

ŽILINSKÁ UNIVERZITA V ŽILINE

FAKULTA RIADENIA A INFORMATIKY

KATEDRA INFORMAČNÝCH SIETÍ

DIPLOMOVÁ PRÁCA

Bc. Jozef Klačko

OPNET – SIMULÁCIA SIETE ŽU

Vedúci práce : Ing. Ľudovít Mikuš, PhD.

Registračné číslo: 6/2009

Máj 2010

ŽILINA 2010

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

OPNET – simulácia siete ŽU
DIPLOMOVÁ PRÁCA

Študijný program:

Informačné systémy

Pracovisko (katedra/ústav):

Katedra informačných sietí

Vedúci diplomovej práce:

Ing. Ľudovít Mikuš, PhD.

Žilina 2010

Bc. Jozef KLAČKO

Registračné číslo diplomovej práce 6/2009
DP zaregistrovaná dňa 30. 9. 2009

DIPLOMOVÁ PRÁCA

Meno a priezvisko Jozef Klačko **šk.rok** 2009/2010

Názov diplomovej práce OPNET – simulácia siete ŽU

Fakulta riadenia a informatiky Katedra Informatiky

Počet strán	56	Počet obrázkov	37	Počet tabuliek	10
Počet grafov	11	Počet príloh	9	Počet použ.lit.	9

ANOTÁCIA (slovenský jazyk)

Cieľom práce je vytvoriť model chrbticovej siete Žilinskej univerzity v Žiline v programe OPNET Modeler. Vo vytvorenom modeli simulovať prevádzku spôsobenú elektronickým vyučovaním a vyhodnotiť výsledky simulácií. Okrem uvedeného je v práci popísaný stav siete, spôsoby modelovania prevádzky v programe OPNET Modeler a spôsoby, ako zrýchliť simuláciu. Zároveň sú uvedené príklady na správanie sa siete pri zlyhaní linky, príklady nastavenia a použitia smerovacích protokolov a modelovanej prevádzky. Výsledky práce slúžia ako príprava na zavedenie elektronického vzdelávania formou e-kurzov a video prednášok.

Kľúčové slová: OPNET Modeler, simulácia, sieť, elektronické vzdelávanie

ANNOTATION (English language)

The aim of diploma thesis is to create model of network backbone of University of Žilina in OPNET Modeler. Then in created model simulate network transmission caused by e-learning and evaluate results of simulations. In addition is in this thesis described actual state of network, methods of modeling network traffic in OPNET Modeler and ways to speed up the simulation. There are examples of the behavior of network in case of line failure, examples of setting and using routing protocols and modeled network transmission. The results of the work are useful for preparation of implementation e-learning through e-courses and video lectures.

Keywords: OPNET Modeler, simulation, network, e-learning

VEDÚCI DIPLOMOVEJ PRÁCE Ing. Ľudovít Mikuš, PhD., Katedra informačných sietí,
Žilinská univerzita

OPONENT DIPLOMOVEJ PRÁCE _____

Dátum 7.5.2010

ŽILINSKÁ UNIVERZITA V ŽILINE, FAKULTA RIADENIA A INFORMATIKY

ZADANIE TÉMY DIPLOMOVEJ PRÁCE

Študijný odbor: Informačné systémy
Zameranie: Informačno-komunikačné siete

Meno a priezvisko

Jozef Klačko

Osobné číslo

551480

Názov práce v slovenskom aj anglickom jazyku

OPNET - simulácia siete ŽU
OPNET - simulation UoZ Network

Zadanie úlohy, ciele, pokyny pre vypracovanie

Cieľ diplomovej práce:

Vytvorenie modelu siete Žilinskej univerzity za účelom simulovania rôznych udalostí

Obsah:

V programe OPNET vytvorte model siete FRI podľa nasledovných požiadaviek:

- model bude čo najvernejšie zodpovedať skutočnej konfigurácii
- pri jednotlivých prvkoch siete uvediete možnosti nastavenia ich parametrov
- na základe informácií od správcu siete nastavíte parametre prvkov tak aby zodpovedali realite
- na vytvorenom modeli spustíte niekoľko simulácií, pri ktorých budete meniť zaťaženie siete
- na vytvorenom modeli spustíte niekoľko simulácií, pri ktorých budete meniť parametre vybraných prvkov siete

Témy z predmetov študijného zamerania

5SI31: 3, 4, 5

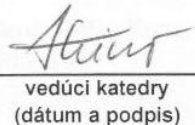
Meno a pracovisko vedúceho DP:

Ing. Ľudovít Mikuš , FRI - KIS , ŽU

Meno a pracovisko tútora DP:

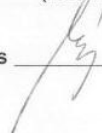

vedúci DP
(dátum a podpis)

tútor
(dátum a podpis)


vedúci katedry
(dátum a podpis)

garant
(dátum a podpis)

Zadanie zaregistrované dňa 30. 09. 2009 pod číslom 6/2009 podpis



Pod'akovanie

Ďakujem Ing. Ľudovítovi Mikušovi, PhD., vedúcemu diplomovej práce a tiež Mgr. Jane Uramovej, PhD., za vedenie, ochotu, pripomienky a návrhy k diplomovej práci.

Prehlásenie

Prehlasujem, že som diplomovú prácu vypracoval samostatne, úplný zoznam použitých prameňov uvádzam v zozname literatúry a zdroje som využil spôsobom vo vede obvyklým.

V Žiline, dňa 7. mája 2010

Podpis.....

ABSTRAKT

KLÁČKO, Jozef: *OPNET - Simulácia siete ŽU* [diplomová práca] – Žilinská univerzita v Žiline. Fakulta riadenia a informatiky, Katedra informačných sietí. – Vedúci: Ing. Ľudovít Mikuš, PhD. – Stupeň odbornej kvalifikácie: Inžinier v odbore Informačné systémy. Žilina: FRI ŽU v Žiline, 2010 – 56 strán.

Cieľom práce je vytvoriť model chrbticovej siete Žilinskej univerzity v Žiline v programe OPNET Modeler. Vo vytvorenom modeli simulovať prevádzku spôsobenú elektronickým vyučovaním a vyhodnotiť výsledky simulácií. Okrem uvedeného je v práci popísaný stav siete, spôsoby modelovania prevádzky v programe OPNET Modeler a spôsoby, ako zrýchliť simuláciu. Zároveň sú uvedené príklady na správanie sa siete pri zlyhaní linky, príklady nastavenia a použitia smerovacích protokolov a modelovanej prevádzky. Výsledky práce slúžia ako príprava na zavedenie elektronického vzdelávania formou e-kurzov a video prednášok.

Kľúčové slová: OPNET Modeler, simulácia, sieť, elektronické vzdelávanie

ABSTRACT

KLÁČKO, Jozef: *OPNET – Simulation of UoZ network* [diploma thesis] – The University of Žilina. Faculty of Management Science and Informatics, Department of Information and Communication Networks. – Leader: Ing. Ľudovít Mikuš, PhD. – Degree of qualification: Engineer in field Information Systems. Žilina: FRI ŽU v Žiline, 2010 – 56 pages.

The aim of diploma thesis is to create model of network backbone of University of Žilina in OPNET Modeler. Then in created model simulate network transmission caused by e-learning and evaluate results of simulations. In addition is in this thesis described actual state of network, methods of modeling network traffic in OPNET Modeler and ways to speed up the simulation. There are examples of the behavior of network in case of line failure, examples of setting and using routing protocols and modeled network transmission. The results of the work are useful for preparation of implementation e-learning through e-courses and video lectures.

Keywords: OPNET Modeler, simulation, network, e-learning

OBSAH

OBSAH	10
ZOZNAM ILUSTRÁCIÍ.....	12
ZOZNAM TABULIEK	15
ZOZNAM SKRATIEK A ZNAČIEK.....	16
ZOZNAM PRÍLOH.....	17
ZOZNAM POUŽITEJ LITERATÚRY	18
ÚVOD.....	1
1 POJEM A VÝZNAM E-VZDELÁVANIA	2
2 SIEŤ ŽILINSKEJ UNIVERZITY.....	3
2.1 Účel zriadenia siete.....	3
2.2 Súčasný stav siete.....	3
3 MODELOVANIE SIEŤOVEJ PREVÁDZKY V PROGRAME OPNET	
 MODELER	6
3.1 Explicitná, diskretná sieťová prevádzka (Explicit traffic).....	6
3.2 Sieťová prevádzka modelovaná na pozadí liniek (Background traffic, analytical traffic).....	7
3.3 Aplikácie (Applications)	8
3.3.1 Model požiadaviek prevádzky aplikácie (Application Demands)	9
3.3.2 Štandardné modely prevádzky	11
3.3.3 Vlastné modely prevádzky	14
3.3.4 Modely prevádzky získané zachytením reálnej prevádzky	15
3.3.5 Modely prevádzky vytvorené nastavením správ (ACE Whiteboard)	17
3.4 Toky sieťovej prevádzky (Traffic flows, Background routed traffic).....	21
3.5 Základné zaťaženie (Baseline Loads).....	21
4 METÓDY ZRÝCHLENIA SIMULÁCIE.....	24
4.1 Paralelná simulácia	24
4.2 Distribuovaná simulácia	24
4.3 Použitie 64-bitového simulačného jadra	29
4.4 Zmena zobrazovania elementov výstupu počas simulácie	31
4.5 Efektívne simulovanie protokolov OSPF, ARP, BGP, EIGRP, IGMP, IGRP, ISIS, RIP a SWITCH (STP)	32
4.6 Vývojové (development) a optimalizované (optimized) simulačné jadro	33

4.7	Optimalizácia kódu.....	33
4.8	Použitie modelu požiadaviek aplikácií (Demands) a modelu prevádzky na pozadí (Background traffic).....	33
4.9	Vypnutie zaznamenávania a zobrazovania animácií.....	33
5	ŠTATISTICKÉ SPRACOVANIE VÝSLEDKOV DISKRÉTNYCH UDALOSTNÝCH SIMULÁCIÍ.....	35
5.1.1	Výpočet intervalov spoľahlivosti.....	35
6	ANALÝZA E-VZDELÁVACÍCH KURZOV.....	39
6.1	Druhy prevádzky	39
6.1.1	Krátke videá alebo vektorové animácie so sprievodným textom	39
6.1.2	Päťdesiatminútové videá.....	39
6.2	Odhad počtu aktívnych študentov za hodinu.....	41
7	POROVNANIE PRVKOV SIETE S MODELMI.....	43
7.1	Redundancia liniek a zariadení.....	43
7.1.1	Možnosti zálohovania predvolenej brány v OPNET Modeler.....	44
8	IMPLEMENTÁCIA CHRBTICOVEJ SIETE V PROGRAME OPNET MODELER.....	45
8.1	Použité modely prevádzky a výsledky simulácií.....	48
8.1.1	Model prevádzky vytváraný profilmi a aplikáciami.....	49
8.1.2	Model požiadaviek dát (Application demands)	50
8.1.3	Modelovanie zlyhania linky.....	52
	ZÁVER	57
	Príloha B: Kalkulačka bitovej rýchlosti.....	1
	Príloha E: Inicializačný blok ACE skriptu	1
	Príloha F: Hlavičkový blok ACE skriptu.....	2
	Príloha G: Funkčný blok ACE skriptu.....	2
	Príloha H: Chybové hlásenie simulácie	3
	Príloha I: Stránka služby youtube.com	4

ZOZNAM ILUSTRÁCIÍ

Obr. 2.1 Stav siete UTC-Net v roku 1998	3
Obr. 2.2 Stav siete v roku 2007	4
Obr. 2.3 Slovenská akademická a dátová sieť. Pripojenie Catalyst 6509 na Martin, Púchov a sieť PIONEER.....	5
Obr. 3.1 Určenie pomeru explicitnej prevádzky a prevádzky na pozadí v modeli požiadaviek aplikácií	7
Obr. 3.2 Spôsoby modelovania prevádzky pomocou modelu aplikácií v OPNET Modeler [7]	9
Obr. 3.3 Topológia siete z príkladu. Šípka predstavuje tok dát modelovaný pomocou AD.....	10
Obr. 3.4 AD modelovaný ako 100% explicitný, graf hore, tok dát na linke medzi switch1 a router1, dole medzi switch1 a router2	11
Obr. 3.5 AD modelovaný ako 100%-ná prevádzka na pozadí, graf hore, tok dát na linke medzi switch1 a router1, dole medzi switch1 a router2, Modelovaný tok napriek očakávaniu nie je ovplyvnený výpadkom linky	11
Obr. 3.6 Spôsob nastavenia vstavaných aplikácií.....	12
Obr. 3.7 Dve podoby modulov konfigurácie aplikácií a modulu konfigurácie profilov.....	12
Obr. 3.8 Aplikácia "HTTP prenos 1kB stránky a 200kB suboru "	13
Obr. 3.9 Nastavenie veľkosti stránky a súbežne sťahovaného súboru	13
Obr. 3.10 Modul Tasks z palety utilities.....	14
Obr. 3.11 Fázy úlohy	14
Obr. 3.12 Application Capture Manager	16
Obr. 3.13 Grafické rozhranie ACE Whiteboard	18
Obr. 3.14 Skript logiky odosielania správ v Python	19
Obr. 3.15 Inicializačný blok	19
Obr. 3.16 Deklarácia parametrov	20
Obr. 3.17 Nástroj QucikPredict	21
Obr. 3.18 Modelovanie priebehu zaťaženia linky pomocou modelu základného zaťaženia	22
Obr. 3.19 Zobrazenie základného zaťaženia liniek v aktuálnom čase.....	22
Obr. 3.20 Nástroj Time Controller.....	23
Obr. 4.1 Priebeh jednotlivých simulácií	25
Obr. 4.2 Topológia siete s distribuovanou simuláciou	26
Obr. 4.3 Nastavenie zdieľania prístupu k priečinku	27

Obr. 4.4 Pridelenie prístupových práv k priečinku pre všetkých používateľov	27
Obr. 4.5 Nastavenie dodatočných práv pre všetkých používateľov	27
Obr. 4.6 Nastavenie parametrov simulácie	29
Obr. 4.7 Výsledky simulácie. Počet aktívnych pripojení (os y) protokolu TCP v závislosti od času (os x) a počtu počítačov v sieti (farebne odlíšené grafy).....	29
Obr. 4.8 Príklad zapnutia 64-bitovej a zároveň paralelnej simulácie spustenej v dvoch vláknach	30
Obr. 4.9 Parametre kompilácie modulu	31
Obr. 5.1 Viac hodnôt násady spôsobí spustenie viacerých simulačných behov	35
Obr. 5.2 Graf využitia linky ne112a ← swi.....	36
Obr. 5.3 Maximálne hodnoty zaťaženia linky	37
Obr. 5.4 Nastavenie osi x grafu	37
Obr. 5.5 Nastavenie grafu na zobrazenie intervalu spoľahlivosti.....	38
Obr. 6.1 Hustota pravdepodobnosti trojuholníkového rozdelenia s parametrami min = 100000, max = 2000000 (os y – pravdepodobnosť hodnôt z osi x).....	40
Obr. 6.2 Nenormalizovaná hustota pravdepodobnosti veľkosti súboru (os x – hodnoty veľkosti súboru, os y – pravdepodobnosť)	40
Obr. 7.1 Modul zlyhanie a obnova linky alebo uzlu.....	44
Obr. 8.1 Topológia siete Žilinskej univerzity v programe OPNET Modeler. Nepoužívané zariadenia sú preškrtnuté červeným krížikom.	46
Obr. 8.2 Objekt 1000BaseX_LAN, v modeli predstavuje aktívne počítače na určenej fakulte.....	47
Obr. 8.3 Graf vytťaženia linky medzi smerovačom fsi a smerovačom v Púchove, os x – čas v hodinách, os y – milióny bitov za sekundu	48
Obr. 8.4 Reprezentácia základnej (baseline) prevádzky z uzla Púchov do uzla fsi v programe OPNET Modeler, os x – čas v hodinách, os y – bity za sekundu	48
Obr. 8.5 Priemerné vytťaženie linky v bitoch za sekundu pre jednotlivé druhy videí	51
Obr. 8.6 Veľkosť toku medzi http_server a ne112a v bitoch za sekundu, os y - bity za sekundu, os - x čas v sekundách.....	52
Obr. 8.7 Topológia časti siete po pridaní prepínačov medzi zariadenia ne112a, ne112e a fsi	53
Obr. 8.8 Grafy vytťaženia liniek medzi zariadeniami ne112a <- fsi, ne112e <- fsi, ne112a -> ne112e. Po výpadku linky fsi – ne112a (prvý graf) v čase 350 s. je celá prevádzka medzi uzlami fsi a ne112a presmerovaná cez uzol ne112e (druhý a tretí graf).	54

-
- Obr. 8.9 Graf sieťovej prevádzky na vnútorných linkách medzi prepínačmi ne112a -> ne112e, ne112e -> fsi, ne112a > fsi. Správanie prevádzky v obidvoch smeroch liniek je okrem veľkosti prevádzky totožné. Os x – množstvo bitov prenesených linkou za sekundu v smere 55
- Obr. 8.10 Graf vytiaženia linky pri serveri http_video, v smere od servera. Prvý graf zobrazuje prenosovú rýchlosť, ak sú medzi prepínačmi pripojené ďalšie prepínače, a teda nedostanú okamžitú informáciu o výpadku linky. Druhý graf zobrazuje priame pripojenie prepínačov. Os x - čas v sekundách, os y - množstvo prenesených bitov za sekundu. 55
-

ZOZNAM TABULIEK

Tab. 3.1 Výhody a nevýhody modelu požiadaviek prevádzky	9
Tab. 3.2 Výhody a nevýhody štandardných modelov prevádzky	12
Tab. 3.3 Výhody a nevýhody vlastného modelu aplikácie	14
Tab. 3.4 Výhody a nevýhody modelovania prevádzky pomocou zachytenej prevádzky	15
Tab. 3.5 Výhody a nevýhody modelovania prevádzky pomocou ACE Whiteboard.....	17
Tab. 6.1 Priemerné veľkosti súborov 50 minútových videí.....	41
Tab. 6.2 Určenie počtu aktívnych študentov	42
Tab. 7.1 Reprezentácia zariadení v simulovanej sieti.....	43
Tab. 8.1 Hodnoty časovačov protokolu OSPF	46
Tab. 8.2 Intervaly spoľahlivosti priemerného vytiaženia linky medzi ne112a a http_server	50

ZOZNAM SKRATIEK A ZNAČIEK

ACE	A pplication p erformance, časť softvérového balíka OPNET Modeler slúžiaca na modelovanie sieťovej prevádzky
API	A pplication P rogramming I nterface, rozhranie pre programovanie aplikácií, množina funkcií poskytovaných programom, knižnicou alebo operačným systémom
CPU	C entral P rocessing U nit, procesor počítača
EIGRP	E nhanced I nterior G ateway P rotocol, smerovací protokol
IP	I nternet P rotocol, komunikačný protokol sieťovej vrstvy
LAN	L ocal A rea N etwork, lokálna počítačová sieť
MAC	M edia A ccess C ontrol, identifikačné číslo sieťového adaptéra
NDIS	N etwork D river I nterface S pecification, rozhranie využívané na programovú komunikáciu aplikácií so sieťovou kartou
OS	o peračný s ystém
OSI	O pen S ystem I nterconnection, abstraktný opis návrhu štruktúry komunikačných protokolov
pixel	picture element, najmenšia zobrazovaná časť obrazovky, videa, alebo obrázku
STP	S panning T ree P rotocol, protokol dátovej vrstvy OSI modelu, ktorý zabraňuje vytváraniu slučiek v prepínanej sieti
mod_dirs	položka globálnych nastavení adresárov (Edit > Preferences, Ctrl+Alt+P) programu OPNET Modeler

ZOZNAM PRÍLOH

- Príloha A: DVD médium – diplomová práca v elektronickej podobe
- Príloha B: Kalkulačka bitovej rýchlosti
- Príloha C: Článok <http://nil.uniza.sk/sk/network-simulation-and-modelling/opnet/priklad-vyvozovania-sietovej-prevadzky-eigrp-loadballancing>: *EIGRP load balancing*
- Príloha D: Článok <http://nil.uniza.sk/sk/network-simulation-and-modelling/opnet/prechod-prevadzky-s-exponencialnymi-medzerami-cez-siet-s-uzlami-s-exponencialnym-oneskore>: *Prechod prevádzky s exponenciálnymi medzerami cez sieť s uzlami s exponenciálnym oneskorením*
- Príloha E: Inicializačný blok ACE skriptu
- Príloha F: Hlavičkový blok ACE skriptu
- Príloha G: Funkčný blok ACE skriptu
- Príloha H: Chybové hlásenie simulácie
- Príloha I: Stránka služby youtube.com
-

ZOZNAM POUŽITEJ LITERATÚRY

- [1] Mikuš, Ľ.: *Perspektívy ICT vo vzdelávaní*, 2008, 12 s.
 - [2] <https://nic.uniza.sk/zuwiki/doku.php?id=zu:net:netinfo>
 - [3] <http://nil.uniza.sk/network-simulation-and-modelling/opnet/opnet>
 - [4] Kavička – Klima – Adamko: *Agentovo orientovaná simulácia dopravných uzlov*. Žilina : ŽU, 2005, 206 s. ISBN 80-8070-477-5
 - [5] http://www.adobe.com/devnet/flash/apps/flv_bitrate_calculator/
 - [6] <http://www.portalvs.sk/sk/vysoka-skola/zilinska-univerzita-v-ziline>, 17.02.2010
 - [7] dokumentácia OPNET Modeler
 - [8] <https://enterprise1.opnet.com/support/product/md>
 - [9] <http://www.winpcap.org/>
-

ÚVOD

Vývoj technológií súvisiacich s prenosom údajov je sprevádzaný rýchlymi zmenami. Zmeny sa prejavujú v oblasti poskytovania služieb a následne aj v oblasti vývoja hardvéru a softvéru. Nový druh služieb, ale aj požiadavky na zvyšovanie kvality v poskytovaní súčasných služieb spôsobujú neustály nárast požiadaviek parametre siete. A to najmä na priepustnosť dosahovanú na ceste od servera ku klientovi. Tá je ovplyvnená aj šírkou pásma liniek a rýchlosťou spracovávaní údajov v zariadeniach.

