



Programová podpora pre sieťový simulátor

Tímový projekt II

Členovia tímu: Bc. Hana Severínová, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Martin Svetlík

Študijný program: Počítačové a komunikačné systémy a siete

Študijný odbor: 9.2.4 Počítačové inžinierstvo

Miesto vypracovania: Ústav počítačových systémov a sietí, FIIT STU Bratislava

Vedúci projektu: Ing. Peter Magula

Akademický rok: 2010/2011

Obsah

Zoznam obrázkov	iv
Zoznam tabuliek.....	iv
Zoznam použitých skratiek	v
Zadanie	1
1 Úvod.....	2
2 Simulácia v počítačových sieťach.....	3
3 Simulátory počítačových sietí.....	5
3.1 Network simulator 2	5
3.1.1 Architektúra ns-2.....	5
3.1.2 Simulácie pomocou ns-2	7
3.1.3 Podporné nástroje pre ns-2.....	15
3.1.4 Zhodnotenie ns-2.....	21
3.2 Network simulator 3	23
3.2.1 Architektúra ns-3.....	23
3.2.2 Simulácia pomocou ns-3	24
3.2.3 Podporné nástroje pre ns-3	30
3.2.4 Zhodnotenie ns-3.....	34
3.3 Zhodnotenie analýzy simulátorov	35
4 Špecifikácia	37
5 Návrh.....	38
5.1 Používatelia	39
5.2 Popis komponentov	40
5.3 Návrh pozorovaných parametrov QoS	43
5.4 Návrh skriptov pre vyhodnotenie pozorovaných parametrov	44
5.4.1 Priepustnosť dátových tokov.....	44
5.4.2 Počet doručených paketov voči počtu odoslaných.....	44
5.4.3 Množstvo stratených paketov	45
5.4.4 Stratovosť paketov	45
5.4.5 Jednosmerné oneskorenie.....	45
5.4.6 Kolísanie oneskorenia	45
5.4.7 Réžia smerovania	46
5.4.8 Normalizovaná réžia smerovania	46
6 Funkcionality podporované prototypom	47
7 Funkcionality podporované finálnou verziou	50
8 Overenie správnosti výstupov	52

9	Záznam o používaní systém	53
10	Čo sme nestihli	54
11	Čo sme sa naučili.....	55
12	Zhodnotenie.....	56
13	Záver.....	57
14	Bibliografia.....	58
	Príloha A – riadenie projektu	A-1
	Príloha B – inštalčná príručka	B-1
	Príloha C – používateľská príručka.....	C-1
	Príloha D – technická dokumentácia.....	D-1
	Príloha E – elektronické médium	E-1

Zoznam obrázkov

Obrázok 1: Zjednodušená architektúra simulátora ns-2 (3)	6
Obrázok 2: Organizácia modulov ns-2.34 v rámci inštaláčného balíka allinone.....	6
Obrázok 3: Organizácie sieťového zásobníka mobilného uzla implementovaného v ns-2.34 ..	8
Obrázok 4: Architektúra programu NS2 TCP Evaluation Tool.....	15
Obrázok 5: Ukážka výstupného grafu simulácie programu NS2 TCP Evaluation Tool.....	16
Obrázok 6: Ukážky grafov z programu Gnuplot (9)	17
Obrázok 7: Ukážka programu XGRAPH (10)	18
Obrázok 8: Ukážky rozhrania programu Ns-nam (11).....	19
Obrázok 9: Ukážka rozhrania programu iNSpect v OS Ubuntu(12)	20
Obrázok 10: Ukážka grafu z programu DrawNetwork (13)	21
Obrázok 11: Schéma simulátora ns-3 (18).....	23
Obrázok 12: Príklad bezdrôtovej topológie	24
Obrázok 13: Grafické rozhranie programu NetAnim	30
Obrázok 14: Grafické rozhranie vizualizéra PyViz	31
Obrázok 15: Grafické rozhranie programu	32
Obrázok 16: Schéma štatistického systému (24).....	33
Obrázok 17: Porovnanie podpory protokolov v jadre ns-2 a ns-3 (28).....	35
Obrázok 18: Architektúra riešenia	39
Obrázok 19: Podrobná schéma riešenia	40
Obrázok 20: Registrácia a prihlásenie používateľa.....	47
Obrázok 21: Zaslanie simulácie na spracovanie	47
Obrázok 22: Stiahnutie výstupného súboru (trace file) zo servera	48
Obrázok 23: Zaslanie výstupného súboru na server za účelom vytvorenia grafického výstupu	48
Obrázok 24: Flow diagram systému.....	50
Obrázok 25: Prípady použitia systému.....	51

Zoznam tabuliek

Tabuľka 1: Zoznam protokolov podporovaných v jadre ns-2.....	22
Tabuľka 2: Zoznam protokolov podporovaných v jadre ns-3.....	34
Tabuľka 3: Zoznam pozorovaných parametrov QoS	43

Zoznam použitých skratiek

ACK	ACKnowledgement
AODV	Adhoc On-demand Distance Vector
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
CBR	Constant Bit Rate
CCNA	Cisco Certified Network Associate
CDMA	Code Division Multiple Access
CSMA	Carrier Sense Multiple Access
CTS	Clear To Send
DCF	Distributed Coordination Function
DiffServ	Differentiated Services
DSDV	Destination Sequence Distance Vector
DSR	Dynamic Source Routing
EPS	Encapsulated PostScript
FTP	File Transfer Protocol
GAF	Geographical Adaptive Fidelity
GOD	General Operations Director
GPL	General Public License
GPRS	General Packet Radio Service
GSM	Global System for Mobile communication
HDLC	High-level Data Link Control
HTML	HyperText Markup Language
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IMEP	Internet MANET Encapsulation Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISI	Information Sciences Institute
LDP	Label Distribution Protocol
LL	Link Layer
LTE	Long Term Evolution
MAC	Medium Access Control
EDCA	Enhanced Distributed Channel Access
MANET	Mobile Adhoc NETWORK
MIF	Maker Interchange Format
MPI	Message Passing Interface
MPLS	Multiprotocol Label Switching
NAM	Network AniMator
ns-2	Network Simulator 2 (NS2, NS-2)
ns-3	Network Simulator 3 (NS3, NS-3)
OFDM	Orthogonal Frequency-Division Multiplexing
OLSR	Optimized Link State Routing
OTcl	Object Tool command language
PCAP	Packet CAPture
PDF	Portable Document Format
PGM	Pragmatic General Multicast
PLM	Packet pair receiver-driven Layered Multicast
PQ	Priority Queuing

PUMA	Protocol for Unified Multicasting through Announcements
RAP	Resource Allocation Protocol
RED	Random Early Detection
REM	Random Exponential Marking
RIO	RED with In/Out bits
RLM	Receiver-driven Layered Multicast
RTP	Real-time Transport Protocol
RTS	Ready To Send
SCTP	Stream Control Transmission Protocol
SINR	Signal to Interference + Noise Ratio
SRM	Scalable Reliable Multicast
SRR	Shaped Round Robin
Tcl	Tool command language
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TFRC	TCP-Friendly Rate Control
Tk	Tool kit
TORA	Temporally Ordered Routing Algorithm
TTL	Time To Live
UAN	Underwater Acoustic Network
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VANET	Vehicular Ad-Hoc Network
VQ	Virtual Queue
WFQ	Weighted Fair Queuing
WiFi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
XCP	eXplicit Control Protocol
XML	eXtensible Markup Language

Zadanie

Analyzujte možnosti súčasných sieťových simulátorov pričom sa zamerajte na popredné simulátory ns-2 a ns-3. Analýzu upriamte na simulačné možnosti vybraných typov počítačových a komunikačných sietí so zameraním sa na prípravu a vyhodnotenie prebehnutých simulácií. Sústreďte sa na typické požiadavky vyhodnocovania simulácií vybraných typov sietí. Na základe vykonanej analýzy navrhните a následne implementujte systém, ktorý používateľovi umožní ľahšiu prípravu a vyhodnotenie prebehnutých simulácií. Systém musí poskytovať čo najuniverzálnejšie prostredie pre vyhodnocovanie a zobrazovanie priebehov a štatistík typických parametrov sledovaných pri simuláciách vybraných typov sietí.

1 Úvod

Tento dokument vznikol ako projektová dokumentácia k predmetu Tímový projekt na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Je súčasťou projektu s názvom Programová podpora pre sieťový simulátor, vedeného Ing. Petrom Magulom. Vypracovaný bol na Ústave počítačových systémov a sietí.

Účelom tohto dokumentu je podrobne oboznámiť čitateľa s problematikou simulácie počítačových sietí pomocou simulačných programov Network Simulator 2 a 3 (ns-2, ns-3). Uvedie čitateľa do funkcionality týchto programov, do štruktúry ich vstupných a výstupných súborov, ako aj možných prídavných rozšírení a programov na spracovanie výstupných súborov. Zároveň upozorní na výhody a nevýhody týchto programov, a odôvodní výber jedného z nich, ktorým sa budeme ďalej v tomto projekte zaoberať. Opíše vnútornú architektúru vybraného simulátora a čitateľovi uvedie návrh riešenia projektu v podobe vstupného a výstupného rozhrania pre daný simulátor. Celkové riešenie programovej podpory pre sieťový simulátor opíše vo viacerých kapitolách, kde sa postupne bude venovať návrhu, špecifikácii aj implementácii navrhovaného riešenia, zároveň spomenie porovnateľné riešenia dostupné na trhu a upozorní na ich silné aj slabé stránky.

Dokument obsahuje v prvej časti analýzu simulátorov ns-2 a ns-3, špecifikáciu a hrubý návrh riešenia. Kapitoly 0 a 2 predstavujú úvod do problematiky simulácií. Nasledujúce kapitoly sú analýzou simulátorov ns-2 a ns-3. Záver je venovaný zhodnoteniu analýzy. V kapitole 4 sú špecifikované požiadavky na systém. V kapitole 4 (Špecifikácia) je uvedený hrubý návrh riešenia. Kapitola 5 predstavuje podrobný návrh riešenia v ktorom sú bližšie opísané jednotlivé moduly, ich vstupy a výstupy. Kapitoly 6 a 7 sa venujú funkcionalite navrhnutých modulov. Ďalšie kapitoly sú venované testovaniu systému a zhodnoteniu celkového výsledku, jeho silných aj slabších stránok. V kapitolách 10 a 11 je zhrnutý prínos tejto práce pre jednotlivých členov tímu a tiež je načrtnutá možnosť pokračovania v tejto práci a doplnenie systému o ďalšie funkcie. K práci je pripojená dokumentácia k riadeniu projektu, používateľská a inštalačná príručka, technická dokumentácia a tiež elektronické médium obsahujúce výsledný produkt.

2 Simulácia v počítačových sieťach

S príchodom informačných technológií sa spoločnosti otvorili široké možnosti v mnohých smeroch, počítače uľahčujú kancelársku prácu, komunikáciu naprieč kontinentmi, poskytujú zábavu aj vzdelávanie a uľahčujú zložité výpočty, ktoré by bez modernej výpočtovej techniky trvali neporovnateľne dlhší čas. Výpočtová kapacita informačných technológií sa dá využiť rôznymi spôsobmi, od vytvárania animovaných filmov, až po meteorologické predpovede. Skutočná sila moderných počítačov spočíva v obrovskom množstve dát, ktoré sú schopné spracovávať. V tomto dokumente sa budeme zaoberať jej využitím na vytváranie, beh a vyhodnocovanie simulácií.

Už od dávnych čias sa rôzne simulácie používali na najrozmanitejšie účely, veliteľ vojska si pred bitkou premyslel rôzne scenáre, akými sa boj môže uberať, a následne zvolil stratégiu, pred stavbou veterného mlynu bolo nutné vypočítať, akým smerom ho treba otočiť tak, aby vietor dul čo najviac do vrtule, a pri búraní budov je nutné vypočítať, kam treba umiestniť nálože, aby budova spadla smerom, ktorým nespôsobí škodu. Pri všetkých týchto úkonoch bolo treba nejakým spôsobom predpokladať, alebo vypočítať, ako sa bude situácia v čase vyvíjať. Odhadnúť smer, ktorým treba natočiť veterný mlyn nepotrebuje zložité výpočty. No výpočet, ktorý určí kam treba umiestniť nálože vo výškovom dome, ktorý stojí v susedstve školy, musí byť úplne presný. Vzhľadom na to, že na pád budovy vplyva obrovské množstvo najrôznejších faktorov, aj výpočet určujúci polohu nálože je značne zložitý a bez použitia výpočtovej techniky značne zdĺhavý, ba až nerealizovateľný. Príchodom moderných informačných technológií sa však situácia obrátila a na všetky podobné úkony sa používajú simulácie.

