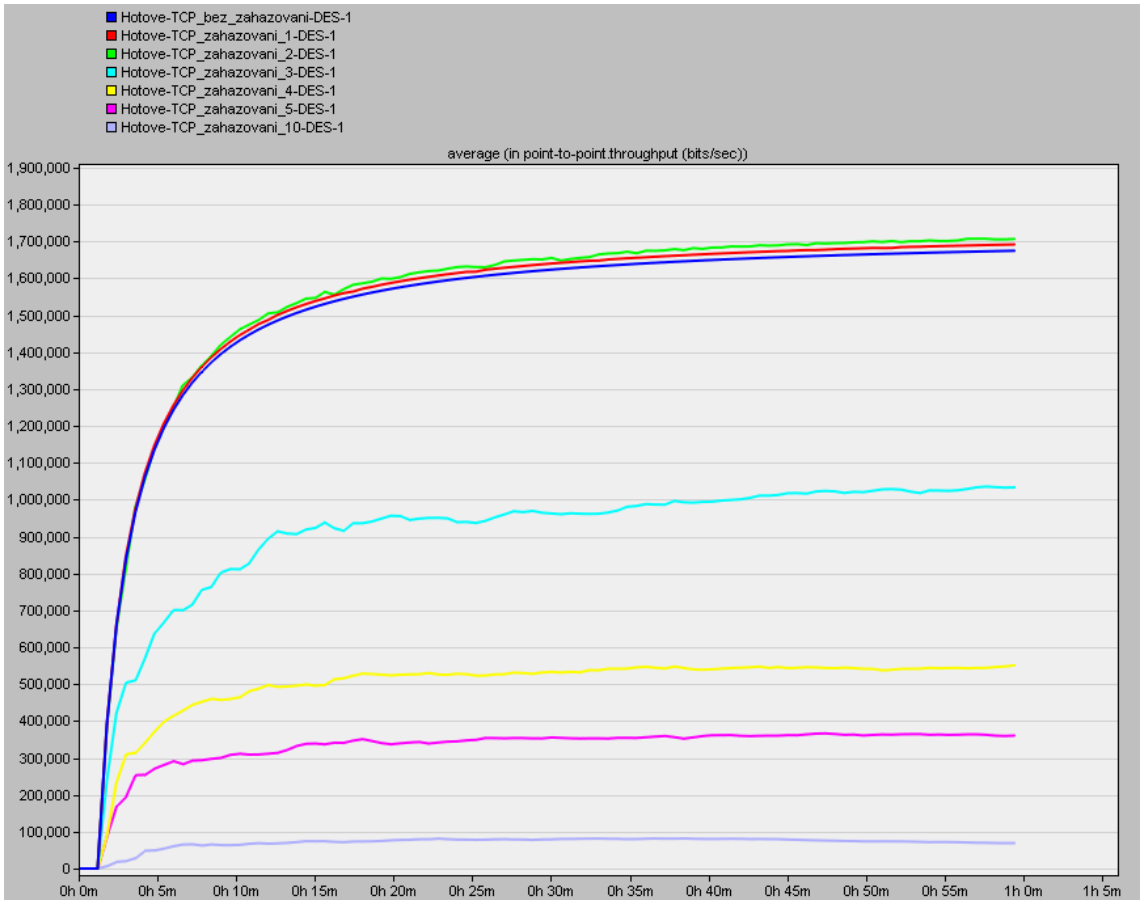
Obr. A.1: Základní srovnání TCP a UPD protokolu.

A.2 Úkol 2

V tomto úkolu jsou vytvořeny scénáře se zahazováním paketů na komponentě *IP_cloud* pro zahazování 1, 2, 3, 4, 5, 10% a scénář bez zahazování paketů, který má sloužit jako srovnávací.

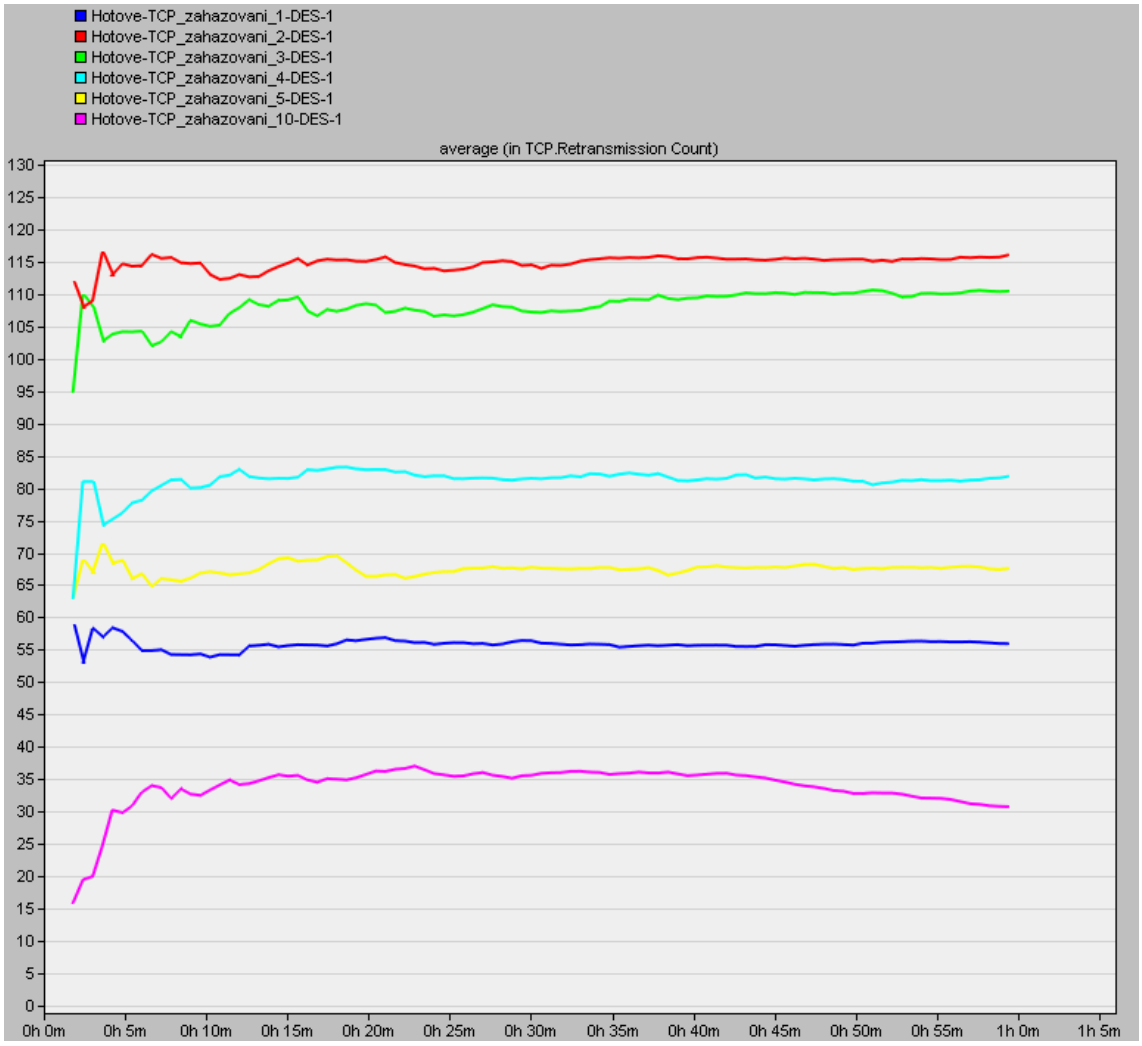
Doplňující otázky a úkoly:

1) Na **Obr. A.2** je vidět statistika pro zahazování paketů s TCP protokolem pro 1, 2, 3, 4, 5, 10% a protokol bez zahazování. Z grafu je patrné, že pro zahazování s 1 a 2 procenty, bude mít negativní vliv na zatížení, které vzroste, protože zde jsou znovu zasílány ztracené pakety. Naopak pro zahazování paketů s 10% je zřejmé, že zahazování je natolik vysoké, že má nejnižší zatížení na linku.



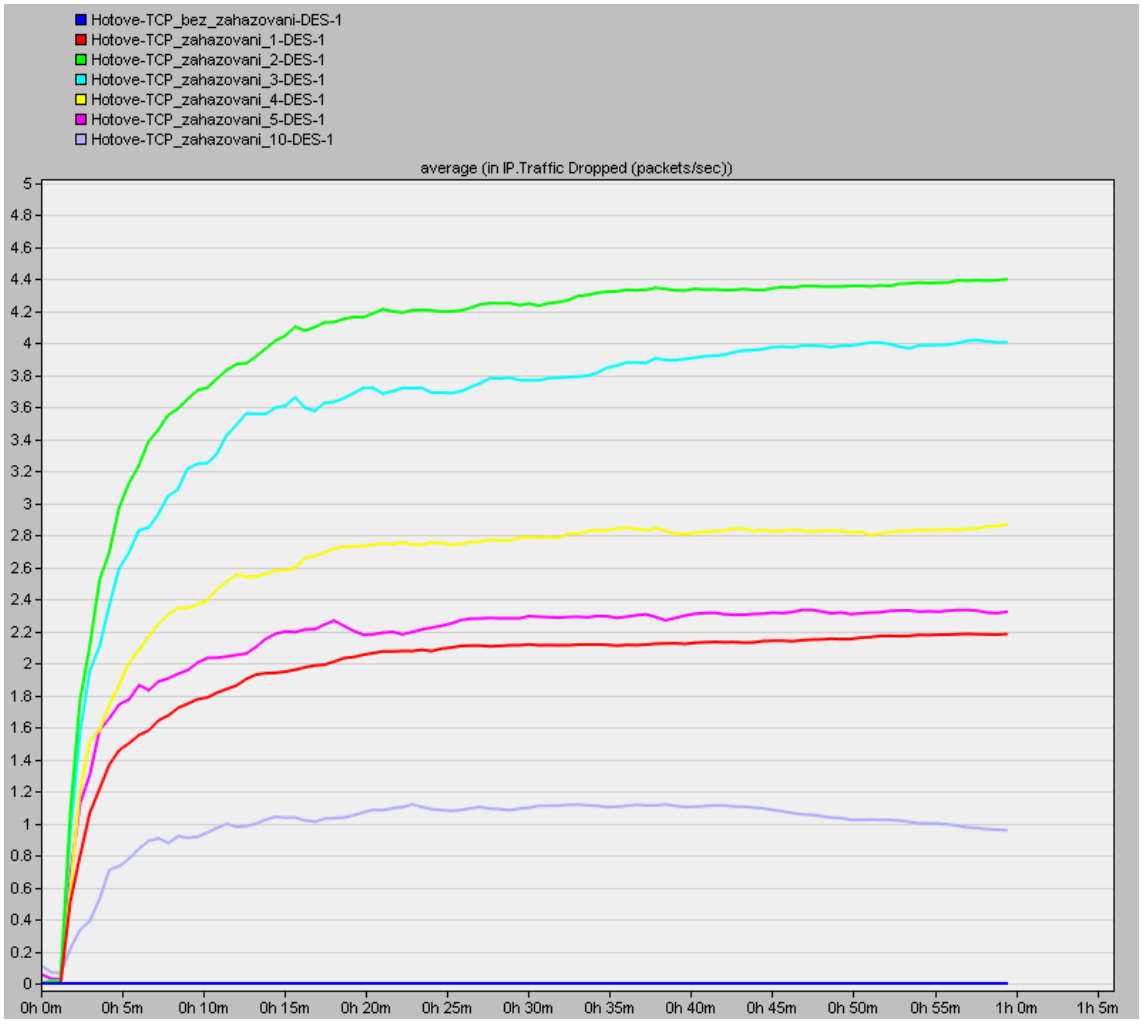
Obr. A.2: Propustnost pro scénáře TCP s různým procentem zahazování paketů v bitech/s.

2) Na **Obr. A.3** je počet opakovaných přenosů TCP pro server neboli *Retransmission Count*.



Obr. A.3: Počet opakovaných přenosů.

3) Statistika udává počet zahozených paketů *(Traffic Dropped)* na prvku *IP_cloud* za sekundu. V **Obr. A.4** je vidět, že s postupným procentuálním navyšováním ztrátovosti na linkách klesá i počet zahozených paketů a to díky tomu, že se zahazováním snižuje i celkový provoz na lince. V těchto dvou statistikách je vidět, že existuje závislost na množství přenášených dat s procentuálním zahazováním pro TCP protokol.



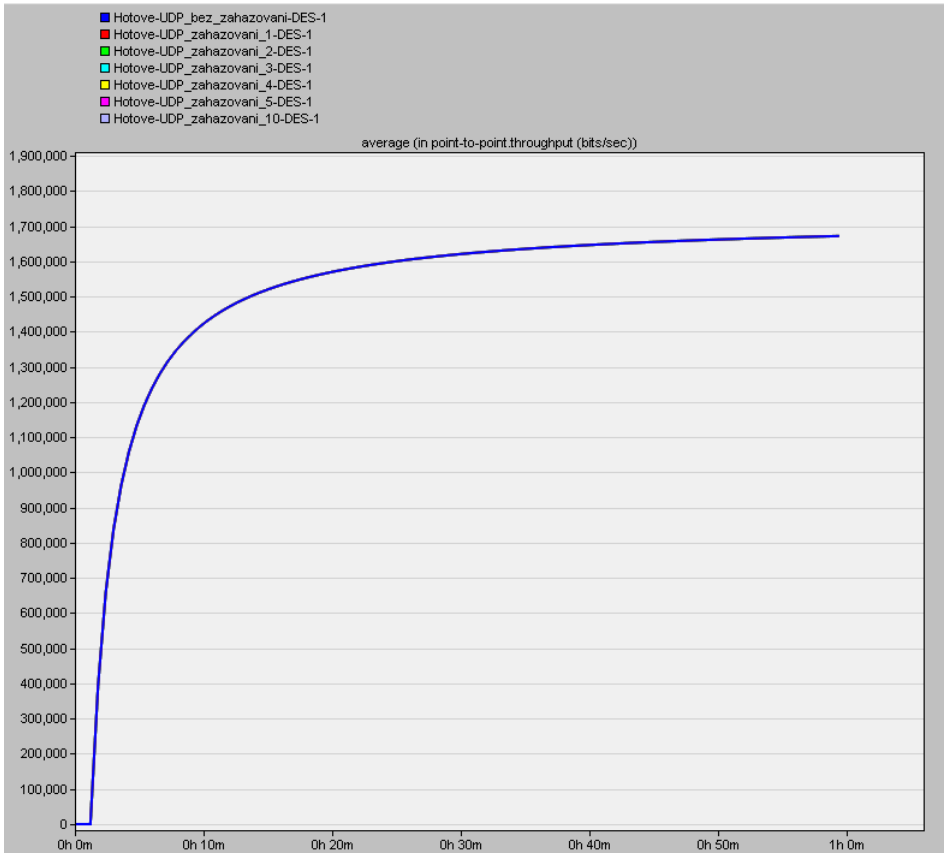
Obr. A.4: Počet zahozených paketů na IP_cloud v paketech/s.

A.3 Úkol 3

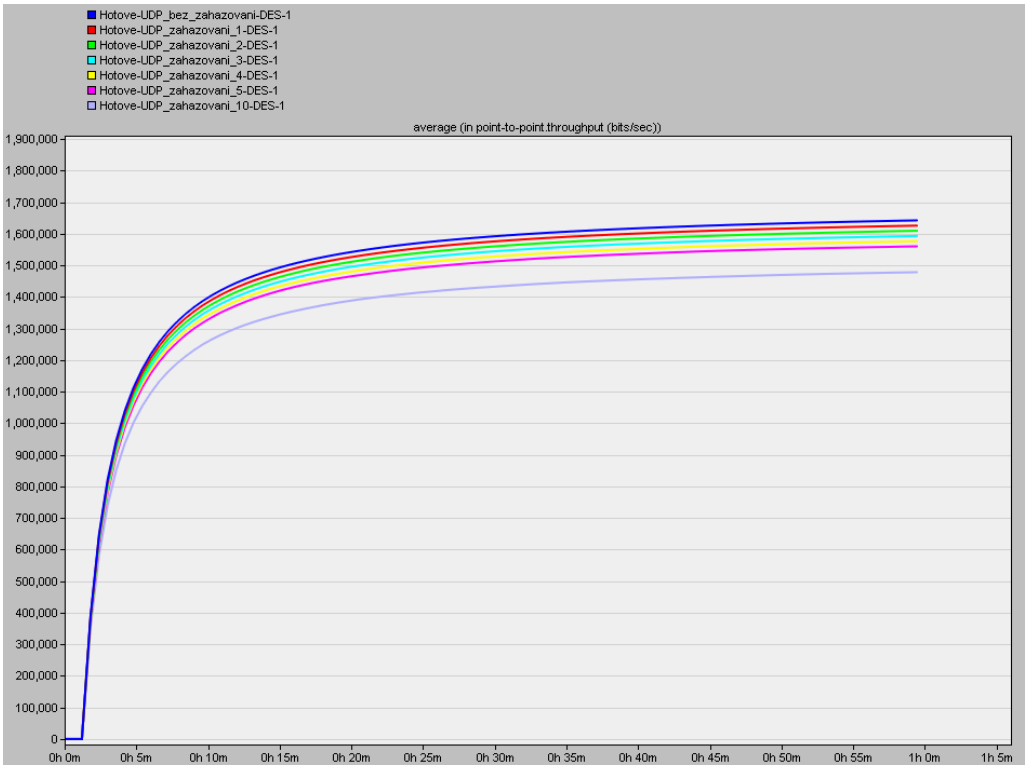
V úkolu tři je zkoumáno, jaké rozdíly jsou v zahazování paketů pro scénáře s protokolem UDP na komponentě *IP_cloud* pro zahazování 1, 2, 3, 4, 5, 10%.