Jednou z novších služieb poskytovaných po sieti je aj služba elektronického vzdelávania. Úlohou elektronického vzdelávania je efektívne vyučovanie z pohľadu študenta, ale aj vyučujúceho. Existuje viac spôsobov, ako vykonávať elektronické vzdelávanie. V práci som testoval dve možnosti. Prednášky, ktoré by boli nahrávané a spracované do videozáznamu a vzdelávanie pomocou vytvorených e-kurzov, obsahujúcich text a krátke animácie. Obidva spôsoby vyučovania majú rozličné nároky na prvky siete, pričom prvá vyžaduje prenos väčších objemov dát prenášaných ako prúdový tok.

Testovať uvedenú prevádzku na produkčných systémoch môže byť nákladné, zdĺhavé a mať nepriaznivé dôsledky na ostatné poskytované služby siete. Preto je vhodné sieťovú prevádzku najprv simulovať a až potom preniesť poznatky zo simulácií do praxe.

Cieľom práce je overiť pripravenosť chrbticovej siete na prevádzku spôsobenú elektronickým vzdelávaním simuláciou v programe OPNET Modeler.

POJEM A VÝZNAM E-VZDELÁVANIA

„E-vzdelávanie je elektronické vzdelávanie, ktoré využíva počítačovú sieť na realizáciu, interakciu alebo podporu výučby. Počítačovou sieťou sa pritom rozumie lokálna sieť intranet alebo celosvetová sieť internet.“ [1]

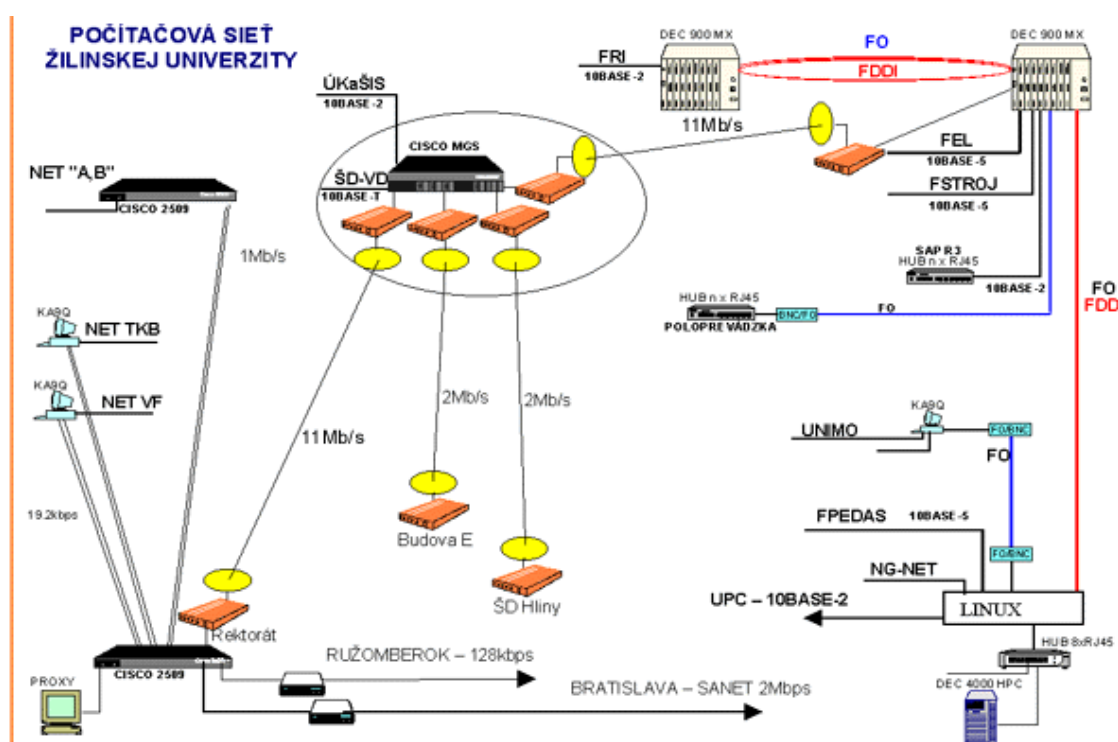
„Oproti iným formám má využitie internetu množstvo výhod. V prvom rade je to mimoriadne rýchla a efektívna forma komunikácie – poskytuje možnosť posielania textovej, obrazovej aj zvukovej formy informácie prakticky v reálnom čase a za veľmi nízku jednotkovú cenu. Vďaka tomu vytvára priestor pre využívanie rôznych foriem učebných materiálov – od textov a obrázkov, cez animácie a audiozáznamy, až po videozáznamy a multimediálne interaktívne aplikácie.“ [1]

Na prenos dát prenosovým médiami je potrebná prenosová kapacita. Tá pôsobí ako limitujúci faktor pri implementácii e–vzdelávania. Znížená prenosová kapacita obmedzuje množstvo dát prenesených médiami, a tak aj kvalitu prenášaného videa a zvuku. Uvedený problém je značný najmä na pomalých pripojeniach. Zároveň znížená kvalita videa a zvuku znižuje kvalitu vzdelávacích materiálov a vzdelávacieho procesu. Preto je pri implementácii e–vzdelávania nutné uvažovať nielen o dopadoch implementácie e–vzdelávania na prenosové médium, ale aj o dopadoch kvality prenosového média na e–vzdelávanie.

SIEŤ ŽILINSKEJ UNIVERZITY

Účel zriadenia siete

Sieť Žilinskej univerzity (UTC-Net) bola zriadená pre výskumné a študijné účely s ohľadom na zameranie jednotlivých pracovísk. Značným prínosom je rozvoj nových aplikácií a foriem vzdelávania, podpora správy a výskumu nových služieb.



Obr. 0.1 Stav siete UTC-Net v roku 1998

Súčasný stav siete

Sieť pozostáva z prístupových LAN sietí fakúlt, internátov a ostatných pracovísk univerzity, prioritne tvorených štruktúrovanou kabelážou, vzájomne prepojených prostredníctvom metropolitnej siete (MAN-ŽU).

„MAN-ŽU spája pomocou optických vlákien jednotlivé lokality univerzity a vytvára vysokorychlostnú chrbticovú sieť využívajúcu technológiu prepínaného gigabitového ethernetu. Aktívne prvky MAN-ŽU sú rozdelené do dvoch základných skupín. Jednu skupinu tvoria chrbticové prepínače zabezpečujúce spojenie medzi jednotlivými lokalitami metropolitnej siete, druhú skupinu aktívnych prvkov tvoria

V rámci siete SANET je tento prepínač pripojený na prepínač v Martine a Púchove a zároveň na prepínač v Poľsku do siete PIONEER. Priepustnosť všetkých liniek v oboch smeroch je 10Gb/s. Názornejšie to zobrazuje Obr. 0.3.



Obr. 0.3 Slovenská akademická a dátová sieť. Pripojenie Catalyst 6509 na Martin, Púchov a sieť PIONEER

MODELOVANIE SIEŤOVEJ PREVÁDZKY V PROGRAME OPNET MODELER

OPNET Modeler je simulačný balík, vyvíjaný spoločnosťou OPNET Technologies, Inc. Hlavným použitím je zrýchlenie procesu vývoja a výskumu sietí, zariadení, protokolov a aplikácií. Ďalej „možno porovnávať vplyv rôznych technológií, modelovať rôzne typy prevádzky (VoIP, TCP, OSPF, MPLS, IPv6, ...)“ [3]. Sieť je v OPNET Modeler simulovaná diskretným udalostným simulačným jadrom s podporou analytickej simulácie.

OPNET Modeler umožňuje modelovať sieťovú prevádzku rôznymi spôsobmi. Po vytvorení topológie siete je ďalším krokom pridanie prevádzky na sieť. Prevádzku do modelu možno importovať z externých súborov (menu Traffic > Import Traffic), generovať alebo pridať ručne nastavovaním atribútov rôznych objektov.

V OPNET Modeler sú dva druhy sieťovej prevádzky: Explicit a Background Traffic, prípadne prevádzka modelovaná ako kombinácia oboch druhov.

Explicitná, diskretná sieťová prevádzka (Explicit traffic)

Explicitná prevádzka znamená, že je simulovaný každý paket a naň nadväzujúce udalosti ako napr. vytvorenie paketu a čakanie paketu vo fronte, ku ktorým dochádza v priebehu simulácie. Explicitné modelovanie poskytuje presnejšie výsledky simulácií, pretože modeluje všetky vplyvy protokolov na model. Zároveň je doba trvania simulácie dlhšia.

V OPNET Modeler existujú tri všeobecné metódy modelovania explicitnej prevádzky:

- **Simulovanie vytvárania paketov:** Jednotlivým uzlom sú priradené zdroje – generátory všeobecných, podrobnejšie nešpecifikovaných paketov. Takéto pakety majú určené len niektoré parametre, napr. veľkosť paketu v bajtoch a interval medzi vyslaním paketu. Neobsahujú dáta z vyššej ako druhej vrstvy. Sú vhodné najmä na skúmanie technológií druhej vrstvy OSI (Ethernet, ATM, Frame Relay). Príkladom je objekt RPG (Raw Packet Generator) a IP_STATION z palety „Shared“.

- **Modely prevádzky aplikácií (Application traffic models):** Sú konfigurované v objekte Application Configuration z palety utilities (Obr. 0.7). Predstavujú najpoužívanejší spôsob modelovania sieťovej prevádzky. OPNET Modeler obsahuje vzorové nastavenia správania sa niektorých aplikácií, napríklad Database, Email, FTP, Print.
- **Modely požiadaviek aplikácií (Application demands):** Požiadavky aplikácií sú používané na modelovanie potenciálne premenlivých vzťahov medzi uzlami. Vtedy musia byť modelované ako výhradne explicitná prevádzka (Obr. 0.1 a príklad v kapitole 1.1.1). Charakterizujú prevádzku len ako veľkosť a rýchlosť požiadaviek a odpovedí (paketov) medzi dvomi uzlami. Výhodou je, že sa vytvárajú a konfigurujú jednoduchšie ako modely aplikácií prevádzky. Model požiadaviek aplikácie je možné konfigurovať ako výhradne explicitnú prevádzku, zmiešanú explicitnú prevádzku s prevádzkou na pozadí, alebo ako prevádzku na pozadí. Pomer druhov prevádzok je určený percentom v nastaveniach.



Obr. 0.1 Určenie pomeru explicitnej prevádzky a prevádzky na pozadí v modeli požiadaviek aplikácií

Sieťová prevádzka modelovaná na pozadí liniek (Background traffic, analytical traffic)

Tento druh prevádzky ovplyvňuje explicitnú prevádzku pridávaním variabilného oneskorenia, ktoré je spôsobené rôznymi dĺžkami vstupných a výstupných frontov. Na rozdiel od explicitnej prevádzky nie sú simulované jednotlivé pakety, a tak je podstatne zvýšená rýchlosť simulácie. Príklad grafu sieťovej prevádzky modelovanej na pozadí je uvedený v Obr. 0.18.

V OPNET Modeler sú tri druhy takejto prevádzky:

- **Toky prevádzky (Traffic Flows):** Predstavuje nastavenie toku dát modelov požiadaviek aplikácií medzi dvomi uzlami siete cez určené medziľahlé uzly.

-
- **Modely požiadaviek aplikácií (Application Demands):** Môžu predstavovať explicitnú prevádzku alebo prevádzku na pozadí. Podrobnejšie sú popísané v explicitnej prevádzke.
 - **Základné zaťaženie (Baseline Loads):** Tento typ prevádzky predstavuje zaťaženie linky alebo uzlu. Na rozdiel od prevádzkového toku (Traffic Flows) sa viaže na jeden objekt, linku, ako základné vytťaženie linky, alebo zariadenie, ako základné vytťaženie CPU.

Jednotlivé druhy explicitnej a prevádzky modelovanej na pozadí sú podrobnejšie popísané v nasledujúcej kapitole.

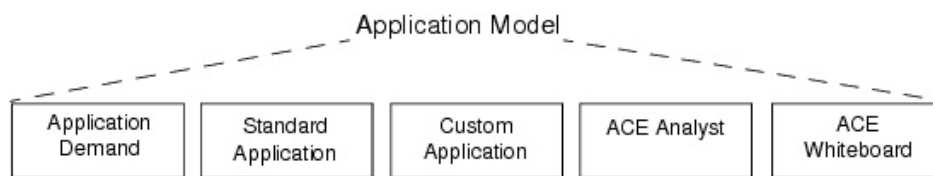
Aplikácie (Applications)

Modely aplikácie sú prevažujúcim zdrojom prevádzky v modelovaných sieťach. Prevádzka generovaná aplikáciami zaťažuje zariadenia a linky siete a vytvára požiadavky na zariadenia. Pre optimálny výkon aplikácie musí byť infraštruktúra siete a serverov navrhnutá tak, aby spĺňali požiadavky na použitie aplikácie. Na tento účel je potrebné najprv vytvoriť presný model aplikácie.

Ak má byť reprezentácia aplikácie presná, model aplikácie by mal mať rovnaké vlastnosti v reálnej prevádzke ako aj v modelovanom príklade. Týka sa to veľkosti generovaných paketov, intenzity odosielania paketov, transportného protokolu pomocou ktorého sú pakety odosielané, počet súčasných aktívnych pripojení, nastavenie časovačov, opätovného prenesenia chybných paketov, správanie pri zlyhaní a obnovení a podobne. Spolu tieto vlastnosti vytvárajú vzor skutočnej prevádzky v modeli. Každá aplikácia má jedinečný vzor prevádzky, alebo popis, a tak tvoria rôzne zaťaženia siete a serverov. Cieľom modelovania aplikácie je čo najvernejšie napodobniť správanie sa cieľenej prevádzky.

Pokiaľ ide o mieru podrobnosti alebo presnosti modelovania sieťovej prevádzky, existujú rôzne spôsoby modelovania. Jeden model môže byť predstavovaný generátorom paketov, ktorý generuje udaný počet paketov za sekundu. Zložitejší spôsob modelu prevádzky môže byť predstavovaný detailným popisom výmeny správ medzi uzlami, vrátane logiky, podľa ktorej sa výmena a parametre správ riadia. OPNET Modeler poskytuje viacero spôsobov ako modelovať prevádzku, od jednoduchých po zložitejšie. Všetky majú svoje výhody aj nevýhody, ktoré môžu výrazne ovplyvniť

správanie siete a rýchlosť simulácie. Preto je dôležité použiť kombináciu vhodných modelov aplikácií.



Obr. 0.2 Spôsoby modelovania prevádzky pomocou modelu aplikácií v OPNET Modeler [7]

1.1.1 Model požiadaviek prevádzky aplikácie (Application Demands)

Modely požiadaviek aplikácie vyžadujú minimálnu konfiguráciu. Je to najrýchlejší spôsob, ako v simulácii vytvoriť prevádzku aplikačnej vrstvy. Modely požiadaviek predstavujú všeobecný model prevádzky typu klient – server. Klient posiela paket so žiadosťou na server a ten na žiadosť odpovedá. Žiadosti a odpovede je možné upraviť napr. zmenou veľkosti paketov alebo intenzity odosielania paketov.

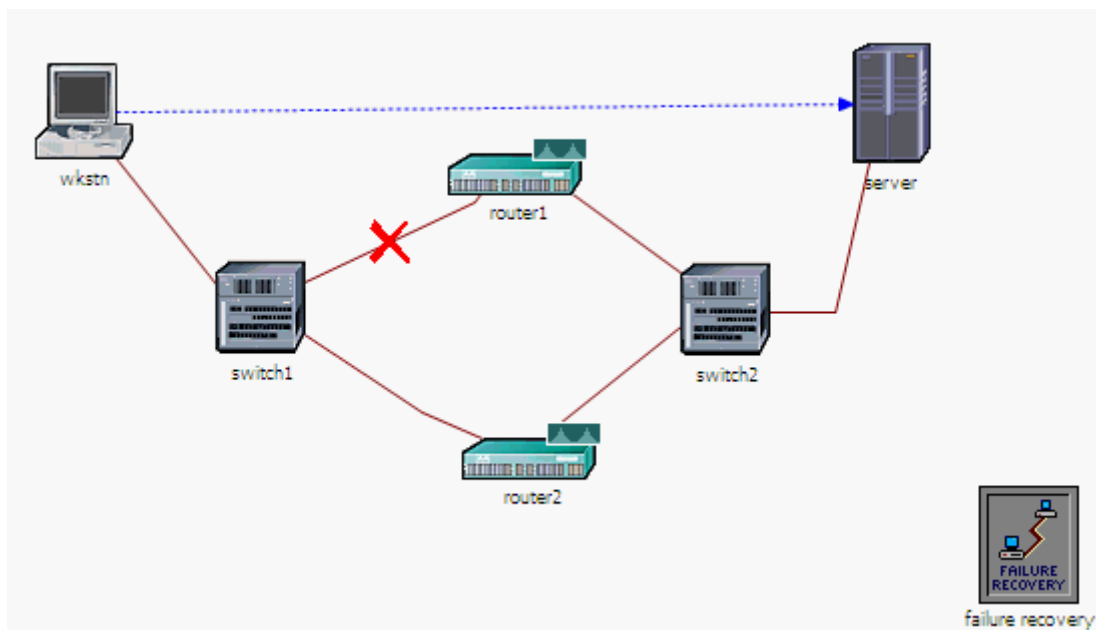
Tento typ modelu je vhodný ako doplnkový typ modelovania prevádzky pri súčasnom použití iných techník modelovania prevádzky. Zároveň je vhodný na skúmanie správania sa uzlov siete s rôznymi smerovacími protokolmi.

Tab. 0.1 Výhody a nevýhody modelu požiadaviek prevádzky

Výhody	Nevýhody
Rýchle nastavenie prevádzky	Prevádzka musí byť typu klient - server
Môžu predstavovať explicitnú prevádzku alebo prevádzku na pozadí (viď kapitola 0)	Nepredstavujú špecifickú prevádzku (napr. FTP)

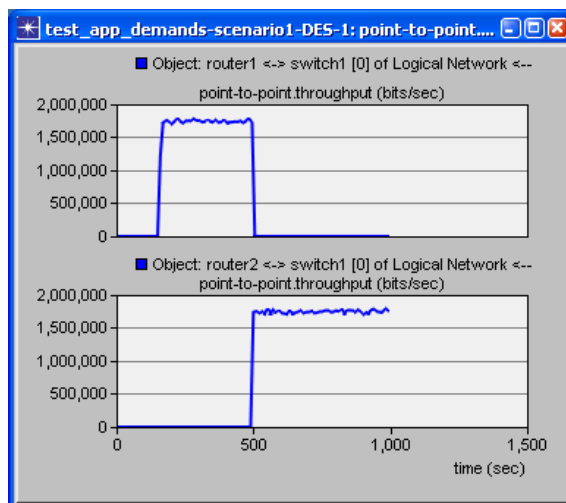
Modely požiadaviek prevádzky je možné vytvoriť použitím menu „Traffic > Create Traffic Flows“ alebo z palety objektov „demands“.

V prílohe A som vytvoril jednoduchý príklad. Je v ňom počítač (wkstn), ktorý posiela IP tok (paleta demands, ip_traffic_flow), ďalej len AD - Application Demand, veľkosti 1,6 Mb/s na server.

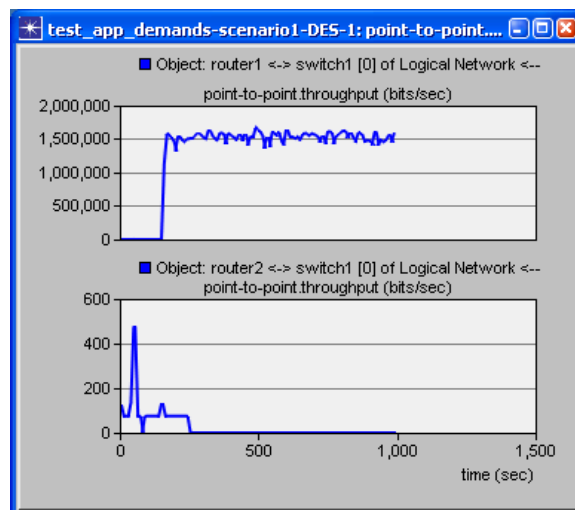


Obr. 0.3 Topológia siete z príkladu. Šípka predstavuje tok dát modelovaný pomocou AD

Kvôli nastavenému zlyhaniu linky medzi smerovačom router1 a prepínačom switch, ku ktorému dôjde v čase 500 sekúnd od začiatku simulácie, je AD nastavený ako 100% explicitný typ prevádzky (Traffic Mix - Obr. 0.1). Ak by som ponechal, aby model požiadavky prevádzky bol modelovaný ako prevádzka na pozadí, vo výsledkoch simulácie pôjde tok aj cez zlyhanú linku. Porovnanie oboch spôsobov modelovania AD je na Obr. 0.4 a Obr. 0.5. Simulácia je nastavená na 1000 sekúnd simulačného času. Pre porovnanie počet udalostí simulačného behu, kedy bol AD modelovaný ako 100%-ný explicitný typ je približne 26 miliónov a trvala približne dve minúty. Naproti tomu simulácia s AD modelovaným ako prevádzka na pozadí vykonala len 46 tisíc udalostí s trvaním menej ako dve sekundy. Príklad na modelovanie zlyhania linky je v kapitole 1.1.12.



Obr. 0.4 AD modelovaný ako 100% explicitný, graf hore, tok dát na linke medzi switch1 a router1, dole medzi switch1 a router2



Obr. 0.5 AD modelovaný ako 100%-ná prevádzka na pozadí, graf hore, tok dát na linke medzi switch1 a router1, dole medzi switch1 a router2, Modelovaný tok napriek očakávaniu nie je ovplyvnený výpadkom linky

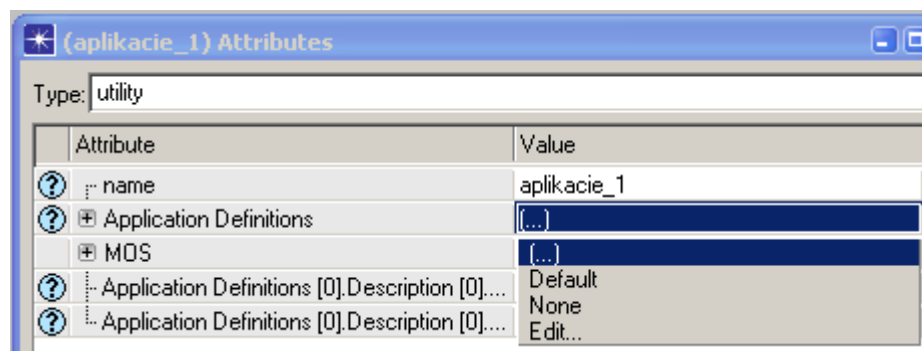
Štandardne je vytvorená prevádzka rozkladaná do viacerých tokov zariadeniami na rozkladanie záťaže (router, prípadne load balancer). Prepnutím nastavenia toku „Flow Type“ na „Microflow“ možno dosiahnuť, že prevádzka týmito zariadeniami nie je ovplyvnená. V príklade žiadne zariadenie nepodporuje funkciu rozkladania záťaže.