Prečo je teda potrebné simulovať? Na to, aby sme si mohli odpovedať na túto otázku si najprv musíme povedať, čo to simulácia je. Simulácia je v podstate prevedenie udalosti, alebo skupiny udalostí podľa daných kritérií. Existujú tri druhy simulácie:

- živá – je realizovaná pomocou ľudí a/alebo zariadení, ktoré v reálnom čase robia činnosť v prostredí, v ktorom sa táto normálne praktizuje. Príkladom sú napríklad vojnové hry, kde skupina ľudí replikuje nejakú historickú bitku.
- konštrukčná – namiesto času je “poháňaná” postupnosťou udalostí. Príkladom je simulácia tornáda, ktoré vzniká určitou postupnosťou a vplyvom rôznych veličín, ako napríklad vietor, tlak, teplota.
- virtuálna – je realizovaná pomocou výpočtovej techniky. Čas prebieha v diskrétnych krokoch, čím dovoľuje používateľovi zamerať sa na dôležité udalosti(1).

V tejto práci sa budeme zaoberať práve tretím typom, teda virtuálnou simuláciou.

Virtuálna simulácia poskytuje používateľovi pohľad na postupnosť udalostí, ktoré sa udejú za daných podmienok v danom prostredí. Dovoľuje používateľovi meniť vstupné dáta, umožňuje mu upriamiť sa iba na niektoré veličiny v určitom čase a keďže sa simulácia deje v počítači, dá sa reprodukovať neobmedzený počet krát. Jednou z hlavných výhod virtuálnej, teda počítačovej simulácie je, že na rozdiel od živej, alebo konštrukčnej simulácie v nej čas beží podľa toho, ako ho používateľ nastaví. Tým umožňuje detailné skúmanie vybraných udalostí, faktorov, či zmien. V súčasnosti sa virtuálne simulácie používajú v rôznych oblastiach vedy, výskumu, ale aj bežného života. V tomto dokumente sa budeme zaoberať simuláciou počítačových sietí a jej vyhodnocovaním.

Simulácia počítačových sietí v sebe zahŕňa tri prvky: topológiu siete, teda usporiadanie a vzájomné prepojenie jednotlivých uzlov siete, použité protokoly a tok dát medzi jednotlivými uzlami. Tieto tri časti tvoria vstupné dáta simulácie, na základe ktorých

potom simulačný program vypočíta, teda nasimuluje, časový priebeh udalostí, ktoré sa v simulovanom modeli udejú. Výstupom simulácie môže byť napríklad textový súbor obsahujúci postupnosť udalostí, ako sú napríklad príjem datagramu uzlom, zahodenie paketu, či zaradenie rámca do rady. Simulácia počítačovej siete nám teda povie, čo sa v danej sieti udeje v určitom časovom úseku.

Simulovanie počítačových sietí má množstvo výhod a kladných dôsledkov, či už pre sieťového administrátora, alebo aj bežného používateľa. Hlavnou výhodou, pre ktorú sa sieťové simulátory používajú je, že môžeme otestovať funkčnosť a správnosť implementácie navrhutej počítačovej siete bez toho, aby sme si museli zadovážiť všetok potrebný hardvér a softvér. Simulátor nám nahradí drahé testovanie skutočných zariadení a značne skráti dobu implementácie navrhutej sieťovej architektúry, keďže všetky nastavenia vykonávame v simulátore a nie vo viacerých sieťových prvkoch a zariadeniach. Zároveň môžeme celý návrh otestovať dôkladnejšie, keďže si môžeme všetky udalosti, ktoré sa v navrhutej sieti prezerat' postupne (nie v reálnom čase, kedy by sme jednotlivé veličiny nestíhali sledovať) a opakovane (vd'aka zaznamenaniu výstupu simulácie). Jednoducho sa tiež môžeme zamerať iba na tie veličiny a udalosti v sieti, ktoré nás zaujímajú, čím sa testovanie stáva efektívnejšie.

Ďalšou výhodou je samotná virtuálnosť počítačovej simulácie, teda to, že na beh simulácie bude mať vplyv iba to, čo nasimulujeme. Ak teda napríklad v sieti nebudeme simulovať rôzne rušenia signálu, či šum bezdrôtového signálu, tento na činnosť siete nebude mať vplyv, na rozdiel od reálnej siete, kde šum úplne eliminovať nevieme. Táto vlastnosť testovanie sieťovej architektúry pomocou počítačovej simulácie značne zvyhodňuje voči testovaniu siete pomocou reálnych zariadení, pretože napríklad pri hľadaní zdroja napríklad zlého smerovania sa nemusíme zaoberať chybami spôsobenými vonkajším prostredím, ktoré by nám hľadanie sťažovali.

Ďalším nemenej významným pozitívom simulácií počítačových sietí je ich vzdelávacia funkcia, keď sa môžu využívať na štúdium a lepšie pochopenie problematiky počítačových sietí. Túto funkciu sa budeme v našom návrhu podporiť intuitívnym a prehľadným rozhraním pre generovanie vstupných súborov simulácie, zároveň však aj prehľadným výstupom simulácie.

V súčasnosti je na internete dostupných viacero programov, poskytujúcich simuláciu počítačových sietí. Medzi najznámejšie špičkové simulátory, ktoré sa ustálili ako štandard v oblasti simulácií, patria simulátory ns-2 a ns-3, analýzou ktorých sa v tomto dokumente budeme zaoberať v najbližších kapitolách.

3 Simulátory počítačových sietí

V nasledujúcej kapitole si priblížime vzhľadom na zadanie projektu simulátory Network simulator 2 (ns-2) a Network simulator 3 (ns-3). Po vzájomnej dohode s vedúcim projektu sme sa rozhodli náš projekt orientovať predovšetkým na simuláciu bezdrôtových sietí, ktorých nasadenie v rôznych oblastiach života sa stáva čoraz väčším trendom. Z tohto dôvodu budú v nasledujúcom texte zhodnotené možnosti vybraných simulátorov najmä v oblasti simulácie bezdrôtových sietí.

3.1 Network simulator 2

Simulátor ns-2 (*Network simulator 2*) bol vyvinutý na University of California, Berkley v roku 1996. Na jeho zdokonaľovaní sa neustále pracuje, pričom pribúdajú nové moduly a rozširuje sa oblasť jeho využitia. Od vzniku prešiel simulátor ns-2 rozsiahlym vývojom a v súčasnosti patrí k špičke v oblasti sieťových simulácií. Od roku 2006 sa paralelne vyvíja simulátor ns-3 (*Network simulator 3*), ktorý bol predstavený v roku 2008 ako eventuálna náhrada populárneho ns-2 simulátora. Starší ns-2 simulátor sa používa už dlhé roky a počas tejto doby z neho boli odstránené mnohé chyby a nedostatky. Súčasná verzia ns-2.34 (predstavená 17.6.2009), je odladená a stabilná, čo je v porovnaní s ns-3 veľkou výhodou. Je dôležité poznamenať, že napriek existencii simulátora ns-3, vývoj ns-2 nebol doposiaľ zastavený a v súčasnosti sa pripravuje verzia ns-2.35 (2).

3.1.1 Architektúra ns-2

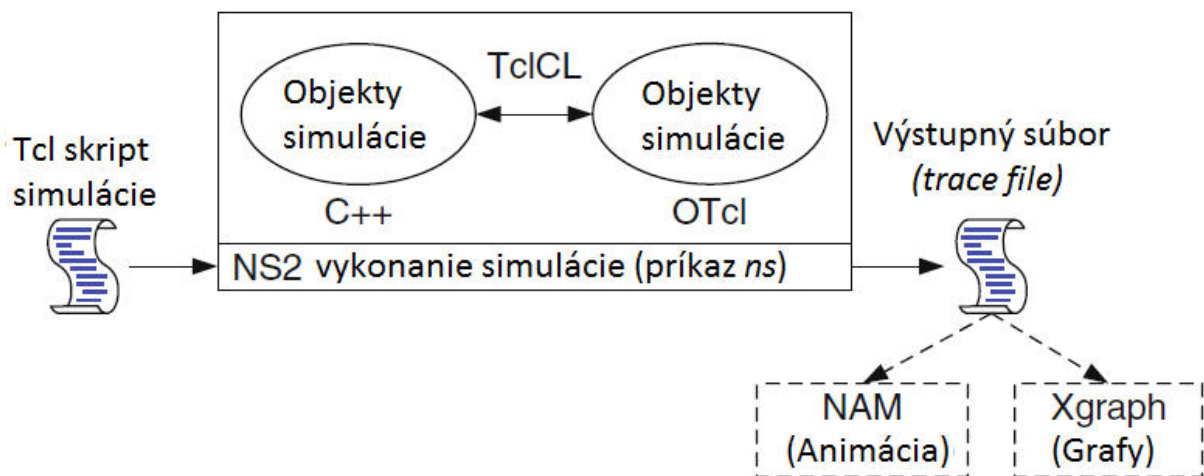
Ns-2 je simulátorom diskretných udalostí a bol vyvinutý pre testovanie a vyhodnocovanie existujúcich prístupov používaných v počítačových sieťach ako aj pre vývoj nových prístupov. Simulátor umožňuje simuláciu TCP/IP sietí, smerovania ako aj rôznych iných protokolov tak v pevných, ako aj v bezdrôtových sieťach.

Simulátor je predmetom licencie GNU GPL (*General Public License*), čo znamená, že je voľne šíriteľný, pričom sú k dispozícii aj jeho zdrojové kódy, ktoré je možné pre vlastné potreby upravovať. Jadro simulátora je napísané v jazyku C++. Keďže programovanie v jazyku C++ je pomerne zložité a časovo náročné, simulačné scenáre sa zvyčajne píšú v objektivej verzii jazyka Tcl (*Tool command language*), OTcl (*Object Tcl*). Skript napísaný v jazyku Tcl je zároveň aj jediným vstupom pre ns-2, potrebným pre uskutočnenie simulácie (3).

Keďže ns-2 bol vyvinutý na platforme Unix (FreeBSD, Linux, SunOS, Solaris), odporúča sa jeho inštalácia pod jedným zo spomenutých operačných systémov. Existuje aj možnosť inštalácie ns-2 pod operačným systémom Windows s využitím programu Cygwin, ktorý dokáže simulovať prostredie operačného systému Linux. Cygwin však v žiadnom prípade nie je úplnou náhradou Linuxového jadra a zďaleka nepodporuje všetky jeho funkcie a aplikácie vyvinuté pre Linux (4).

Samotné jadro ns-2 tvoria dva kľúčové jazyky: C++ a OTcl. V jazyku C++ sú definované vnútorné mechanizmy simulačných objektov. OTcl slúži na nastavenie simulačných objektov a ich parametrov a tiež na plánovanie diskretných udalostí. Tieto dva jazyky sú spolu prepojené pomocou TclCL, ktorý zabezpečí mapovanie C++ objektov s OTcl parametrami. Objekty v OTcl sú len jednoduché premenné, ktorých funkcionality zabezpečujú až C++ objekty, na ktoré sú mapované. Jednotlivé OTcl objekty tvoria doménu,

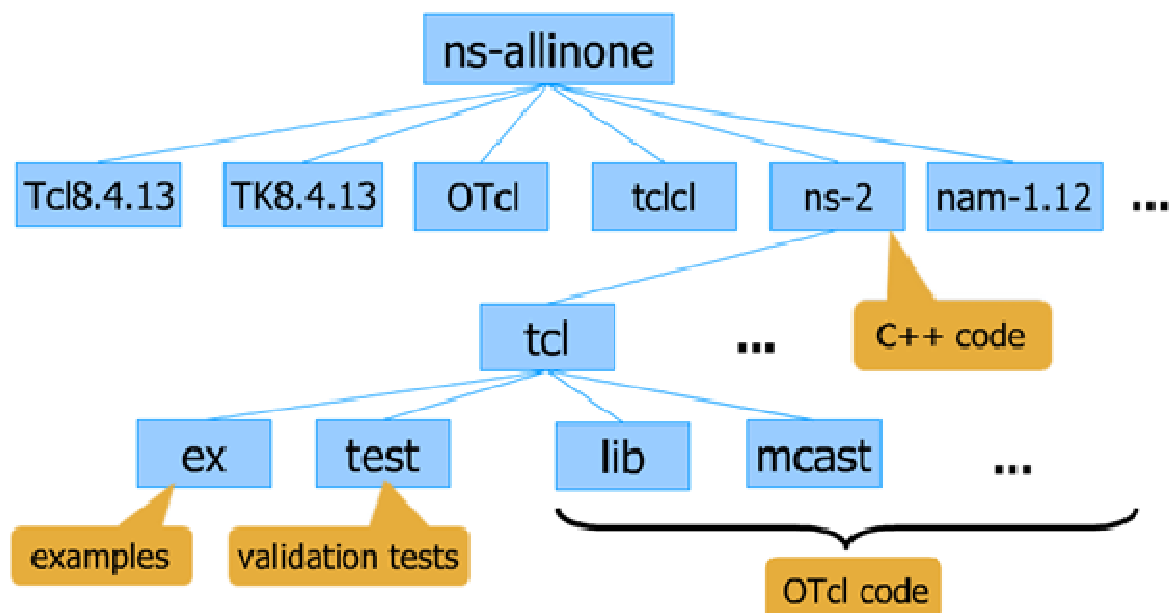
ktorá umožňuje jednoduchšiu interakciu používateľa s inštanciami C++ objektov. Zjednodušenú architektúru simulátora ilustruje Obrázok 1.