Doplňující otázky a úkoly:

1) Na **Obr. A.5** jsou statistiky s různým procentem zahazování pro UDP protokol. Protože se jedná o propustnost na lince před místem zahazovaní paketů, budou přenosy stejné. Na **Obr. A.6** je propustnost pro linku, která je za místem zahazování paketů a můžeme zde pozorovat postupný pokles.

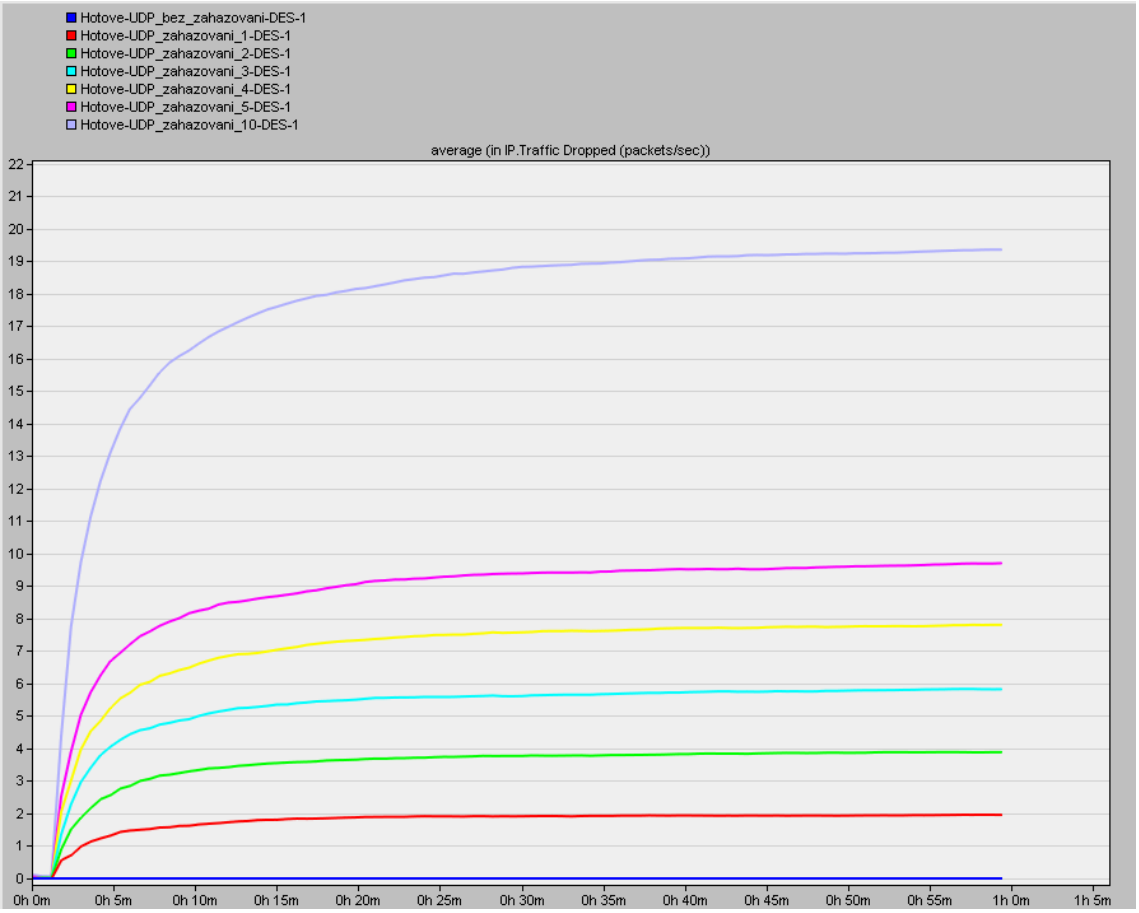


Obr. A.5: Propustnost pro scénáře UDP před místem zahazování paketů v bitech/s.



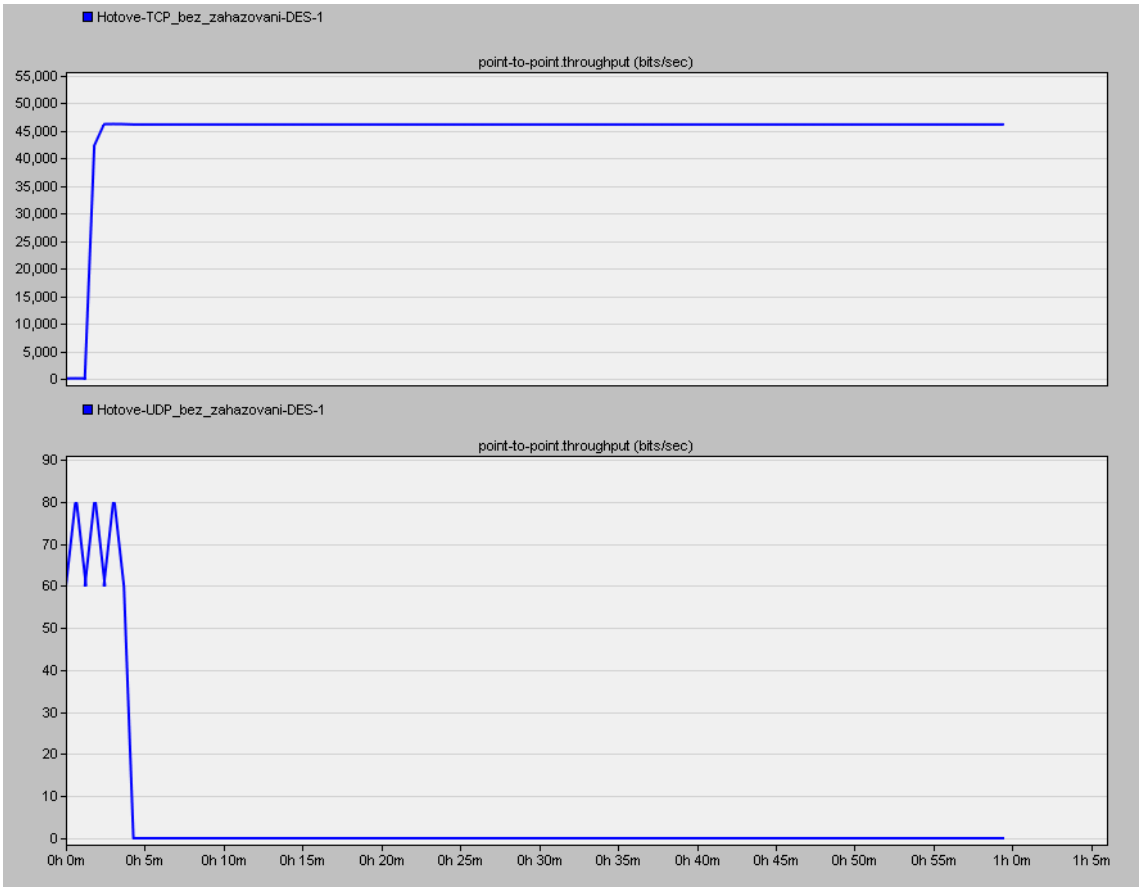
Obr. A.6: Propustnost pro scénáře UDP za místem zahazování paketů v bitech/s.

2) Počet zahozených paketů na **Obr. A.7** je u UDP oproti TCP přímo úměrný ztrátovosti paketů. Tedy čím více se ve scénáři na komponentě *IP_cloud* zvětší zahazování paketů, tím více se zahodí paketů. Průměrně se například pro scénář se zahazováním 10% zahodí 19 paketů/s.



Obr. A.7: Počet zahozených paketů UDP v paketech/s.

3) Na **Obr. A.8** je možné vidět, že TCP na rozdíl od UDP zpětně potvrzuje přijatá data, UDP je naopak protokol nepotvrzovaný.



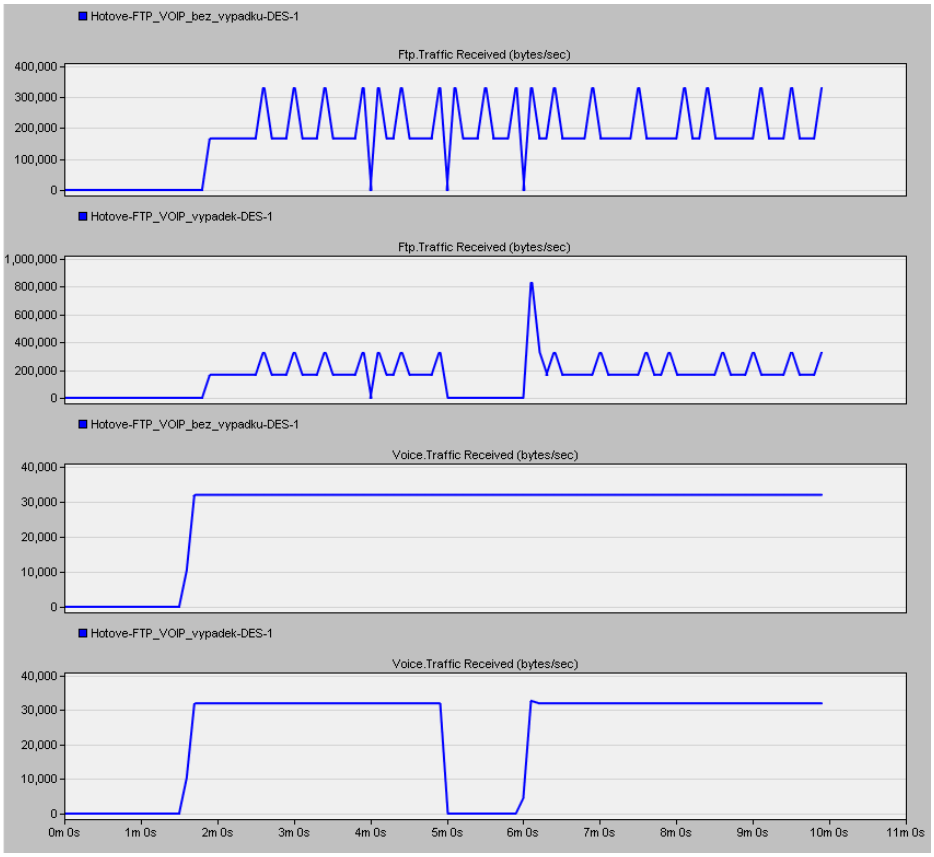
Obr. A.8: Srovnání TCP a UDP.

A.4 Úkol 5

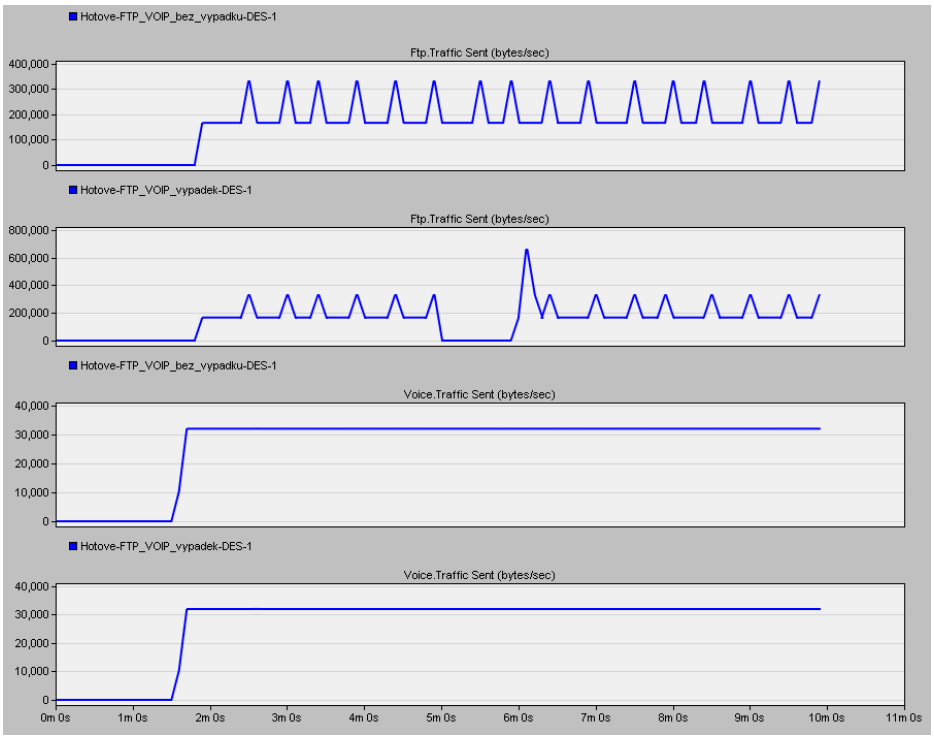
V tomto úkolu se zkoumají aplikace FTP a Voice při výpadku na lince pomocí komponenty *Failure Recovery*.

Doplňující otázky a úkoly:

1) Z **Obr. A.9** je možné pozorovat, že při výpadku se FTP aplikace bude snažit ztracená data znovu poslat, protože je zde používán protokol TCP. Na **Obr. A.10** je naproti tomu vidět, že hlasová aplikace, která používá protokol UDP, jež je zároveň nepotvrzovaný, se pro odeslaný provoz *(Traffic Sent)* nemá jak dozvědět, že došlo na lince k výpadku, a tak bude neustále data vysílat.



Obr. A.9: Traffic Received (bytes/sec) pro FTP a Voice.



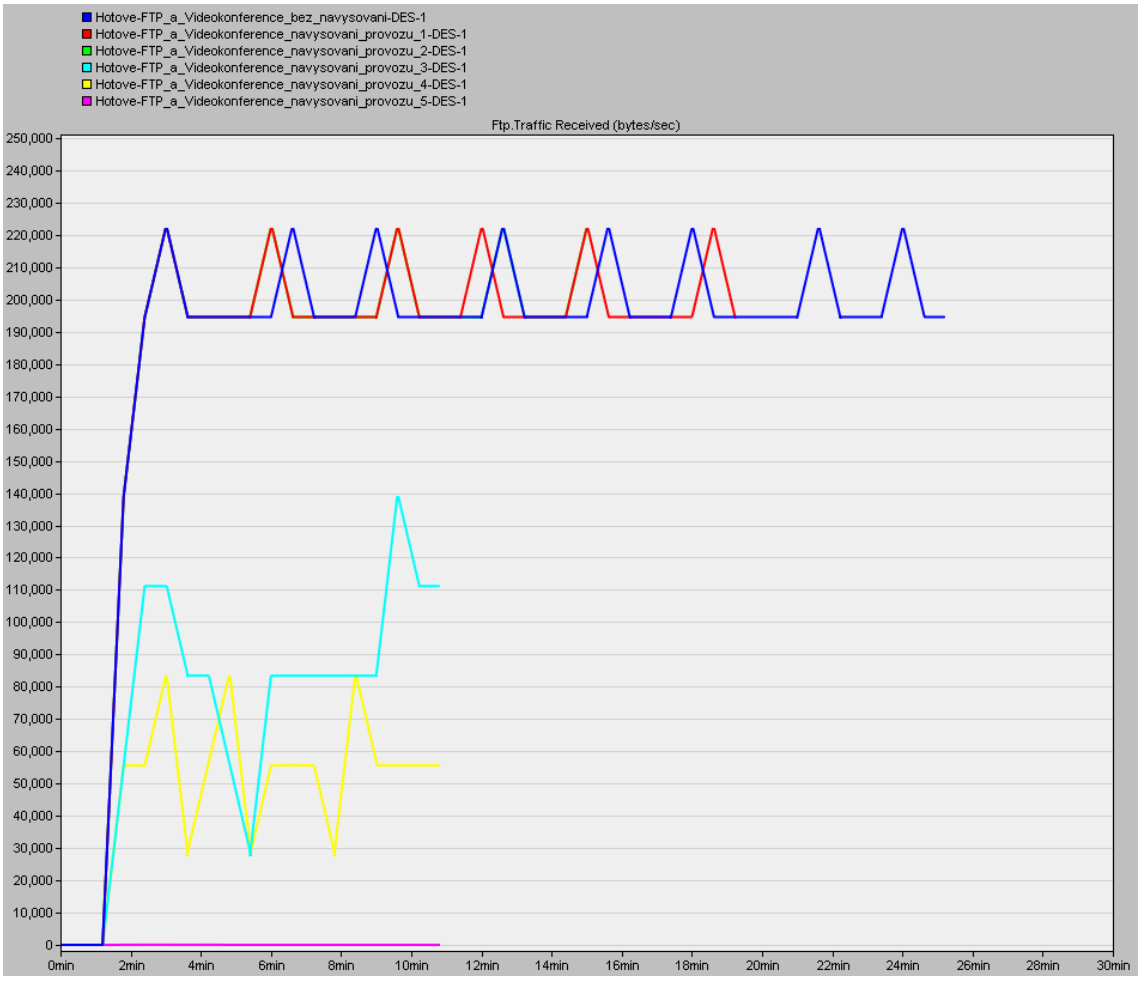
Obr. A.10: Traffic Sent (bytes/sec) pro FTP a Voice.