1.1.2 Štandardné modely prevádzky

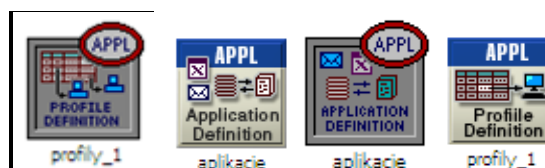
Sú to vstavané modely prevádzky. Výhodou oproti AD je, že model presnejšie zodpovedá reálnemu stavu a oproti ACE sa jednoduchšie implementuje. Obsahuje

modely prevádzky databázových aplikácií, elektronickej pošty, FTP, http, tlačenia, vzdialeného prihlásenia, VoIP a videokonferencií. Každý model prevádzky má špecifické nastavenia. Napríklad v aplikácii HTTP je možné z pohľadu klienta modelovať veľkosti sťahovaných objektov, počet týchto objektov a interval medzi požiadavkami. Štandardné modely prevádzky je vhodné použiť na modelovanie prevádzky typického používateľa, na skúmanie doby odozvy. V dokumentácii OPNET Modeler je uvedený príklad „Tutorial: Small Internetworks“.

Aplikácie sú používané modulmi „Profile Definition“ z palety „utilities“, ktoré predstavujú správanie klienta. OPNET Modeler obsahuje 16 vstavaných nastavení aplikácií. Ich použitie predchádza nastavenie položky „Application Definitions“ na hodnotu „Default“, ako je to v Obr. 0.6. Na jednoduchšiu implementáciu nastavení slúži menu „Protocols > Applications“.



Obr. 0.6 Spôsob nastavenia vstavaných aplikácií



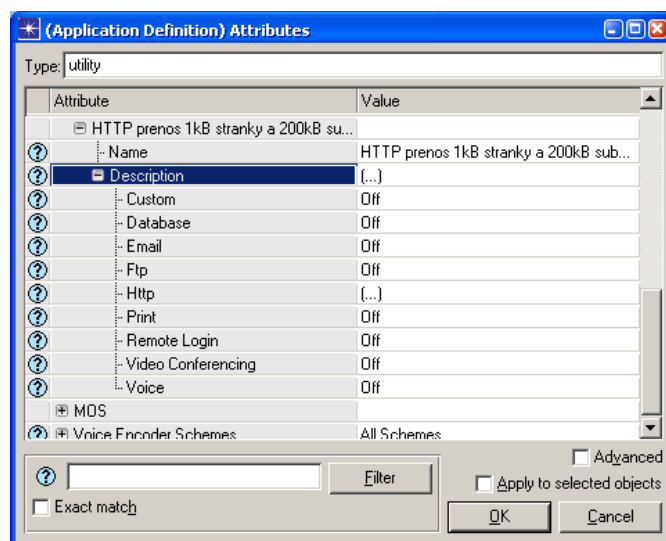
Obr. 0.7 Dve podoby modulov konfigurácie aplikácií a modulu konfigurácie profilov

Tab. 0.2 Výhody a nevýhody štandardných modelov prevádzky

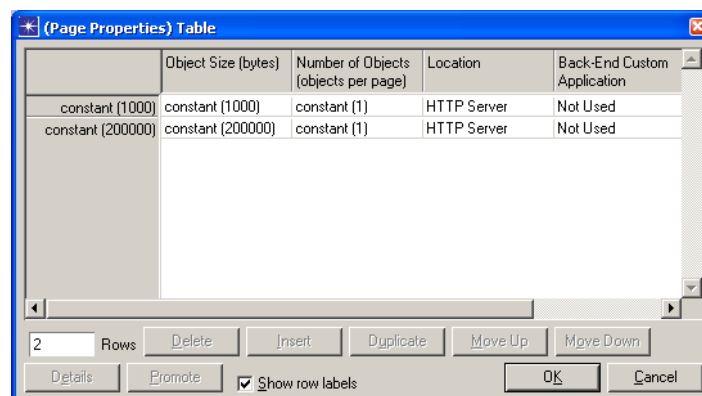
Výhody	Nevýhody
Rýchle vytvorenie modelu správania sa bežných sieťových aplikácií	Nepredstavujú model napodobňujúci špecifické implementácie od rôznych výrobcov

Prednastavené konfigurácie, napr. prezeranie obrázkov	Neposkytujú možnosť modelovať aplikáciu vo viacerých vrstvách, ako napr. klient - web server – databázový server, ale len na úrovni dvoch vrstiev uzlov klient – server
--	---

Obr. 0.8 a Obr. 0.9 zobrazujú príklad nastavenia modelu aplikácií. Vytvoril som aplikáciu s názvom „HTTP prenos 1kB stránky a 200kB suboru“.



Obr. 0.8 Aplikácia "HTTP prenos 1kB stránky a 200kB suboru "



Obr. 0.9 Nastavenie veľkosti stránky a súbežne sťahovaného súboru

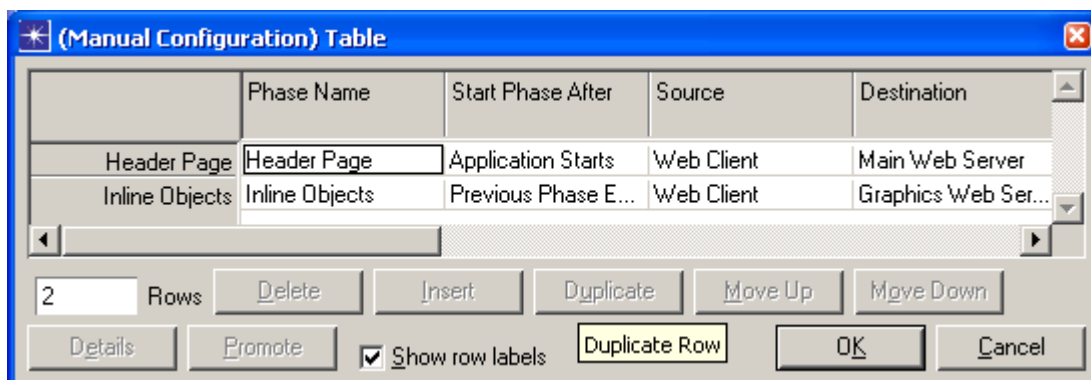
1.1.3 Vlastné modely prevádzky

Umožňujú podrobnejšie modelovať prevádzku ako štandardné modely prevádzky. Aplikácia je delená na úlohy a tie prípadne až na fázy (Obr. 0.11). Úlohy sú zastúpené objektom Tasks (Obr. 0.10) z palety utilities. OPNET Modeler obsahuje príklad Custom_Application_Examples s podrobnejším opisom nastavenia úloh.

Tento spôsob modelovania prevádzky je vhodný pri modelovaní jednoduchšej logiky nadväznosti správ, ale aj ak aplikácia zahŕňa viac spolupracujúcich uzlov.



Obr. 0.10 Modul Tasks z palety utilities



Obr. 0.11 Fázy úlohy

Tab. 0.3 Výhody a nevýhody vlastného modelu aplikácie

Výhody	Nevýhody
Škálovateľný, presné modelovanie prevádzky	Nutnosť nastaviť všetky podrobnosti aplikácie
Podporuje viac klientov a serverov	
Podpora logiky nadväzností správ	

V nasledujúcich dvoch kapitolách 1.1.4 a 1.1.5 popisujem ďalšie dve možnosti ako okrem použitia fáz úlohy modelovať sieťovú prevádzku. OPNET Modeler obsahuje

modul ACE, ktorý slúži na modelovanie prevádzky na úrovni zasielania jednotlivých paketov, pričom poradie správ je určené používateľom určenou logikou, alebo zachytenou komunikáciou reálnej sieťovej prevádzky.

1.1.4 Modely prevádzky získané zachytením reálnej prevádzky

Okrem nahradenia fáz v úlohách, prostredie ACE poskytuje aj iné pokročilejšie funkcie, napr. prognózu oneskorenia na linkách s rôznou priepustnosťou bez použitia diskretnej simulácie. ACE Analyst je vhodný na vytvorenie presného modelu prevádzky z reálnej prevádzky, bez podrobného skúmania.

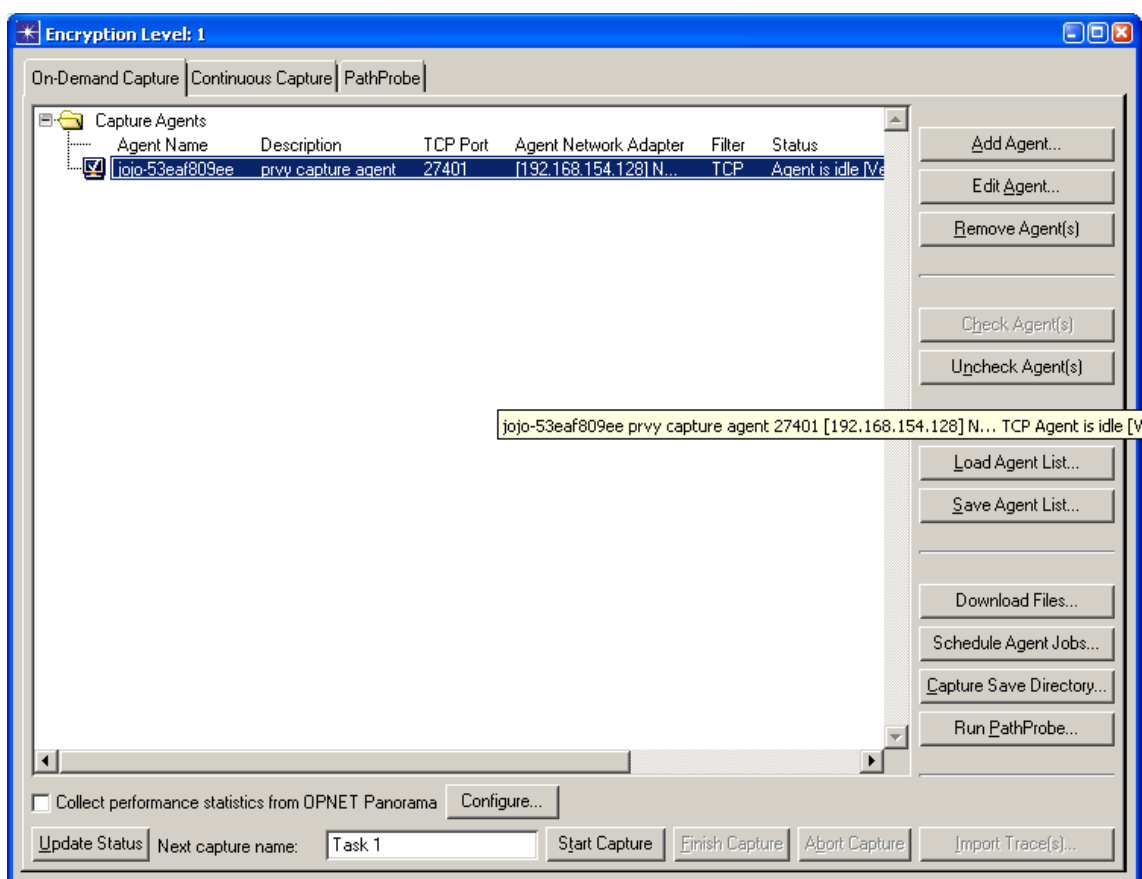
Tab. 0.4 Výhody a nevýhody modelovania prevádzky pomocou zachytenej prevádzky

Výhody	Nevýhody
Verne modeluje sieťovú prevádzku aplikácie	Potreba licencie
Jednoduchá konfigurácia	Len na simulovanie existujúcej aplikácie
	Model aplikácie je vždy ten istý, hoci v reálnom prostredí sa parametre menia

Modely prevádzky sú v module ACE Analyst vytvárané transformovaním zachytenej prevádzky siete. Na prácu s ACE Analyst je potrebná licencia a funkčný Wireshark. [8] Odporúčaná verzia je 0.98. Mne správne pracovala aj verzia 1.26. Na zachytenie prevádzky je potrebná knižnica Pcap. Keďže som pracoval v OS Windows, použil som implementáciu WinPcap, ktorá slúži na zachytávanie paketov v pôvodnej forme obídením funkčnosti protokolov. Okrem toho podporuje filtrovanie a zber štatistík. WinPcap pozostáva z ovládača, ktorý rozširuje možnosti OS a poskytuje priamy prístup k nižším sieťovým vrstvám. WinPcap ako API k sieťovej karte využíva NDIS. [9] Okrem načítania prevádzky zachytenej s použitím Wireshark existuje aj možnosť nainštalovať na počítače produkt ACE Capture Agent, ktorý ma podobnú funkciu. Navyše podporuje funkciu centrálnej správy nainštalovaných agentov. (Hlavné okno OPNET Modeler > File > Application Capture Manager). Na Obr. 0.12 je zobrazené ovládanie ACE Capture Manager s nastaveným jediným agentom, pripraveným na začatie zachytávania. Okrem ukladania prijatej prevádzky pre potreby ACE Analyst je ACE Capture Agent schopný priamo odosielať informácie o prevádzke

do ACE Live, ktoré slúži na meranie odoziev transakcií. Pomáha identifikovať problémy s výkonom aplikácií v reálnom čase.

Na nasledovnom Obr. 0.12 je v Application Capture Manager nastavený jeden agent. Stlačením tlačidla Start Capture dôjde k monitorovaniu prevádzky na určenej sieťovej karte. Po ukončení zachytávania prevádzky je súbor s prevádzkou uložený na vzdialenom počítači. Skopírovanie súboru na lokálny počítač je zabezpečené stlačením Download Files. O interpretáciu uloženého binárneho súboru do ACE Analyst sa stará Wireshark. Pred transformovaním binárneho súboru je potrebné nastaviť v globálnej konfigurácii hodnotu atribútu `ace_external_decoder_prog` na cestu k Wireshark, napr. `C:\Program Files\wireshark\wireshark.exe` a zároveň `ace_use_external_decoder` na hodnotu 1.



Obr. 0.12 Application Capture Manager

Implementovať zachytenú prevádzku do modelovanej siete je možné manuálne nastavovaním parametrov zariadení, a objektov, alebo pomocou menu `Protocols >`

Applications > Deploy ACE Analyst Application on Existing Network > as Discrete Traffic.

1.1.5 Modely prevádzky vytvorené nastavením správ (ACE Whiteboard)

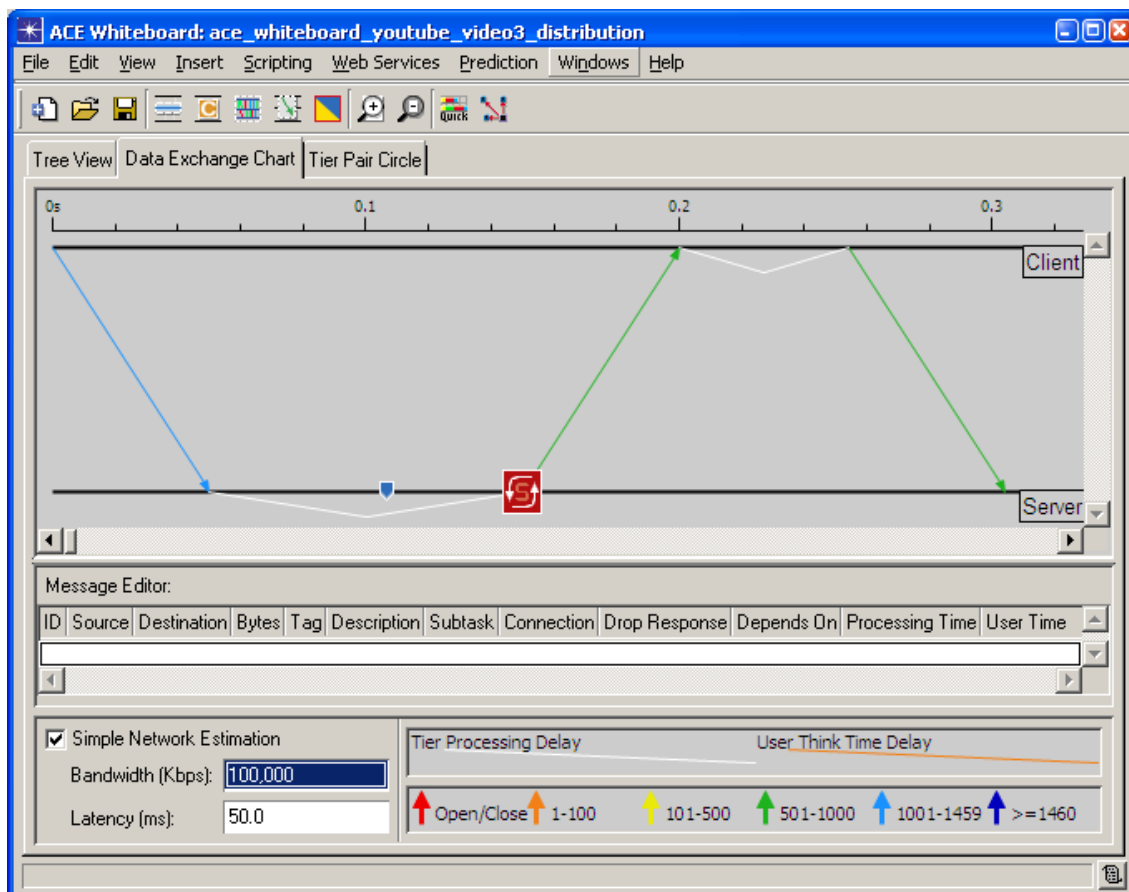
Predstavuje oproti modelu prevádzky získanej zachytením možnosť vytvoriť si vlastný podrobný model prevádzky tvorený správami medzi účastníkmi komunikácie, ktorý môže byť navyše ovládaný skriptom napísanom v programovacom jazyku Python. Pri tomto spôsobe vytvorenia modelu je navyše možné určiť veľkosť a čas odosielaných správ, čas spracovania správy v uzle a vytvoriť bloky správ, ktoré duplikujú odoslanú a prijatú správu ako dvojicu správ.

Tab. 0.5 Výhody a nevýhody modelovania prevádzky pomocou ACE Whiteboard

Výhody	Nevýhody
Zložitejšia logika nadväznosti správ pomocou skriptov v jazyku Python	Vykonávanie modelu aplikácie obsahujúcej Python skript môže spomaliť vykonávanie simulácie
Možnosť upraviť v ACE Whiteboard aj súbor z ACE Analyst	

V nasledovnom príklade uvádzam vytvorenie modelu prevádzky pomocou ACE Whiteboard. Prevádzka je modelovaná ako prenesenie určeného počtu súborov s veľkosťou generovanou náhodne ako číslo z rovnomerného rozdelenia, zo servera na klienta. V príklade sú vynechané správy otvorenia a ukončenia spojenia.

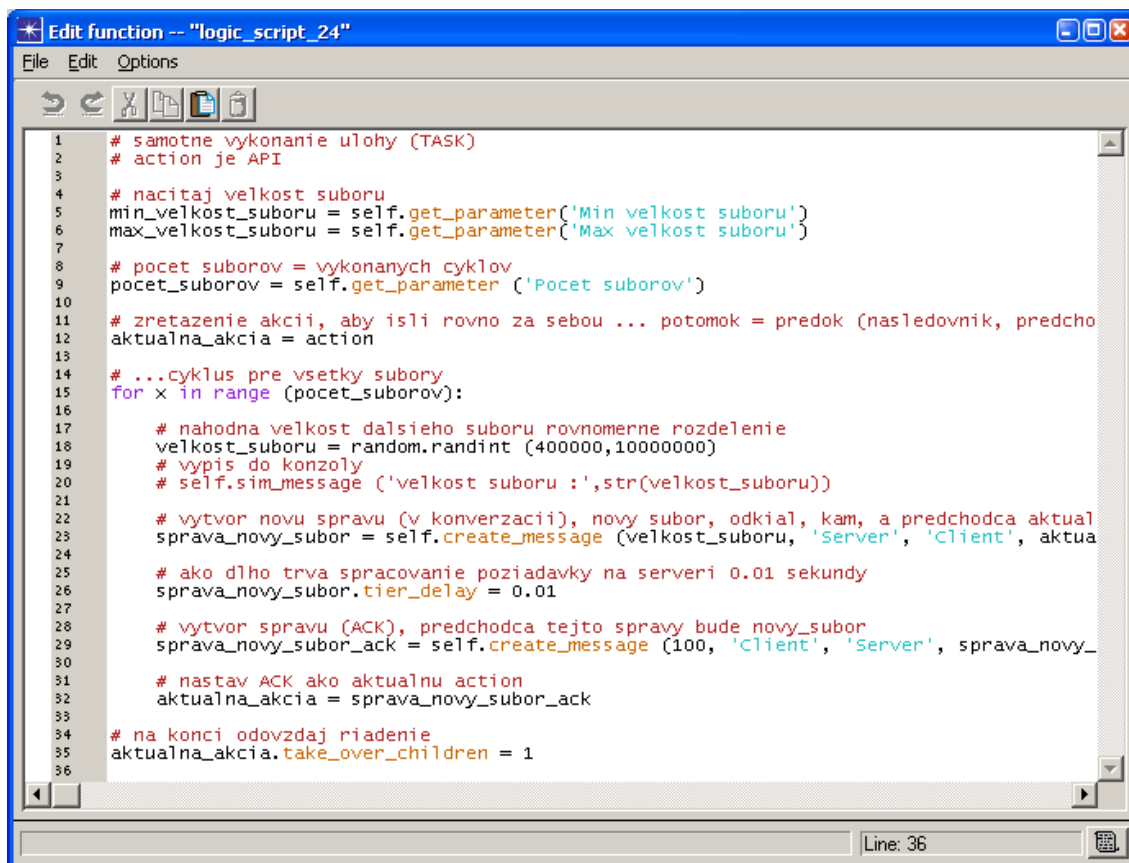
Vytvoril som nový prázdny súbor typu ACE Whiteboard. S pomocou sprievodcu som pridal dve komunikačné strany s názvom Client a Server. Názvy sú dôležité, lebo sú používané v nastavení simulácie. Zobrazia sa dve pomenované vodorovné osi a jedna os s časom v sekundách. Pomocou menu Insert > Message som vytvoril sériu troch správ, ako v Obr. 0.13. Parametre správy sa nastavujú v Message Editor. Modrá správa s veľkosťou 1116 bajtov predstavuje úvodnú požiadavku klienta na server. Veľkosť správy som určil pozorovaním požiadaviek na stiahnutie súboru. Biele lomené čiary predstavujú dĺžku spracovania správy v uzle. Takisto aj pomenovanie správ je dôležité pri využívaní skriptov. V mojom príklade nie. Ďalšie dve správy predstavujú odosielanie súboru a potvrdenie prijatia súboru. Veľkosť oboch správ je 1000 bajtov.