Obrázok 1: Zjednodušená architektúra simulátora ns-2 (3)

Simulátor ns-2 sa spúšťa príkazom *ns*, ktorého argumentom je názov vstupného Tcl skriptu. V Tcl skripte je potrebné definovať sieťovú topológiu, dátové toky i sieťové prístupy, ktoré zamýšľame simulovať. Okrem toho je potrebné v tomto skripte definovať informácie, ktoré sa majú zaznamenávať do výstupných súborov v stanovených časových intervaloch. Samozrejme nesmie chýbať časovanie a riadenie priebehu celej simulácie. Simulátor ns-2 ponúka štandardný výstup v textovej podobe. Textový výstupný súbor je potrebné analyzovať a na základe zistených informácií je možné vytvoriť grafy alebo animáciu simulácie.

Hierarchické usporiadanie jednotlivých modulov simulátora ns-2 ilustruje Obrázok 2.



Obrázok 2: Organizácia modulov ns-2.34 v rámci inštalačného balíka allinone

3.1.2 Simulácie pomocou ns-2

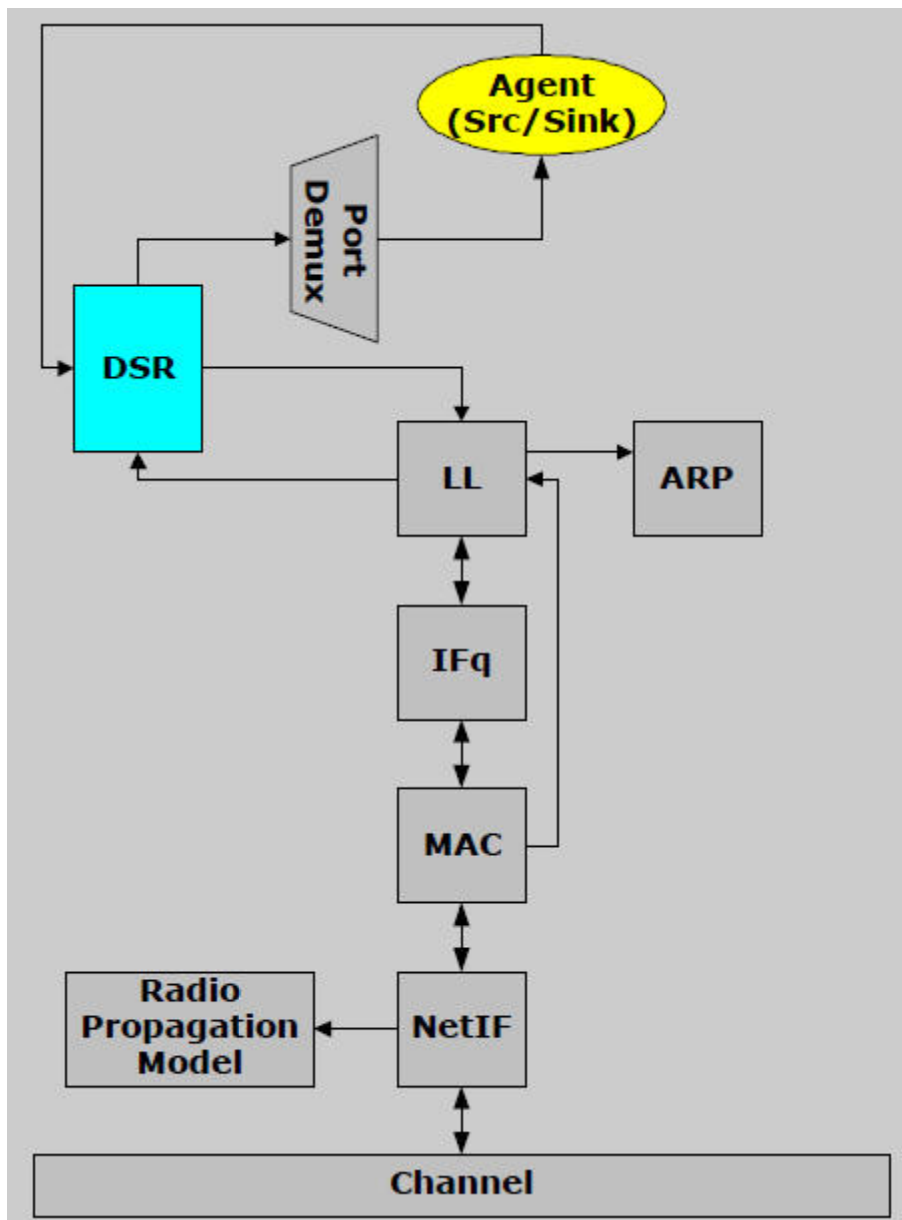
Nasledujúca kapitola je venovaná možnostiam simulátora ns-2 v oblasti simulácie bezdrôtových sietí, na ktoré je projekt prevažne zameraný. Podrobnejšie sú tu rozobrané jednotlivé entity prítomné v tomto druhu simulácií, podpora protokolov druhej a tretej vrstvy RM OSI a tiež špeciálny formát výstupného súboru určený pre bezdrôtové siete.

Mobilný uzol

Mobilný uzol je reprezentovaný triedou *MobileNode* odvodenej od triedy *Node*. Je to teda základný uzol v sieti s pridanou funkcionalitou ako pohyblivosť v priestore, počúvanie a vysielanie na kanáloch, periodické oznamovanie pozície, ohraničenie topológie a podobne. Táto prídavná funkcionalita umožňuje prostredníctvom takýchto uzlov simulovať mobilné bezdrôtové prostredie (5). Sieťový zásobník mobilného uzla je znázorňuje

Obrázok 3 a tvoria ho nasledovné moduly :

1. Agent (AGT)
 - Reprezentuje transportnú a aplikačnú vrstvu
 - Zabezpečuje generovanie a prijímanie sieťovej komunikácie
 - Podpora rôznych typov komunikácie (TCP,CBR,FTP a iné)
2. Smerovací agent (RTagent)
 - Zabezpečuje smerovanie podľa vybraného protokolu
 - Vypočítava najvhodnejšie cesty
3. Linková vrstva (LL)
 - Zabezpečuje funkcionalitu protokolu linkovej vrstvy
 - Zabezpečuje fragmentáciu a skladanie rámcov
4. ARP modul
 - Zabezpečuje preklad IP adres na MAC adresy
 - Je priamo pripojený na linkovú vrstvu
5. MAC vrstva
 - Realizuje funkcionalitu podľa štandardu IEEE 802.11
6. Sieťové rozhranie (netIF)
 - Reprezentuje fyzické rozhranie zabezpečujúce prístup ku kanálu
 - Simuluje integritu signálu, kolízie a chyby pri vysielaní
 - Vyslaným paketom pridá informáciu o vlnovej dĺžke a sile signálu
7. Prioritný rad rozhrania (netQ)
 - Obsahuje prioritný rad pre smerovacie protokoly
 - Umožňuje filtrovanie paketov
8. Model šírenia signálu
 - Simuluje útlm signálu v závislosti od vzdialenosti
 - Určuje rozlíšiteľnosť signálu na základe jeho kvalitatívnych vlastností
9. Anténa
 - Spravidla všesmerová anténa.



Obrázok 3: Organizácie sieťového zásobníka mobilného uzla implementovaného v ns-2.34

Kanál

Pri bezdrôtových sieťach sa namiesto klasických prenosových liniek na komunikáciu využíva prenosový kanál. Prenosový kanál simuluje vysielenie paketu na fyzickej vrstve, pričom realizuje mechanizmy súperenia o zdieľané médium. Na detekciu kolízií sa využíva identifikácia nosnej vlny. Pri zaznamenaní kolízie sa nastaví príznak, prostredníctvom ktorého sa o kolízii informuje protokol vyššej vrstvy (MAC), ktorý sa s ňou vysporiada.

Nakoľko čas vysielenia paketu je daný jeho veľkosťou a rýchlosťou modulácie závislou od konkrétneho rozhrania (MAC), objekt kanála jednoducho nastaví signál vyťaženia po dobu požadovanú vrstvou MAC. Taktiež naplánuje doručenie paketu do cieľa za čas vysieleniasčítaný s časom prenosu.

Smerovacie protokoly

Simulátor ns-2 podporuje nasledovné ad-hoc smerovacie protokoly :

- DSDV (*Destination Sequence Distance Vector*)
- AODV (*Adhoc On-demand Distance Vector*)
- TORA (*Temporally Ordered Routing Algorithm*)
- PUMA (*Protocol for Unified Multicasting through Announcements*)
- DSR (*Dynamic Source Routing*)

MAC protokoly

Simulátor ns-2 podporuje dva základné MAC štandardy pre bezdrôtové siete, pričom pre štandard 802.11 existuje aj niekoľko rozšírení :

1. IEEE 802.11

a. 802.11 DCF (*Distributed Coordination Function*)

- Model vysielania typu unicast RTS/CTS/DATA/ACK
- Model vysielania typu broadcast DATA
- Možnosť prepojenia bezdrôtových a klasických sietí

b. 802.11 infraštruktúrne rozšírenia

- Aktívne a pasívne skenovanie
- Autentifikácia
- Komunikácia medzi prístupovými bodmi
- Podpora mobility (*handoff*)

c. 802.11Ext (*extended*)

- Štruktúrovaný návrh modulov funkcionality na MAC vrstve: vysielanie, prijímanie, koordinácia vysielania.
- Monitorovanie stavu kanála, riadenie stiahnutia sa z vysielania
- Súhrnné výpočty SINR (*Signal to Interference + Noise Ratio*)
- Schopnosť zachytávania rámcov
- Podpora viacerých schém modulácie
- Sledovanie strát dátových jednotiek na fyzickej vrstve
- Nakagamiho model slabnutia signálu

d. 802.11mr (*multirate*)

- Podpora viacerých prístupov na fyzickej vrstve
- Simulovanie vysielania na rôznych frekvenciách
- Modulácia a kódovanie signálu podľa štandardu IEEE802.11b/g.
- Model chybovosti paketov založený na SINR, zohľadňujúci silu signálu, šum a rušenie.