A.5 Úkol 6

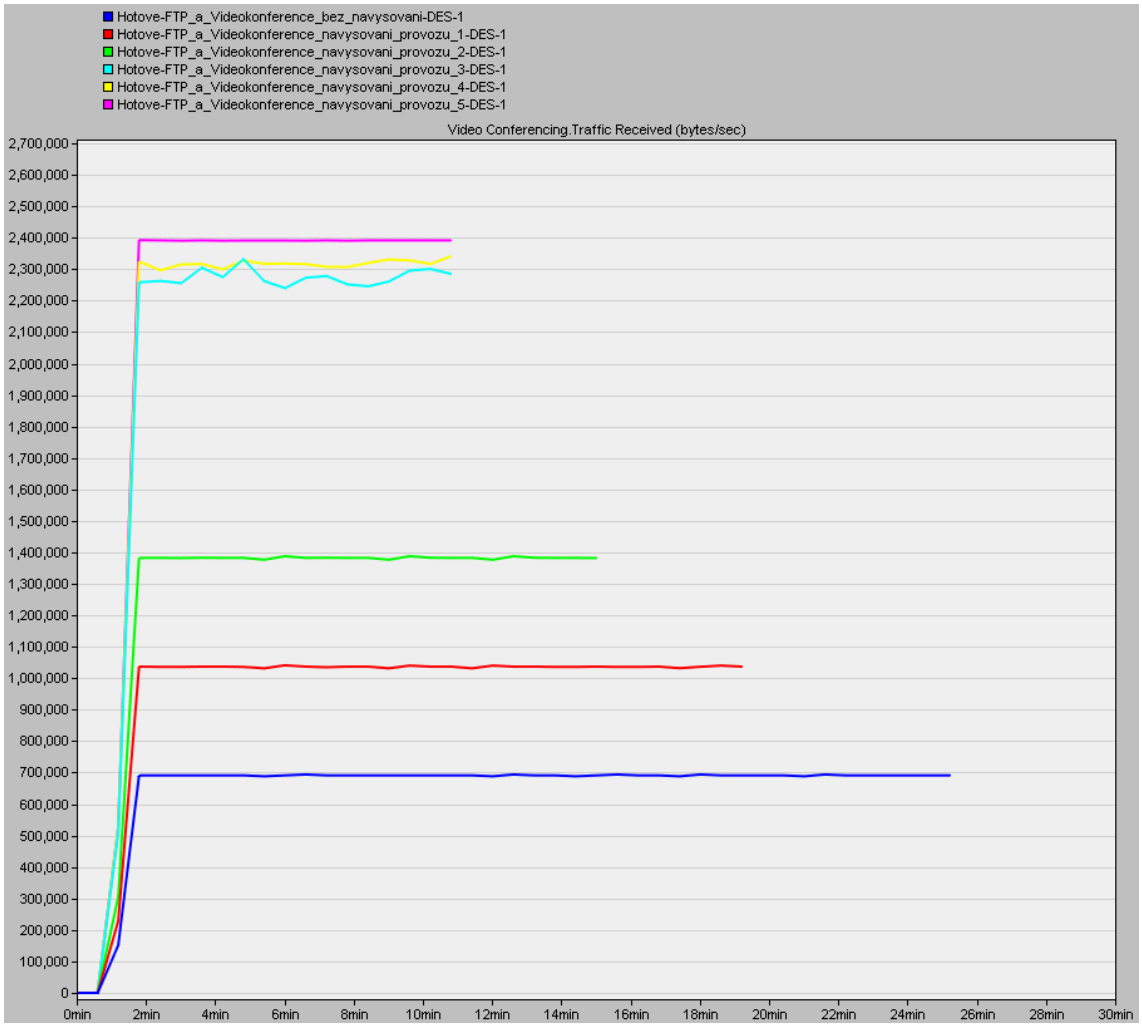
Tento úkol je zaměřen na postupné navyšovaní provozu a tím i zátěže na lince pro Videokonferenci, která má společnou přenosovou linku s FTP aplikací.

Doplňující otázky a úkoly:

1) Z obrázků s přijatým provozem (**Obr. A.11** a **Obr. A.12**) je patrné, že zatímco *Traffic Received (bytes/sec)* pro Videokonferenci narůstá, FTP s narůstajícím provozem klesá.

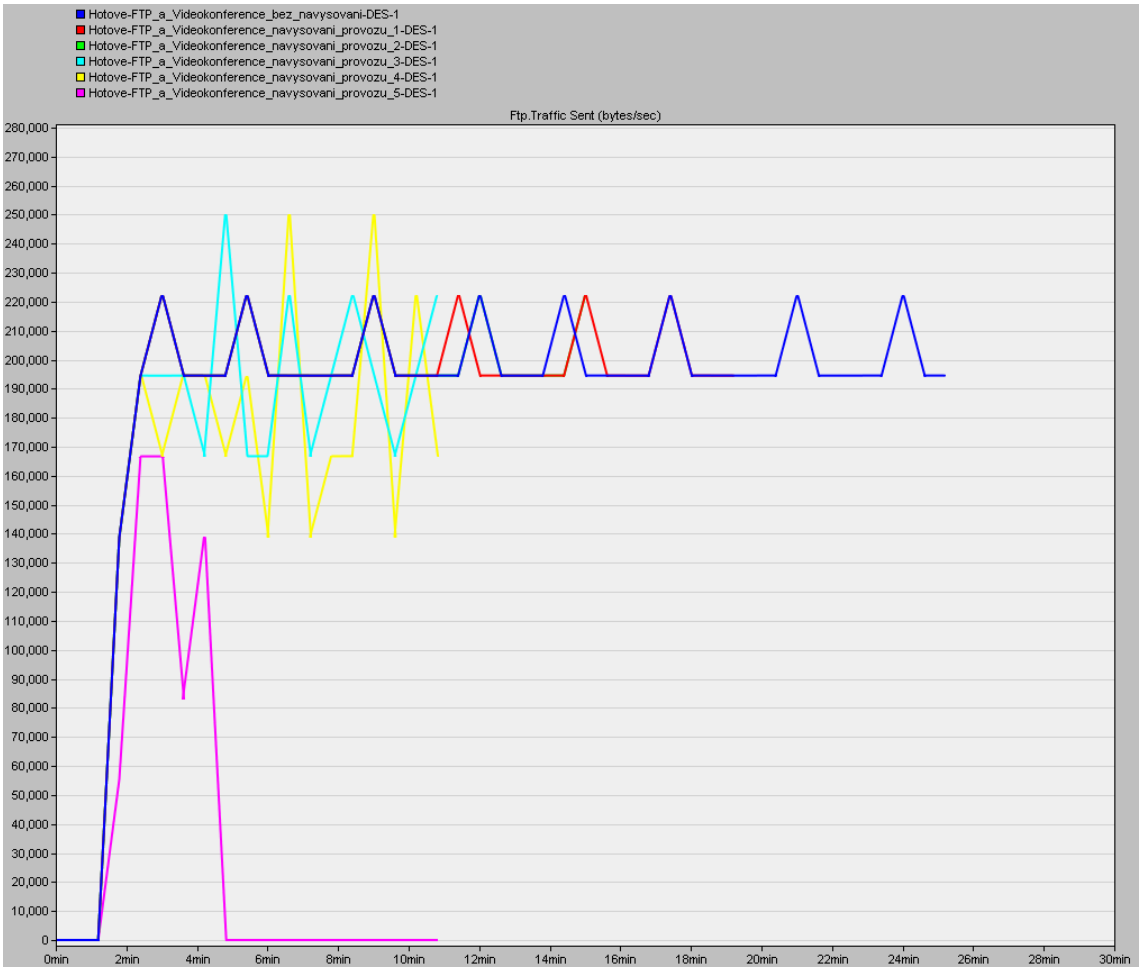


Obr. A.11: Traffic Received (bytes/sec) pro FTP.

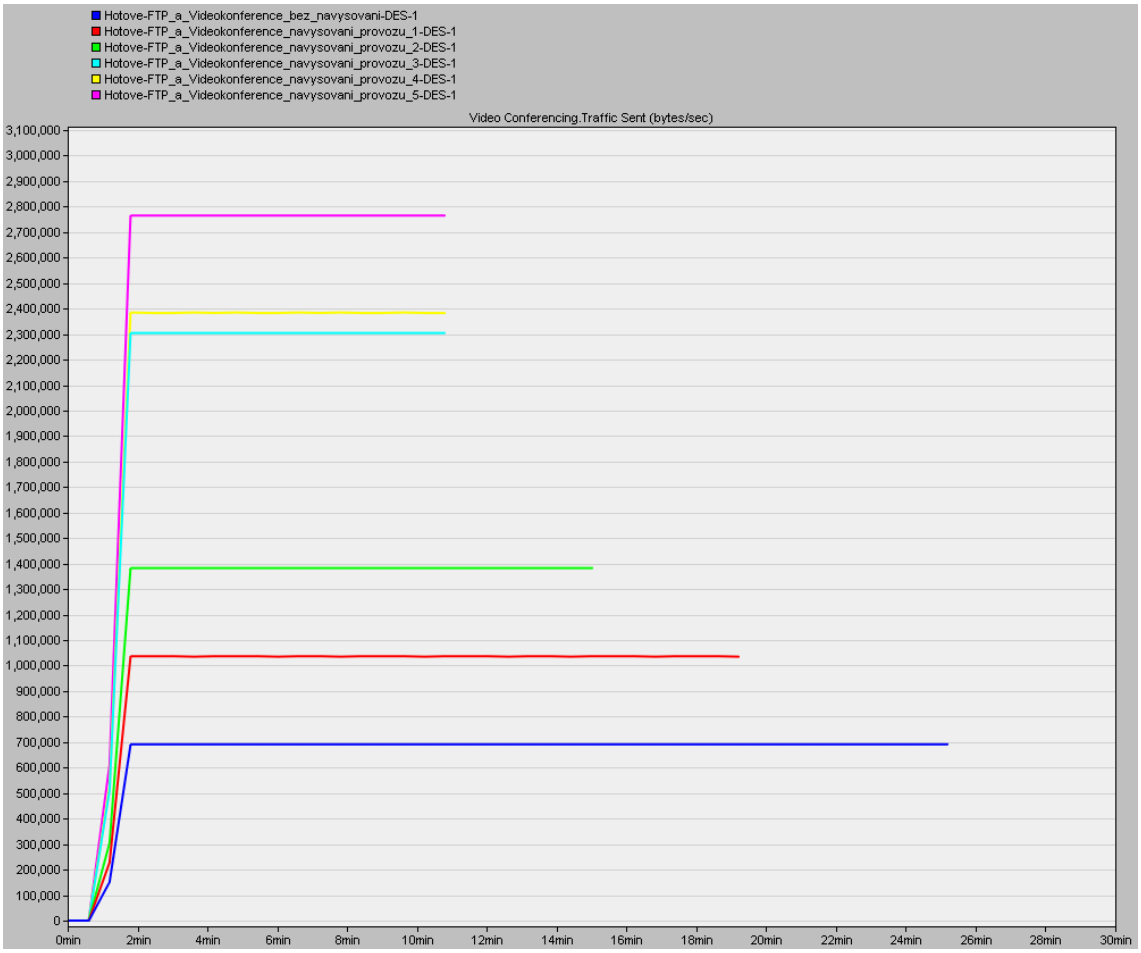


Obr. A.12: Traffic Received (bytes/sec) pro Videokonferenci.

2) Při pohledu na odeslaný provoz (**Obr. A.13** a **Obr. A.14**) pro Videokonferenci je vidět, že se vysílač snaží vyslat mnohem více dat než je přijímač a daná linka schopná zvládnout, naopak *Traffic Sent* pro FTP s ohledem na vytíženost linky a navyšující se provoz, posílá dat čím dál méně.



Obr. A.13: Traffic Sent (bytes/sec) pro FTP.



Obr. A.14: Traffic Sent (bytes/sec) pro Videokonferenci.

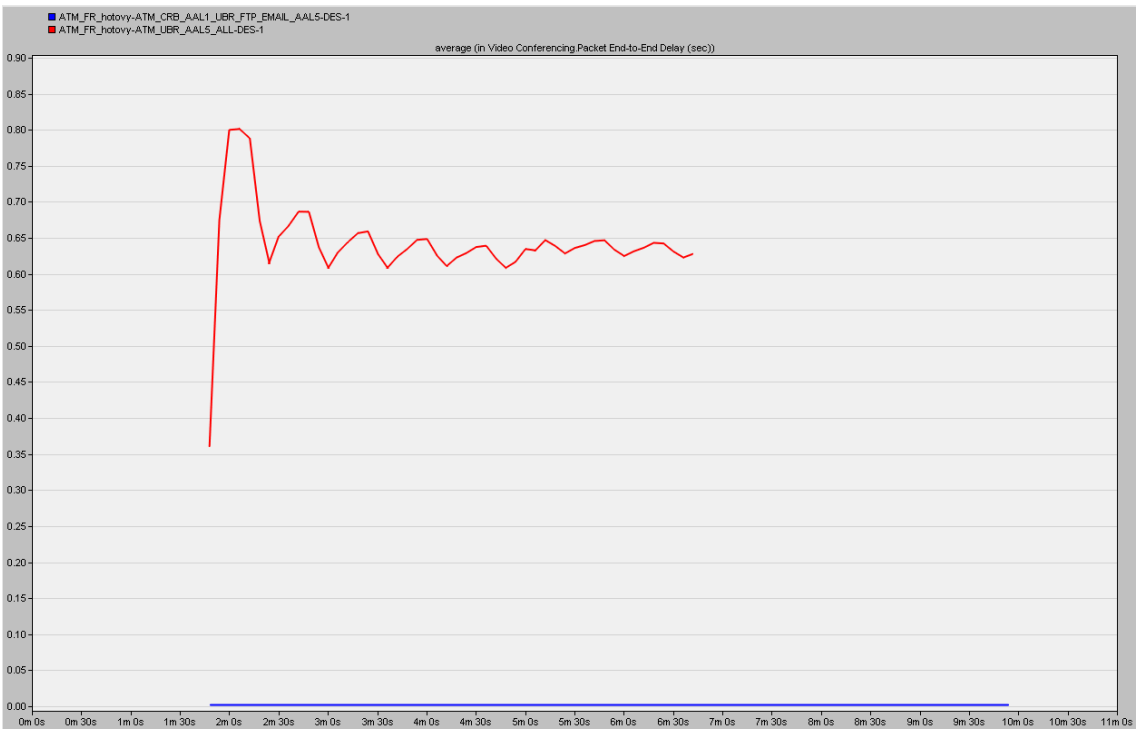
B ŘEŠENÍ OTÁZEK A ÚKOLŮ PRO SROVNÁNÍ TECHNOLOGIÍ ATM A FRAME RELAY

Laboratorní úloha je zaměřena na srovnání dvou technologií pro rozsáhlé sítě typu WAN, kterými jsou ATM a Frame Relay.

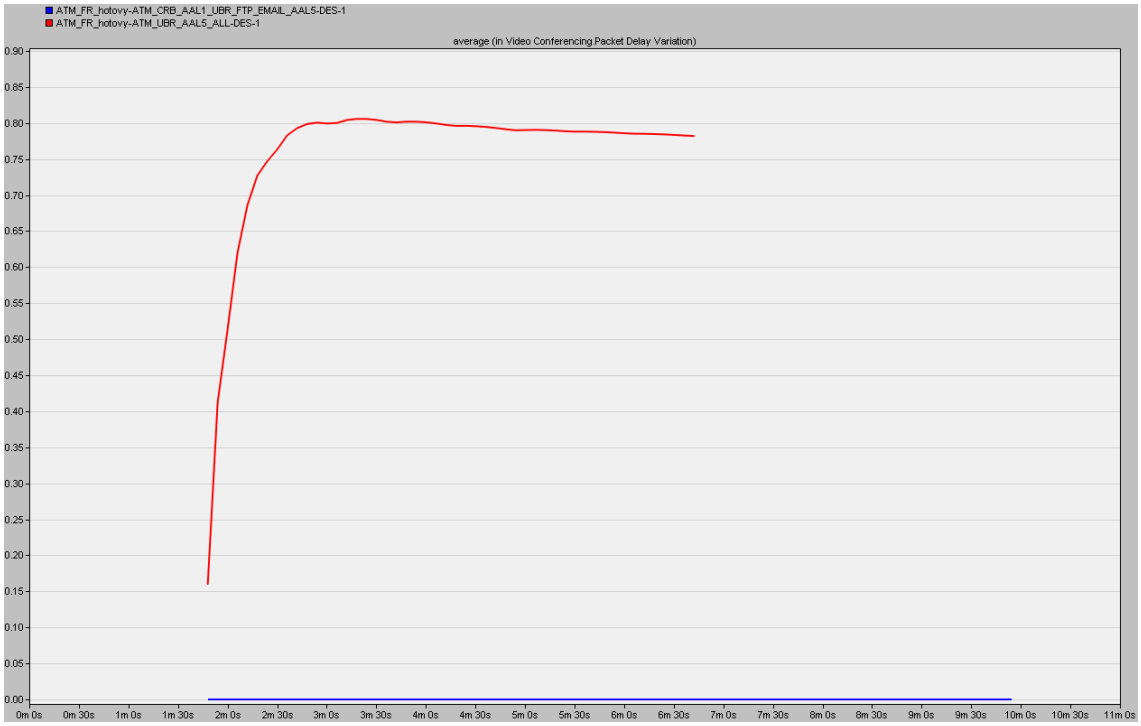
B.1 Úkol 1

První úloha je zaměřena na srovnání dvou tříd služeb s video konferencí v ATM. Studenti budou porovnávat třídy CBR a UBR v předpřipraveném projektu. Pro první scénář je na konferenci nastavena třída služeb CBR s adaptační vrstvou na AAL1, která je pro CBR přímo určena. Ve druhém scénáři je na konferenci nastaveno UBR s AAL5. Následně budou studenti duplikovat tyto dva scénáře, kde bude jejich úkolem postupně navyšovat provoz na konferenci.