Obr. 0.13 Grafické rozhranie ACE Whiteboard

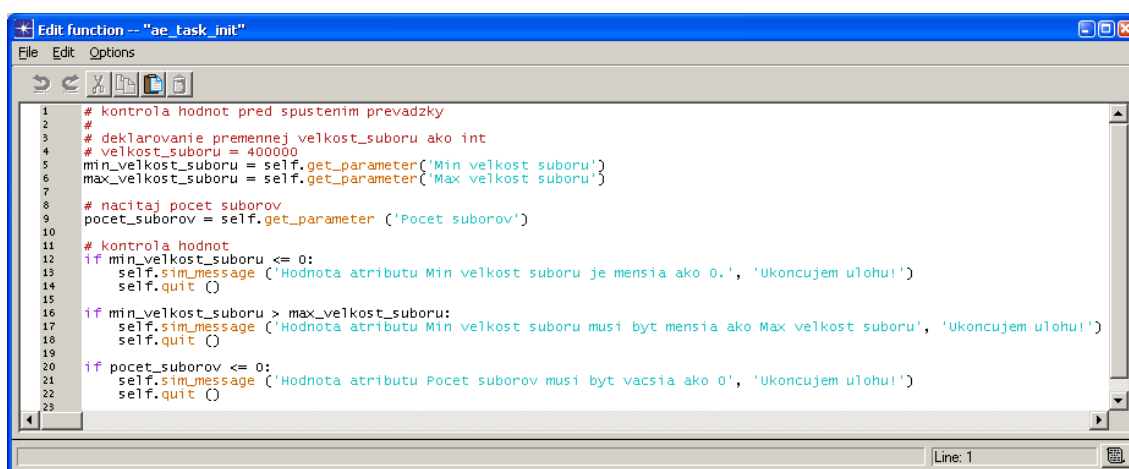
V ďalšom kroku som vytvoril a upravil skript, ktorý zabezpečí odosielanie určeného počtu súborov, ktoré budú predstavované správami. Po označení prvej zelenej šípky a vybratím voľby Add Logic Script at Start z menu pravého tlačidla sa zobrazí nový skript.

Naprogramoval som jednoduchú logiku odosielania správ (Obr. 0.14). V cykle, ktorý sa opakuje určený počet krát sú vytvárané správy s náhodnou veľkosťou. Doba spracovania správy v uzle má konštantnú hodnotu 0,01 sekundy. Riadky s komentármi začínajú znakom #. Slovo action je vyhradené a zabezpečuje nadväznosť správ. Príkaz get_parameter slúži na získanie hodnoty definovaného parametra.



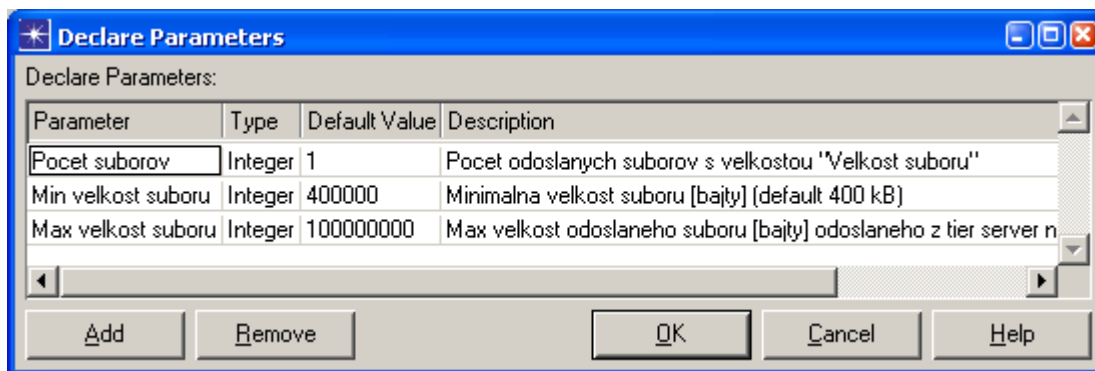
Obr. 0.14 Skript logiky odosielania správ v Python

Okrem tohto skriptu je nutné hlavičke (Scripting > Header Block) definovať použitú knižnicu `random` s generátorom pseudonáhodných čísel príkazom `import random`. V inicializačnom bloku (Scripting > Initialization Block) načítať a ošetriť parametre (Obr. 0.15).



Obr. 0.15 Inicializačný blok

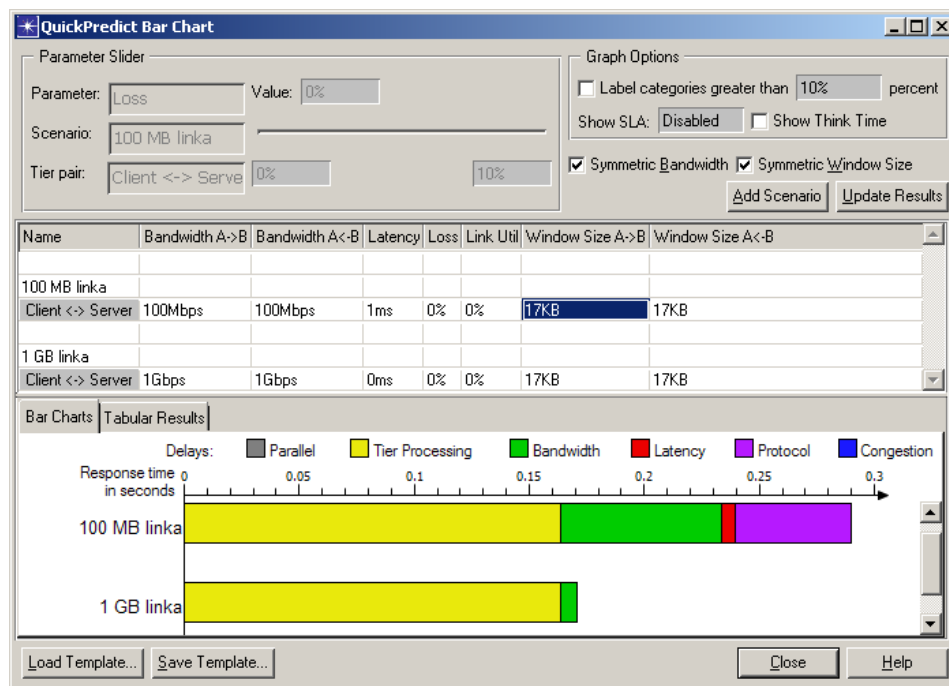
V neposlednom rade je potrebné deklarovať parametre, ktoré budú odovzdávané cez nastavenie objektu Tasks (Scripting > Declare Parameters). Hodnoty parametrov sú v skripte načítané cez funkciu `get_parameter('nazov deklarovaného parametra')`.



Obr. 0.16 Deklarácia parametrov

Podrobnosti k skriptom sú rozobrané v dokumentácii OPNET Modeler: Modules > Ace Analyst > ACE Scripts for Filtering, Analyzing, and Visualizing. Texty skriptov sú v prílohách E, F, G. Uvedený príklad ACE modelu je v súbore `ace_whiteboard_model_prenosu_suborov.aed` v prílohe DVD.

Prostredie ACE umožňuje vytvorený model voľbou Prediction z menu vyskúšať pomocou diskkrétnej simulácie, alebo odhadnúť zdroje zmeny dĺžky odozvy v rôznych prostrediach pomocou nástroja QuickPredict. Na nastavenie ACE modelu uzlom v sieti v projekte je vhodné použiť objekty z palety Shared Objects Palettes > ACE a menu Protocols > Applications. Príklad k implementácii ACE modelu prevádzky v sieti je súčasťou praktickej časti diplomovej práce, v prílohe na DVD, súbor `uniza_core.opcfa` projektu siete a v scenári `scenario 50min ACE Whiteboard`, prípadne `scenario 6MB subor ACE Capture`.



Obr. 0.17 Nástroj QuickPredict

Toky sieťovej prevádzky (Traffic flows, Background routed traffic)

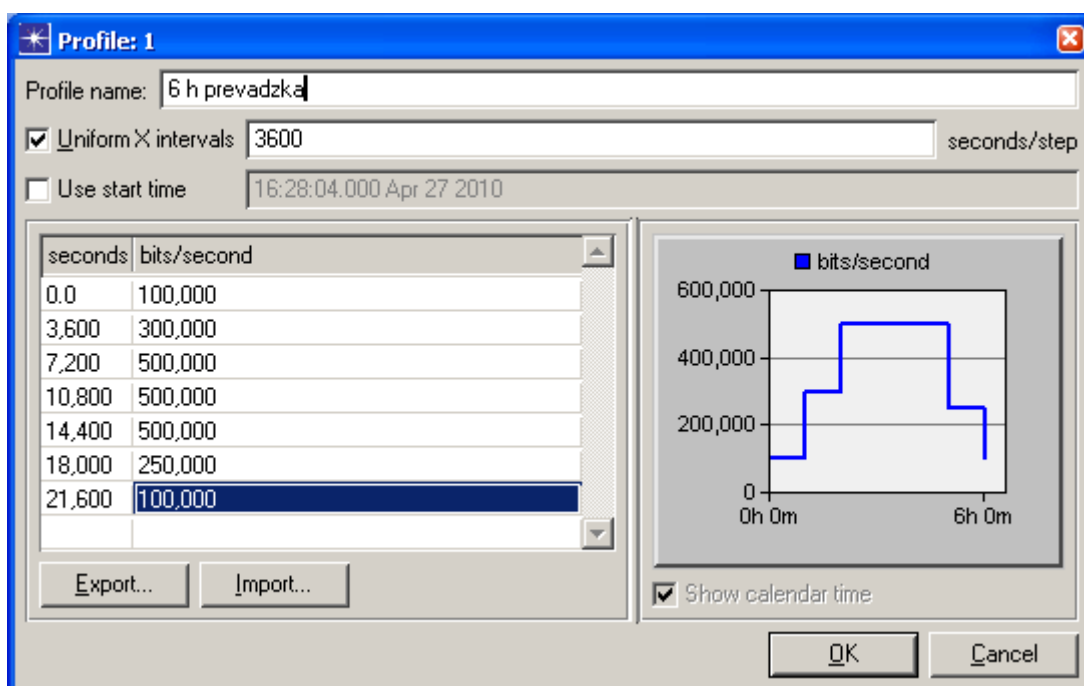
Toky sieťovej prevádzky, ďalej len TF. Predstavujú spojenie AD a základného zaťaženia linky (baseline load). Na rozdiel od základného zaťaženia, ktoré sa viaže len na jeden objekt, TF prechádza sieťou od zdroja k cieľu cez linky a uzly, ktoré ovplyvňuje. TF môže mať pomocou objektu path z palety definovanú presnú cestu. Uvedená možnosť má význam napr. pri MPLS.

Základné zaťaženie (Baseline Loads)

Predstavuje záťaž, ktorá je aplikovaná len na jeden objekt – linku, alebo zariadenie. Základné zaťaženie aplikované na linku nemá v diskkrétnej simulácii vplyv na využívanie systémových prostriedkov modelovaného zariadenia k linke pripojeného.

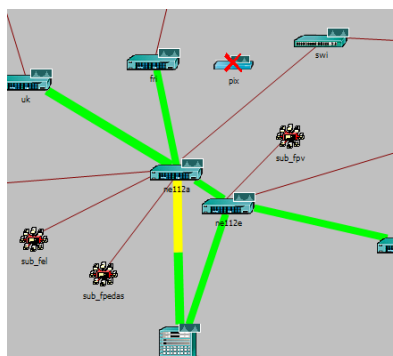
Základné zaťaženie linky sa nastavuje vo vlastnostiach linky Traffic Information, resp. v Traffic Center. Zaťaženie CPU zariadenia vo vlastnostiach zariadenia > CPU > CPU Background Utilization. Na určenie zaťaženia linky, alebo CPU sú používané vlastné, prípadne štandardne vytvorené profily, ktoré predstavujú časový priebeh vytťaženia.

Modelovanie prevádzky pomocou základného zaťaženia môže zrýchliť simuláciu.



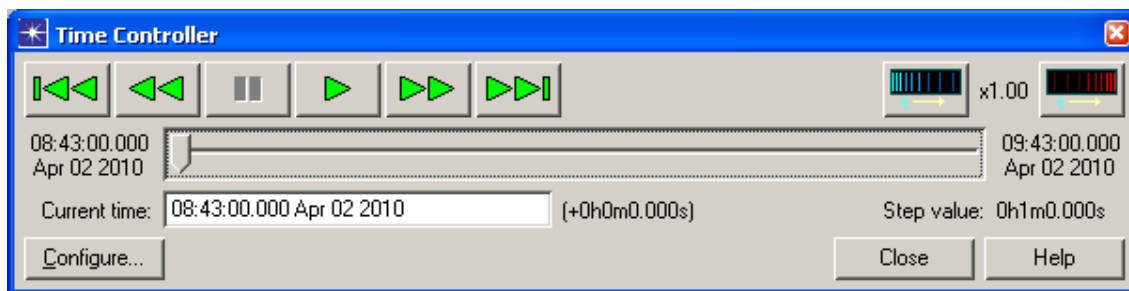
Obr. 0.18 Modelovanie priebehu zaťaženia linky pomocou modelu základného zaťaženia

Najvyššiu, alebo priemernú záťaž liniek v čase, počas ktorého záťaže trvajú je možné zobrazit' v topológii pomocou menu View > Visualize Link Usage > Color by Link Usage > Baseline Link Utilization.



Obr. 0.19 Zobrazenie základného zaťaženia liniek v aktuálnom čase

Rôzne farby predstavujú rôzne percentuálne vyt'azenie liniek v oboch smeroch, hrúbka čiar objem dát prenesených za sekundu. Čas, v ktorom je vyt'azenie zobrazované je možné menit' nástrojom „Time Controller“ (View > Show Time Controller).



Obr. 0.20 Nástroj Time Controller

Ak sú v profiloch priebehu zaťaženia linky alebo modeloch požiadaviek aplikácií (Obr. 0.18) a v simulácií (Time Controller) nastavené dátumy a časy spustenia profilov a tie majú byť spustené v rovnakom čase ako začiatok simulácie je dôležité, aby bol rovnaký čas nastavený aj v nástroji Time Controller (View > Show Time Controller > Configure > Start modeling at time).

Na správu prevádzky modelovanej na pozadí slúži „Traffic Center“ (Traffic > Open Traffic Center).

METÓDY ZRÝCHLENIA SIMULÁCIE

Nie vždy je simulačný beh vykonaný dostatočne rýchlo. Rýchlosť simulácie je ovplyvnená rýchlosťou hardvéru. Vykonaním uvádzaných opatrení sa mi podarilo zrýchliť simuláciu.

Paralelná simulácia

Pod pojmom paralelná simulácia uvádzam koncept vykonávania simulačného behu na viacerých procesoroch, resp. procesorových jadrách súčasne, pričom výsledky simulácie sú rovnaké ako pri vykonávaní na jednom jadre. Výhodou oproti sekvenčnému vykonávaniu je zvýšenie rýchlosti simulácie. Pre paralelné vykonávanie simulácie je nutná podpora simulačného jadra a kód optimalizovaný pre vykonávanie paralelnej simulácie. Príkladmi, kedy paralelná simulácia profituje z paralelizmu sú udalosti, pri ktorých doba výpočtu je vyššia a udalosti, ktoré sa vykonávajú v iných častiach siete a navzájom sa neovplyvňujú.

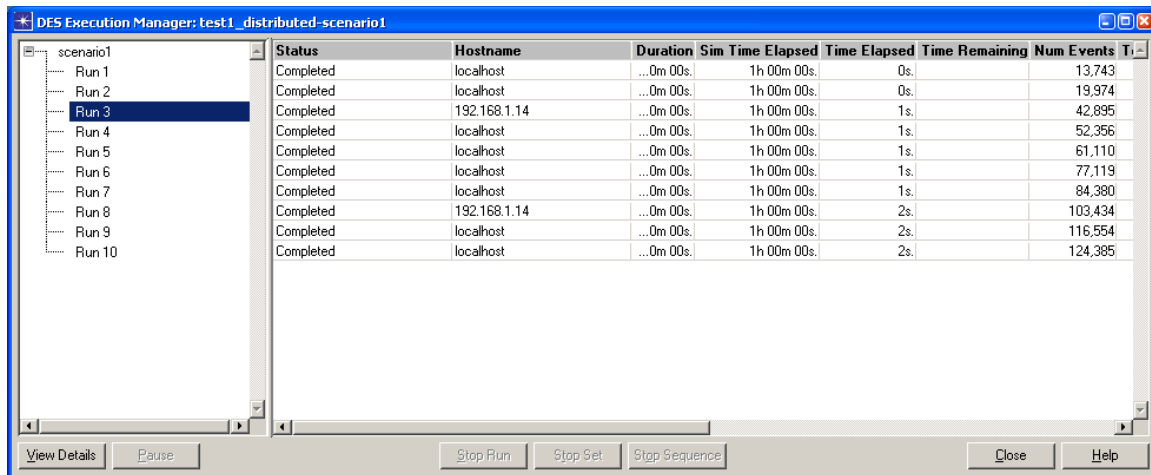
Nastavenie výhľadu [4], resp. okna (množstva udalostí, ktoré sa môžu vykonávať samostatne) a iných položiek je možné v nastavení simulačného behu Execution > Advanced > Kernel Preferences. Na sledovanie zmeny rýchlosti paralelnej simulácie oproti sekvenčnej je nutné zapnúť Potential execution speedup (Execution > Advanced > Kernel Reports) a sekvenčné vývojové simulačné jadro. Dôjde k vygenerovaniu tabuľky zobrazujúcej odhad zrýchlenia paralelnej simulácie oproti sekvenčnej. Po vykonaní sekvenčnej simulácie je možné zmeniť typ simulačného jadra na paralelné a porovnať rýchlosti.

Distribuovaná simulácia

OPNET Modeler dokáže využiť viac jadier procesora nielen na paralelnú simuláciu popísanú vyššie, ale aj na vykonanie viacero simulačných behov s rôznymi parametrami v rovnakom časovom okamihu na viacerých počítačoch. Distribuovaná simulácia môže byť použitá aj na spustenie simulácie na výkonnejšom vzdialenom počítači.

Keď sa vykonáva distribuovaná simulácia, OPNET Modeler spusti určené množstvo simulácií na každom uvedenom počítači. Po skončení simulácie na procesore je využitý na simulovanie ďalšej simulácie, pokiaľ nie sú všetky simulácie ukončené.

Priebeh jednotlivých simulácií je zobrazovaný v okne DES Execution Manager podobne, tak ako v Obr. 0.1.



The screenshot shows the DES Execution Manager window with the title 'DES Execution Manager: test1_distributed-scenario1'. On the left, a tree view shows 'scenario1' expanded, listing 'Run 1' through 'Run 10', with 'Run 3' selected. The main area displays a table of simulation results.

Status	Hostname	Duration	Sim Time Elapsed	Time Elapsed	Time Remaining	Num Events	T...
Completed	localhost	...0m 00s.	1h 00m 00s.	0s.		13,743	
Completed	localhost	...0m 00s.	1h 00m 00s.	0s.		19,974	
Completed	192.168.1.14	...0m 00s.	1h 00m 00s.	1s.		42,895	
Completed	localhost	...0m 00s.	1h 00m 00s.	1s.		52,356	
Completed	localhost	...0m 00s.	1h 00m 00s.	1s.		61,110	
Completed	localhost	...0m 00s.	1h 00m 00s.	1s.		77,119	
Completed	localhost	...0m 00s.	1h 00m 00s.	1s.		84,380	
Completed	192.168.1.14	...0m 00s.	1h 00m 00s.	2s.		103,434	
Completed	localhost	...0m 00s.	1h 00m 00s.	2s.		116,554	
Completed	localhost	...0m 00s.	1h 00m 00s.	2s.		124,385	

At the bottom, there are buttons: 'View Details', 'Pause', 'Stop Run', 'Stop Set', 'Stop Sequence', 'Close', and 'Help'.