2. TDMA (*Time Division Multiple Access*)

- Alokácia časových slotov pre komunikujúce uzly, pričom množina slotov tvorí časový rámec
- Rozpoznávanie časových rámcov na základe preamble

3.1.2.1 Formát výstupného súboru

Pre simuláciu bezdrôtových sietí je v simulátore ns-2 implementovaný tzv. nový formát výstupného súboru. Tento formát sa aktivuje príkazom `$ns use-newtrace` a pozostáva z nasledovných položiek :

1. Typ udalosti v danom uzle :

s odoslanie paketu
r prijatie paketu
d zahodenie paketu
f preposlanie paketu

2. Všeobecná značka :

-t čas
-t *(globálne nastavenie)

3. Značka nastavenia uzla :

-Ni: identifikátor uzla
-Nx: x-ová súradnica uzla
-Ny: y-ová súradnica uzla
-Nz: z-ová súradnica
-Ne: energetická úroveň uzla
-NI: úroveň sledovania uzla (napr.: AGT - agent, RTR - router, MAC)
-Nw: dôvod udalosti :

"END"	DROP_END_OF_SIMULATION	(ukončenie simulácie)
"COL"	DROP_MAC_COLLISION	(kolízia na MAC vrstve)
"DUP"	DROP_MAC_DUPLICATE	(kolízia MAC adries)
"ERR"	DROP_MAC_PACKET_ERROR	(porušenie paketu)
"RET"	DROP_MAC_RETRY_COUNT_EXCEEDED	
"STA"	DROP_MAC_INVALID_STATE	(neplatný stav)
"BSY"	DROP_MAC_BUSY	(uzol je zaneprázdnený)
"NRTE"	DROP_RTR_NO_ROUTE	(nedostupnosť cesty)
"LOOP"	DROP_RTR_ROUTE_LOOP	(smerovacia slučka)
"TTL"	DROP_RTR_TTL	(vypršanie TTL)
"TOUT"	DROP_RTR_QTIMEOUT	(vypršanie platnosti paketu)
"CBK"	DROP_RTR_MAC_CALLBACK	
"IFQ"	DROP_IFQ_QFULL	(plný rad)
"ARP"	DROP_IFQ_ARP_FULL	(zahodené protokolom ARP)
"OUT"	DROP_OUTSIDE_SUBNET	(správa z inej domény)

4. Informácia o pakete na sieťovej vrstve :

-Is: zdrojová IP adresa . zdrojový port
-Id: cieľová adresa . cieľový port
-It: typ paketu
-Il: veľkosť paketu
-If: identifikátor toku
-Ii: unikátny identifikátor
-Iv: hodnota časovača TTL(*Time To Live*)

5. Informácia o nasledujúcom skoku :
- Hs**: identifikátor nasledujúceho uzla
 - Hd**: identifikátor pre nasledujúci uzol vzhľadom na cieľ
6. Informácia o pakete na linkovej vrstve :
- Ma**: doba trvania
 - Md**: cieľová MAC adresa
 - Ms**: zdrojová MAC adresa
 - Mt**: typ rámca
7. Informácia o pakete na aplikačnej vrstve :
- P arp** ARP (*Address Resolution Protocol*)
 - Po**: ARP dotaz/odpoveď
 - Pm**: zdrojová MAC adresa
 - Ps**: zdrojová IP adresa
 - Pa**: cieľová MAC adresa
 - Pd**: cieľová IP adresa
 - P dsr** DSR (*Dynamic Source Routing*)
 - Pn**: počet prejdených uzlov
 - Pq**: príznak pre smerovací dotaz
 - Pi**: sekvenčné číslo smerovacieho dotazu
 - Pp**: príznak pre odpoveď na smerovací dotaz
 - Pl**: dĺžka odpovede
 - Pe**: zdroj zo zdroja smerovanie/cieľ z cieľa smerovanie
 - Pw**: chybový príznak
 - Pm**: počet chýb
 - Pc**: komu oznámiť správu
 - Pb**: chyba pri prenose z A do B
 - P cbr** CBR (*Constant Bit Rate*)
 - Pi**: sekvenčné číslo
 - Pf**: počet preposlaní paketu
 - Po**: optimálny počet preposlaní
 - P tcp** Informácie o TCP toku
 - Ps**: sekvenčné číslo
 - Pa**: číslo potvrdenia ACK
 - Pf**: počet preposlaní paketu
 - Po**: optimálny počet preposlaní

3.1.2.2 Konfigurácia simulácie bezdrôtovej siete

V tejto kapitole si uvedieme, ktoré rozdiely je potrebné definovať pri bezdrôtových sieťach. NS-2 zabezpečuje mobilitu pre jednotlivé komunikujúce stanice. Pri simulácií takýchto sietí je potrebné definovať viacero parametrov, ktoré pri klasických pevných sieťach nie je potrebné definovať. Tieto parametre sú napríklad anténa, smerovací protokol, propagačný rádio model, kanál cez ktorý budú zariadenia komunikovať, typ sieťového rozhrania, typ MAC adresy a iné. Tieto parametre aj s popisom sú uvedené v nasledujúcom kóde(6).

```
set val(chan) Channel/WirelessChannel # typ kanálu
set val(prop) Propagation/TwoRayGround # model šírenia
signálu
set val(ant) Antenna/OmniAntenna # typ antény
set val(ll) LL # typ linkovej vrstvy
set val(ifq) Queue/DropTail/PriQueue # typ radu na
rozhraní
set val(ifqlen) 50 # dĺžka radu
set val(netif) Phy/WirelessPhy # typ sieťového
rozhrania
set val(mac) Mac/802_11 # typ MAC
set val(rp) DSDV # smerovací protokol
set val(nn) 2 # počet uzlov
```

Keďže ide o mobilnú sieť, je potrebné zadefinovať taktiež rozsah mobilnej siete. V našej ukážke si zadefinujeme rozsah 500m × 500m. Pri písaní Tcl skriptu bude najjednoduchšie, ak si najprv vytvoríme objekt a potom definujeme jeho hodnotu.

```
set topo [new Topography]
topo load_flatgrid 500 500
```

Číslo 500 sa nachádza ako parameter dvakrát, pretože definujeme najprv x-ovú a potom y-ovú os. V ďalšom kroku je nutné vytvoriť objekt označený ako „GOD“.

```
create-god $val(nn)
```

Objekt „GOD“ sa používa na uloženie informácií o danej simulácii. Najmä o type siete, prostredia, jednotlivých zariadeniach a taktiež aj o potrebných skokoch (hops) medzi zariadeniami pri smerovaní. Toto sa vypočítava počas behu simulácie, a preto to môže byť veľmi náročné na čas. Tento objekt je volaný interne objektom MAC. Globálne však známy byť nemôže. V tomto príklade zatiaľ tento objekt využívať nebudeme, avšak je potrebné ho vytvoriť.

Predtým, než vytvoríme samotné zariadenia, musíme ich najskôr nakonfigurovať. Pod konfiguráciou rozumieme zadanie parametrov, ktoré boli vyššie spomenuté a sú potrebné pri bezdrôtových sieťach. Príklad konfigurácie je uvedený v nasledujúcom kóde (6).

```

$ns_ node-config
-addressingType      flat/hierarchical/expanded
-adhocRouting        DSDV or DSR or TORA
-llType              LL
-macType              Mac/802_11
-propType             Propagation/TwoRayGround"
-ifqType              "Queue/DropTail/PriQueue"
-ifqLen              50
-phyType              "Phy/WirelessPhy"
-antType              "Antenna/OmniAntenna"
-channelType          "Channel/WirelessChannel"
-topoInstance         $topo
-energyModel           "EnergyModel"
-initialEnergy        (in Joules)
-rxPower              (in W)
-txPower              (in W)
-agentTrace           ON or OFF
-routerTrace          ON or OFF
-macTrace             ON or OFF
-movementTrace        ON or OFF

```

Väčšina parametrov má nedefinovanú základnú hodnotu (NULL).Následne si jednotlivé zariadenia vytvoríme. V našom prípade vytvárame dve zariadenia nasledovným kódom:

```

for {set i 0} {$i < $val(nn) } {incr i} {
    set node_($i) [$ns_ node ]
    $node_($i) random-motion 0
}

```

Tým, že máme mobilnú sieť, vieme vopred definovať, kde jednotlivé zariadenia budú začínať a ako sa budú v simulácií pohybovať. Toto je najväčší rozdiel oproti simulácií pevných sietí. Definovali sme si sieť 500m x500m. Je teda potrebné definovať začiatkové pozície napríklad nasledovným kódom:

```

$node_(0) set X_ 5.0
$node_(0) set Y_ 2.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 390.0
$node_(1) set Y_ 385.0
$node_(1) set Z_ 0.0

```

A následne definovať pohyb zariadenia:

```

$ns_ at 50.0 "$node_(1) setdest 25.0 20.0 15.0"
$ns_ at 10.0 "$node_(0) setdest 20.0 18.0 1.0"

# Node_(1) sa následne začne pohybovať od node_(2)
$ns_ at 100.0 "$node_(1) setdest 490.0 480.0 15.0"

```

\$ns_ at 50.0 "\$node_(1) setdest 25.0 20.0 15.0"znamená, že v čase 50.0 sekúnd sa node1 začne pohybovať smerom k cieľu (x=25,y=20) a to rýchlosťou 15m/s.

Ďalej je potrebné zdefinovať tok dát medzi zariadeniami a iné parametre. Tie sa definujú aj pri pevných sieťach. Účelom tohto dokumentu je priblížiť nastavenia a nové možnosti pri bezdrôtových sieťach a preto sa ďalšími nastaveniami zaoberať nebudeme.

Simulátor ns-2 nám ponúka ešte ďalšie možnosti. Zadávať vždy pohyb jednotlivých zariadení manuálne by bolo veľmi zdĺhavé a nie príliš triviálne. Preto využijeme vopred pripravený generátor náhodného pohybu mobilných zariadení a taktiež generátor náhodného toku dát.

Tieto generátory nám umožnia simulovať sieť s vopred nakonfigurovanými zariadeniami, ktoré sa pohybujú náhodne a medzi nimi existuje tok dát. Pakety sa medzi jednotlivými zariadeniami posielajú, prijímajú a preposielajú. Cesty, ktorými budú pakety putovať je možné vygenerovať skriptom TCP/CBR(angl.: constant bit rate, sk.: konštantná prenosová rýchlosť).

Simulátor ns-2 ponúka samozrejme niekoľko vopred vygenerovaných skriptov, avšak tieto skripty si môžeme vygenerovať aj my sami a to nasledovným spôsobom.

Najprv si uvedieme aké možnosti máme a potom ukázkový príkaz.

```
ns cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]
```

- -type: typ protokolu
- -nn počet mobilných zariadení
- -seed počiatok
- -mc počet konekcií
- -rate prenosová rýchlosť

Príklady:

```
ns cbrgen.tcl -type cbr -nn 10 -seed 1.0 -mc 8 -rate 4.0 > cbr-10-test
ns cbrgen.tcl -type tcp -nn 25 -seed 0.0 -mc 8 > tcp-25-test
```

Obdobne si môžeme vygenerovať pohyb jednotlivých zariadení. Najprv si uvedieme naše možnosti a potom príklady ako na to. Generovanie týchto skriptov sa deje pomocou príkazu „setdest“.

```
./setdest [-n num_of_nodes] [-p pausetime] [-s maxspeed] [-t simtime] \
```

```
[-x maxx] [-y maxy] > [outdir/movement-file]
```

Príklad:

```
./setdest -n 20 -p 2.0 -s 10.0 -t 200 -x 500 -y 500 > scen-20-test
```

Príklad ukazuje, ako môžeme vytvoriť náhodný pohyb 20tich zariadení s rýchlosťou 10 m/s. Pauza medzi jednotlivými presunmi je 2 sekundy, pričom chceme simuláciu nechať bežať 200 sekúnd. Topológia je veľkosti 500m × 500m.

3.1.3 Podporné nástroje pre ns-2

Ns-2 sieťový simulátor je silný nástroj pre simulovanie sietí, no neposkytuje v základnom balíku nástroje pre čítanie zaujímavých výsledkov tejto simulácie. Jednoducho iba odsimuluje sieť a poskytne súbor sieťových udalostí, ktoré nastali (trace file). To spôsobilo, že vzniklo množstvo podporných nástrojov, ktoré tieto udalosti analyzujú a poskytujú lepšie čitateľné výstupy a rôzne štatistiky. A práve to je tiež poslaním tohto projektu. No na rozdiel od predchádzajúcich vývojárov, sa pokúsime zahrnúť do nášho komplexného nástroja na simuláciu siete existujúce nástroje tretích strán. Táto kapitola sa teda venuje prehľadu takýchto existujúcich nástrojov.

Po zvážení zadania a orientácie nášho projektu sme sa rozhodli zanalyzovať podporné nástroje v týchto oblastiach:

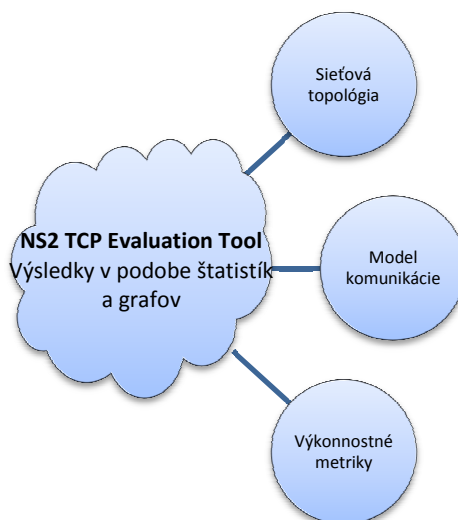
- Nástroje pre štatistické vyhodnotenie simulácie
- Kreslenie grafov výstupných štatistík
- Kreslenie topológií sietí s podporou bezdrôtových sietí
- Nástroje pre štatistické vyhodnotenie

3.1.3.1 NS2 TCP Evaluation Tool

Mnohí vývojári a výskumníci používajú sieťový analyzátor ns-2 na vyhodnotenie výkonu ich protokolov, a to nielen finálnych verzií, ale aj v rôznych štádiách vývoja. Jedna špecifická oblasť sú práve tzv. TCP alternatívy, protokoly pre riadenie zahltienia vo vysokorýchlostných sieťach. A práve pre takéto oblasti vývoja (okrem iných) bol vytvorený hodnotiaci nástroj *NS2 TCP Evaluation Tool*(7). Je vyvíjaný tímom NEC Labs China ako softvér s otvoreným obsahom (Open-source).