Z **Obr. B.1** a **Obr. B.2** je pro výchozí scénáře s konferencí patrné, že třída CBR s adaptační vrstvou AAL1 má oproti UBR s AAL5 skoro nulové zpoždění i kolísání zpoždění. V tomto případě je tedy ověřena teorie, že pro hlasové služby, které vyžadují malé zpoždění a kolísání zpoždění s pevným časováním, je mnohem lepší třída CBR s adaptační vrstvou AAL1, která podporuje CBR se synchronním a spojově orientovaným přenosem dat.



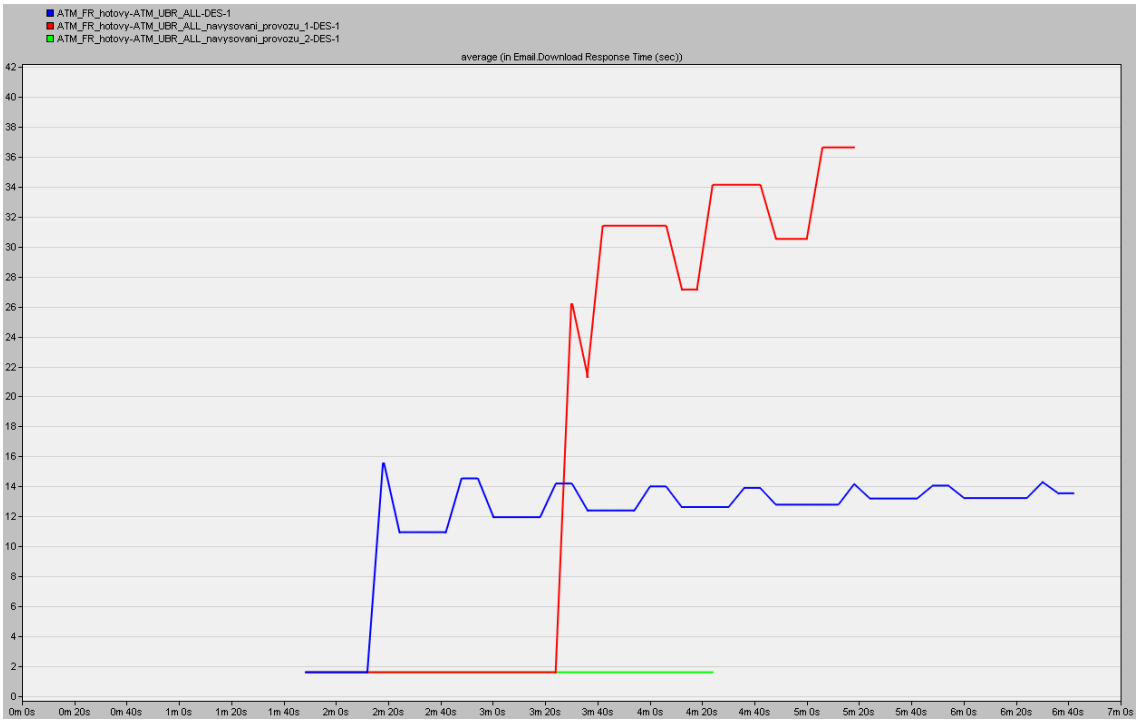
Obr. B.1: Zpoždění pro konferenci v předvytvořených scénářích.



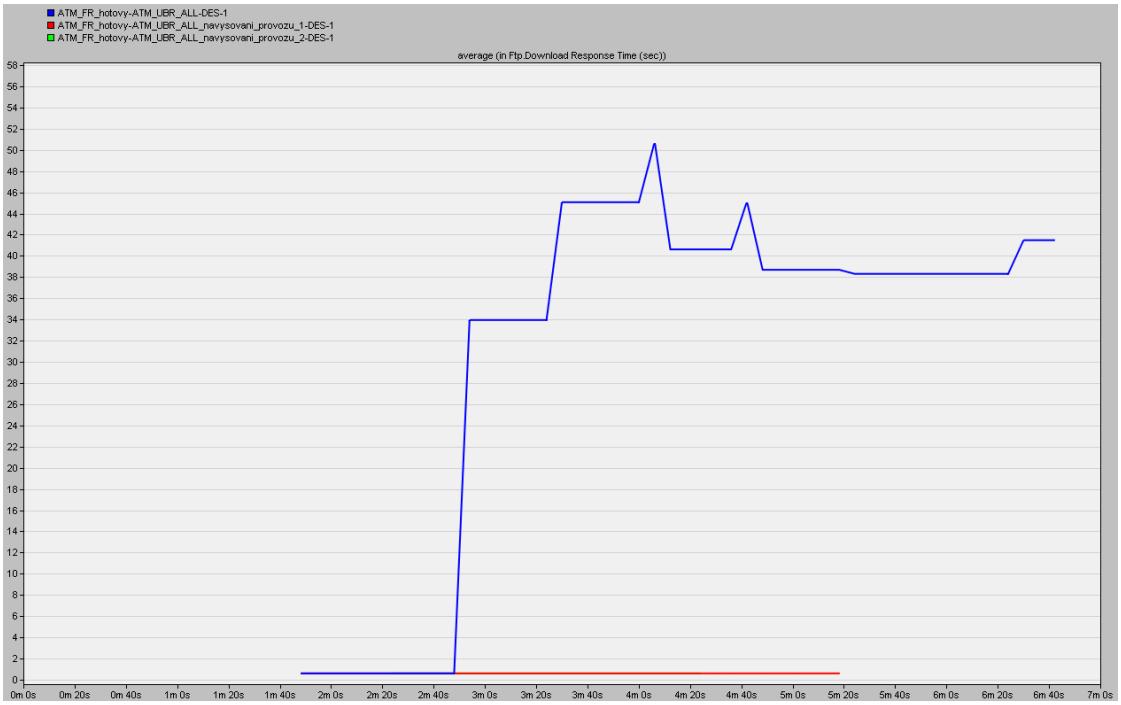
Obr. B.2: Kolísání zpoždění pro konferenci v předvytvořených scénářích.

Doplňující otázky a úkoly:

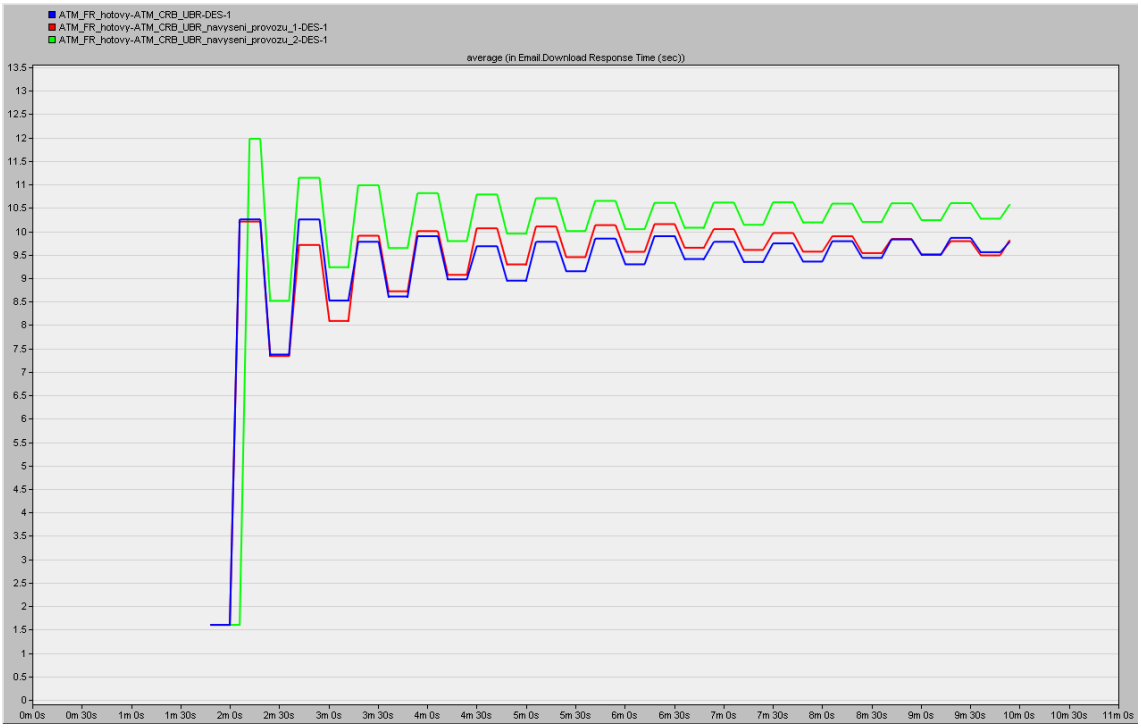
1) Na následujících obrázcích - **Obr. B.3** a **Obr. B.4** je možné pozorovat odezvu pro aplikace FTP a Email s nastaveným UBR pro všechny aplikace. Na **Obr. B.5** a **Obr. B.6** je naopak pro konferenci nastaveno CBR a pro aplikace FTP a Email je nastaveno UBR, což se projeví postupným zvyšováním odezvy.



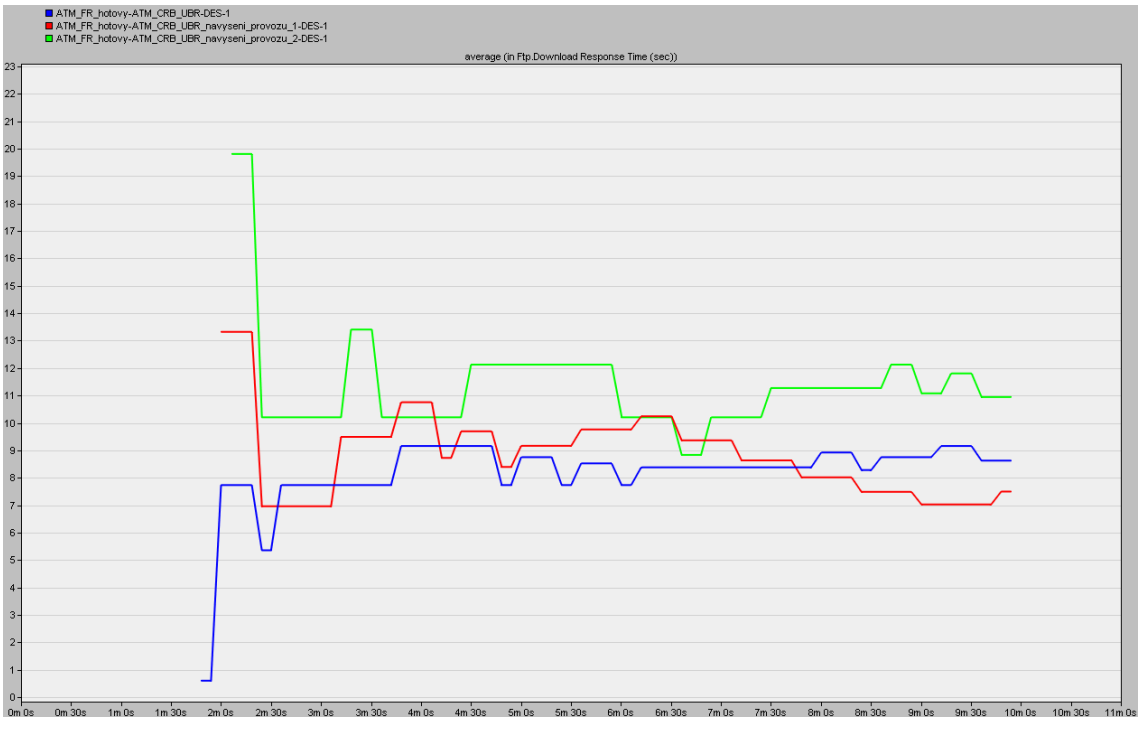
Obr. B.3: Download Response Time (sec) pro email.



Obr. B.4: Download Response Time (sec) pro FTP.

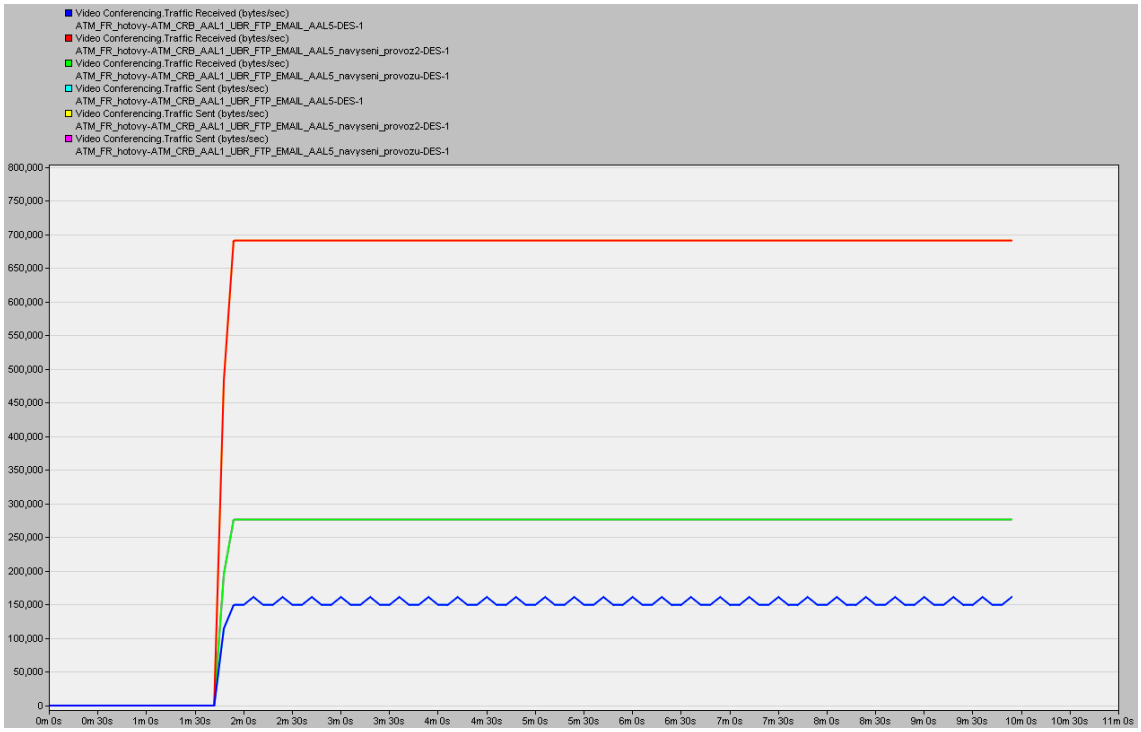


Obr. B.5: Download Response Time (sec) pro email.



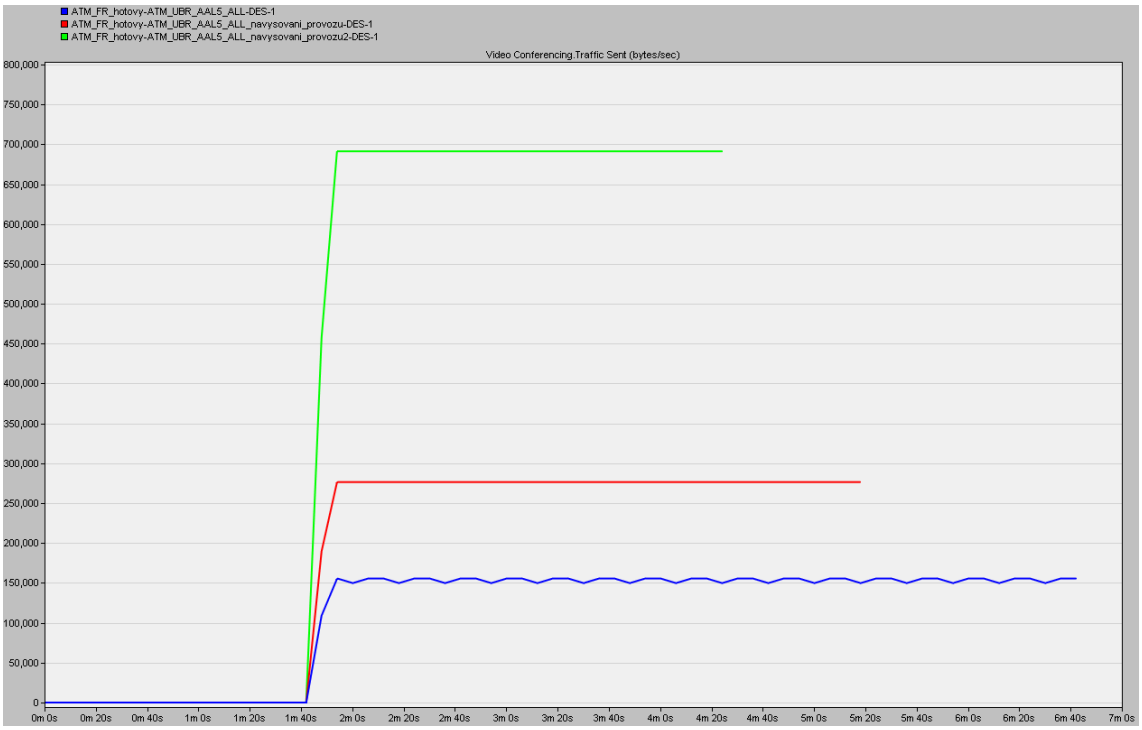
Obr. B.6: Download Response Time (sec) pro FTP.