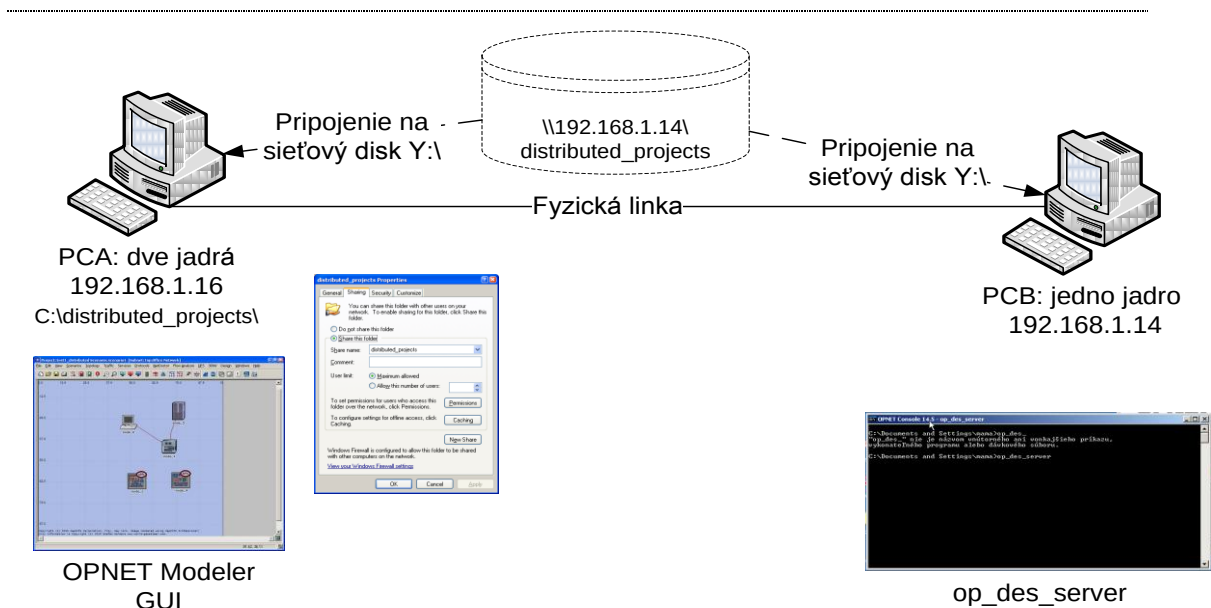
Obr. 0.1 Priebeh jednotlivých simulácií

Na vykonanie distribuovanej simulácie je nutné splniť niektoré predpoklady:

- Je nutné, aby model, resp. projekt bol umiestnený v tom istom zdieľanom sieťovom priečinku. Vhodné je priečinok pripojiť ako sieťový disk s úplnými právami na prístup a zapisovanie. Tento priečinok musí byť rovnako nastavený v mod_dirs vo všetkých počítačoch.
- Ostatné priečinky v nastavení mod_dirs môžu mať rôzne umiestnenie.
- Výsledky sú ukladané do zdieľaného sieťového priečinku.
- Každému využívanému procesoru zodpovedá platná licencia.

Uvádzam príklad nastavenia. V nasledovnom príklade distribuovanej simulácie je nastavené jednoduché prostredie typu klient-server. Budem spúšťať simuláciu s rôznym počtom klientov aplikácie 10-krát. Použité sú predvolené aplikácie a profiley.

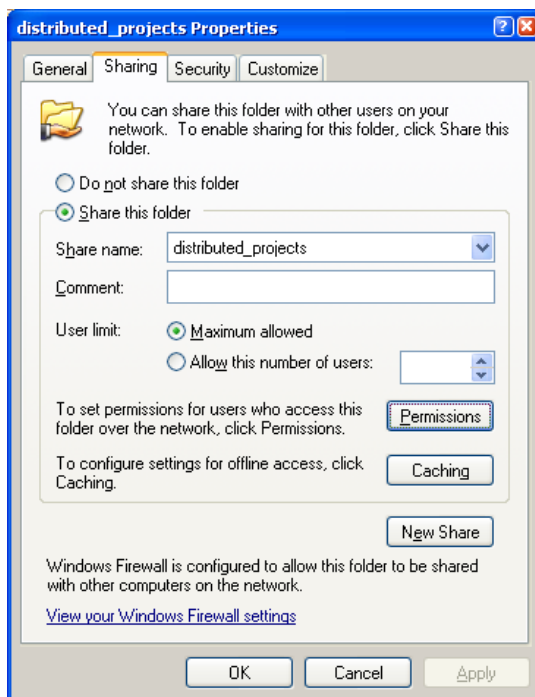
Distribuovanú simuláciu som pustil na dvoch počítačoch s OS Windows XP. Počítač s IP adresou 192.168.1.16 (localhost) v Obr. 0.2 obsahuje dvojjadrový procesor. Počítač slúži na normálnu prácu a vytváranie modelov.



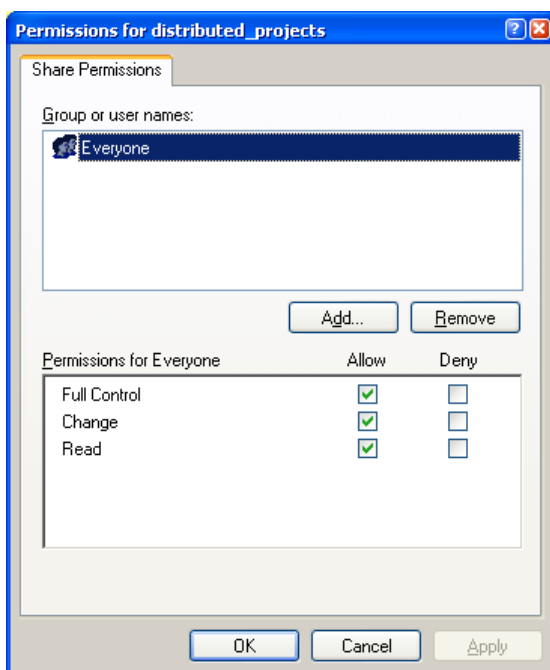
Obr. 0.2 Topológia siete s distribuovanou simuláciou

Na tomto som vytvoril zdieľaný priečinok `c:\distributed_projects\`, do ktorého som uložil model `test1_distributed`. Priečinok má nastavené zdieľanie a prístupové práva je to zobrazené v Obr. 0.3, Obr. 0.4 a Obr. 0.5. Tento priečinok som na oboch počítačoch pripojil ako sieťový disk Y:\ príkazom:

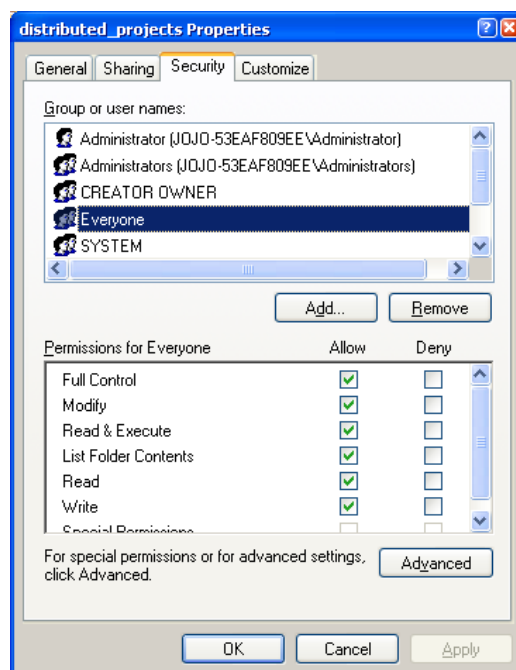
```
net use Y: \\192.168.1.16\distributed_projects /persistent:YES
```



Obr. 0.3 Nastavenie zdieľania prístupu k priečinku



Obr. 0.4 Pridelenie prístupových práv k priečinku pre všetkých používateľov



Obr. 0.5 Nastavenie dodatočných práv pre všetkých používateľov

Súbor test1_distributed.opcf s projektom sa nachádza v DVD prílohe. Je zabalený. Postup rozbalenia je nasledovný. V menu File > Manage Model Files > Expand Project File Archive do C:\distributed_projects\, resp. Y:\ a otvoríme v PCA z disku Y:\.

Na oboch počítačoch som zmenil položku mod_dirs a pridal som na druhé miesto priečinok Y:\.

V počítači PCA som zmenil hodnoty parametrov Allow Simulation Runs on Multiple Hosts na TRUE a do hodnoty atribútu Distributed Simulation Hosts som pridal dve položky localhost:7007:2, 192.168.1.14:7007:1. Parametre predstavujú adresu počítača, port a počet použitých procesorov na simulácie.

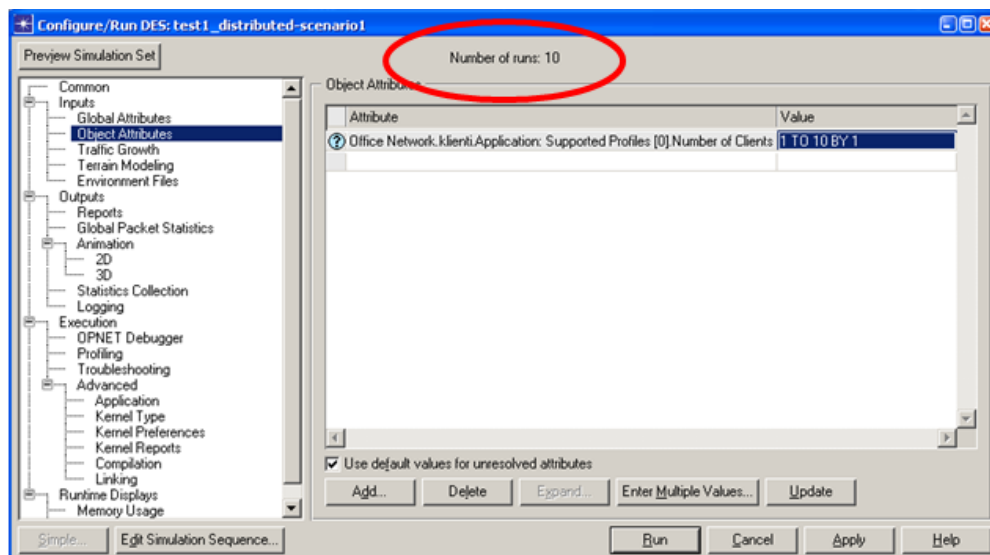
Druhý počítač PCB s IP adresou 192.168.1.14 má procesor s jedným jadrom. Výkon jednotlivých jadier je porovnateľný. Na ňom je spustený príkazový riadok OPNET Console a simulačný server op_des_server.exe.

Na oboch počítačoch je nainštalovaný samostatne použiteľný OPNET Modeler.

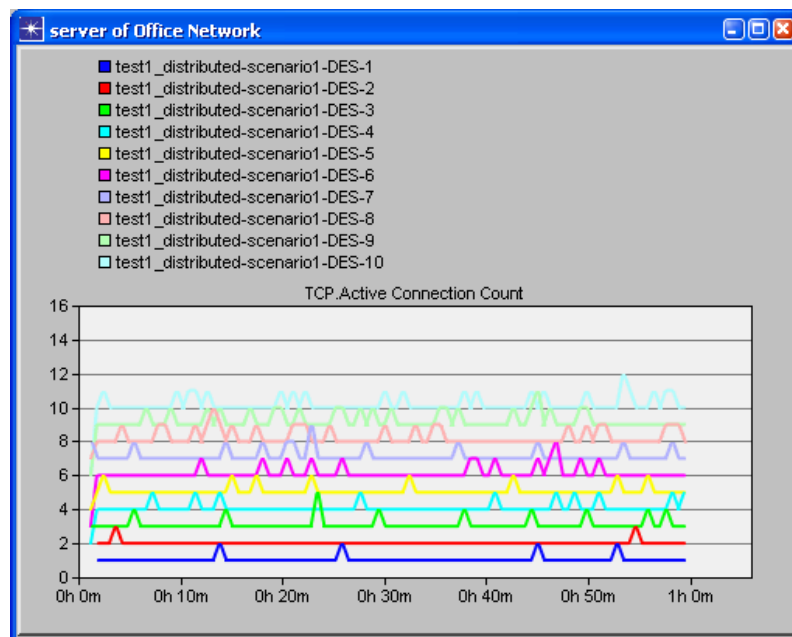
V projekte otvorenom na počítači PCA som zmenil atribút klienta Applications > Application: Supported Profiles > Engineer > Number of Clients na promoted. To mi umožní meniť počet klientov pri každej simulácii.

V nastavení simulácie (DES > Configure/Run Discrete Event Simulation) nastavíme vstupy pre jednotlivé simulačné behy a to v menu Inputs > Object Attributes pridaním atribútu Office Network.klienti.Application: Supported Profiles [0].Number of Clients a zadaním viacerých hodnôt počtu klientov od 1 až po 10 s krokom 1 (Obr. 0.6).

Počet simulačných behov sa zmenil z jedného na desať, ako v Obr. 0.6. Tie budú postupne rozkladané na jednotlivé procesory v počítačoch. V Obr. 0.7 je zobrazená závislosť počtu aktívnych pripojení od počtu pripojených klientov pre jednotlivé simulačné behy s rôznym počtom klientov.



Obr. 0.6 Nastavenie parametrov simulácie

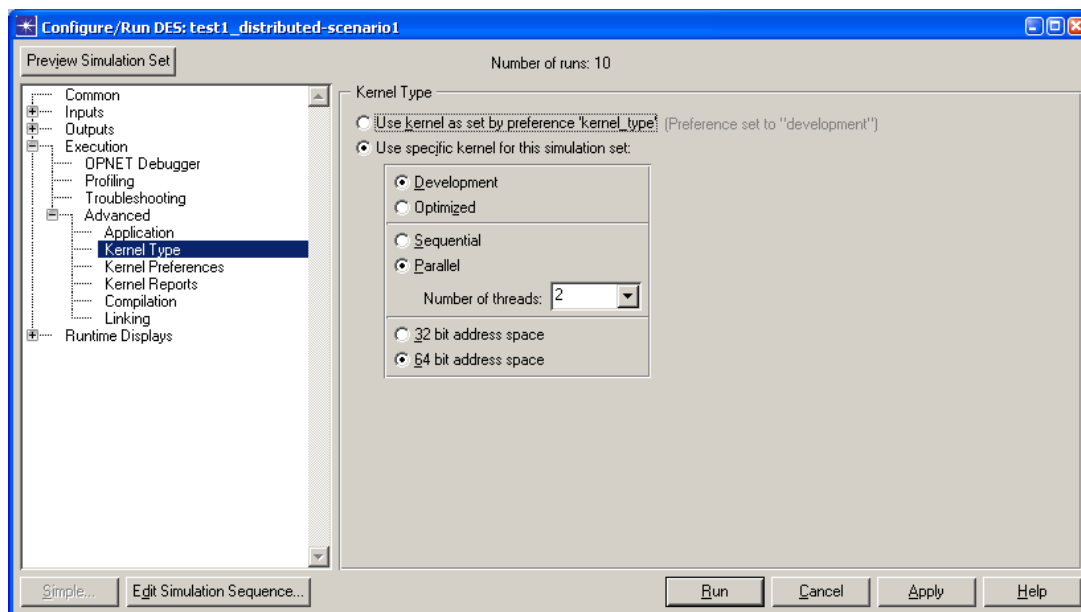


Obr. 0.7 Výsledky simulácie. Počet aktívnych pripojení (os y) protokolu TCP v závislosti od času (os x) a počtu počítačov v sieti (farebne odlišené grafy)

Použitie 64-bitového simulačného jadra

Využitím väčšieho adresného priestoru je možné zrýchliť sekvenčné aj paralelné vykonávanie simulácie. Pri použití 64-bitového simulačného jadra OPNET je potrebné zvážiť rozdiely v implementácii jeho interných dátových typov na OS Windows a Linux. Pri používaní 64-bitového simulačného jadra je nutné mať 64-bitový OS,

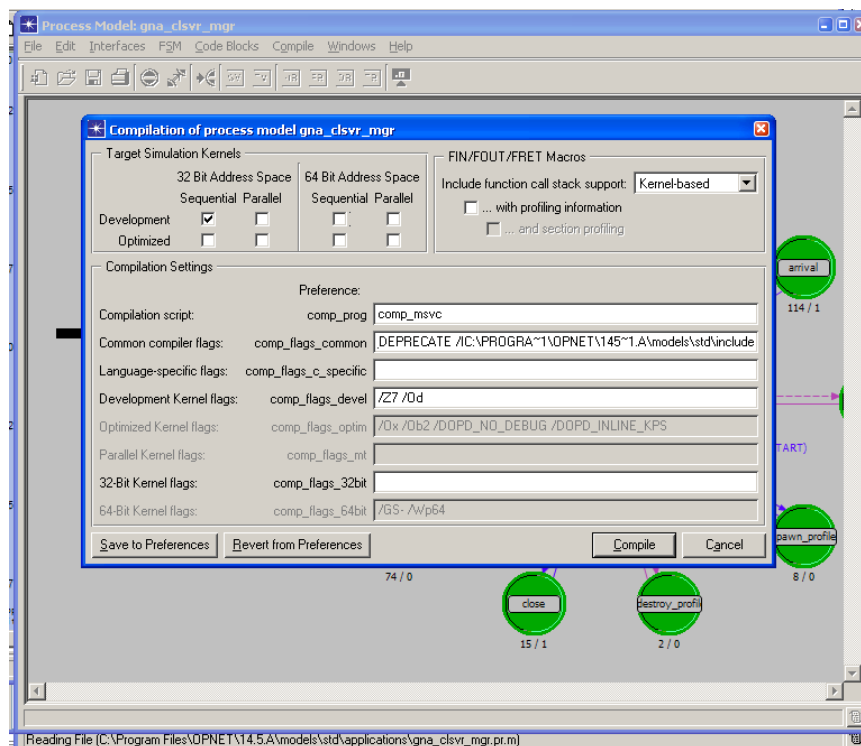
kompilátory a v neposlednom rade aj podporujúci hardvér. Zapnúť 64-bitovú simuláciu je možné trvalo v nastaveniach, zmenením položky Compilation Flags for 64-Bit Code na hodnotu /GS- /Wp64. Systém potom automaticky vyberá medzi 32-bitovým a 64-bitovým jadrom. Vynútenie použitia 64-bitového jadra je možné nastavením položky Kernel Type. Vid' nasledujúci obrázok (Obr. 0.8).



Obr. 0.8 Príklad zapnutia 64-bitovej a zároveň paralelnej simulácie spustenej v dvoch vláknach

Ak jednotlivé moduly neboli kompilované do 64-bitového adresného priestoru, dôjde ku kompilácii zdrojových kódov jednotlivých modulov. Dôvodom je, že tieto moduly sú štandardne kompilované len ako 32-bitové a sekvenčné. Kompilovanie modulov je možné zmeniť hodnotou jednotlivých položiek v globálnej konfigurácii > Discrete Event Simulation > Code Generation > Compilation > ... Compile for 32-Bit, Development, Parallel Kernel; Compile for 32-Bit, Development, Parallel Kernel; Compile for 32-Bit, Development, Parallel Kernel; Compile for 32-Bit, Development, Parallel Kernel; Compile for 64-Bit, Development, Parallel Kernel; Compile for 64-Bit, Development, Sequential Kernel; Compile for 64-Bit, Development, Sequential Kernel; Compile for 64-Bit, Optimized, Sequential Kernel na TRUE.

Pri vytváraní modulov môžeme nastaviť parametre kompilácie aj zmenou v menu Compile > Compile Code Advanced (Obr. 0.9).



Obr. 0.9 Parametre kompilácie modulu

Zmena zobrazovania elementov výstupu počas simulácie

OPNET Modeler počas simulácie priebežne informuje o aktuálnom stave simulácie. Príkladom môže vypisovanie odhadovaného času a počtu spracovaných udalostí. Zobrazovanie priebežných výsledkov spomaľuje simuláciu. Vypnutím ich zobrazovania alebo ukladania sa simulácia zrýchli. Niekedy výrazne, inokedy len veľmi málo. Položky, ktorých zobrazovanie spomaľuje simuláciu a ako ich vplyv možno zmenšiť:

- Interval aktualizácie priebehu simulácie - Configure/Run Discrete Event Simulation > Common > Update Interval nastaviť na vyššiu hodnotu, napríklad 10 miliónov.

Ostatné nastavenia sú jednotne v Configure/Run Discrete Event Simulation > Runtime displays a Other displays:

- Vykresľovanie grafu využitia pamäte - Graph total memory usage during simulation.
- Vypisovanie štatistík pamäte. Nie je štandardne zapnuté - Show memory statistics panel.

-
- Zrušenie vypisovania rýchlosti spracovávaní udalostí za sekundu – Graph event processing speed.
 - Zobrazenie parametrov príkazového riadku simulácie - Display parameters passed to each simulation run.
 - Zobrazenie štatistík počas behu simulácie, tzv. live statistics - Display live statistics panel. Takýto typ štatistík musí byť správne definovaný.
 - Vytvorenie zoznamu súborov použitých v simulácii - Generate list of component file dependencies.

Efektívne simulovanie protokolov OSPF, ARP, BGP, EIGRP, IGMP, IGRP, ISIS, RIP a SWITCH (STP)

V konvergovanej sieti tvorí prevádzka smerovacích protokolov a protokolov iných vrstiev, napr. protokolu STP len malú časť šírky pásma. Zapnutím efektívneho simulovania protokolov dôjde po určenom čase k vypnutiu ich aktualizácie, avšak smerovacie tabuľky ostanú zachované. Efektívne simulovanie je vhodné používať len ak nie je pomocou modulu Failure recovery nastavené zlyhanie linky alebo zariadenia (Obr. 0.1).

Nastavenie efektívneho simulovania je v menu Configure/Run DES > Inputs > Global Attributes > Simulation Efficiency.

Druhou možnosťou, ako zrýchliť simuláciu siete použitím efektívneho simulovania protokolov je využiť exportované smerovacie tabuľky. Výhodou je, že každá aplikácia (modul application) pre riadenie prevádzky alebo požiadavky (demand) môžu byť smerované okamžite bez čakania konvergencie siete a naplnenie smerovacích tabuliek. Zrýchlenie nastáva najmä pri väčších sieťach. Pri udalostnej simulácii na to slúži nastavenie Configure/Run DES > Inputs > Global Attributes > IP > IP Routing Table Export/Import. Prvý krát som spustil simuláciu s nastavením Export. Smerovacie tabuľky boli zapísané do súboru nazovprojektu-scenario.ot v priečinku modelu. Druhý krát som spustil simuláciu s nastavením Import. Na začiatku simulácie sú smerovacie tabuľky načítané do jednotlivých simulovaných zariadení bez potreby opätovného simulovania konvergencie siete. Po zmene topológie simulácia pracovala správne, až po opätovnom exportovaní a importovaní smerovacích tabuliek.

Oba popísané postupy nie je vhodné používať, ak sú predmetom skúmania smerovacie protokoly alebo je nastavené pomocou modulu Failure Recovery (Obr. 0.1) simulované zlyhanie, prípadne zotavenie linky alebo zariadenia.