Medzi hlavné vlastnosti a výhody nástroja patrí:

- Možnosť merať mnohé základné veličiny používané pri hodnotení TCP
- Automatické generovanie štatistík zo simulácie
- Automatické generovanie grafov (export do formátu html a latex)
- Rozšíriteľné voľne dostupné rozhranie
- Zahnuté viaceré typické topológie a modely



Obrázok 4: Architektúra programu NS2 TCP Evaluation Tool

Samotný nástroj pozostáva z troch hlavných komponentov (Obrázok 4):

1. Sieťová topológia

- Výber z troch preddefinovaných topológií bežne používaných pri výkonových testoch protokolov počítačových sietí. Preddefinované topológie obsahujú rôzne počty koncových používateľov, sieťových zariadení a úzkych hrdiel siete (communication bottleneck).
- Výber vlastnej topológie

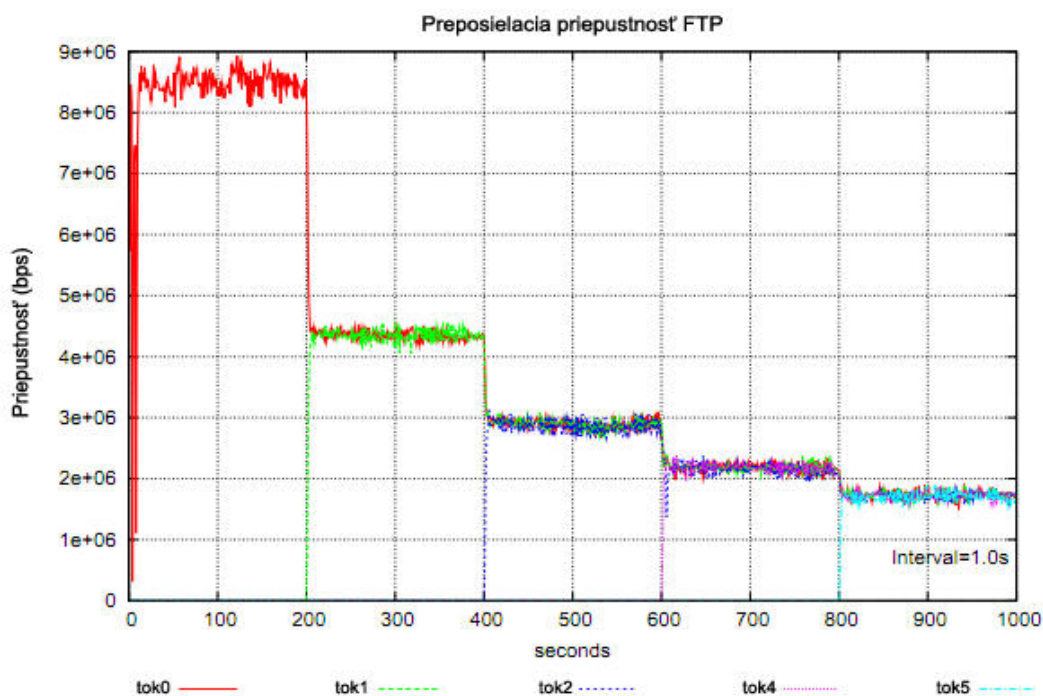
1. Model komunikácie

- Dlhodobá FTP komunikácia s veľkým objemom dát
- Krátko FTP komunikácia za použitia generátora komunikácie *PackMimeHTTP Traffic Generator*
- Krátkodobá webová komunikácia
- Vysielanie video komunikácie (video streaming)
- Interaktívna hlasová komunikácia

2. Výkonnostné metriky

- Priepustnosť (throughput)
- Oneskorenie (delay)
- Kolísanie oneskorenia (jitter)
- Stratovosť (loss rate)
- Doba konvergencie
- Mnohé ďalšie alebo vlastné metriky

Spojením týchto hlavných komponentov nám nástroj dáva dokopy finálnu podobu štatistík a grafov. Tento nástroj nám okrem klasického textového výstupu ponúka aj výstupy vo formáte HTML, latex a EPS. Vygenerované výstupy sú ukladané v dočasnom priečinku pre ďalšie použitie. Formát výstupného grafu znázorňuje Obrázok 5.



Obrázok 5: Ukážka výstupného grafu simulácie programu NS2 TCP Evaluation Tool

3.1.3.2 GnuPlot

GnuPlot(8) je multiplatformový (Linux, OS/2, MS Windows, OSX, VMS) nástroj na kreslenie grafov. Zdrojový kód nástroja je voľne šíriteľný. Pôvodne bol vytvorený na interaktívne aplikácie pre študentov a vedcov na vizualizáciu matematických funkcií a interakcií. Neskôr sa rozšíril do viacerých oblastí, ako napr. web skriptovanie. Takisto sa dá využiť ako nástroj na tvorbu grafov pre aplikácie tretích strán ako napr. *Octave* alebo *NS-2*. Ukážku vybraných funkcií programu GnuPlot ilustruje Obrázok 6.

Medzi výhody nástroja patrí:

- Možnosť tvorby grafov pre aplikácie tretích strán
- Možnosť tvorby širokej palety 2D a 3D grafov
- Kreslenie grafov rôznymi spôsobmi (čiary, body, boxy, texty)
- Rôzne druhy výstupu – terminál, tlačiarne, rôzne typy súborov, ľahko rozšíriteľný o nové typy
- K posledným zmenám patrí podpora interaktívnych grafov prístupných cez HTML5 element `<canvas>`

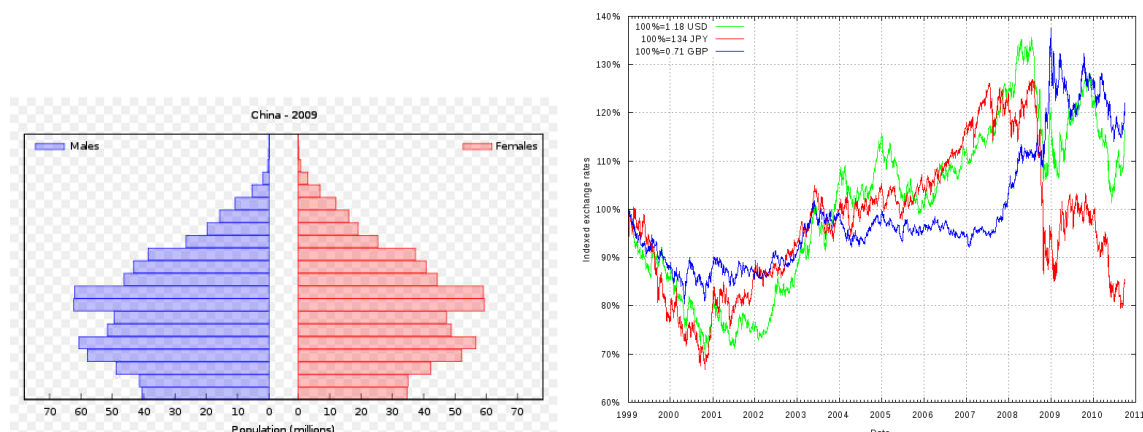
Ukážka skriptu pre generovanie grafu:

```
if len(sys.argv) != 3:
    print "example.py <trace file><output eps file>"
    sys.exit(1)

input = file(sys.argv[1])
flows = flowanalysis.parseFlowsFromTrace(input)
input.close()

data = throughputOverTime.throughputOverTime(flows)

stupidplot.gnuplotTable(data, sys.argv[2])
```



Obrázok 6: Ukážky grafov z programu GnuPlot (9)

3.1.3.3 Xgraph

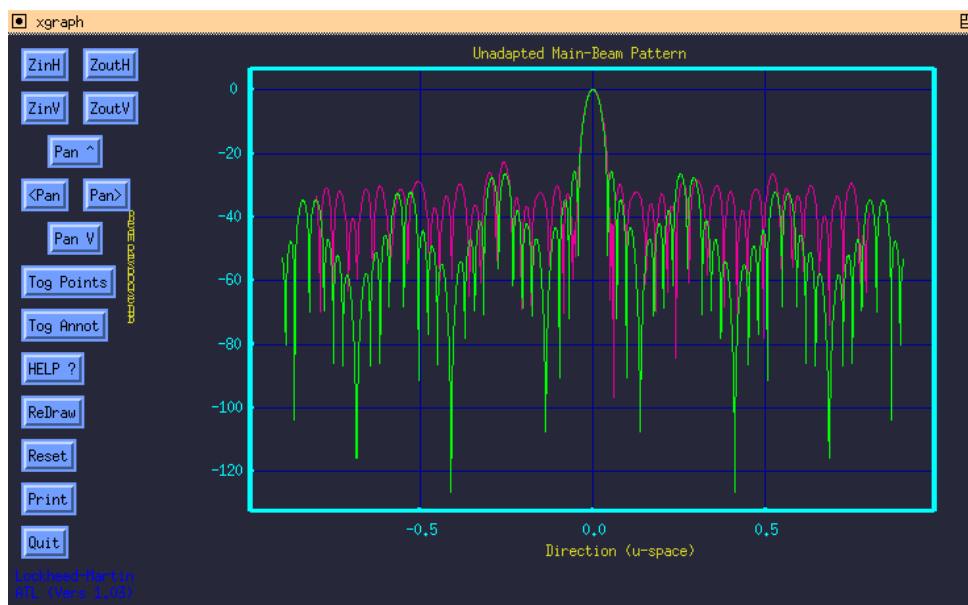
Xgraph je komplexný nástroj na vytváranie dvojrozmerných grafov s podporou mnohých interaktívnych prvkov. Nástroje je vytvorený na všeobecné použitie.

Medzi jeho nesporné výhody patrí:

- tvorba grafov z neobmedzeného počtu a veľkosti dát a súborov
- produkovanie PostScript, PDF, and MIF výstupov na tlač, ukladanie a zdieľanie grafov. Podpora importovania grafov do textových formátovacích nástrojov.
- Množstvo spôsobov kreslenia grafov, podpora rôznych farieb, tvaru a hrúbky čiar, interaktívnych tlačidiel a pod.
- Automatické rozširovanie okna a možnosť približovania sa v grafe
- Je podporovaný na každej Unix platforme s nainštalovaným programom X Windows

Xgraph môže byť použitý dvoma spôsobmi:

- Interaktívne – štandardné nastavenie. Poskytuje grafické rozhranie v okne s viacerými podpornými tlačidlami. Používateľ sa môže v grafe približovať, vzdiaľovať alebo grafom hýbať. Kliknutím na niektoré miesto v grafe sa zobrazia súradnice vybraného bodu. Kliknutím na ďalšie miesto sa zobrazí rozdiel/vzdialenosť týchto bodov (Obrázok 7).
- Bez oknového rozhrania – vhodný pri vzdialenej práci alebo na tvorbu serverových reportov. Vygenerovaný graf sa nezobrazí na obrazovku, ale sa pošle do postscript alebo MIF súboru, kde sa uloží na ďalšie alebo neskoršie spracovanie.



Obrázok 7: Ukážka programu XGRAPH (10)

3.1.3.4 ThroughputOverTime

Malý, no veľmi nápomocný nástroj *ThroughputOverTime*, nám poskytuje informácie o priepustnosti uzlov v jednotlivých časoch simulácie. Takto získané informácie sa dajú ľahko posunúť do niektorého z ďalej opisovaných nástrojov na prehľadné grafické znázornenie(7).