2) Z **Obr. B.7** je vidět, že pro CBR je odeslaný provoz stejný jako přijatý a s postupně zvyšující se zátěží na konferenci se provoz konstantně zvyšuje.

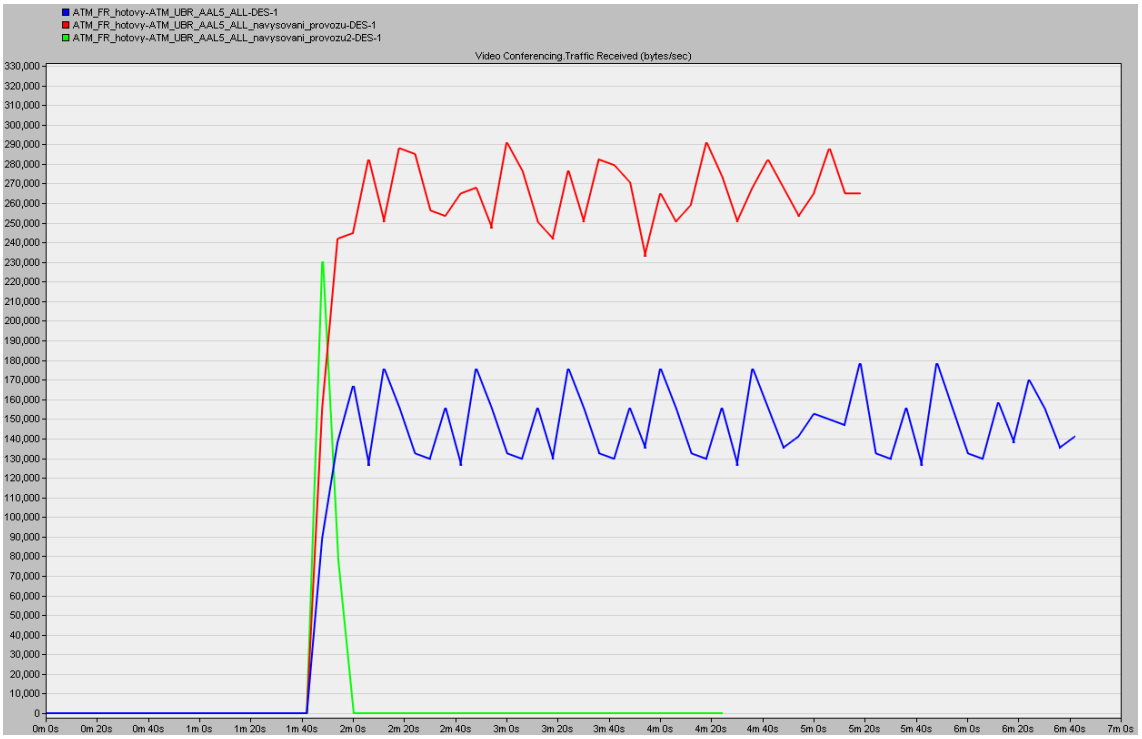


Obr. B.7 Odeslaný a přijatý provoz pro CBR v bytes/s.

3) V těchto scénářích jsou všechny aplikace (FTP, EMAIL a konference) nastaveny s třídou služeb UBR. V UBR se nesestavuje spojení a nezaručuje se žádná kvalita služeb, označuje se také jako Best effort. Protože nemá Video konference zaručenou kvalitu služeb jako je tomu v CBR, je zde mnohem větší zátěž na lince, což způsobuje i menší propustnost dat. To je možné vidět na **Obr. B.8** a **Obr. B.9**. Můžeme si všimnout, že odeslaný provoz je stejný jako v CBR, ale přijatý provoz se s navyšující zátěží na video konferenci postupně snižuje, protože začínají být zahlceny síťové prostředky, což nakonec způsobí, že je linka natolik vytížena, že nezvládá přijímat odeslaná data.



Obr. B.8: Odeslaný provoz pro Video konferenci s UBR v bytes/s.



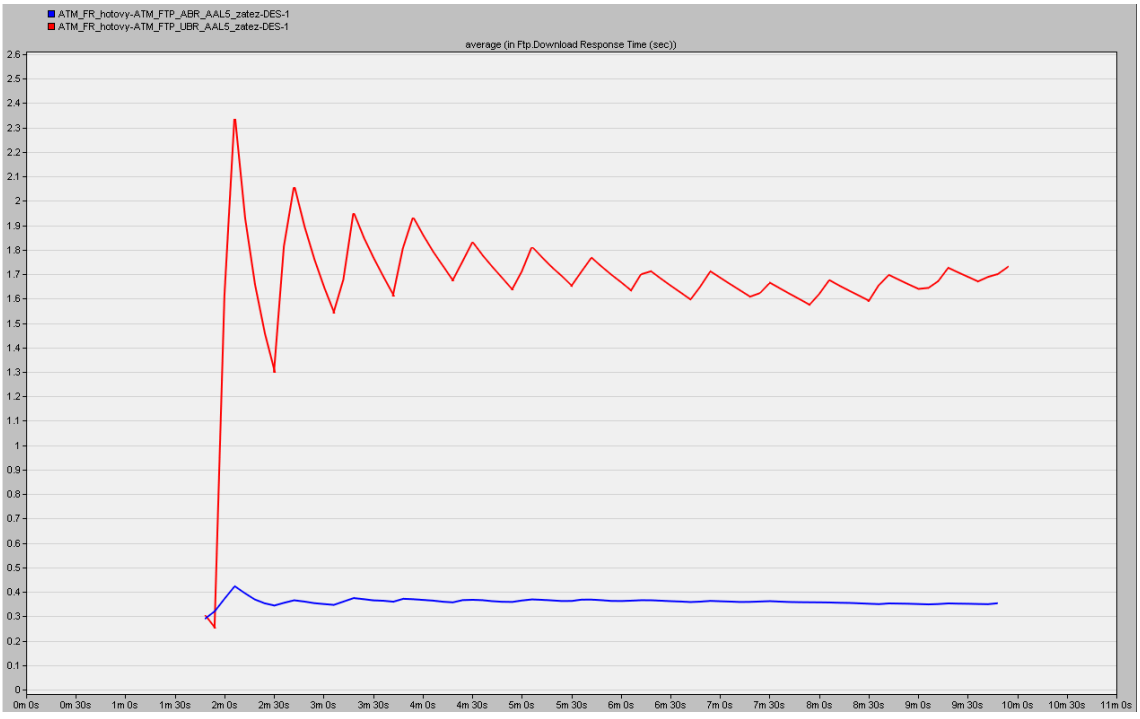
Obr. B.9: Přijatý provoz pro Video konferenci s UBR v bytes/s.

B.2 Úkol 2

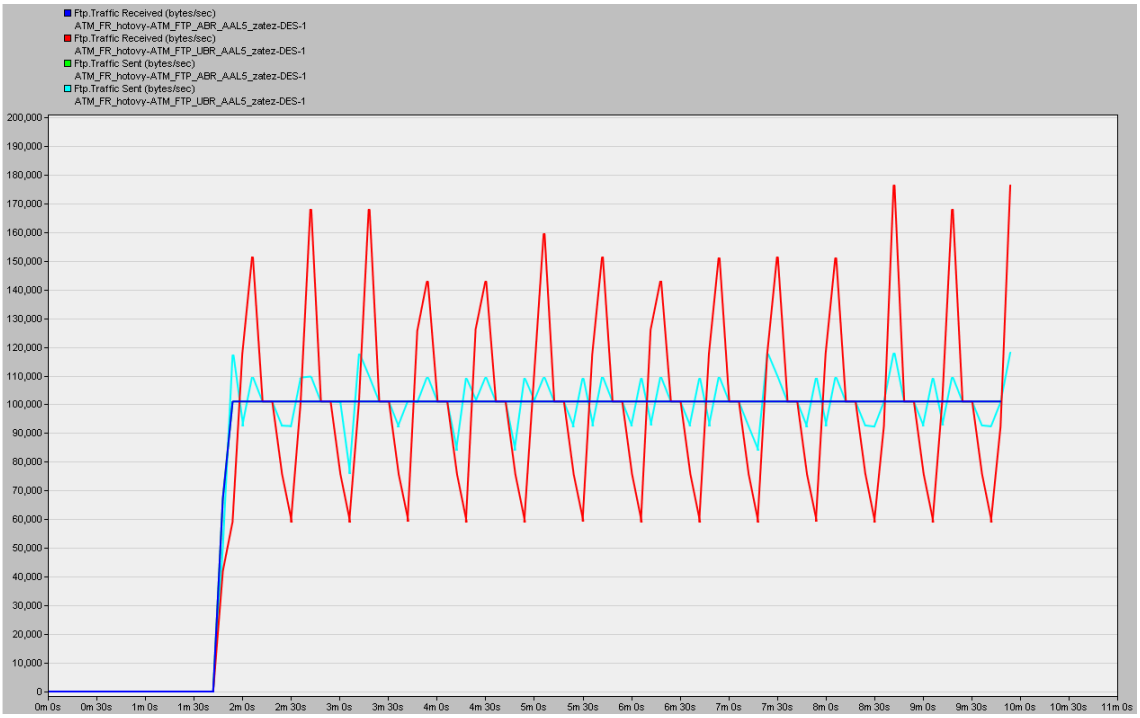
Druhá část úlohy je zaměřena na rozdíly tříd ABR a UBR na FTP aplikaci.

Doplňující otázky a úkoly:

1) Statistika *Global Statistics/FTP/Download Response Time(sec)* udává dobu odezvy při stahování FTP souborů. Z grafu **Obr. B.10** je vidět, že doba odezvy je zhruba dvojnásobná pro FTP s nastavenou třídou služeb UBR, která oproti ABR nemá žádnou definovanou záruku na kvalitu služeb. ABR je tedy pro FTP aplikaci mnohem výhodnější a rychlejší z hlediska doby odezvy. Při pohledu na odeslaný a přijatý provoz (**Obr. B.11**) můžeme pozorovat výrazné kolísání pro FTP s nastaveným UBR, což je způsobeno tím, že UBR může využít jen takovou rychlost, která je momentálně k dispozici a ve scénářích jsou spolu s FTP použity aplikace konference s CBR a email s nastaveným UBR.



Obr. B.10: Download Response Time(sec) pro FTP aplikaci.



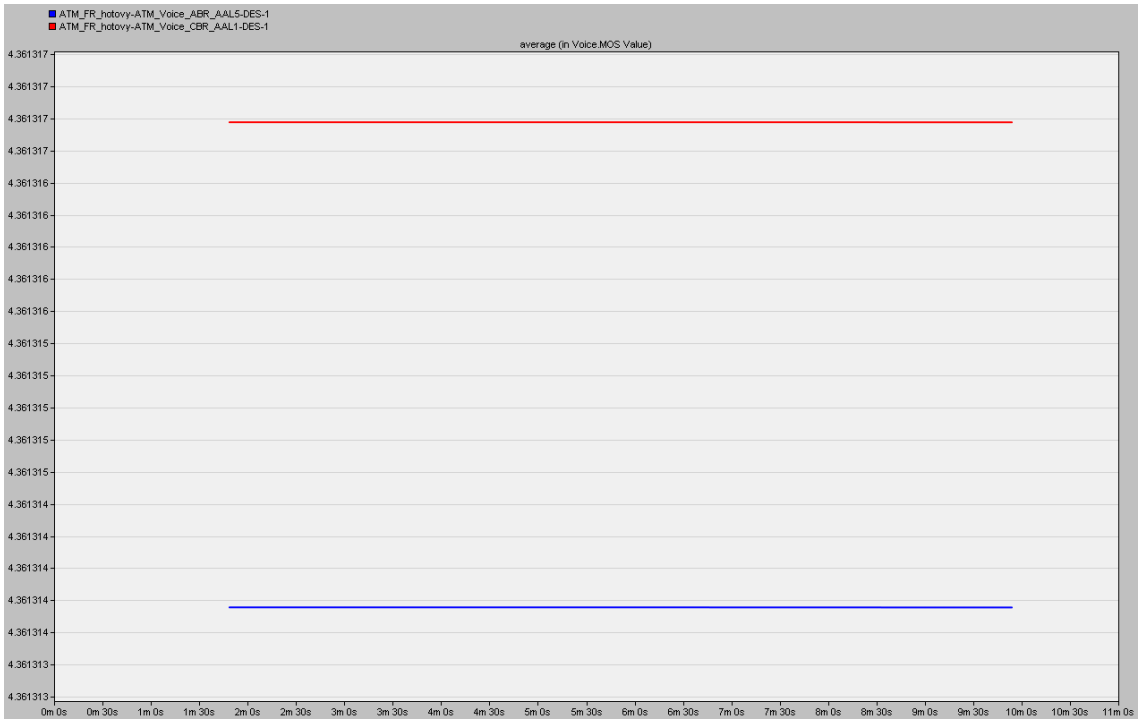
Obr. B.11: Odeslaný a přijatý provoz pro FTP aplikaci s ABR a UBR v bytes/s.

B.3 Úkol 3

Tento úkol je zaměřen na rozdíly mezi adaptačními vrstvami AAL pro hlasovou aplikaci. Jsou zde porovnány třídy CRB s nastavenou adaptační vrstvou AAL1 a ABR s AAL5.

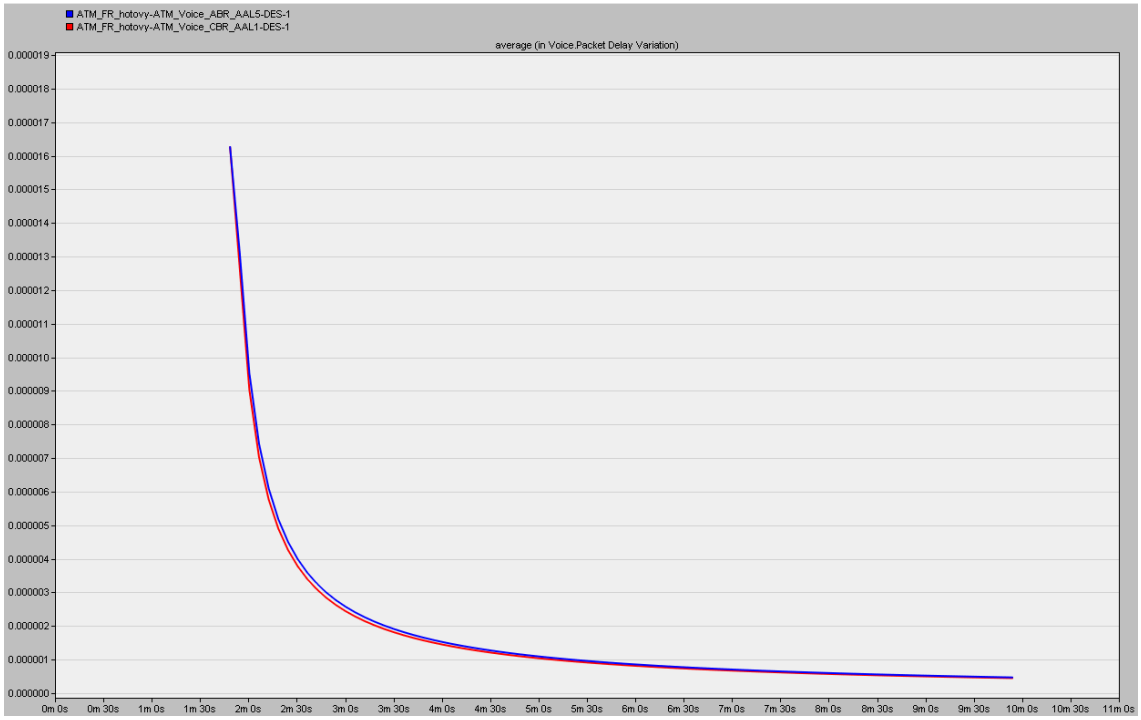
Doplňující otázky a úkoly:

1) MOS – *(Mean opinion score)* vyjadřuje přenosové rychlosti kodeků a jejich kvalitu, která je ohodnocena parametrem MOS. MOS může nabývat maximálně hodnoty 5. Na **Obr. B.12** je použit pro hlasovou aplikaci kodek G.711. Zde opět můžeme říct, že CBR s AAL1, který se primárně pro hlasové služby v ATM využívá, má lepší hodnotu MOS než ABR s AAL1.