Vývojové (development) a optimalizované (optimized) simulačné jadro

Vývojové jadro je určené na ladenie kódu (debugging). Prepnutím simulačného jadra na optimalizované dôjde ku kompilovaniu modulov. Vynechaním réžie potrebnej na ladenie spôsobí, že udalosti sú spracovávané rýchlejšie. Dosahované zrýchlenie bolo na úrovni 50% až 300%.

Optimalizácia kódu

Zrýchliť simuláciu je možné aj optimalizáciou zdrojových kódov jednotlivých modulov. Tematika optimalizácie kódu a algoritmov je obsiahla a nie je účelom tejto práce vysvetľovať jej princípy.

Ako príklad uvádzam použitie hešovacej tabuľky pri práci so záznamami smerovej tabuľky smerovača.

Na účely optimalizácie algoritmov sú určené vypisované štatistiky popísané v kapitole 0. OPNET Modeler obsahuje aj profiler, podporný program na zobrazenie častí kódu, ktoré sú vykonávané pomaly.

Použitie modelu požiadaviek aplikácií (Demands) a modelu prevádzky na pozadí (Background traffic)

Nahradenie explicitne modelovanej prevádzky modelom požiadaviek aplikácií, prípadne modelom prevádzky na pozadí, zmenší počet vykonávaných udalostí a zároveň zrýchli vykonávanie simulácie. Podrobnejší popis zaoberajúci sa modelovaním sieťovej prevádzky je v kapitole 0.

Vypnutie zaznamenávania a zobrazovania animácií

Počas simulácie je možné zobrazovať a zaznamenávať pohyb modelov zariadení v priestore a pohyb dátových jednotiek. Zaznamenávanie animácie pri priemerných 120 tisíc spracovaných udalostí za sekundu výrazne nespomaľuje simuláciu, naproti tomu, hoci zobrazovanie dokáže pomôcť pri ladení modelu, niekoľkonásobne ho spomaľuje. Zaznamenávanie je implicitne vypnuté.

Vypnutie zaznamenávania a zobrazovania sa nachádza v menu DES > Record Packet Flow, Record Node Movement a v Congirugre/Run DES > Outputs > Global Animation > 2D

Obsah zaznamenaného súboru je binárny a je možné ho transformovať do textovej formy a späť pomocou programu op_cvanim. Binárna podoba je nazývaná animation history, textová animation script. Skonvertoval som jeden binárny súbor. Kompletný súbor je uložený v prílohe dvd v súbore test1_distributed-scenario1-DES-1.as.

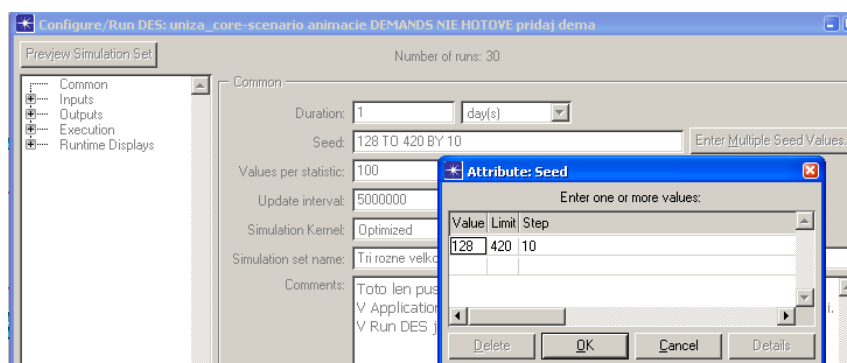
Z menu štart som otvoril som OPNET príkazový riadok. Pomocou príkazov pre zmenu adresára cd som zmenil aktuálny adresár na adresár modelu so súborom zaznamenatej animácie. Príkaz op_cvanim -reverse TRUE -m test1_distributed-scenario1-DES-1 konvertoval animáciu.

Na opätovnú konverziu textového súboru zaznamenatej animácie na binárny som zmenil hodnotu -reverse z TRUE na FALSE.

Okrem uvedených princípov a postupov, ako zrýchliť simuláciu existuje aj SMART MAC model, ktorý poskytuje abstrakciu MAC a fyzickej vrstvy bezdrôtových zariadení.

ŠTATISTICKÉ SPRACOVANIE VÝSLEDKOV DISKRÉTNÝCH UDALOSTNÝCH SIMULÁCIÍ

Jeden simulačný beh na určenie výsledných hodnôt nestačí. Zle zvolená násada generátora pseudonáhodných čísel môže spôsobiť značné odchýlky presnosti. Preto, ak som chcel určiť správnosť výsledkov, čiže odhadu, s nejakou mierou pravdepodobnosti, určil som intervaly spoľahlivosti nameraných údajov. Intervaly spoľahlivosti som vypočítal pre každú zmenu parametrov veľkosti prenášaného súboru, alebo veľkosti prevádzky generovanej pomocou modelov požiadaviek aplikácií. Na výpočty intervalov spoľahlivosti som použil odporúčaných 30 simulácií s rôznymi násadami, pričom hodnoty ostatných parametrov ostali bez zmeny (Obr. 0.1). Viac simulačných behov s rovnakou topológiou a konfiguráciou siete mi zabezpečilo jedinečnosť štatistických výsledkov pre rôzne rozdelenia pravdepodobnosti a rôzne výsledky náhodných premenných.

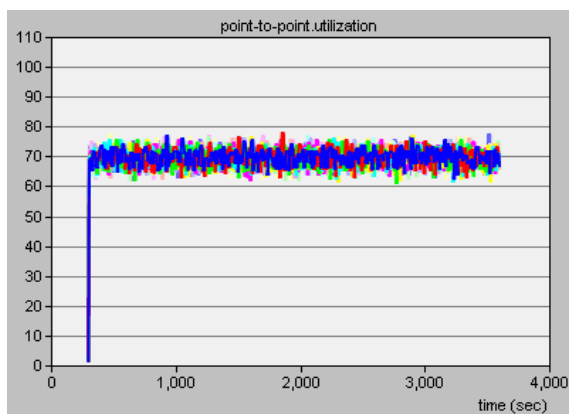


Obr. 0.1 Viac hodnôt násady spôsobí spustenie viacerých simulačných behov

1.1.6 Výpočet intervalov spoľahlivosti

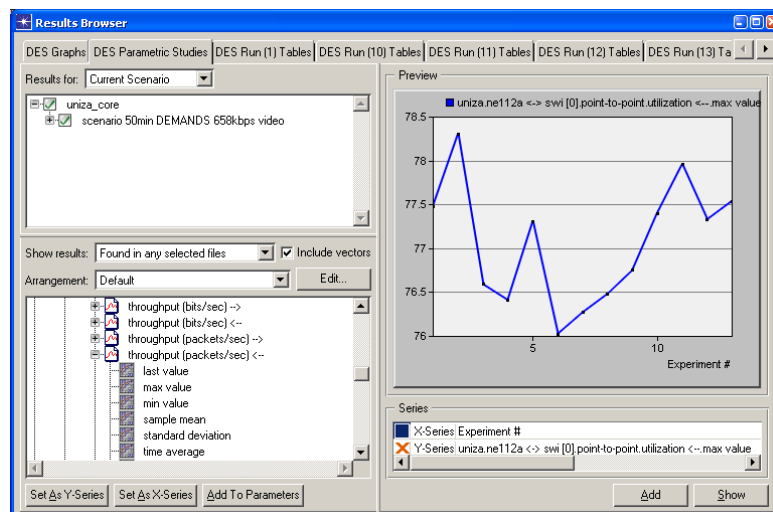
V nasledovnom príklade ukážem postup výpočtu intervalu spoľahlivosti maximálneho využitia linky, ktorý som využil v diplomovej práci. Pre vyskúšanie výpočtu intervalov spoľahlivosti je v DVD prílohe súbor uniza_core.opcfa. Ten si rozbalím (File > Manage Project Files > Extract Project Archive). Teraz sa v určenom priečinku nachádza projekt. Postup je ukázaný na scenári s názvom 50min DEMANDS 658kbps video. V modeli sú nastavené požiadavky aplikácií do každej fakulty. Ich veľkosti sú vypočítané ako suma požiadaviek študentov fakulty za hodinu. Popis výpočtu veľkosti tokov je v Tab. 0.1.

Aby som spustil viac simulačných behov, nastavil som viac hodnôt násad v nastavení simulačného behu (Configure/Run DES) ako v Obr. 0.1. Problematikou distribuovanej simulácie sa zaoberám v kapitole 0. Po vykonaní všetkých simulácií sú vo výsledkoch simulácie zozbierané štatistiky využitia linky medzi prepínačmi ne112a a swi (utilization ←) z projektu uniza_core v prílohe A. Porovnaním grafov zo simulačných behov s rôznymi násadami vznikol nasledovný graf:



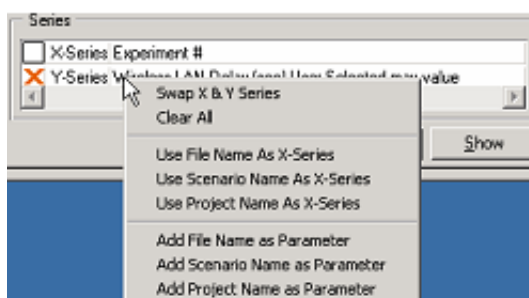
Obr. 0.2 Graf využitia linky ne112a ← swi

Interval spoľahlivosti som vypočítal v záložke DES Parametric Studies (Results Browser). V rozbalenej položke utilization ← som nastavil atribút „max value“ na os y (Set As Y-Series). Na osi y grafu sú maximálne hodnoty zaťaženia linky v percentách pre jednotlivé simulačné behy na osi x. Hodnoty maximálneho vyťaženia linky sú rozťahnuté po celej osi x, pretože v grafe sú štandardne zobrazované čísla experimentov.



Obr. 0.3 Maximálne hodnoty zaťaženia linky

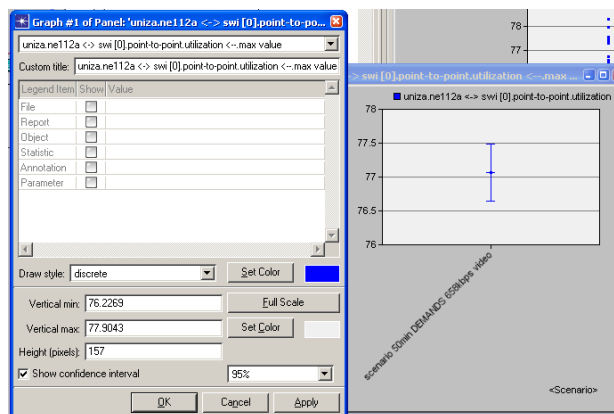
Na zobrazenie intervalu spoľahlivosti sú potrebné také body grafu, ktoré majú spoločnú hodnotu na osi x. Na zobrazenie takéhoto grafu som nastavil na os x názov scenára (Pravým tlačidlom myši na X-Series > Use Scenario Name As X-Series).



Obr. 0.4 Nastavenie osi x grafu

Teraz majú body rovnakú hodnotu na osi x. Tlačidlom Show sa graf zobrazí v samostatnom okne, hodnoty vytťaženia linky sú približne 76% až 78,3%. Interval spoľahlivosti je možné zobraziť v grafe, alebo vypísať ako hodnoty. Pre vypísanie hodnôt stačí kliknúť pravým tlačidlom myši na sivú časť grafu a zobraziť dáta grafu (Show Statistics Data). V riadkoch zobrazených odspodu sú zobrazené 80%-né až 99%-né intervaly spoľahlivosti.

Pre zobrazenie intervalu spoľahlivosti v grafe je možné vykonať úpravou grafu (kliknutím pravého tlačidla myši na plochu grafu a vybratím položky Edit Graph Properties. V novom okne je položka Show confidence interval.



Obr. 0.5 Nastavenie grafu na zobrazenie intervalu spoľahlivosti

ANALÝZA E-VZDELÁVACÍCH KURZOV

Pri analýze kurzov elektronického vzdelávania a množstva prenesených dát som vychádzal z práce [1]. Na strane 64 je vypočítaná priemerná hodinová sieťová prevádzka spôsobená kurzami na Fakulte riadenia a informatiky. Tieto údaje mi slúžili a podklad určenie hodnôt prevádzky.

Druhy prevádzky

Cieľom diplomovej práce bolo modelovať správanie chrbticovej časti siete pri nasadení elektronických vzdelávacích kurzov. Tieto zastupujú vyhľadávaciu triedu, prípadne prúdovou triedou (streaming). Tá predpokladá služby toku v reálnom čase.

Jednalo sa jednalo o dva druhy realizácie elektronického vzdelávania:

- Krátke videá alebo vektorové animácie so sprievodným textom
- Päťdesiatminútové videá

1.1.7 Krátke videá alebo vektorové animácie so sprievodným textom

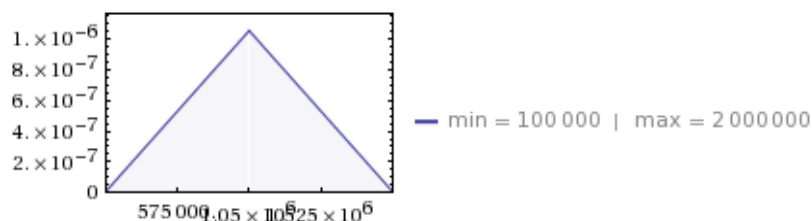
Pri vytváraní simulačných modelov som vychádzal z predpokladu, že jeden kurz obsahuje približne 60 animácií s veľkosťou 100kB až 2MB. 90% animácií je veľkosti maximálne 500kB. Podľa [1], strana 80.

Veľkosť jedného súboru je modelovaná distribučnou funkciou trojuholníkového rozdelenia pravdepodobnosti s parametrami (100000,2000000), ktoré predstavujú minimálnu a maximálnu veľkosť súboru v bajtoch. V druhom prípade som veľkosť súboru modeloval vlastným empirickým rozdelením, ktoré viac zodpovedá realite. Empirické rozdelenie je možné modelovať v OPNET Modeler pomocou PDF Editoru, ako je v Obr. 0.2. Pravdepodobnosti (os y) hodnôt na osi x kvôli názornému zobrazeniu nezodpovedajú skutočnosti a nie sú normalizované na hodnotu súčtu 1. Preto súčet pravdepodobností jednotlivých hodnôt $\neq 0$.

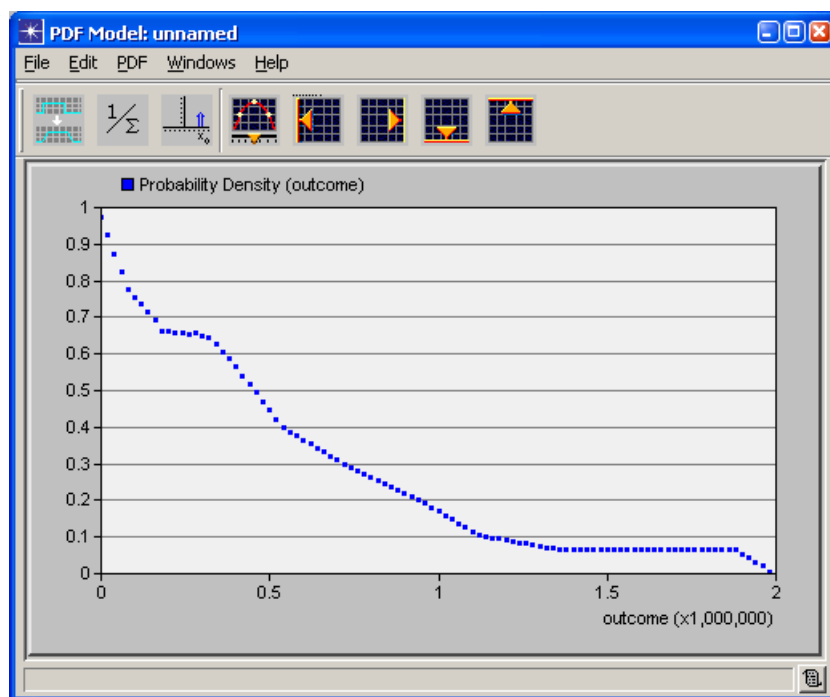
1.1.8 Päťdesiatminútové videá

Druhá, zatiaľ nerealizovaná možnosť vzdelávania, ktorú som simuloval predstavuje vzdelávanie pomocou prednášok pozostávajúcich zo súvislých nahraných videí prenášaných po sieti, podobne ako služba na stránke www.youtube.com. Obrázok zo stránky služby youtube.com je v prílohe I. Trvanie jedného videa je približne 50 minút. Súbor s videom je zo servera na klienta prenášaný ako unicast, protokolom

HTTP, ako Flash video. Veľkosť súborov 50 minútových videí som ako určil pomocou oficiálnej kalkulačky bitovej rýchlosti Flash videa [5] a porovnal s bitovou rýchlosťou desiatich videí dĺžky približne 50 minút. Obrázok tejto kalkulačky je v prílohe B.



Obr. 0.1 Hustota pravdepodobnosti trojuholníkového rozdelenia s parametrami min = 100000, max = 2000000 (os y – pravdepodobnosť hodnôt z osi x)



Obr. 0.2 Nenormalizovaná hustota pravdepodobnosti veľkosti súboru (os x – hodnoty veľkosti súboru, os y – pravdepodobnosť)

Tab. 0.1 Priemerné veľkosti súborov 50 minútových videí

Rozlíšenie videa [pixelov]	Počet obrázkov za sekundu	Zmena obsahu	Video kódok	Priemerná bitová rýchlosť videa [Kbps]	Zvukový kódok	Vzorkovacia frekvencia [kHz]	Kanály	Kvalita kódovania a zvuku	Sumárna priemerná bitová rýchlosť [Kbps]	Odhad veľkosti súboru 50 min. videa [MB]
320 na 240	15	malá	H.264	84	AAC	11025	Mono	Stredná	100	36,62
320 na 240	25	malá	H.263	160	MP3	11025	Mono	Stredná	178	65,19
320 na 240	30	malá	H.263	192	MP3	11025	Mono	Stredná	210	76,9
448 na 336	25	malá	H.263	314	MP3	11025	Mono	Stredná	332	121,58
640 na 480	25	malá	H.263	640	MP3	11025	Mono	Stredná	658	240,97

V simulácií sú na modelovanie veľkosti videí podobne ako v modeli so šesťdesiatimi animáciami použité hustoty pravdepodobnosti s trojuholníkovými rozdelením. Trojuholníkové rozdelenie som zvolil preto, že je z možností, ktoré sa dali v tomto prípade v programe OPNET Modeler použiť na modelovanie veľkosti súborov najviac vhodné. Parametre minimum a maximum trojuholníkového rozdelenia sú vypočítané z odhadu veľkosti súboru 50 min. videa [MB] ± 5 MB a zaokrúhlené na celé MB nahor alebo nadol. Napríklad video s veľkosťou 36,62 MB bude v zastúpené trojuholníkovým rozdelením s parametrami 31000000 a 42000000. Výsledné veľkosti súborov s videom použitých v simulácií sú overené na základe priemernej veľkosti desiatich prednášok s dĺžkou v rozmedzí 48 až 53 minút.

Výsledky pre obidve možnosti zloženia kurzov - šesťdesiat animácií, alebo jedno päťdesiatminútové video sú v kapitole 0.

Odhad počtu aktívnych študentov za hodinu

Počet aktívnych študentov za hodinu predstavuje počet študentov, ktorí v danej hodine využívajú nejaký vzdelávací kurz. Výpočet počtu aktívnych študentov na Fakulte riadenia a informatiky za hodinu je určený z približného počtu počítačových učební, desať a približného počtu študentov v študijnom krúžku 20. Z pomeru počtu študentov FRI v dennej forme štúdia 1270 a počtu aktívnych študentov 200 som vypočítal percento aktívnych študentov za hodinu zo všetkých študentov. Je to približne 15%. Z uvedených čísel som odhadom určil pomer množstva vytvorených kurzov voči FRI. Vyjadruje mieru využívania elektronického vzdelávania. Vynásobením počtu študentov danej fakulty, pomerného množstva elektronických kurzov a percenta

aktívnych študentov som dostal odhad počtu aktívnych študentov danej fakulty. Výsledky sú v Tab. 0.2.

Tab. 0.2 Určenie počtu aktívnych študentov

Fakulta	Počet študentov v dennej forme štúdia	Počet elektronických vzdelávacích kurzov	Pomerné množstvo vytvorených elektronických kurzov voči FRI	Počet aktívnych študentov za hodinu
Elektrotechnická fakulta	1583 [6]	157	0,8	199
Fakulta prevádzky a ekonomiky dopravy a spojov	2316	317	0,7	255
Fakulta prírodných vied	1074	91	0,6	101
Fakulta riadenia a informatiky	1270	112	1	200
Fakulta špeciálneho inžinierstva	921	112	0,6	87
Stavebná fakulta	902	45	0,6	85
Strojnícka fakulta	1175	76	0,7	130
pomer aktívnych študentov v jednej hodine na celkový počet študentov	200 / 1270 $\approx$ 15%			

POROVNANIE PRVKOV SIETE S MODELMI

OPNET Modeler obsahuje rozsiahlu databázu modelov zariadení rôznych výrobcov. Niektoré, najmä novšie modely neobsahuje, preto som ich substituoval najlepším možným variantom modelu.

Tab. 0.1 Reprezentácia zariadení v simulovanej sieti

Zariadenie v sieti UNIZA	Reprezentácia v OPNET Modeler
Cisco Catalyst Switch 3750	Cisco Catalyst Switch 3560
Cisco Catalyst Switch 3550	Cisco Catalyst Switch 3560
Cisco Catalyst Switch 6509	
Cisco Router 1720	
Cisco PIX 515E	

Redundancia liniek a zariadení

Zlyhanie ktoréhokoľvek prepínača z modelovanej siete spôsobí výpadok spojenia až do opravy, resp. nahradenia prepínača. Takéto výpadky bývajú zväčša viachodinové. Jediná výnimka sa týka prepínačov ne112a a ne112e. V prípade výpadku ľubovoľného jedného z nich je personál schopný do polhodiny manuálne zmeniť konfigurácie a substituovať jeden prepínač druhým.

Takisto žiadna relevantná linka medzi fakultami nie je zálohovaná. V prípade výpadku dôjde vo väčšine prípadoch k prerušeniu spojenia na vzdialený uzol. Obnova spojenia je možná až po oprave linky, ktorá na trase zlyhala. Zriadenie zálohovania trasy je do budúcnosti plánované.