Ukážka výstupu nástroja:

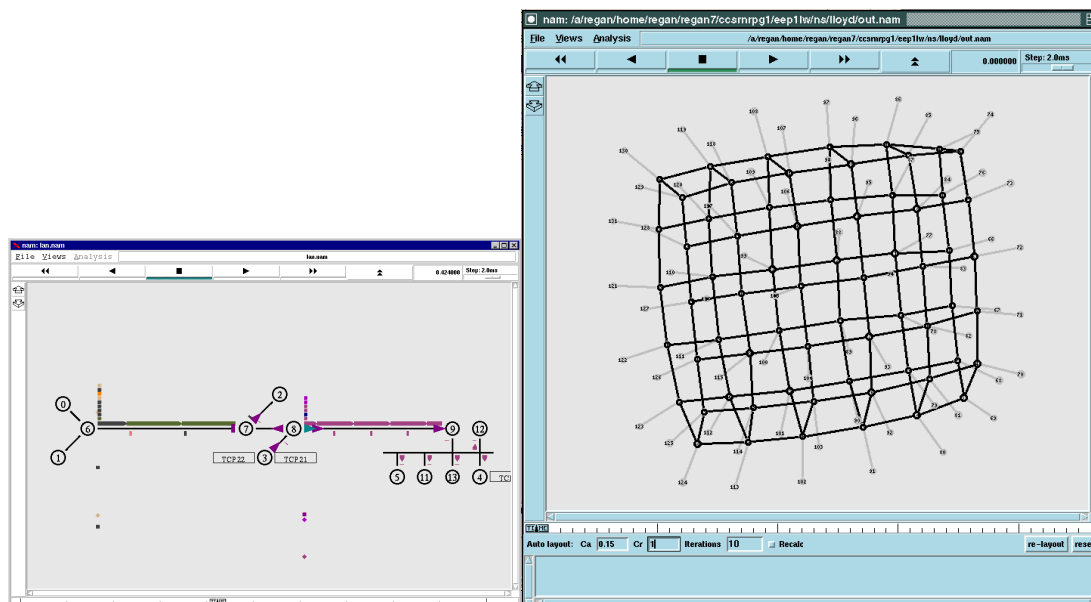
Time	0:0	->	4:0	4:0	->	0:0
1.0			0.0			0.0
2.0			0.0			0.0
3.0			0.0			0.0

3.1.3.5 Ns-nam

Sieťový animátor pre NS (Ns-nam, network animator for ns) bol vytvorený začiatkom deväťdesiatych rokov ako jednoduchý nástroj pre animovanie sietí a toku rámcov zo simulačných súborov (trace files). Tie sa dajú získať ako výstupy sieťového simulátora *ns-2* alebo odchytením reálnej prevádzky programom *TcpDump*. V súčasnosti je stále vo vývoji na vedeckom inštitúte *ISI (Information Sciences Institute)* (11).Obrázok 8 znázorňuje rozhranie programu NS-nam.

Hlavné vlastnosti:

- Vizualizačný nástroj výstupných súborov sieťovej simulácie založený na Tcl/Tk (Tk je grafické používateľské rozhranie na tvorbu aplikácií. Okrem mnohých iných je využitý ako štandardné rozhranie pre Tcl)
- Veľká podpora pre simulácie TCP, smerovania, komunikácie jeden - viacerí (multicasting) a pod. ako pre klasické pevné tak aj bezdrôtové siete



Obrázok 8: Ukážky rozhrania programu Ns-nam (11)

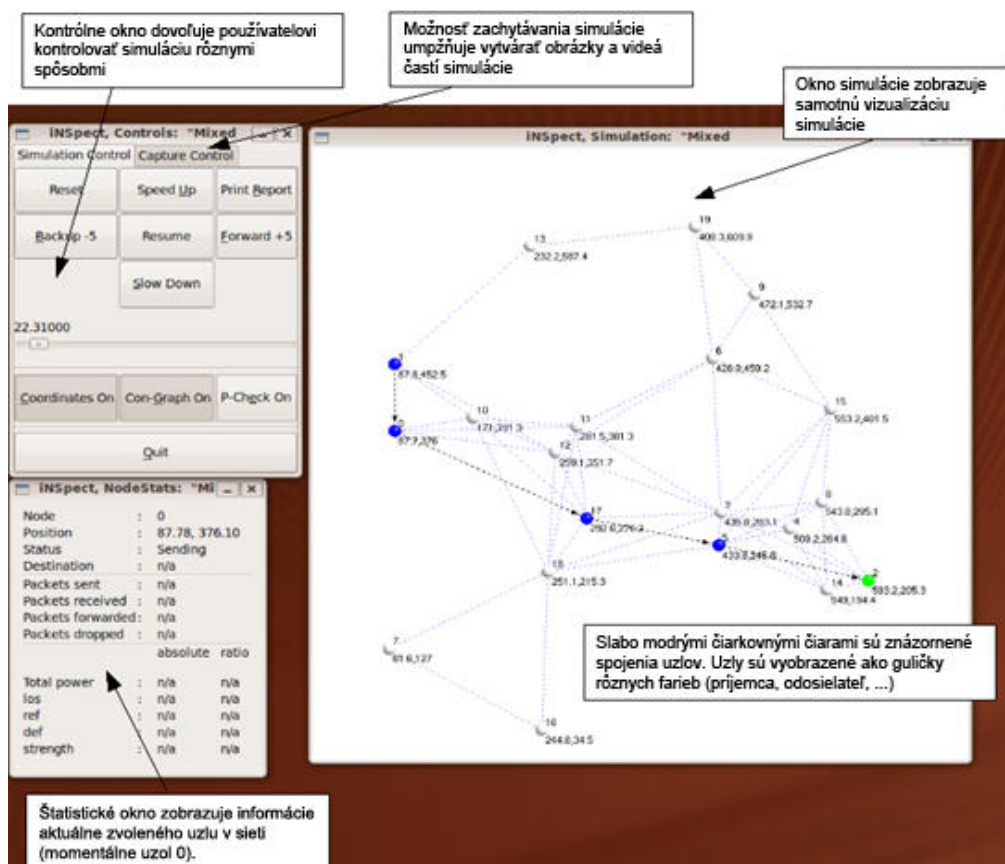
3.1.3.6 iNSpect

iNSpect, v súčasnej verzii 4, je analyzačný a vizualizačný nástroj pre drôtové a bezdrôtové siete. Je podporovaným nástrojom pre viaceré simulačné prostredia. Nástroj pracuje s výstupnými súbormi (trace files) napr. simulátora ns-2 a vytvorí vizuálnu prezentáciu tejto simulácie. V súčasnosti je vyvíjaný ako spoločný projekt škôl *ColoradoSchool of Mines* a *Fraunhofer Institute for Computer Graphics Research*(12).

Medzi jeho hlavné vlastnosti a prednosti patrí:

- Veľká podpora klasických pevných sietí
- Vynikajúca podpora pre bezdrôtové, mobilné, ad-hoc aj senzorové siete (na rozdiel od vyššie opísaného Ns-nam)
- Presné a rýchle vykresľovanie topológie
- Možnosť pohybovania sa vo vizualizácii
- Možnosť zobrazenia podrobnejšej analýzy simulácie
- Vytvorenie obrázka jedného okamihu simulácie alebo nahrávania video sekvencií častí simulácie v známych formátoch
- Podpora pre ns-2 simulátor - *Saeed Khalafinejad* z univerzity *Sharif University of Technology* napísal podporný program na konvertovanie výstupných simulačných súborov simulátora NS (NS trace files) na formát čitateľný pre *iNSpect*.

Popis rozhrania programu iNSpect znázorňuje Obrázok 9.

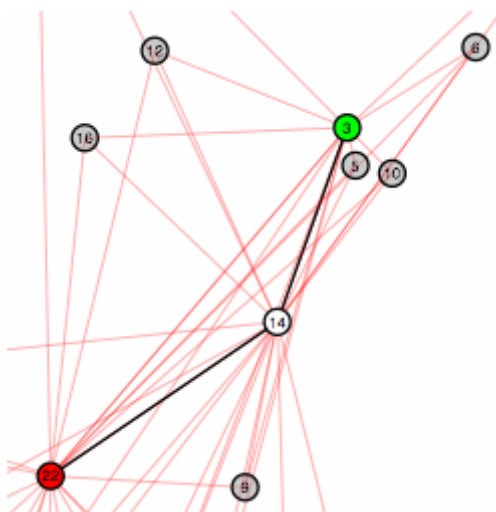


Obrázok 9: Ukážka rozhrania programu iNSpect v OS Ubuntu(12)

3.1.3.7 DrawNetwork

DrawNetwork (13) je jednoduchý program vytvorený v grafickom prostredí *GnuStep* na vykresľovanie topológie pre beh simulácie (

Obrázok 10). Program je zameraný hlavne na kreslenie bezdrôtových sietí.



Obrázok 10: Ukážka grafu z programu DrawNetwork (13)

3.1.4 Zhodnotenie ns-2

Nasledujúca kapitola sumarizuje zistené vlastnosti simulátora ns-2 a jeho možnosti v oblasti simulácie bezdrôtových sietí. Tiež je tu uvedený zoznam protokolov, ktoré sú implementované priamo v jadre simulátora a pri implementácii boli riadne validované.

Výhody ns-2

- Stabilita, odladenosť Podpora širokého spektra sieťových protokolov (Tabuľka 1)
- Podpora 802.15.4 (ZigBee), Bluetooth, 802.11, senzorové siete
- Existuje množstvo rozširujúcich modulov a podporných aplikácií
- Možnosť simulácie pohybu uzlov v 3D priestore
- Špeciálny formát výstupného súboru pre bezdrôtové siete

Nevýhody ns-2

- Niektoré moduly neboli riadne otestované
- Väčšia hardvérová náročnosť simulácií (14)
- Chýba podpora smerovacieho protokolu OLSR
- Chýba podpora pre 802.16 (WiMax), CDMA, GPRS, GSM

Zoznam podporovaných služieb, protokolov a štandardov podporovaných priamo v jadre simulátora ns-2 ilustruje Tabuľka 1.

Tabuľka 1: Zoznam protokolov podporovaných v jadre ns-2

Aplikačná vrstva	Ping, vat, telnet, FTP, multicast FTP, HTTP, vybrané generátory sieťovej premávky.
Transportná vrstva	TCP, UDP, SCTP, XCP, TFRC, RAP, RTP. Vysielanie typu multicast: PGM, SRM, RLM, PLM.
Sieťová vrstva	IP, Mobile IP, IPinIP, Distance vector smerovanie, Link state smerovanie, Smerovanie na báze zdroja. Vysielanie typu multicast: SRM, všeobecný centralizovaný. MANET: AODV, DSR, DSDV, TORA, IMEP
Linková vrstva	ARP, HDLC, GAF, MPLS, LDP, Diffserv. Správa radov: DropTail, RED, RIO, WFQ, SRR, REM, PQ, VQ. MAC: CSMA, 802.11b, 802.15.4, Aloha
Fyzická vrstva	Energetický model, Všesmerová anténa, Satelitný opakovač a iné.

3.2 Network simulator 3

Network simulator 3 (*ns-3*) je simulátor diskretných udalostí v počítačových sieťach, ktorý je orientovaný na výskum a vzdelávanie. Vývoj *ns-3* simulátora začal 1. júla 2006. Podieľajú sa na ňom odborníci z University of Washington, Georgia Institute of Technology a ICSI Center for Internet Research. Okrem spomínaných organizácií sa rozrastá aj komunita používateľov, ktorí prispievajú k tvorbe simulátora (napr. Google summer of code). Dôvodom pre začiatok vývoja nového simulátora bola technická zaostalosť network simulátora 2 (*ns-2*) a nedostatok integrovaných a správne otestovaných modulov pre nové sieťové technológie.

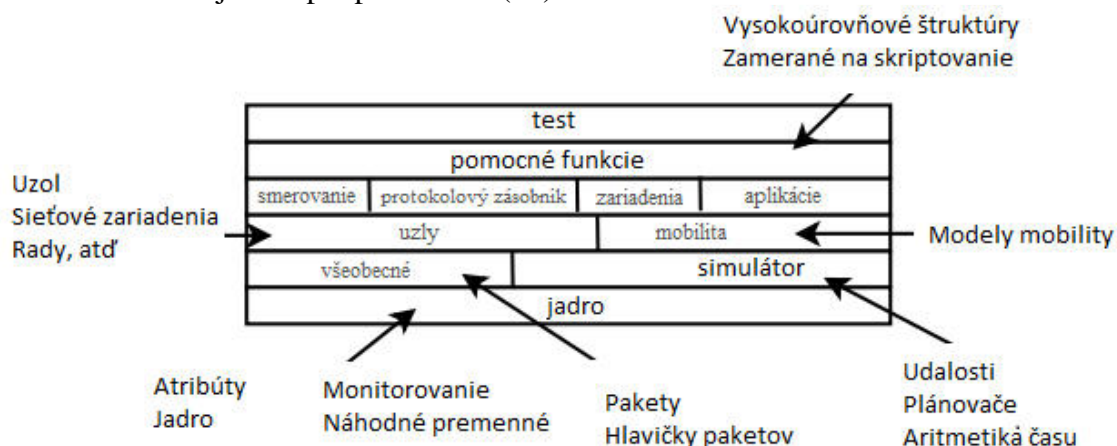
3.2.1 Architektúra ns-3

Simulátor *ns-3* je primárne určený pre linuxové a unixové systémy, možné je aj spustenie pod OS Windows cez program Cygwin, uvažuje sa taktiež o natívnom behu v prostredí Windows (15).