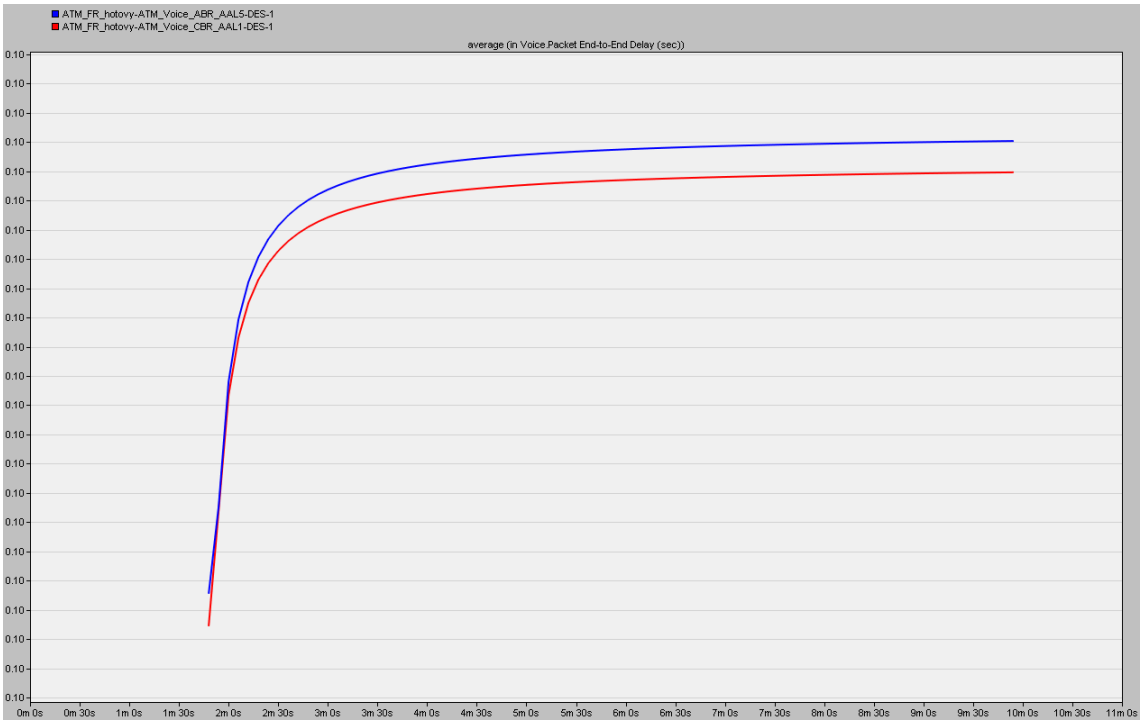


Obr. B.12: Hodnota MOS pro AAL1 a AAL5.

2) Na **Obr. B.13** a **Obr. B. 14** je srovnání hlasové aplikace pro zpoždění a kolísání zpoždění, kde i pro minimální zátěže v hlasové službě jsou vidět rozdíly v třídách služeb s různými adaptačními vrstvami, kde CBR s AAL1 vykazuje viditelně lepší výsledky než ABR s AAL5.



Obr. B.13: Kolísání zpoždění pro hlasovou aplikaci v s/min.



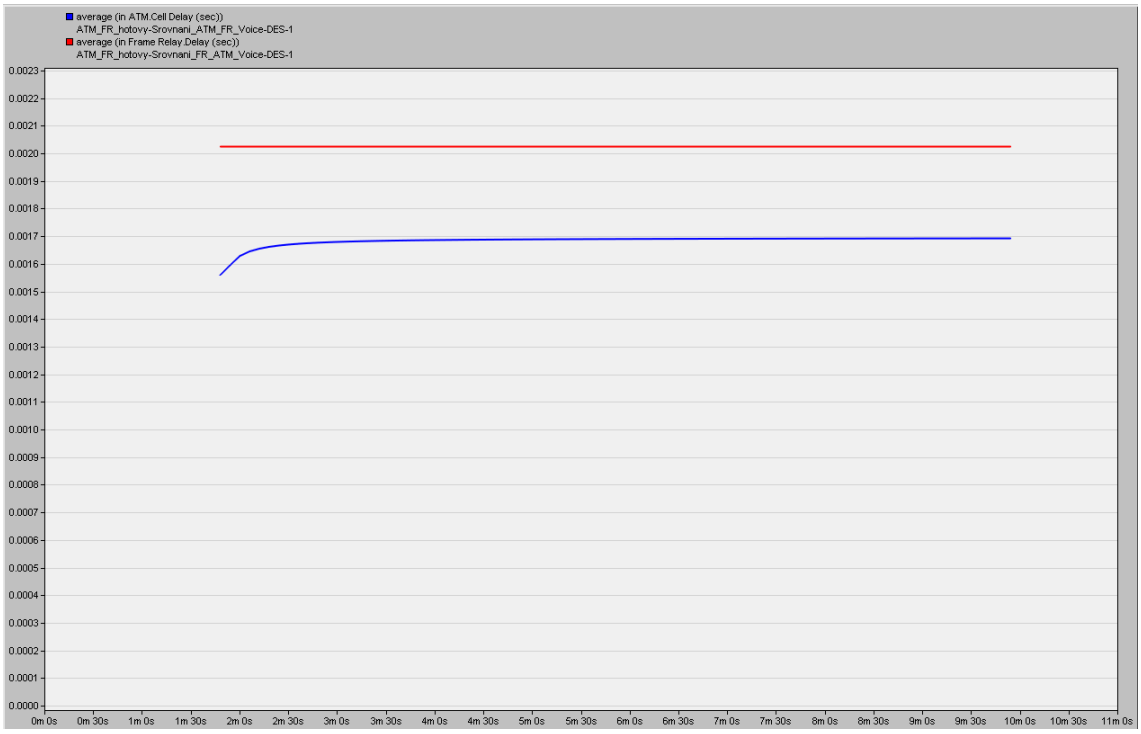
Obr. B. 14: Zpoždění pro hlasovou aplikaci v s/min.

B.4 Úkol 4

Čtvrtý úkol se zabývá srovnáním technologií Frame Relay a ATM s nakonfigurovanou hlasovou aplikaci, přenosovými linkami E1 a jejich následnou změnou pro pomalejší DS0 linku.

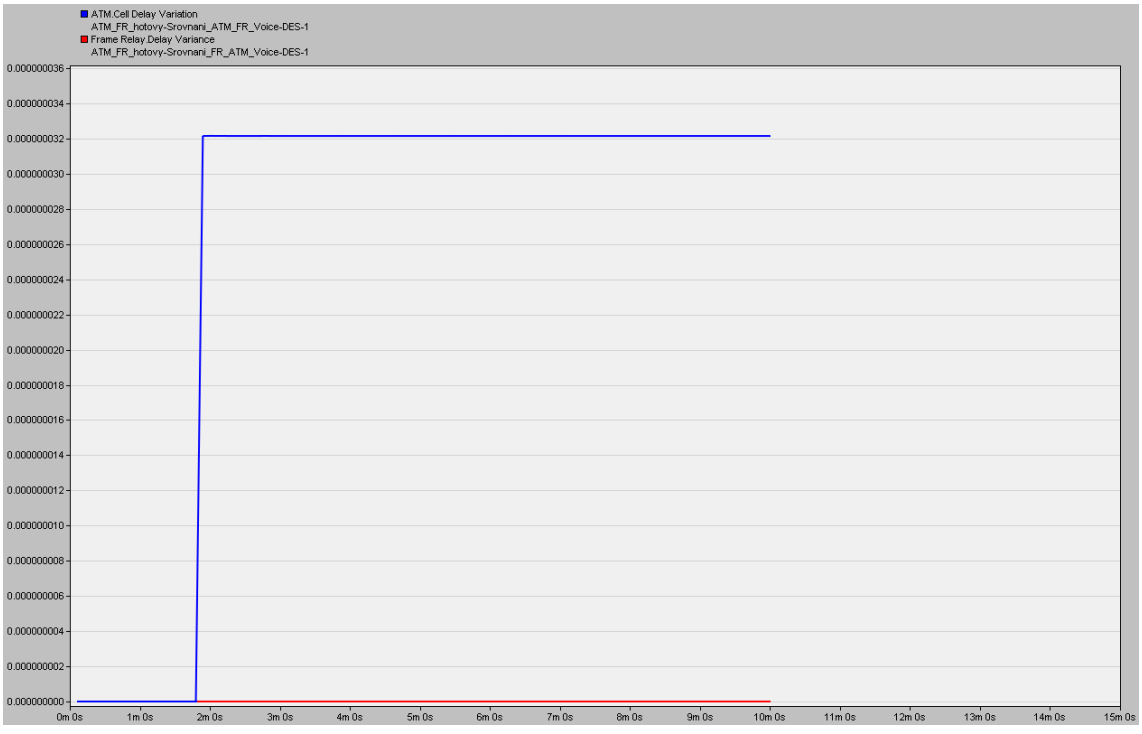
Doplňující otázky a úkoly:

1) *Cell Delay (sec)* - jedná se o zpoždění přijatých ATM buněk všemi vrstvami v ATM síti a je zde měřeno od okamžiku odeslání ATM buňky ze zdrojové ATM vrstvy do okamžiku, kdy je přijata vrstvou ATM v cílovém uzlu. V případě Frame Relay značí Delay (sec) zpoždění rámců (frames) v celé Frame Relay síti, od okamžiku odeslání rámce ze zdrojového uzlu do okamžiku jeho přijetí na cílovém uzlu. Z **Obr. B.15** je vidět, že technologie Frame Relay má větší zpoždění než stejně nastavená síť s ATM technologií.



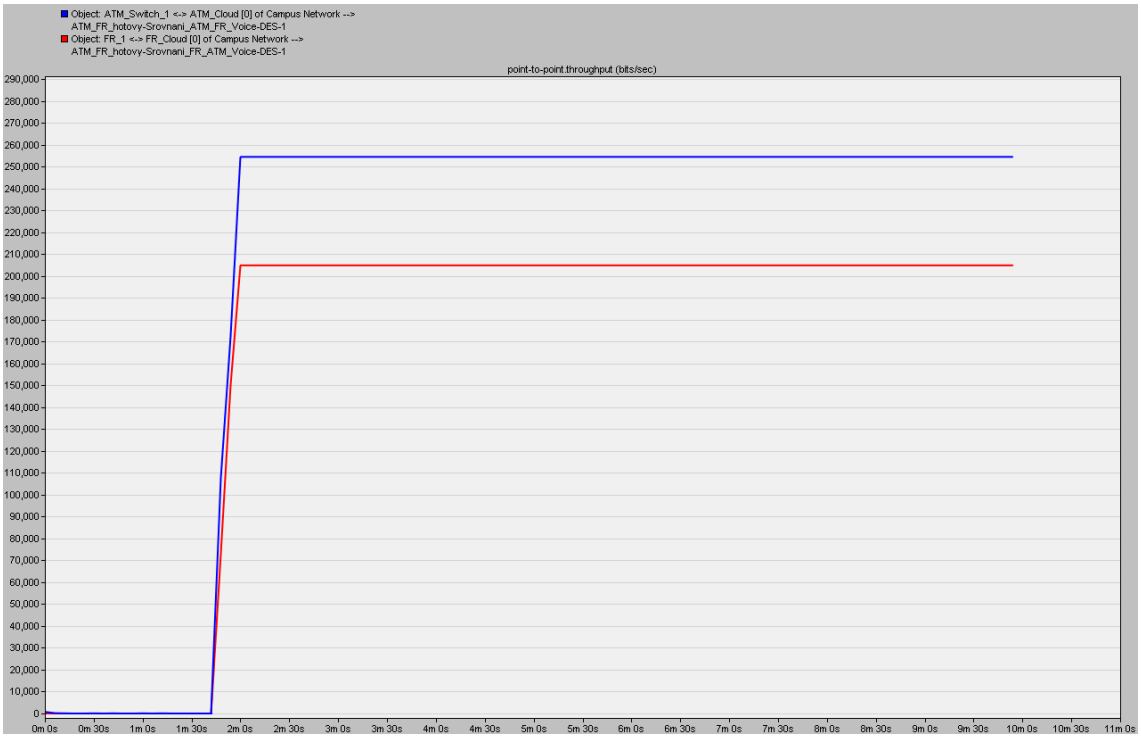
Obr. B.15: Srovnání Cell Delay pro ATM a Delay pro Frame Relay v s/min.

2) Cell Delay Variation - udává odchylku mezi zpožděními pro ATM buňky, které byly přijaté v ATM síti. Delay Variance je odchylka mezi zpožděními pro Frame Relay rámce v síti. Frame Relay má Delay Variance nulovou, naproti tomu ATM má Cell Delay Variation mnohonásobně vyšší (**Obr. B.16**).



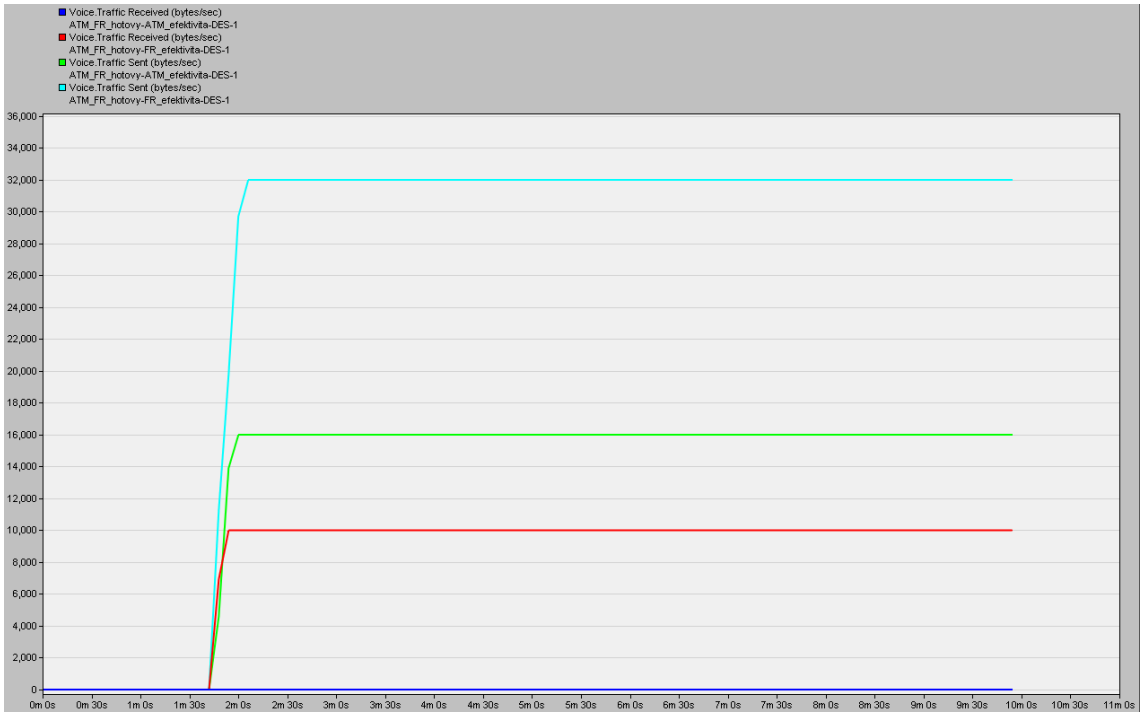
Obr. B.16: ATM Cell Delay Variation a Delay Variance pro Frame Relay v s/min.

3) Srovnání propustnosti linek ATM a Frame Relay.

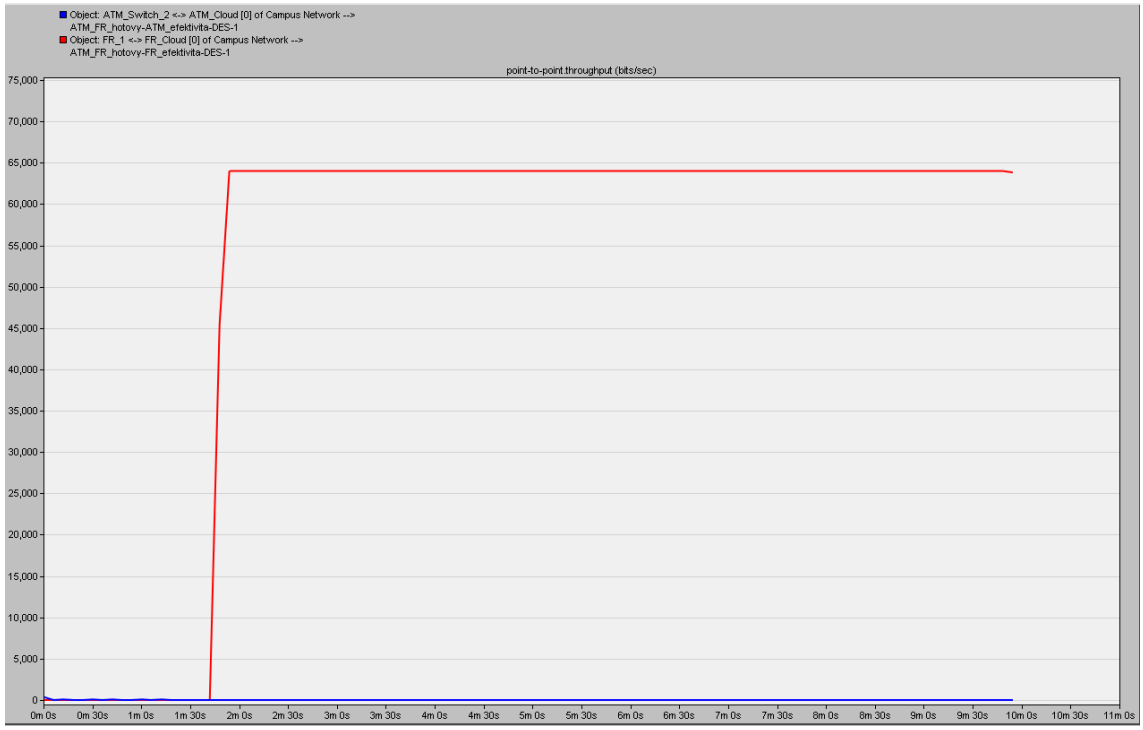


Obr. B.17: ATM a FR propustnost linky v bitech/s.