V OPNET Modeler je možné simulovať výpadok, prípadne obnovenie linky pomocou objektu Failure Recovery z palety utilities.



Obr. 0.1 Modul zlyhanie a obnova linky alebo uzlu

Využitie objektu Failure Recovery je v kapitolách 1.1.1 a 1.1.12

1.1.9 Možnosti zálohovania predvolenej brány v OPNET Modeler

Hot standby router protocol (HSRP), proprietárny protokol, predstavuje mechanizmus IP prevádzky bez prerušenia na sieťach podprujúcich multicast, alebo broadcast, ako napríklad Ethernet sieť. Protokol chráni proti zlyhaniu najbližšieho smerovača. Používa sa na ochranu proti výpadku predvolenej brány. V sieti ŽU nie je použitý. Podpora HSRP sa nachádza v OPNET Modeler.

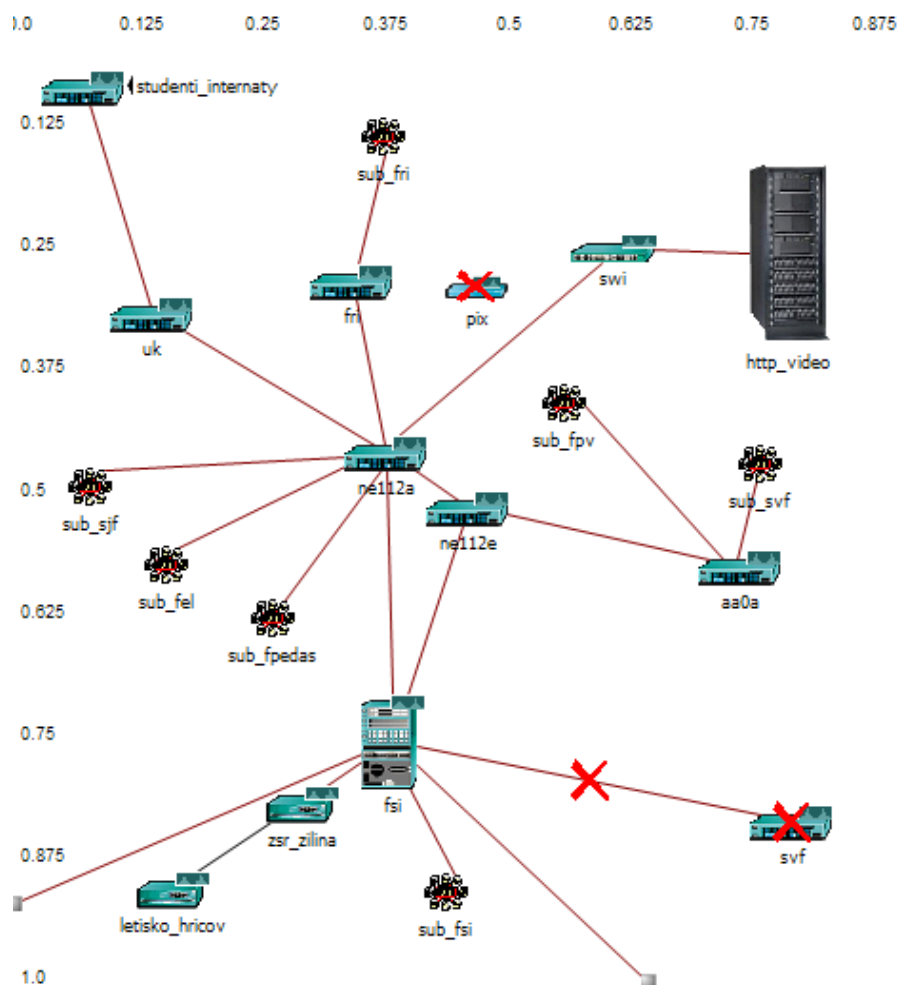
IMPLEMENTÁCIA CHRBTICOVEJ SIETE V PROGRAME OPNET MODELER

Pri implementácii jednotlivých prvkov siete som musel abstrahovať od niektorých parametrov. Dôvodmi boli najmä veľká časová náročnosť konfigurácie, výpočtová zložitosť a množstvo získaných údajov. Problémom zrýchlenia simulačných behov sa zaoberám v kapitole 0. Nastavovanie siete je možné zrýchliť importovaním topológie siete a konfigurácie jednotlivých prvkov importovaním z VNE servera. Je to samostatne bežiacia aplikácia, ktorá pomocou agentov zisťuje topológiu, toky a aktuálny stav zariadení siete. Používaný softvér OPNET Modeler neobsahuje VNE server. Výhody použitia importovania sa prejavia pri importovaní veľkého počtu zariadení.

V praktickej časti práce som jednotlivé prvky pridával a konfiguroval manuálne podľa dostupných informácií.

V prvom kroku som pridal uzly a prepojil ich linkami. Chrbticová sieť je modelovaná šiestimi prepínačmi Cisco Catalyst Switch 3560 (uzly studenti_internaty, uk, fri, ne112a, ne112e, aa0a), prepínačom Cisco Catalyst Switch 6509 (uzol fsi), smerovačmi Cisco Router 1720 (uzly zsr_zilina a letisko_hricov) a zabezpečovacím systémom - firewall Cisco PIX 515E (uzol pix).

Uzol predstavujúci server s uloženými elektronickými výučbovými materiálmi má názov http_server. Podľa požiadaviek vedúceho práce je nadimenzovaný tak, aby aj relatívne veľký objem požiadaviek bol bez problémov spracovaný. Tento uzol je predstavovaný modelom skutočného servera z knižnice objektov - IBM p690 Turbo so šiestimi dvojjadrovými procesormi taktovanými na frekvenciu 1700MHz.



Obr. 0.1 Topológia siete Žilinskej univerzity v programe OPNET Modeler. Nepoužívané zariadenia sú preškrtnuté červeným krížikom.

V ďalšom kroku nasledovalo pridanie uzlov, ktoré predstavujú jednotlivé fakulty, nastavenie jednotlivých uzlov IP adries (Protocols > IP > Addressing) a overenie dostupnosti medzi uzlami pomocou modelu požiadaviek aplikácií (IP Ping).

Smerovací protokol OSPF som nastavil podľa nastavení v sieti ŽU. Ocenenie každého rozhrania, v súvislosti s vytvorením grafu s najkratšími cestami medzi uzlami, je vypočítavané automaticky podľa referenčnej šírky pásma nastavenej na 10000000000. Ostatné relevantné hodnoty časovačov sú v Tab. 0.1.

Tab. 0.1 Hodnoty časovačov protokolu OSPF

Hodnota	Čas v sekundách
Hello Interval	10

Router Dead Interval	40
Transmission Delay	1
Retransmission Interval	5

Hoci je v sieti použité rozdelenie na oblasti (area), v modeli som ich pre malý počet uzlov nenastavoval.

OPNET Modeler nemá úplnú podporu sietí VLAN. Neúplnou podporou myslím, že všetky porty medzi prepínačmi, ktoré majú zapnutú podporu VLAN, musia byť tzv. trunk. Nie je možné nastaviť prechod tromi VLAN sieťami, ak sú tieto siete na troch rôznych linkách. Preto som každému L3 prepínaču pridal fyzické porty, ktoré je možné smerovať. Na nich sú nastavené IP adresy. Pre modelovanie funkcionality siete nebolo potrebné nastaviť IP podľa skutočného stavu.

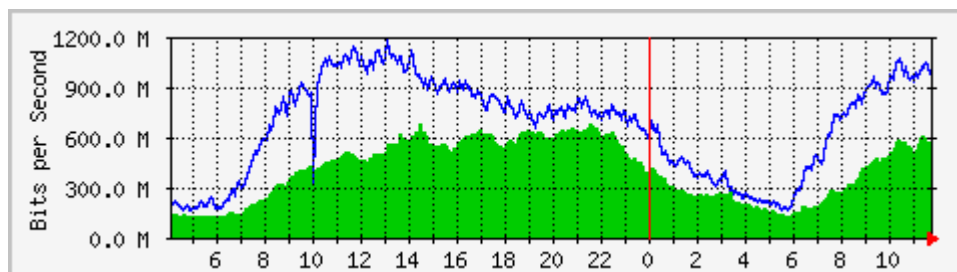
Na hromadné nastavenie smerovacieho protokolu na všetkých zariadeniach som použil položku Configure routing protocol v menu Protocols > IP > Routing a Protocols > OSPF.

Jednotlivé siete fakúlt sú modelované ako skupina počítačov pripojených na prepínače. Počítače predstavujú počet aktívnych používateľov v hodine na danej fakulte. Takýto model je v OPNET Modeler zastúpený objektom s názvom 1000BaseX_LAN z palety LANs (Obr. 0.2). Každá fakulta je pripojená do chrbticovej siete linkou s priepustnosťou 1000 Mbit za sekundu. Prevádzka typu broadcast predstavuje oproti ostatnej prevádzke na sieti veľmi malý objem. Preto nehrozí, že by umiestnenie viacerých počítačov do jednej broadcast domény výrazne ovplyvnilo výsledky simulácie.

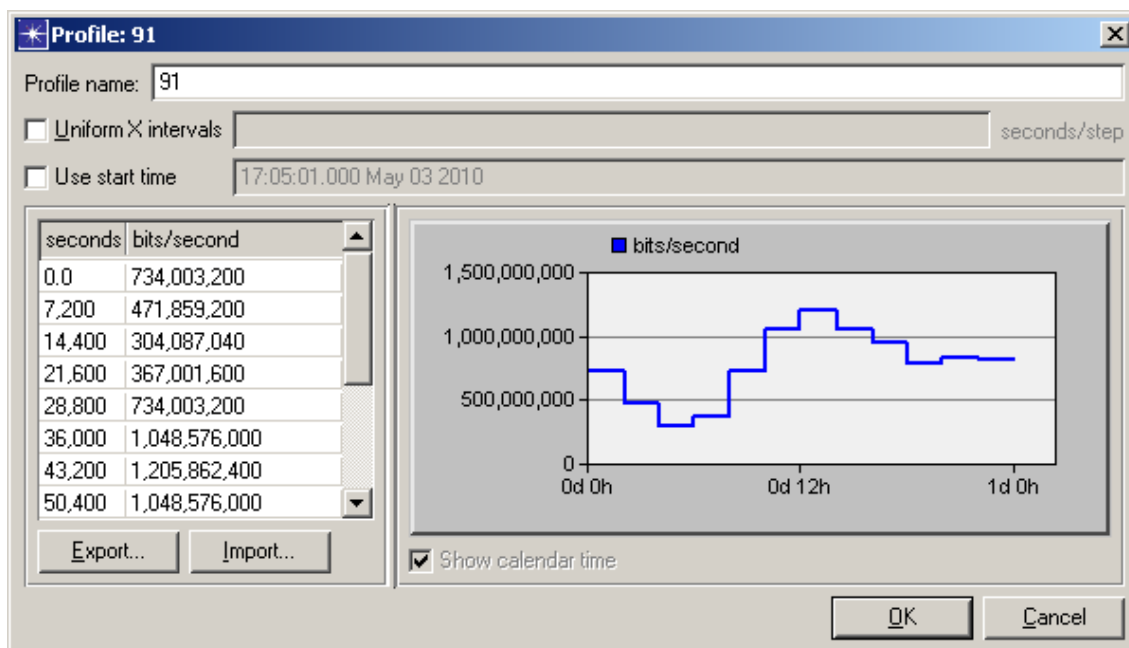


Obr. 0.2 Objekt 1000BaseX_LAN, v modeli predstavuje aktívne počítače na určenej fakulte

V druhom kroku som na linky pridal pomocou prevádzky modelovanej na pozadí (kapitola 0 a 0) základnú prevádzku. Tá odráža súčasný stav prevádzky medzi jednotlivými fakultami.



Obr. 0.3 Graf vyt'azenia linky medzi smerovačom fsi a smerovačom v Púchove, os x – čas v hodinách, os y – milióny bitov za sekundu



Obr. 0.4 Reprezentácia základnej (baseline) prevádzky z uzla Púchov do uzla fsi v programe OPNET Modeler, os x – čas v hodinách, os y – bity za sekundu

Použité modely prevádzky a výsledky simulácií

Toky dát zo servera k používateľom som modeloval dvomi spôsobmi:

- Nastavením parametrov v objektoch profily a aplikácie
- Modelom požiadaviek dát (Application demands)

1.1.10 Model prevádzky vytváraný profilmi a aplikáciami

Simulácia s využitím tohto spôsobu modelovania prevádzky je výrazne pomalšia a spotrebuje veľa fyzickej pamäte. Výhodou je, že poskytuje napr. oproti modelu požiadaviek dát presnejšie výsledky. Projekt modelovania prevádzky vytváraný profilmi a aplikáciami je v prílohe na DVD. Je to súbor `uniza_core.opcfs` a scenáre s názvami `scenario 60 animacii EVENTS`, `scenario 50 min EVENTS`.

1.1.10.1 Profily a aplikácie: Simulácia prenosu 50 minútových videí

Ani po mnohých pokusoch sa mi nepodarilo simulovať ani len pár sekúnd prevádzky 50 minútových videí. Na všetky simulácie som používal počítač s dvojjadrovým procesorom, fyzickou pamäťou 2,5 GB a celkovou virtuálnou pamäťou 15 GB. Príčinou zlyhania simulácie bol nedostatok pamäte. Simulácia prenosu dát presahovala možnosti hardvéru. Vykonať simuláciu by bolo možné s výrazne nižšími modelovanými objemami dát. Zobrazenie chyby je v prílohe H.

1.1.10.2 Profily a aplikácie: Simulácia prenosu krátkych animácií

Z jednej hodiny, počas ktorej jeden študent preniesie 60 krátkych animácií som z dôvodu dĺžky trvania simulačného behu simuloval len jednu minútu. Medzery medzi požiadavkami na stiahnutie animácie sú exponenciálne s parametrom 60 sekúnd. To znamená, že za jednu hodinu stiahne študent v priemere 60 animácií. Z dôvodu konverencie siete je vo všetkých simuláciách uvedených v nasledovnom texte nastavený čas prvej požiadavky študenta na animáciu v rozmedzí 300 až 400 sekúnd. V čase 460 sekúnd končí simulácia. Súbor s projektom sú v prílohe na DVD v súbore `uniza_core.opcfa`.

Výsledok 90%-ného intervalu spoľahlivosti maximálneho času odozvy na jednu stránku, pozostávajúcu z dvoch objektov – html stránka a krátka animácia, z 30 simulačných behov s rôznymi nasadami pre pseudonáhodné generátory je $\langle 0,13; 0,2 \rangle$ sekundy. Hodnoty 90%-ného intervalu priemernej odozvy HTTP prevádzky sú v rozmedzí $\langle 0,04; 0,2 \rangle$ sekúnd. Priemerne teda 0,09 s. so smerodajnou odchýlkou 0.0259. Najvyťaženejšia linka, medzi `http_server` a prepínačom `ne112a`, cez ktorú idú všetky požiadavky a odpovede z jednotlivých fakúlt mala hodnoty intervalu maximálneho zaťaženia v rozmedzí $\langle 103; 145 \rangle$ Mbit za sekundu, čo predstavuje 10 až 15 percentnú šírku pásma. Takýto typ prevádzky by po sieti bez väčších problémov prešiel.

1.1.11 Model požiadaviek dát (Application demands)

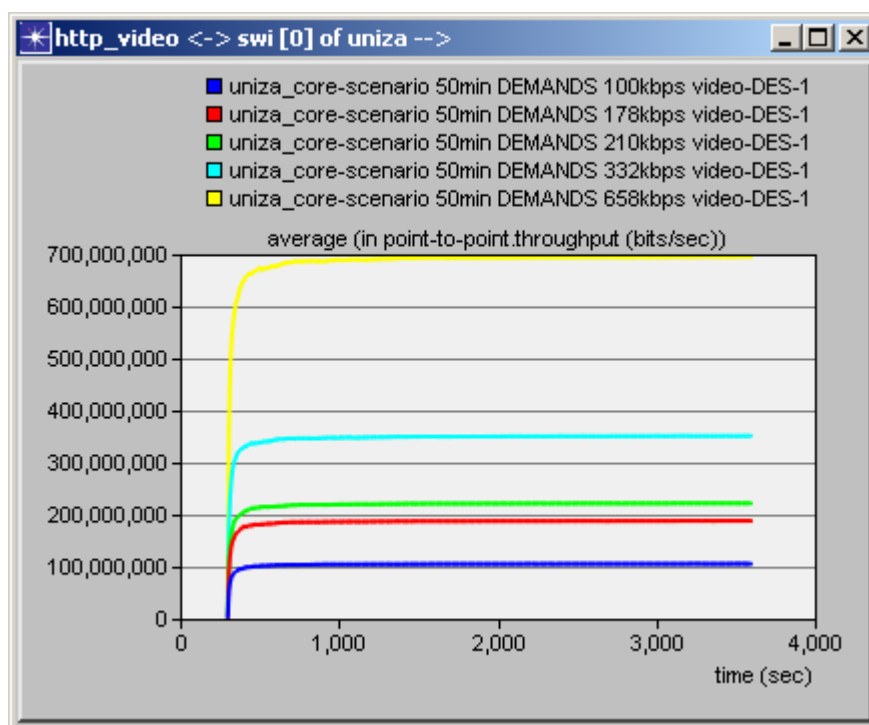
Modelom požiadaviek aplikácií sa mi podarilo modelovať oba spôsoby vyučovania - 50 minútové videá a tiež krátke animácie. Simulovaná časť predstavuje jednu hodinu. Nasledujúce príklady sú v scenároch s názvami scenario 50min DEMANDS 100kbps video, scenario 50min DEMANDS 178kbps video, scenario 50min DEMANDS 210kbps video, scenario 50min DEMANDS 332kbps video, scenario 50min DEMANDS 658kbps video, scenario 60 animácii DEMANDS.

1.1.11.1 Model požiadaviek dát: Simulácia prenosu 50 minútových videí

Prenos videí pomocou tohto modelu bolo nutné simulovať ako tok dát s určitým objemom za časovú jednotku. V Tab. 0.1 sú v stĺpci uvedené priemerné bitové rýchlosti potrebné na prenos jedného videa uvedeného typu. Celkový objem dát prichádzajúcich do fakultnej podsiete je násobkom počtu aktívnych študentov za hodinu. Simuláciu som spustil pre päť druhov videí s rôznou bitovou rýchlosťou. V tabuľke sú zobrazené hodnoty 90%-ného intervalu spoľahlivosti vytiaženia linky medzi uzlom http_video a prepínačom ne112a. V ostatných uzloch je podstatne nižšia prevádzka, preto ich vo výsledkoch neuvádzam.

Tab. 0.2 Intervaly spoľahlivosti priemerného vytiaženia linky medzi ne112a a http_server

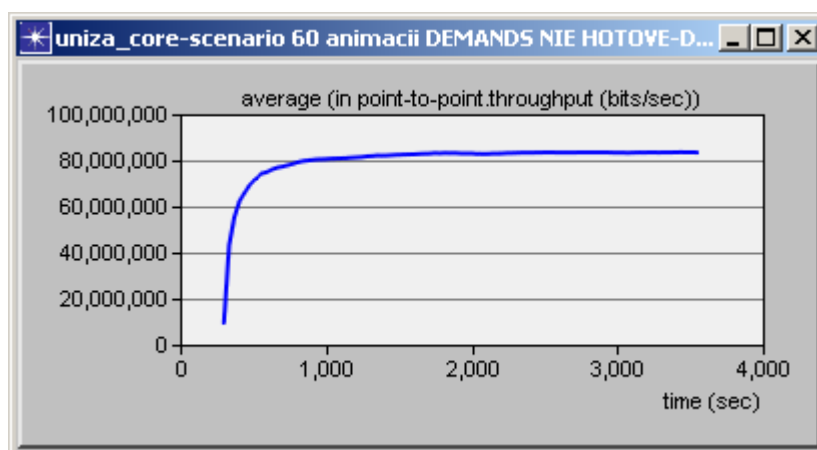
Bitová rýchlosť videa [Kbps]	90%-ný interval spoľahlivosti vytiaženia linky [Mbps]
100	< 103,713; 104,642 >
178	< 184,862; 186,539 >
210	< 217,896; 219,862 >
332	< 345,172; 348,234 >
658	< 683,250; 689,372 >



Obr. 0.5 Priemerné vyt'aženie linky v bitoch za sekundu pre jednotlivé druhy videí

1.1.11.2 Model požiadaviek dát: Simulácia prenosu krátkych animácií

Údaje o veľkosti toku, ktorý bude predstavovať sťahovanie animácií som určil z výsledkov simulácie, v ktorej som nastavil, aby 100 počítačov pripojených v sieti s jednoduchou topológiou sťahovalo súbory s veľkosťou animácií. Intervaly medzi sťahovaním jedného súboru jedným študentom sú z exponenciálneho rozdelenia s parametrom 60 (sekúnd). Približná veľkosť toku na 100 vychádza 8 Mbit/s. Podľa veľkosti výsledného toku som proporcionálne, podľa počtu aktívnych študentov na fakultách nastavil veľkosti tokov modelov požiadaviek dát zo servera http_server.



Obr. 0.6 Veľkosť toku medzi http_server a ne112a v bitoch za sekundu, os y - bity za sekundu, os x čas v sekundách

Výsledky pre určenie toku spôsobeného sťahovaním krátkych animácií na 100 študentov je v DVD prílohe [A], súbor s projektom uniza_core, scenár s názvom scenario 60 animacii DEMANDS skusam. Scenár simulácie prenosu krátkych animácií pomocou modelu požiadaviek dát má názov scenario 60 animacii DEMANDS.

Z hľadiska vyťaženia sieťových prvkov je výhodnejšie použiť vzdelávanie formou krátkych animácií v porovnaní s ľubovoľným typom 50 minútových videí. Avšak vytváranie videí sa zdá byť časovo menej náročné na spracovanie a pri použití vysielania typu multicast, je možné výrazne znížiť prevádzku v sieti.