Ns-3 je vyvíjaný ako voľne šíriteľný program pod licenciou GNU GPLv2 alebo inou GPL kompatibilnou licenciou, hlavne pre účely výskumu. Je napísaný v jazyku C++ a simulačné scenáre je možné písať v jazyku C++ alebo pre jednoduchšie používanie v skriptovacom jazyku Python. Toto rozhranie jazyka Python však momentálne nepodporuje všetky funkcionality dostupné z rozhrania v jazyku C++. *Ns-3* nie je priamym nástupcom simulátora *ns-2*, preto ani simulácie a moduly z *ns-2* na *ns-3* nie sú spustiteľné. Na druhej strane treba spomenúť, že autori simulátora si dali za úlohu použiť čo najviac komponentov z *ns-2*, poprípade ich modifikovať, aby spĺňali požiadavky *ns-3*. (16)

Vývojári *ns-3* simulátora pri jeho návrhu identifikovali päť základných oblastí, na ktoré sa vzhľadom na vlastnosti *ns-2* treba sústrediť (Obrázok 11):

- **Jadro** – Zvýšiť modularitu a rozšíriteľnosť. Navrhnuť triedy tak, aby viac pripomínali objekty z reálneho sveta
- **Integrácia** – V čo najvyššej miere použiť voľne dostupný softvér, a orientovať sa na virtualizáciu
- **Bezdrôtové siete** – Viac modelov pre bezdrôtové siete
- **Vzdelávanie** – Animácie, simulácie určené na vzdelávanie
- **Údržba** – Validácia modelov, kvalitná dokumentácia, distribúcia originálneho kódu aj kódu prispievateľov (17)



Obrázok 11: Schéma simulátora ns-3 (18)

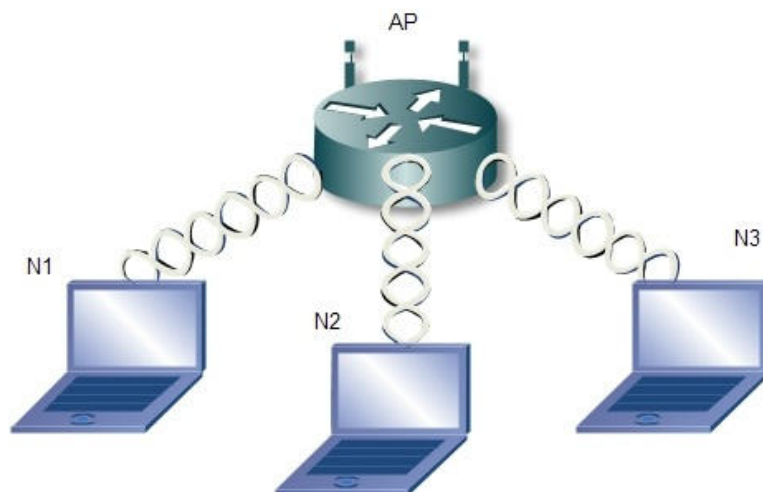
3.2.2 Simulácia pomocou ns-3

Simulátor ns-3 podporuje simulovanie klasických pevných a bezdrôtových sietí, pričom obsahuje knižnice poskytujúce dátové typy a funkcie potrebné na tieto simulácie. Vzhľadom na to, že vstupné súbory simulátora sú písané v jazyku C++ (prípadne Python), jednotlivé prvky a funkcie používané vo vstupných súboroch simulátora sú značne abstrahované, čím sa písanie tohto súboru výrazne sprehľadnilo a používateľ sa môže plne sústrediť na opis topológie, protokolov a toku dát a nemusí sa zaoberať rôznymi implementačnými problémami.

V ns-3 sa simulácia klasických sietí realizuje podobne ako bezdrôtových. Funkcie poskytujúce modelovanie napríklad uzlov, alebo protokolov v sieti majú rovnakú kostru, rovnako sa zapisuje aj topológia (samozrejme so správnymi funkciami a prvkami, v klasickej sieti sa nebudú vytvárať prístupové body a podobne). Táto skutočnosť taktiež veľmi uľahčuje prácu so simulátorom (19).

3.2.2.1 Vstupný súbor simulátora ns-3

Vstupný súbor simulátora ns-3 obsahuje viacero častí, v ktorých používateľ opíše topológiu navrhovanej siete, protokoly v nej použité a tok dát, ktorý chce simulovať. V tejto časti dokumentu si jednotlivé časti opíšeme na príklade topológií, ktorú znázorňuje Obrázok 12.



Obrázok 12: Príklad bezdrôtovej topológie

V ukážkovej topológii vidíme bezdrôtovú sieť založenú na technológii IEEE 802.11. Sú v nej štyri zariadenia: tri počítače s bezdrôtovou kartou a jeden statický prístupový bod (*angl. access point*). Dĺžka simulácie bude 10 sekúnd. IP adresa siete bude 192.168.10.0/24. Počítač N3 (klient) bude posielať od 6. do 10. sekundy protokolom UDP dáta počítaču N1 (server). Na začiatku každého vstupného súboru si musíme prilinkovať (*angl. include*) knižnice potrebné pre našu simuláciu a zdefinovať menný priestor (*angl. namespace*) ns-3.

```
#include "ns3/core-module.h"
#include "ns3/simulator-module.h"
#include "ns3/node-module.h"
#include "ns3/helper-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
using namespace ns3;
```

Vo funkcii main() si ako prvé zadefinujeme počet počítačov do premennej pocetPocitacov, ktoré budú v sieti.

```
uint32_t pocetPocitacov = 3;
```

Nastavíme vypisovanie sledovacích výpisov (angl. log) počas simulácie pre server a klienta. Tieto sa vypíšu do príkazového riadku.

```
LogComponentEnable("UdpEchoClientApplication",  
LOG_LEVEL_INFO);  
LogComponentEnable("UdpEchoServerApplication",  
LOG_LEVEL_INFO);
```

Vytvoríme si tri uzly v sieti, ktoré budú predstavovať tri počítače.

```
NodeContainer wifiPocitace;  
wifiPocitace.Create(3);
```

Vytvoríme si ďalší uzol v sieti pre prístupový bod.

```
NodeContainer wifiAP;  
wifiAP.Create(1);
```

Vytvoríme si bezdrôtový kanál.

```
YansWifiChannelHelper kanal =  
YansWifiChannelHelper::Default();  
YansWifiPhyHelper phy = YansWifiPhyHelper::Default();  
phy.SetChannel(kanal.Create());
```

Vytvoríme si pomocný objekt pre wifi spojenie (*angl. wifi helper*).

```
WifiHelper wifi = WifiHelper::Default();  
wifi.SetRemoteStationManager("ns3::AarfWifiManager");
```

Vytvoríme si pomocný objekt pre MAC adresáciu (*angl. MAC helper*).

```
NqosWifiMacHelper mac = NqosWifiMacHelper::Default();
```

Pomenujeme sieť.

```
Ssid ssid = Ssid("ns-3-ssid");
```

Nastavíme správne hodnoty pre pomocný objekt MAC adresácie pre počítače v sieti. Vytvoríme bezdrôtové zariadenia v sieti z vytvorených uzlov (počítačov). Priradíme im kanál a pomocný objekt MAC adresácie.

```
mac.SetType("ns3::NqstaWifiMac", "Ssid", SsidValue(ssid), "Ac  
tiveProbing", BooleanValue(false));  
NetDeviceContainer staDevices;  
staDevices = wifi.Install(phy, mac, wifiPocitace);
```


Pomocnému objektu MAC adresácie nastavíme správne hodnoty pre prístupový bod. Vytvoríme bezdrôtové zariadenie pre uzol (prístupový bod). Priradíme mu pomocný objekt pre MAC adresáciu a kanál.

```
mac.SetType("ns3::NqapWifiMac", "Ssid", SsidValue
(ssid), "BeaconGeneration", BooleanValue(true), "BeaconInterval
", TimeValue (Seconds(2.5)));
NetDeviceContainer apDevices;
apDevices = wifi.Install (phy, mac, wifiAP);
```

Vytvoríme pomocný objekt pre mobilitu (*angl. mobility helper*), ktorému nastavíme mobilitu pre počítače v sieti, ktoré môžu meniť svoju polohu.

```
MobilityHelper mobility;
mobility.SetPositionAllocator("ns3::GridPositionAllocator"
, "MinX", DoubleValue(0.0), "MinY", DoubleValue(0.0),
"DeltaX", DoubleValue(5.0), "DeltaY", DoubleValue(10.0),
"GridWidth", UIntegerValue(3), "LayoutType",
StringValue("RowFirst"));
mobility.SetMobilityModel
("ns3::RandomWalk2dMobilityModel", "Bounds", RectangleValue
(Rectangle (-50, 50, -50, 50)));mobility.Install
(wifiPocitace);
```

Nastavíme mobilitu pre prístupový bod, ktorý je statický a nehýbe sa.

```
mobility.SetMobilityModel
("ns3::ConstantPositionMobilityModel");
mobility.Install (wifiAP);
```

Vytvoríme si balík internetových protokolov a nainštalujeme ho na prístupový bod a počítače.

```
InternetStackHelper stack;
stack.Install (wifiAP);
stack.Install (wifiPocitace);
```

Vytvoríme si pomocný objekt pre Ipv4 adresáciu (*angl. Ipv4AddressHelper*), nastavíme ho, a priradíme ho najprv bezdrôtovému zariadeniu prístupového bodu, následne aj bezdrôtovým zariadeniam počítačov. IP adresy sa pridelia v tomto poradí, teda prístupový bod bude mať prvú adresu z rozsahu a počítače postupne ďalšie.

```
Ipv4AddressHelper address;
Ipv4InterfaceContainer wifiInterfaces;
address.SetBase ("192.168.10.0", "255.255.255.0");
address.Assign (apDevices);
wifiInterfaces = address.Assign (staDevices);
```

Na prvom počítači si vytvoríme prijímač (UDP echo server), ktorý bude posilať dáta od 6. do 10. sekundy simulácie.

```
UdpEchoServerHelper echoServer (9);
ApplicationContainer serverApps =
echoServer.Install(wifiPocitace.Get(0));
serverApps.Start (Seconds (6.0));
```



```
serverApps.Stop (Seconds (10.0));
```

Na poslednom počítači (N3, Obrázok 12) vytvoríme vysielateľ (UDP echo klient), ktorý bude posielať od 6. do 10. sekundy simulácie dáta na počítač číslo 1 (N1), ktorý sme si vyššie zadefinovali ako prijímač.

```
UdpEchoClientHelper  
echoClient(wifiInterfaces.GetAddress(0), 9);  
ApplicationContainer clientApps =  
echoClient.Install(wifiPocitace.Get(pocetPocitacov - 1));  
clientApps.Start (Seconds (6.0));  
clientApps.Stop (Seconds (10.0));
```

Naplníme smerovacie tabuľky.

```
Ipv4GlobalRoutingHelper::PopulateRoutingTables();
```

Po desiatich sekundách simuláciu zastavíme.

```
Simulator::Stop (Seconds (10.0));
```

Nastavíme zariadenie, ktorého komunikácia sa bude sledovať a tiež názov výstupného súboru. Náš súbor sa bude volať *“example“* a sledovať budeme prístupový bod.

```
phy.EnablePcap ("example", apDevices.Get (0));
```

Spustíme simuláciu a po skončení dealokujeme pamäť použitú pre simulátor.

```
Simulator::Run ();  
Simulator::Destroy ();
```

3.2.2.2 Výstupný súbor simulátora ns-3

Sledovanie simulácie sa v simulátore ns-3 môže realizovať niekoľkými spôsobmi. Buď sa použijú už implementované prostriedky na sledovanie simulácie, ako je napríklad ASCII sledovanie, alebo výstup simulácie vo forme súboru typu .pcap, ktoré si popíšeme neskôr. Ďalšou možnosťou sledovania simulácie sú výpisy implementované priamo vo vstupnom súbore simulácie, v ktorých si môžeme vypisovať ľubovoľne zadefinované hlásenia, napríklad pri vytvorení nového uzla, alebo podobne. Poslednou možnosťou je upraviť priamo jadro simulátora, kde si môžeme nechať vypisovať hlásenia priamo vo vstavaných funkciách simulátora. (20)

Našou úlohou je však poskytnúť rozhranie pre používateľa, ktoré mu uľahčí vytváranie vstupných súborov a zároveň interpretáciu výstupných súborov, preto nepokladáme za dôležité venovať sa v tomto dokumente úpravám jadra simulátora ns-3, keďže ich neplánujeme vykonávať.

V nasledujúcom texte sa oboznámime s už implementovanými spôsobmi sledovania behu simulácie, ako je sledovanie pomocou ASCII výstupu či PCAP výstupu. ASCII aj PCAP sledovanie sa definuje priamo vo vstupnom súbore simulátora, pričom sa musí určiť, na ktorom uzle v simulovanej sieti sa toto sledovanie bude vykonávať. Sledovanie každej entity v simulácii sa ukladá do samostatného súboru, ktorý sa neskôr spracováva.