4) Zde se porovnává odeslaný a přijatý provoz v ATM a Frame Relay. Pro přenos hlasu je použita co nejpomalejší linka DS0 a to shodně na obou scénářích. Tato linka má maximální propustnost 64 kbit/s. PCM, které je rovněž využito v obou scénářích, má shodnou bytovou rychlost 64 kbit/s, viz **Obr. B.19**. U Frame Relay je možné uskutečnit přenos dat již od rychlosti 64 kbit/s, i když bude kapacita linky maximálně vytížena, viz **Obr. B.18**. Naopak ATM takto nízké rychlosti nepodporuje. V grafu je možné pro ATM pozorovat narůstající Traffic Sent (bytes/sec), ale Traffic Received (bytes/sec) již neproběhne. Z tohoto důvodu je na nižších rychlostech Frame Relay efektivnější než ATM.



Obr. B.18: Odeslaný a přijatý provoz pro hlasovou aplikaci s minimální rychlostí linky v bytes/s.



Obr. B.19: Throughput pro Voice v ATM a Frame Relay v bytes/sec.

C ŘEŠENÍ OTÁZEK A ÚKOLŮ PRO ÚLOHU PRÁCE S PROTOKLY IPV4 A IPV6

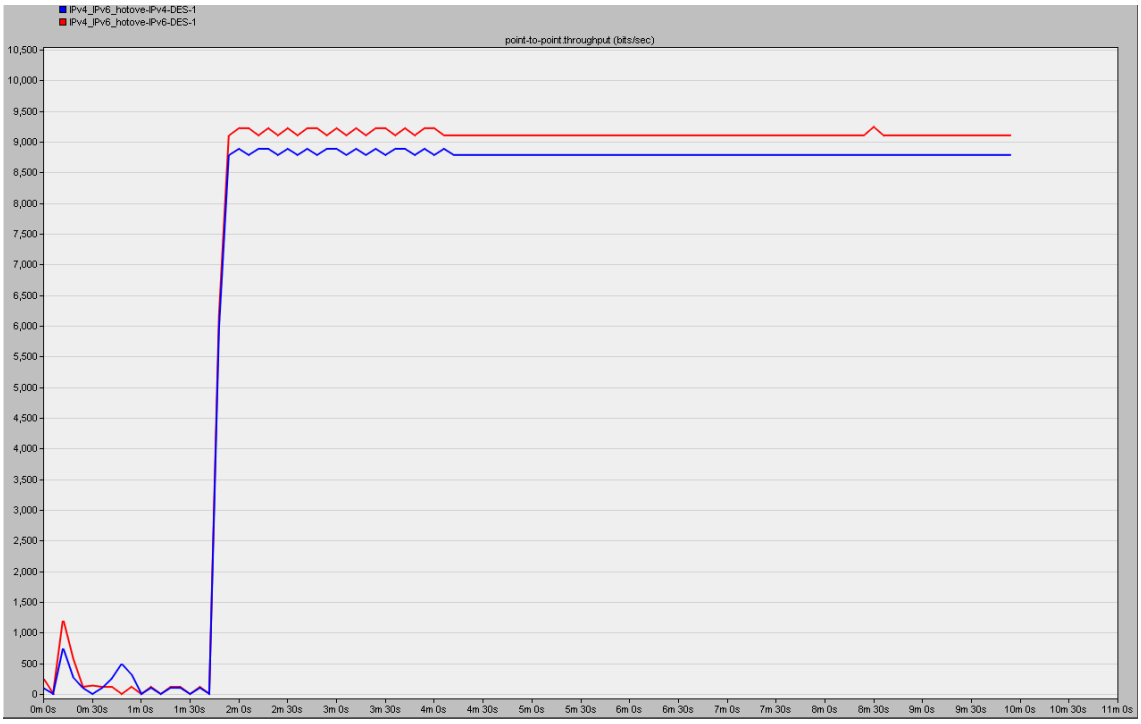
Poslední laboratorní úloha je zaměřena na konfiguraci a srovnání protokolů IPv4 a IPv6 z hlediska přenášených objemů dat, rychlosti odezvy a fragmentace paketů.

C.1 Úkol 1

Po konfiguraci protokolů IPv4 a IPv6 jsou úkoly zaměřeny na porovnání objemů přenášených dat, která souvisí s rozdílnou velikostí záhlaví těchto protokolů.

Doplňující otázky a úkoly:

1) V **Obr. C.1** je vidět větší objem přenesených dat u protokolu IPv6.



Obr. C.1: Propustnost linek IPv4 a IPv6.

2) Čím nižší bude datový tok, tím více se projeví rozdíly v hlavičkách v IPv4 a IPv6, viz **Obr. C.2**.

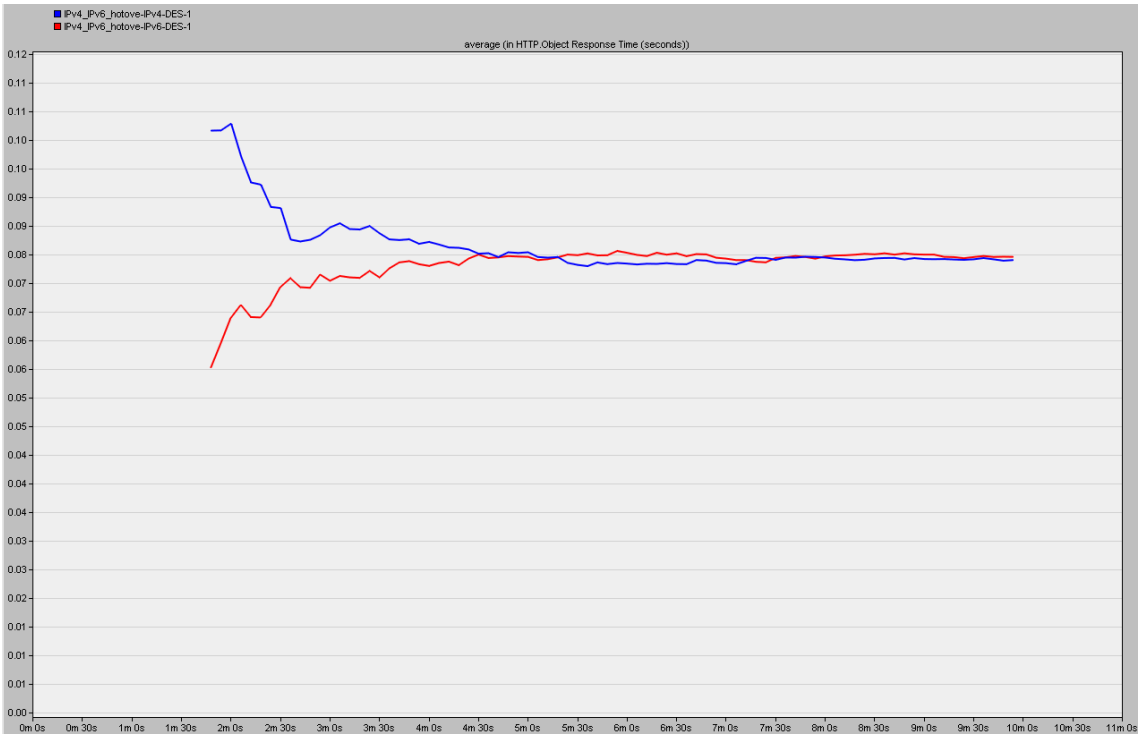
Obr. C.2: Propustnost linek pro scénáře s nižším provozem.

C.2 Úkol 2

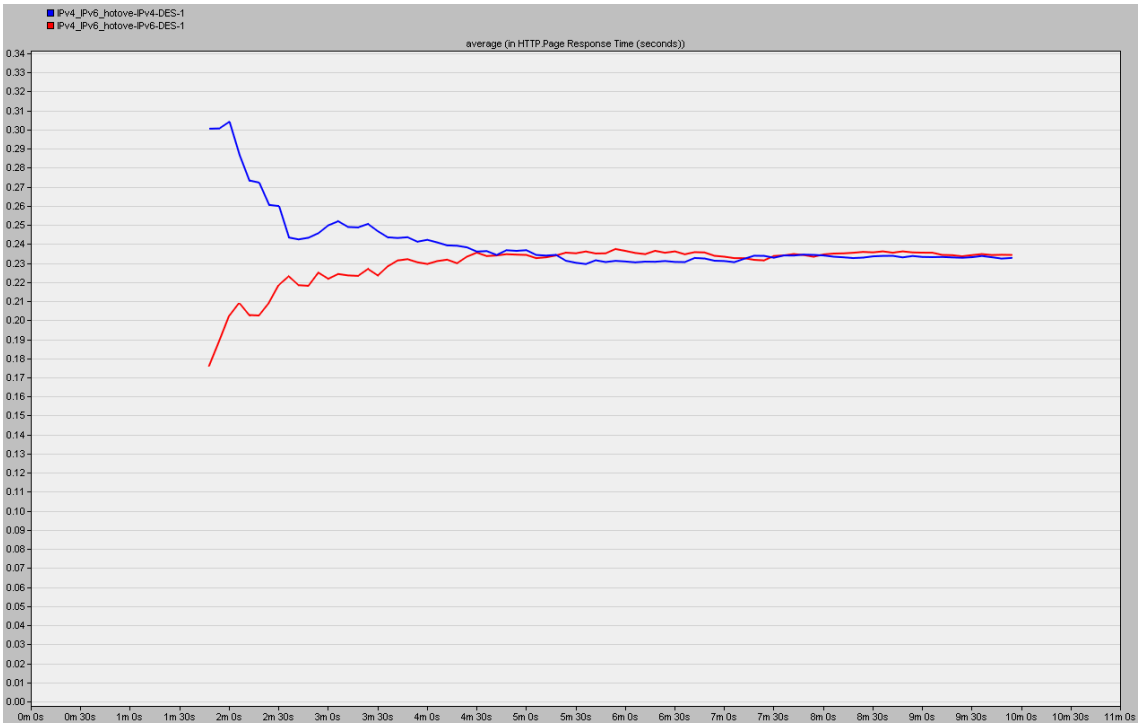
V tomto úkolu jsou předvytovřeny dva scénáře s nakonfigurovanou rozsáhlejší sítí pro IPv4 a IPv6 protokol, které budou srovnány z hlediska odezvy v aplikaci HTTP a porovnány s odezvou v menší síti.

Doplňující otázky a úkoly:

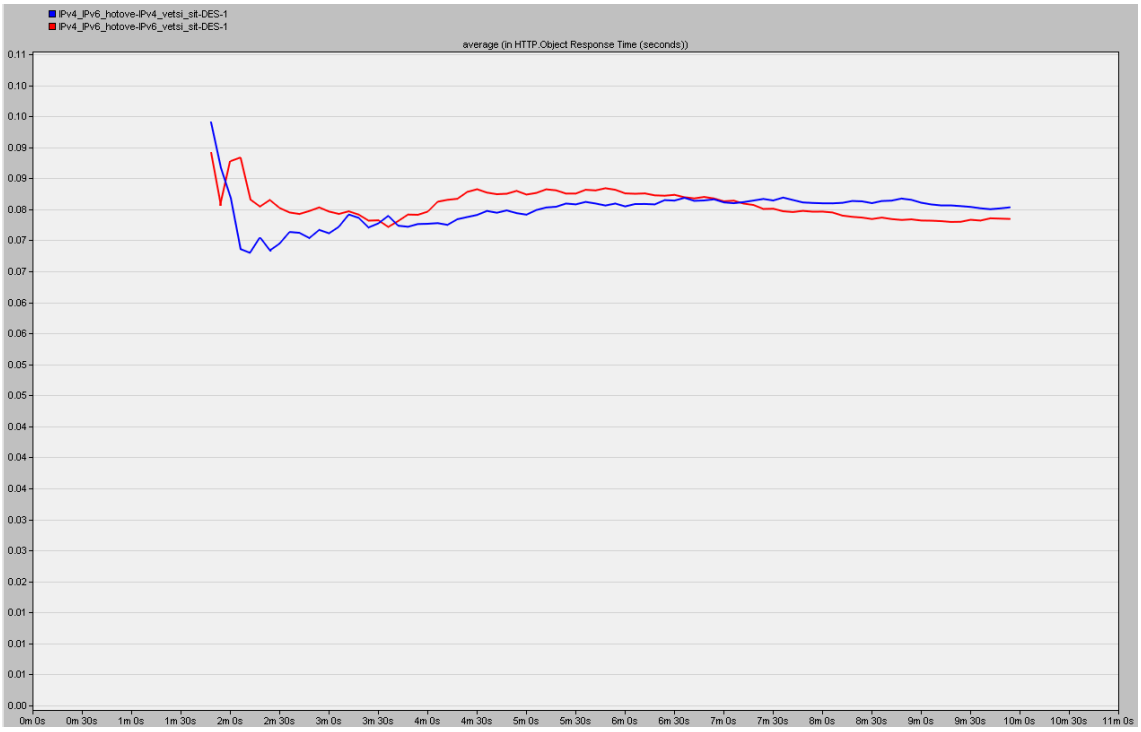
1) Na **Obr. C.3** a **Obr. C.4** je pro malou síť z hlediska odezvy lepší IPv4, naopak v rozsáhlejší větší síťi bude lépe vycházet IPv6.



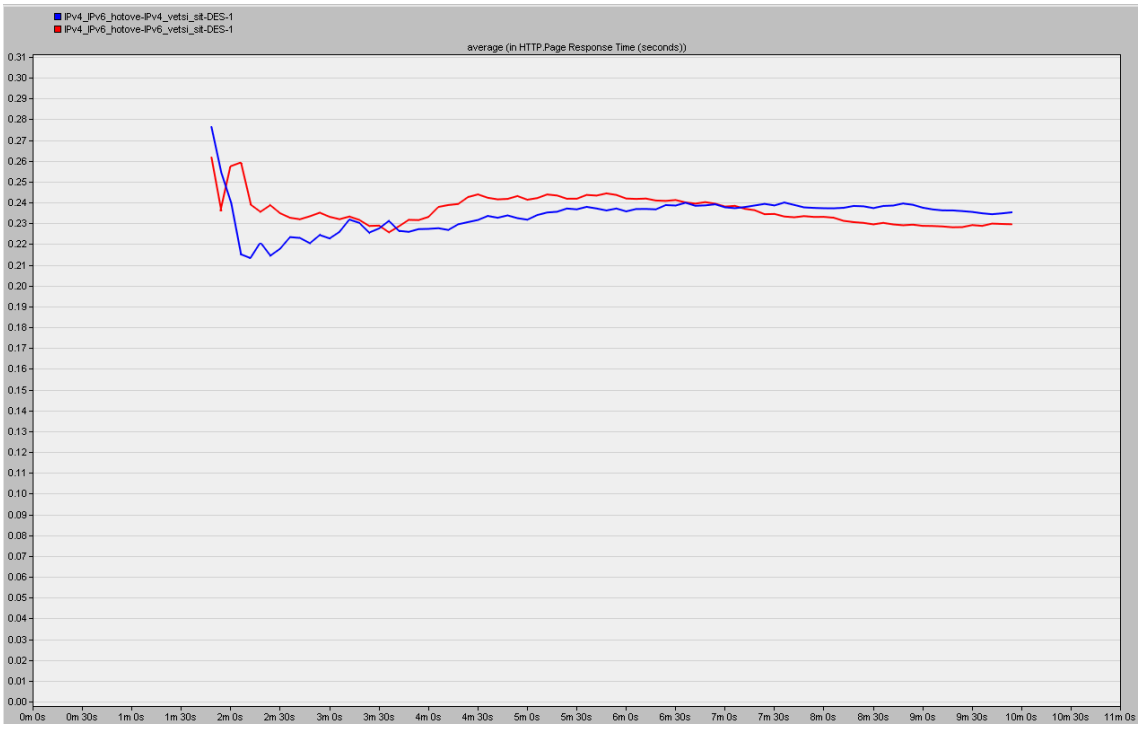
Obr. C.3: Srovnání IPv4 a IPv6 v malé síti pro HTTP s Response Time (seconds).



Obr. C.4: Srovnání IPv4 a IPv6 v malé síti pro HTTP s Page Response Time (seconds).



Obr. C.5: Srovnání IPv4 a IPv6 ve velké síti pro HTTP s Response Time (seconds).