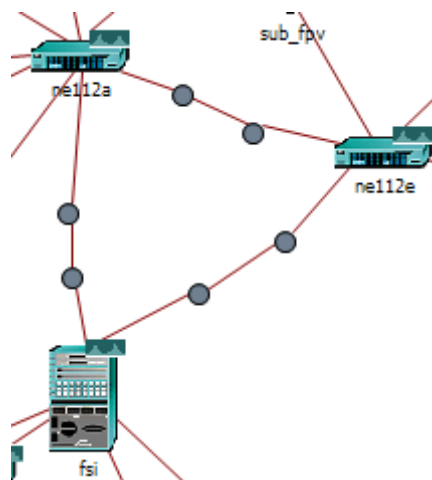
1.1.12 Modelovanie zlyhania linky

V práci som okrem príkladu v kapitole 1.1.1, kde rozoberám postup simulácie zlyhania linky, simuloval aj zlyhanie linky na chrbticovej sieti ŽU. Pred spustením simulácií som vypol efektívne simulovanie protokolu OSPF (kap. 0) V tejto časti sa venujem dopadu zlyhania liniek medzi prepínačmi fsi, ne112a a ne112e, postupne v poradí:

1. fsi – ne112a
2. fsi – ne112e
3. ne112a – ne112e

V prvom prípade uvažujem, že zariadenia sú prepojené priamo. Tento model zodpovedá realite. V praxi sa často používa aj iný spôsob zapojenia. Preto som vytvoril model, v ktorom sú prepínače prepojené cez ďalšie dva prepínače. Príklady sú v prílohe

DVD, súbor s projektom uniza_core v scenároch s názvami: scenario OSPF spamatanie, scenario OSPF spamatanie plus medzilahle.



Obr. 0.7 Topológia časti siete po pridaní prepínačov medzi zariadenia ne112a, ne112e a fsi

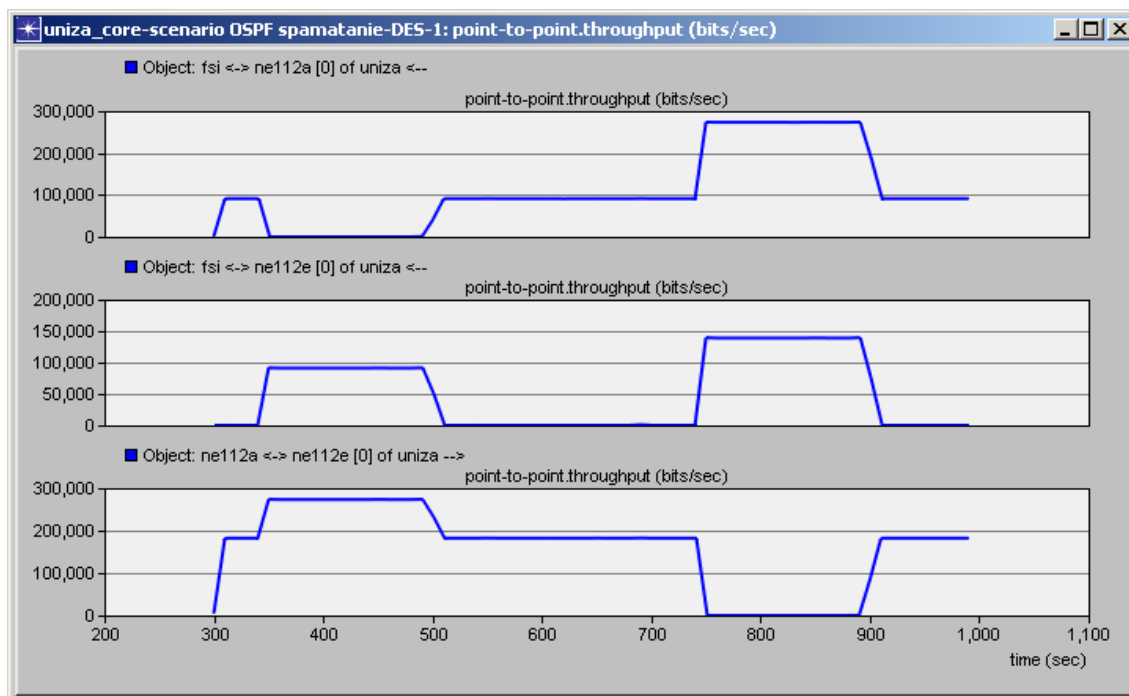
Prevádzka je simulovaná objektmi 1000BaseX_LAN, podobne ako v predchádzajúcich modeloch fakulty a sú umiestnené na rovnakých pozíciách s rovnakým pripojením. Rozdiel je v tom, že v jednom objekte je naraz aktívny len jeden počítač. Prevádzka jedného počítača predstavuje 10 požiadaviek za sekundu s veľkosťou 700 bajtov, na ktoré server http_video odpovedá paketmi veľkosti 1000 bajtov. To predstavuje sumárnu prevádzku v 10000 bajtov za sekundu v smere ku fakulte a 7000 bajtov za sekundu v opačnom smere.

Po konvergencii siete je v čase 300 sekúnd spustená prevádzka zo všetkých fakúlt. V čase 350 sekúnd dôjde k výpadku prvej linky. Následne je po 100 sekundách linka obnovená a o ďalších 100 sekúnd nastáva zlyhanie ďalšej linky v poradí popísanom vyššie.

1.1.12.1 Analýza výsledkov simulácií

V prvom prípade, keď sú prepínače fsi, ne112a a ne112e prepojené priamo je v prípade zlyhania ktorejkoľvek linky medzi nimi prevádzka okamžite presmerovaná. Tento stav je dôsledkom toho, že v prepínačoch dôjde k okamžitému zablokovaniu rozhraní, ktoré sú pripojené na zlyhanú linku. Namiesto použitia zablokovaného portu je prevádzka presmerovaná iným výstupným portom a inou cestou cez ktorý je miesto

určenia prevádzky dostupné. Z tohto dôvodu a v tomto prípade výpadok spôsobí stratu len veľmi malého množstva paketov.



Obr. 0.8 Grafy vytťaženia liniek medzi zariadeniami ne112a <- fsi, ne112e <- fsi, ne112a -> ne112e. Po výpadku linky fsi – ne112a (prvý graf) v čase 350 s. je celá prevádzka medzi uzlami fsi a ne112a presmerovaná cez uzol ne112e (druhý a tretí graf).

V druhom prípade, vďaka tomu, že zariadenia nie sú prepojené priamo sa v prepínačoch informácia o prerušení linky zostaví až po uplynutí času 40 sekúnd, keď nedôjde k prijatiu štyroch hello paketov protokolu OSPF. Dovtedy prepínač posielala pakety aj po ceste, ktorá obsahuje zlyhanú linku.

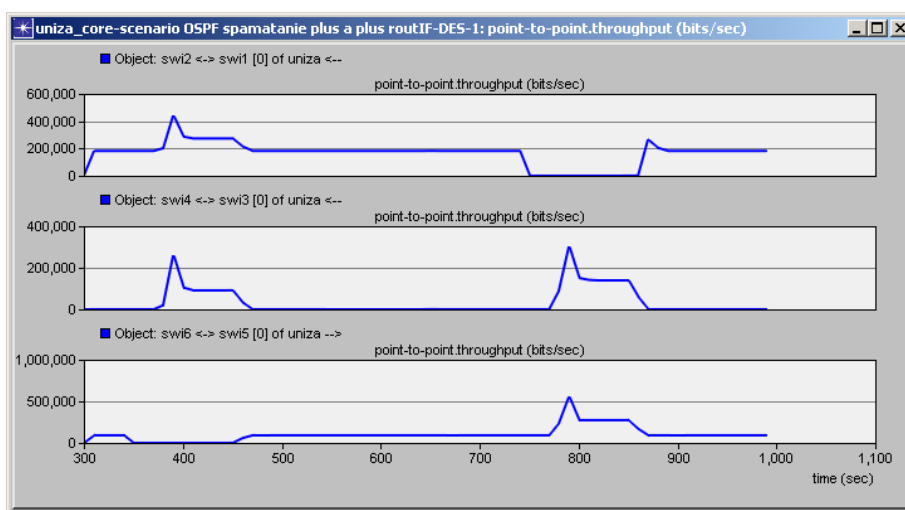
V simulácii zlyhá linka medzi pridanými prepínačmi v rovnakom poradí ako v príklade bez medziľahlých prepínačov.

V čase 350 sekúnd dôjde k zlyhaniu linky medzi uzlom fsi a ne112a, tak ako to vidno na poslednom grafe. Touto linkou nie je možné preniesť žiadne dáta, aj keď z oboch menovaných uzlov sú vysielané pakety žiadajúce o prenos, prípadne pakety predstavujúce samotný prenos.

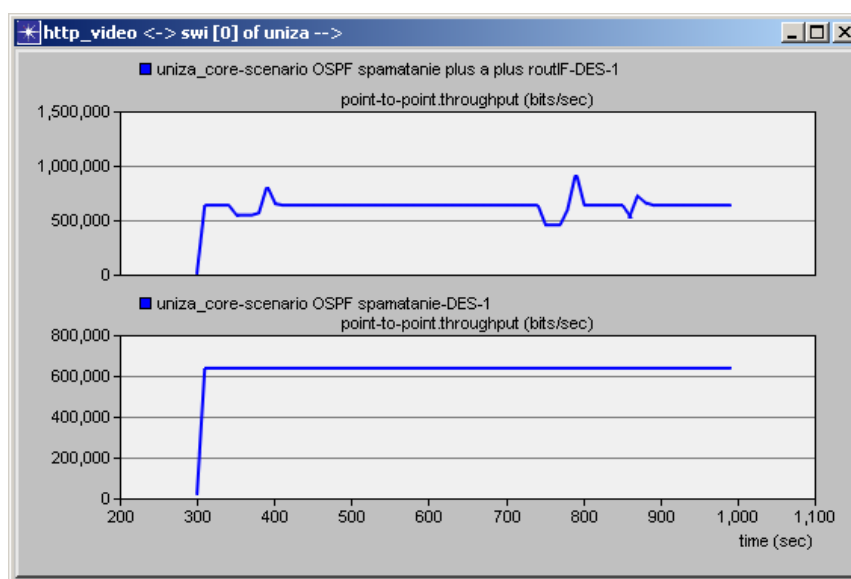
V čase 390 sekúnd dôjde k vypršaniu času (Router Dead interval), počas ktorého medzi fsi a ne112a z dôvodu zlyhania linky nedochádza k prenosu Hello paketov smerovacieho protokolu OSPF, ktoré dokazujú, že linka aj zariadenia sú v

poriadku. Po tomto čase je v prepínačoch dosiahnutý stav, že uvedená cesta nie je použiteľná. Preto je celá prevádzka z uzla sub_fsi smerujúca na http_server presmerovaná uzlom fsi, v smere na uzol ne112e. V rozmedzí 350 a 390 sekúnd nie je možné prepraviť prevádzku z uzla sub_fsi do uzla http_server a naopak.

V čase 450 dôjde k obnoveniu linky a k obnoveniu posielania Hello paketov. Posielanie údajov v sieti prebieha ako pred prvým zlyhaním. Nasledovné zlyhania majú veľmi podobný charakter priebehu, ako vyššie uvedené zlyhanie. Situáciu ozrejmjuje Obr. 0.9.



Obr. 0.9 Graf sieťovej prevádzky na vnútorných linkách medzi prepínačmi ne112a -> ne112e, ne112e -> fsi, ne112a > fsi. Správanie prevádzky v oboch smeroch liniek je okrem veľkosti prevádzky totožné. Os x – množstvo bitov prenesených linkou za sekundu v smere



Obr. 0.10 Graf vyťaženia linky pri serveri http_video, v smere od servera. Prvý graf zobrazuje prenosovú rýchlosť, ak sú medzi prepínačmi pripojené ďalšie prepínače, a teda nedostanú

okamžitú informáciu o výpadku linky. Druhý graf zobrazuje priame pripojenie prepínačov. Os x - čas v sekundách, os y - množstvo prenesených bitov za sekundu.

V prílohe C a D uvádzam ďalšie dva mnou vytvorené príklady modelovania sieťovej prevádzky.

ZÁVER

Prevádzkovatelia sietí pri uvažovaní o poskytovaní nových služieb a rozširovaní infraštruktúry stoja pred voľbou, ako sa ochrániť pred stratami spôsobenými zníženou kvalitou poskytovaných služieb. Jednou z možností, ako predísť takejto situácii je simulovať správanie siete. Účelom zriadenia siete Žilinskej univerzity nie je zvýšenie zisku, ale je integrálnou súčasťou vzdelávacieho procesu a administratívnej činnosti súvisiacej so vzdelávaním. Sieť sa postupne vyvíja a pribúdajú do nej aj nové služby.

Simulácia sieťovej prevádzky spôsobenej elektronickým vzdelávaním v chrbticovej sieti v programe OPNET Modeler sa na prvý pohľad zdá ako jednoduchý problém. Prvky siete a ich konfigurácia je s narastajúcim počtom zložitejšia. Pomerne veľký problém je aj určenie počtu prenesených údajov.

V práci som preveril pripravenosť chrbticovej siete pri záťaži spôsobenej elektronickým vzdelávaním, stručne som porovnal použité modely zariadení so skutočnosťou a testoval následky výpadku linky na spojenie.

PRÍLOHY

Príloha B: Kalkulačka bitovej rýchlosti

VIDEO	 	Aspect Ratio 4:3	Width 320	Height 240	Rate 29.97	307 } Video Kbps																																																		
		Motion Avg	Video Codec On2 VP6-E		Grid Quality: Best																																																			
AUDIO		Audio Codec MP3	Sampling Rate 22.050 kHz	Channels Mono	Quality Medium	40 } Audio Kbps																																																		
RESULTS		Encoding Method: 2-pass VBR				TOTAL FLV BITRATE: 347 Kbps RECOMMENDED CONNECTION SPEED: 416 Kbps																																																		
ADJUSTMENTS		<table border="0"><tr><td>Which type of content describes your video best?</td><td>Talking head</td><td>Mixed</td><td>Action trailer</td><td>N/A</td></tr><tr><td></td><td>☐</td><td>☐</td><td>☐</td><td>☒</td></tr><tr><td></td><td>Narration</td><td>Mixed</td><td>Music video</td><td>N/A</td></tr><tr><td></td><td>☐</td><td>☐</td><td>☐</td><td>☒</td></tr><tr><td>What is the soundtrack of your video?</td><td>6 or 7</td><td>8 - 9.0.48</td><td>9.0.115+</td><td></td></tr><tr><td></td><td>☐</td><td>☒</td><td>☐</td><td></td></tr><tr><td>Which Flash Player are you targeting?</td><td>HTTP</td><td>RTMP</td><td></td><td></td></tr><tr><td></td><td>☒</td><td>☐</td><td></td><td></td></tr><tr><td>How will you deploy the Flash Video file?</td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>☐</td><td>☐</td><td></td><td></td></tr></table>					Which type of content describes your video best?	Talking head	Mixed	Action trailer	N/A		☐	☐	☐	☒		Narration	Mixed	Music video	N/A		☐	☐	☐	☒	What is the soundtrack of your video?	6 or 7	8 - 9.0.48	9.0.115+			☐	☒	☐		Which Flash Player are you targeting?	HTTP	RTMP				☒	☐			How will you deploy the Flash Video file?						☐	☐		
Which type of content describes your video best?	Talking head	Mixed	Action trailer	N/A																																																				
	☐	☐	☐	☒																																																				
	Narration	Mixed	Music video	N/A																																																				
	☐	☐	☐	☒																																																				
What is the soundtrack of your video?	6 or 7	8 - 9.0.48	9.0.115+																																																					
	☐	☒	☐																																																					
Which Flash Player are you targeting?	HTTP	RTMP																																																						
	☒	☐																																																						
How will you deploy the Flash Video file?																																																								
	☐	☐																																																						

Príloha E: Inicializačný blok ACE skriptu

```
# kontrola hodnot pred spustenim prevadzky
#
# deklarovanie premennej velkost_suboru ako int
velkost_suboru = 400000
min_velkost_suboru = self.get_parameter('Min velkost suboru')
max_velkost_suboru = self.get_parameter('Max velkost suboru')

# nacistaj pocet suborov
pocet_suborov = self.get_parameter ('Pocet suborov')

# kontrola hodnot
if min_velkost_suboru <= 0:
    self.sim_message ('Hodnota atributu Min velkost suboru je mensia
ako 0.', 'Ukoncujem ulohu!')
    self.quit ()

if min_velkost_suboru > max_velkost_suboru:
    self.sim_message ('Hodnota atributu Min velkost suboru musi byt
mensia ako Max velkost suboru', 'Ukoncujem ulohu!')
    self.quit ()
```

```
if pocet_suborov <= 0:
    self.sim_message ('Hodnota atributu Pocet suborov musi byt
    vacsia ako 0', 'Ukoncujem ulohu!')
    self.quit ()
```

Príloha F: Hlavičkový blok ACE skriptu

```
# import kniznice random
import random
```

Príloha G: Funkčný blok ACE skriptu

```
# samotne vykonanie ulohy (TASK)
# action je API

# nacitaj velkost suboru
min_velkost_suboru = self.get_parameter('Min velkost suboru')
max_velkost_suboru = self.get_parameter('Max velkost suboru')

# pocet suborov = vykonanych cyklov
pocet_suborov = self.get_parameter ('Pocet suborov')

# zretazenie akcii, aby isli rovno za sebou ... potomok = predok
(nasledovnik, predchodca)
aktualna_akcia = action

# ...
for x in range (pocet_suborov):

    # nahodna velkost dalsieho suboru rovnomerne rozdelenie
    velkost_suboru = random.randint (400000,10000000)
    # vypis do konzoly
    # self.sim_message ('velkost suboru :',str(velkost_suboru))

    # vytvor novu spravu (v konverzácii), novy subor, odkial, kam, a
    predchodca aktualnej akcie
    sprava_novy_subor = self.create_message (velkost_suboru,
    'Server', 'Client', aktualna_akcia)

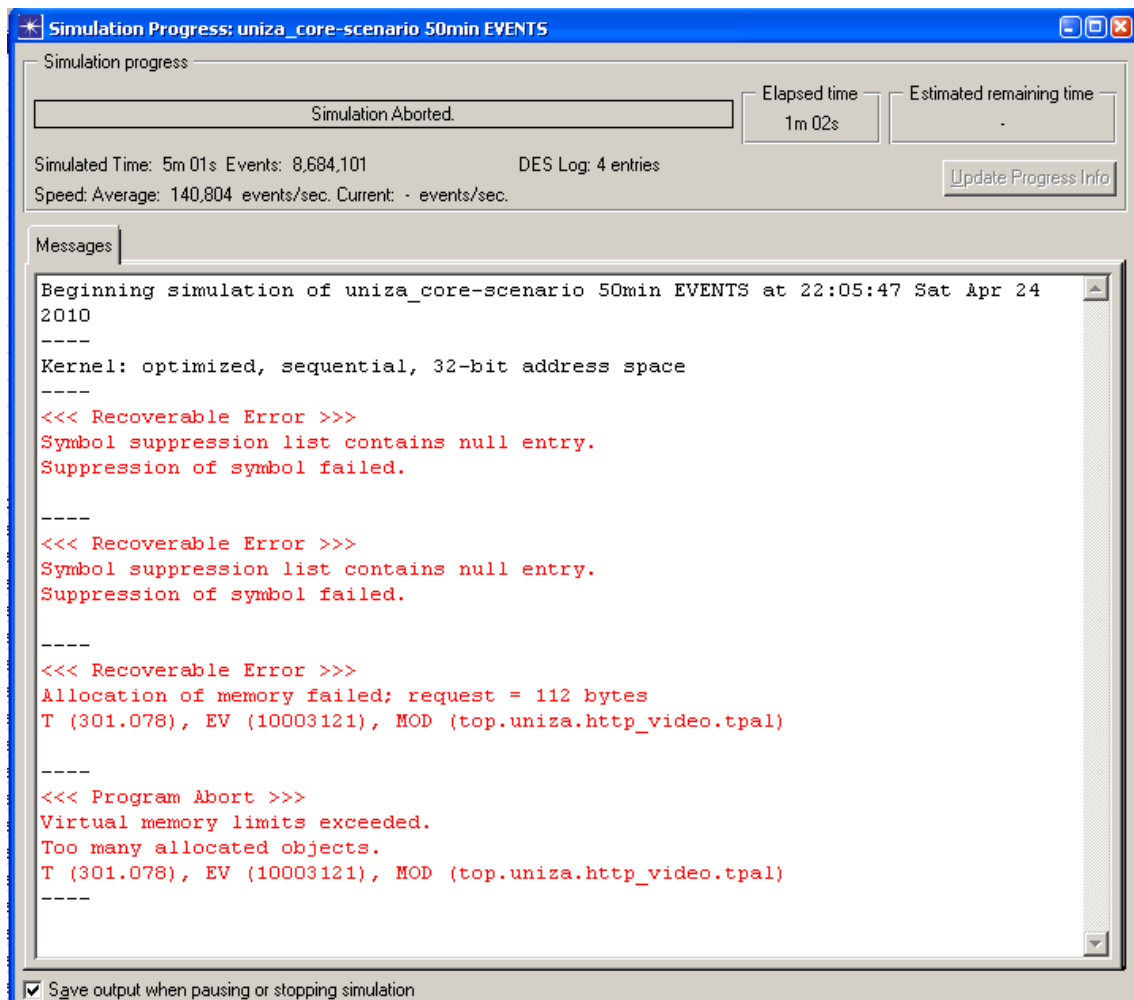
    # ako dlho trva spracovanie poziadavky na serveri 0.01 sekundy
    sprava_novy_subor.tier_delay = 0.01

    # vytvor spravu (ACK), predchodca tejto spravy bude novy_subor
    sprava_novy_subor_ack = self.create_message (100, 'Client',
    'Server', sprava_novy_subor)

    # nastav ACK ako aktualnu action
    aktualna_akcia = sprava_novy_subor_ack

# na konci odovzdaj riadenie
aktualna_akcia.take_over_children = 1
```

Príloha H: Chybové hlásenie simulácie



Príloha I: Stránka služby youtube.com

Vyhledat Procházet Nahrát

Přednáška V3C - Sítě ve světě virtualizace (část 7.)

[jprodelal](#) Počet videí: 44 Odběr



II 0:15 / 3:18 360p ↺ ↻ ⌵

[jprodelal](#) — 4. leden 2010 — Přednáška "Virtualizace, cluster a cloud computing" pořádaná společností OldanyGroup pro vysoké školy po celé Č...

30 zhlédnutí