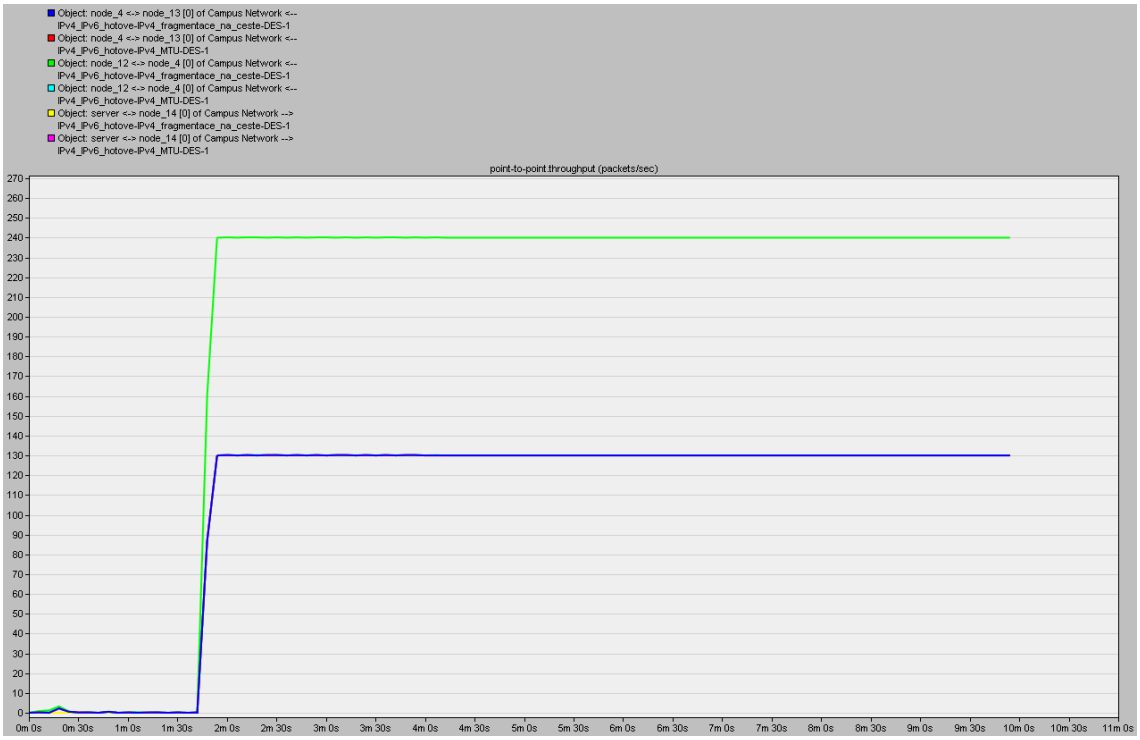
Obr. C.6: Srovnání IPv4 a IPv6 ve velké síti pro HTTP s Page Response Time (seconds).

C.3 Úkol 3

Poslední úkol je zaměřen na rozdíly ve fragmentaci paketů v protokolu IPv4 a IPv6. Další část úkolu se zaměřuje na zpoždění paketů při zpracování na směrovačích a na rozdíly ve zpoždění end-to-end delay následkem fragmentování pro aplikaci video konference.

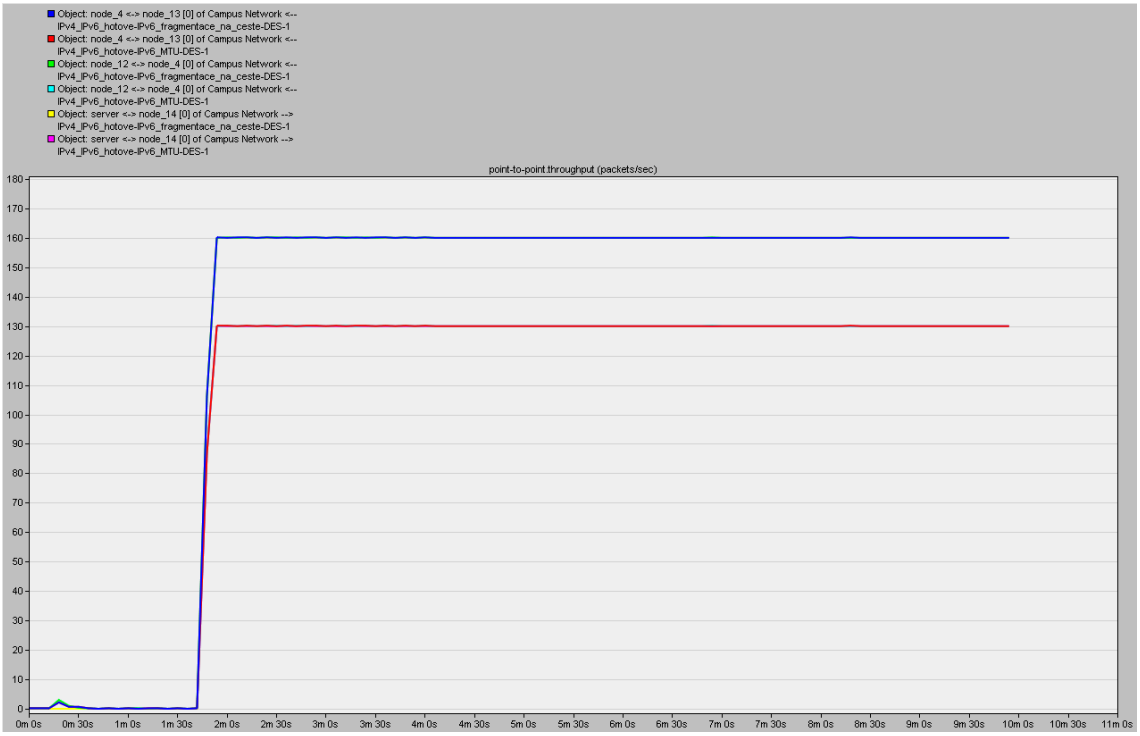
Doplňující otázky a úkoly:

1) Na **Obr. C.7** je možné pozorovat scénář s defaultně nastaveným MTU na 1500 bajtů a scénář, kde byla na uzlu node_4 nastavená jednotka MTU na 1300 bajtů. V obou scénářích je nastaven protokol IPv4.



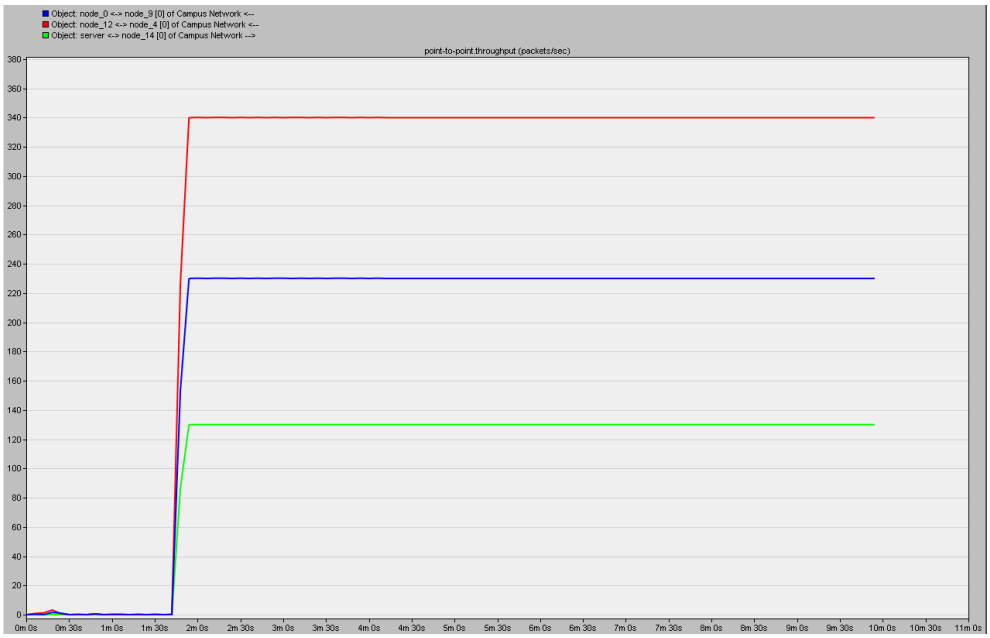
Obr. C.7: Fragmentace v různých uzlech pro IPv4.

2) Tento úkol byl nastaven stejně jako v přechozím bodě opět pro dva scénáře s defaultně nastavenou jednotkou MTU a MTU 1300 bajtů na node_4 a to s tím rozdílem, že v tomto případě je použit protokol IPv6, viz **Obr. C.8**.

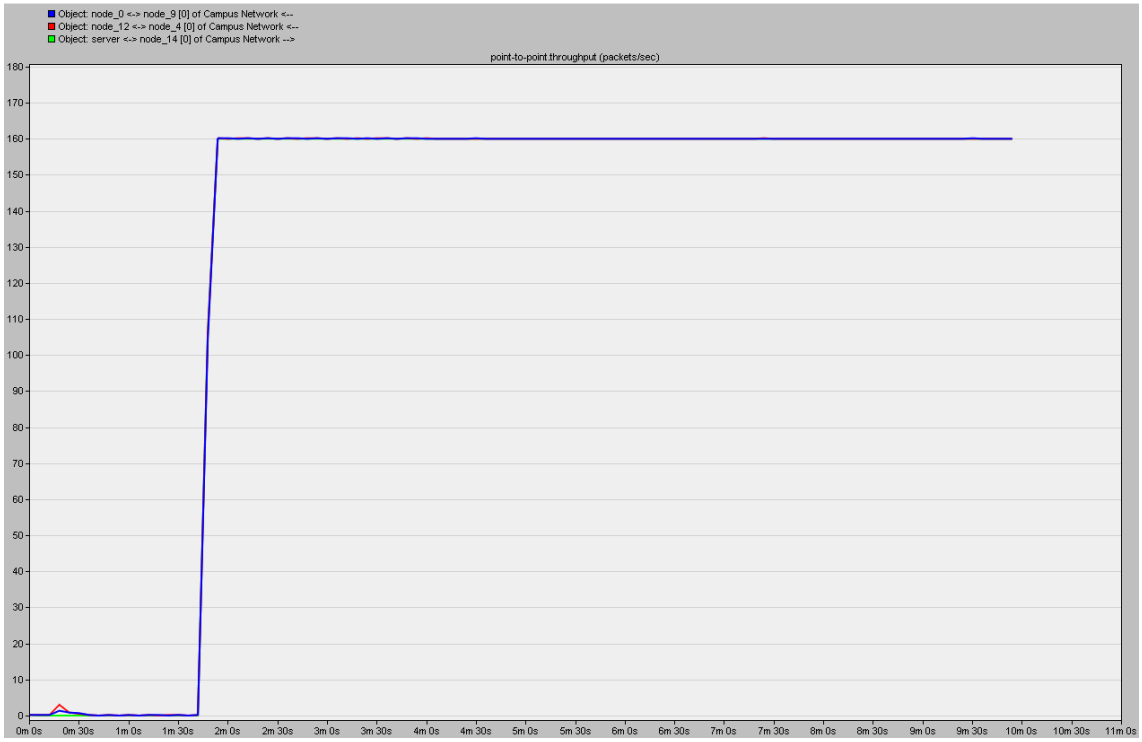


Obr. C.8: Fragmentace v různých uzlech pro IPv6.

3) V **Obr. C.9** a **Obr. C.10** byla opět ověřena teorie. Jak pro IPv4, tak pro IPv6 protokoly byly nastaveny na dvou různých směrovačích rozdílné jednotky MTU. Na node_9 bylo nastaveno 1400 bajtů a na node_4 1300 bajtů. Opět je možné pozorovat pro oba protokoly rozdílné chování při fragmentaci paketů. V IPv4 je možné fragmentovat pakety jak u odesilatele, tak i na jednotlivých směrovačích. Naopak u IPv6 je fragmentace možná jen u odesilatele.

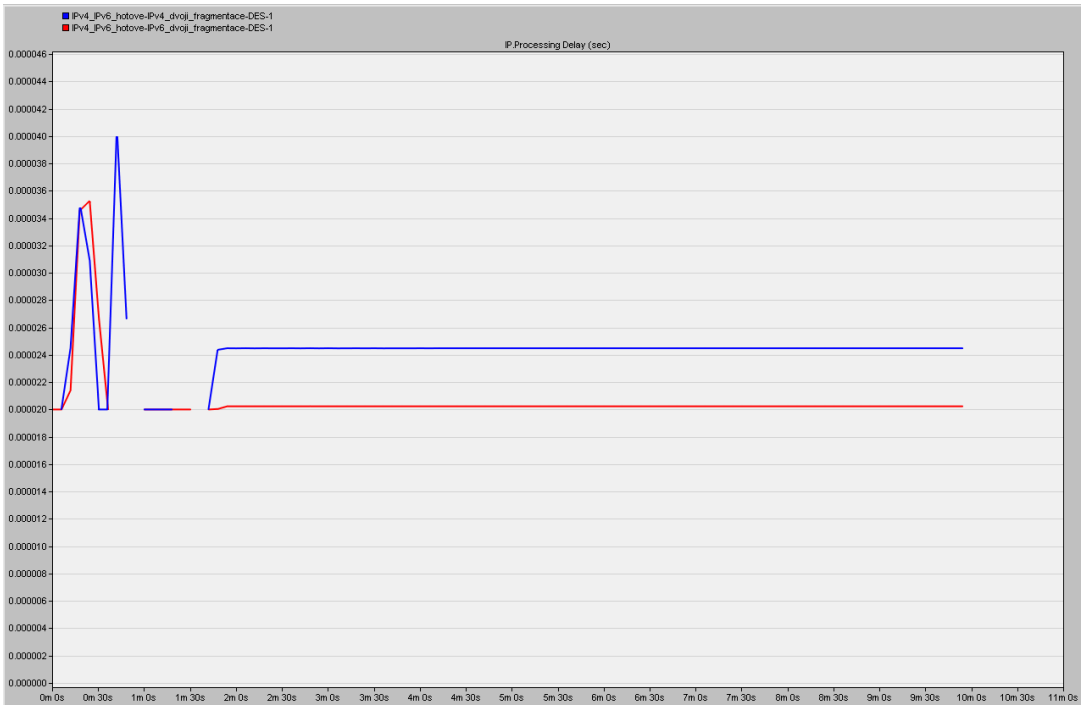


Obr. C.9: Dvojí fragmentace na různých směrovačích v IPv4.



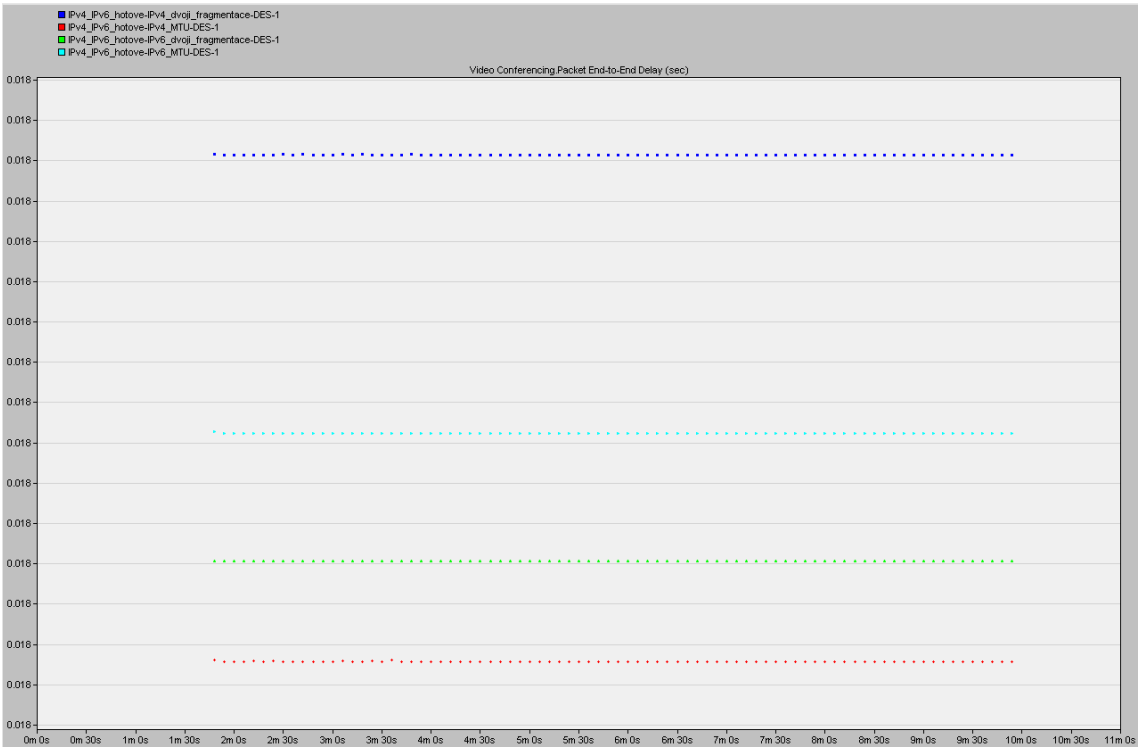
Obr. C.10: Dvojí fragmentace na různých směrovačích v IPv6.

4) Na **Obr. C.11** byl pro scénáře s IPv4 a IPv6 s dvojí fragmentací měřen Processing Delay, který udává, jak velké měl paket zpoždění od doby, kdy dorazil na IP vrstvu až do jeho zpracování. Z grafu je možné vypozorovat, že IPv6 má toto zpoždění mnohem nižší.



Obr. C.11: Processing Delay (sec) pro IPv4 a IPv6.

5) Poslední doplňující úkol byl zaměřen na rozdíly ve zpoždění end-to-end delay následkem fragmentování na dvou různých místech pro aplikaci video konference. Opět se srovnávají statistiky pro protokoly IPv4 a IPv6 se scénáři bez fragmentování. Zatímco, pro scénáře bez fragmentace vychází lépe IPv4. Naopak pokud bude k fragmentaci docházet, bude mít menší zpoždění IPv6, viz **Obr. C.12**.



Obr. C.12: Packet End-to-End Delay (sec) pro konferenci.