To, že vo vstupnom súbore simulátora povieme, ktoré komunikácie na ktorých uzloch chceme zaznamenávať do akého súboru, nám značne uľahčuje orientáciu vo výstupných

súboroch. Vždy vieme, v ktorom nájdeme požadované dáta a preto nemusíme práčne prehľadávať záznamy z rôznych uzlov, aby sme našli komunikáciu, ktorú potrebujeme.

ASCII sledovanie simulácie

Formát výstupného súboru z ASCII sledovania je nasledovný. Na každom riadku sa nachádzajú tieto polia:

1. Typ udalosti na uzle, udalosti sa týkajú paketu: pridanie do radu (+), odobranie z radu (-), prijatie (r), zahodenie (d).
2. Časová pečiatka udalosti.
3. Ďalší reťazec znakov je ns-3 špecifická cesta k udalosti, ktorá dáva informáciu o tom, na ktorom sieťovom rozhraní ktorého uzlu sa udalosť stala, zároveň aj poskytuje informáciu o type udalosti.
4. Ďalší reťazec je zoznam typov hlavičiek, zakódovaných ako názvy tried ns-3 simulátora, ktoré sa v pakete/rámci nachádzajú. Každá hlavička je nasledovaná informáciou indikujúcou informácie špecifické pre daný protokol.

Príklad ASCII .tc výstupného súboru simulátora:

```
t 0.00162933 /NodeList/1/DeviceList/0/$ns3::WifiNetDevice/Phy/State/Tx
ns3::WifiMacHeader (CTL_ACK Duration/ID=0us, RA=00:00:00:00:00:04)
ns3::WifiMacTrailer ()

r 0.00167335 /NodeList/0/DeviceList/0/$ns3::WifiNetDevice/Phy/State/RxOk
ns3::WifiMacHeader (CTL_ACK Duration/ID=0us, RA=00:00:00:00:00:04)
ns3::WifiMacTrailer ()

r 0.00167335 /NodeList/2/DeviceList/0/$ns3::WifiNetDevice/Phy/State/RxOk
ns3::WifiMacHeader (CTL_ACK Duration/ID=0us, RA=00:00:00:00:00:04)
ns3::WifiMacTrailer ()

r 0.00167337 /NodeList/3/DeviceList/0/$ns3::WifiNetDevice/Phy/State/RxOk
ns3::WifiMacHeader (CTL_ACK Duration/ID=0us, RA=00:00:00:00:00:04)
ns3::WifiMacTrailer ()

t 2.5 /NodeList/3/DeviceList/0/$ns3::WifiNetDevice/Phy/State/Tx
ns3::WifiMacHeader (MGT_BEACON ToDS=0, FromDS=0, MoreFrag=0, Retry=0,
MoreData=0 Duration/ID=0us, DA=ff:ff:ff:ff:ff:ff, SA=00:00:00:00:00:04,
BSSID=00:00:00:00:00:04, FragNumber=0, SeqNumber=1)
ns3::MgtProbeResponseHeader (ssid=ns-3-ssid, rates=[*6mbs 9mbs 12mbs 18mbs
24mbs 36mbs 48mbs 54mbs]) ns3::WifiMacTrailer ()
```

PCAP sledovanie simulácie

PCAP sledovanie behu simulácie poskytuje skutočne prehľadné informácie o dianí v simulovanej sieti, neposkytuje však toľko informácií ako ASCII sledovanie. Výstupný súbor má príponu .pcap a dá sa otvoriť v programe Wireshark aj tcpdump.

Príklad výstupného súboru s príponou .pcap, ktorý sme otvorili programom tcpdump:

```
0.000025 Beacon () [6.0* 9.0 12.0 18.0 24.0 36.0 48.0 54.0
Mbit] IBSS
```

```

0.000263 Assoc Request () [6.0 9.0 12.0 18.0 24.0 36.0 48.0
54.0 Mbit]
0.000279 Acknowledgment RA:00:00:00:00:00:01
0.000357 Assoc Response AID(0) :: Succesful
0.000501 Acknowledgment RA:00:00:00:00:00:04
0.000748 Assoc Request () [6.0 9.0 12.0 18.0 24.0 36.0 48.0
54.0 Mbit]
0.000764 Acknowledgment RA:00:00:00:00:00:02
0.000842 Assoc Response AID(0) :: Succesful
0.000986 Acknowledgment RA:00:00:00:00:00:04
0.001242 Assoc Request () [6.0 9.0 12.0 18.0 24.0 36.0 48.0
54.0 Mbit]
0.001258 Acknowledgment RA:00:00:00:00:00:03
0.001336 Assoc Response AID(0) :: Succesful
0.001480 Acknowledgment RA:00:00:00:00:00:04
2.500000 Beacon () [6.0* 9.0 12.0 18.0 24.0 36.0 48.0 54.0
Mbit] IBSS
5.000000 Beacon () [6.0* 9.0 12.0 18.0 24.0 36.0 48.0 54.0
Mbit] IBSS
6.000112 ARP, Request who-has 192.168.10.2
(ff:ff:ff:ff:ff:ff) tell 192.168.10.4, length 32
6.000128 Acknowledgment RA:00:00:00:00:00:03
6.000206 ARP, Request who-has 192.168.10.2
(ff:ff:ff:ff:ff:ff) tell 192.168.10.4, length 32
6.000464 ARP, Reply 192.168.10.2 is-at 00:00:00:00:00:01,
length 32

```

3.2.2.3 Podporované bezdrôtové technológie

Vývoj simulátora ns-3 ide relatívne rýchlym tempom, nové verzie vychádzajú približne každé tri až štyri mesiace. Nasleduje zoznam podporovaných technológií podľa jednotlivých verziách simulátora.

ns-3.5 (Júl 2009)

- 802.11e MAC EDCA
- 802.11n A-MSDU frame aggregation
- 802.11b PHY
- Nakagami loss
- Gamma, Erlang, Zipf random variables

ns-3.6 (Október 2009)

- Minstrel rate control
- WiFi Athstats and 5/10MHz channels
- IPv6 radvd, ICMP
- 802.11s mesh
- Nix-vector routing
- Flow Monitor

ns-3.7 (Január 2010)

- 802.11p PHY
- AODV
- Waypoint mobility

- NetAnim
- IPv6 Extension and Option headers

ns-3.8 (Máj 2010)

- MPI-based sims
- WiMAX
- 802.11n Block Ack
- Gauss-Markov and steady state random waypoint mobility models
- Matrix prop. loss mode
- Two-way ray prop model

ns-3.9 (August 2010)

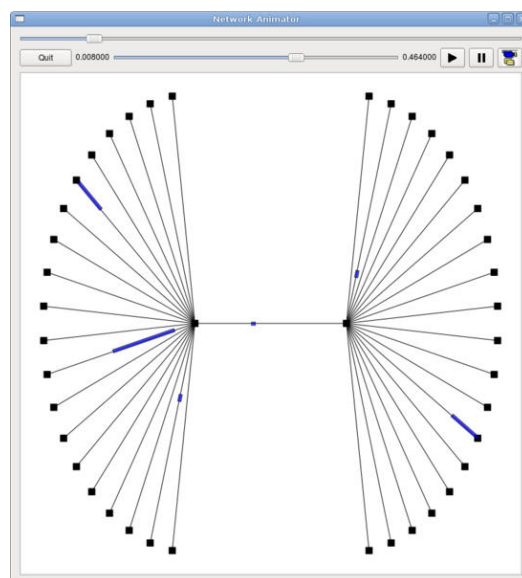
- OFDM
- UAN (Underwater Acoustic Network)
- Energy model(16)

3.2.3 Podporné nástroje pre ns-3

Ns-3 je v mnohom podobný a v mnohom odlišný od simulátora ns-2. Charakter prístupu k simuláciám je rovnaký – simulátor poskytuje výstup vo formáte pcap a v textovej forme, čo je silný podnet na vznik viacerých vizualizačných nástrojov. Väčšina z nich si vyžaduje špeciálny formát výstupu, čo nie je zložité dosiahnuť, keďže výstup simulátora je prispôsobiteľný.

3.2.3.1 NetAnim

Tento nástroj je založený na vývojovom prostredí QT4 a animuje toky dát medzi jednotlivými uzlami. Tento program vyžaduje výstup v špeciálnom formáte, generovanom animačným rozhraním (angl. animation interface) simulátora. Toto rozhranie je od verzie 3.6 štandardnou súčasťou simulátora(21).



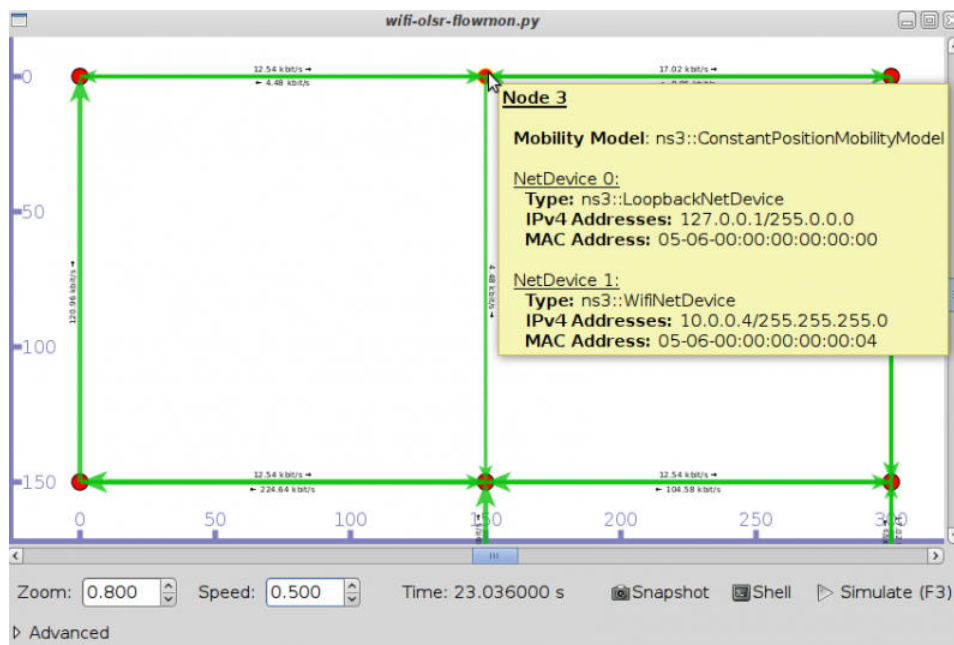
Obrázok 13: Grafické rozhranie programu NetAnim

Zaujímavou vlastnosťou tohto vizualizéra je špecifikácia umiestnenia uzlov na schéme (Obrázok 13). Potreba špecifikovať umiestnenie jednotlivých uzlov naznačuje, že program neobsahuje modul na samostatné generovanie schémy (t.j. nemá implementované algoritmy optimalizujúce zobrazenie grafu ako planárneho a pod., čo je z pohľadu záťaže na hardvér pozitívne), a rovnako to poskytuje možnosti zachovania schémy v prípade, že by ju do systému zadával používateľ cez vizuálny editor. Kód na umiestnenie objektu do schémy vyzerá takto (22):

```
Ptr<Node> hub = nodes.Get (0);
Ptr<CanvasLocation> hubLoc = hub->GetObject<CanvasLocation>
();
if (hubLoc == 0)
{
    hubLoc = CreateObject<CanvasLocation> ();
    hub->AggregateObject (hubLoc);
}
Vector hubVec (5, 7, 0);
hubLoc->SetLocation (hubVec);
```

3.2.3.2 PyViz

PyViz je vizualizér simulácií v reálnom čase, t.j. nepoužíva žiadne výstupné súbory a do simulácie je možné vstupovať prostredníctvom konzoly, cez ktorú je možné meniť stav simulovaných objektov za behu. PyViz podporuje simulácie napísané v jazyku Python aj v C++ (23). Tento vizualizér nie je súčasťou hlavnej vetvy simulátora, ale existujú balíky, kde sú PyViz a ns-3 zlúčené. Rozhranie vizualizéra znázorňuje Obrázok 14.



Obrázok 14: Grafické rozhranie vizualizéra PyViz

Na spustenie vizualizácie simulácie stačí obyčajná simulácia (napísaná v Pythone alebo C++), s minimálnymi úpravami – namiesto štartu simulácie sa zavolá vizualizér. Odporúčaný spôsob v Pythone je nahradiť riadok:

```
ns3.Simulator.Run()
```

nasledujúcim kódom

```
try:
    import visualizer
except ImportError:
    print "no visualizer"
    ns3.Simulator.Run()
else:
    visualizer.start()
```

a v C++ simulácii vložiť riadok

```
#include "ns3/visualizer.h"
```

a nahradiť

```
Simulator::Run ()
```

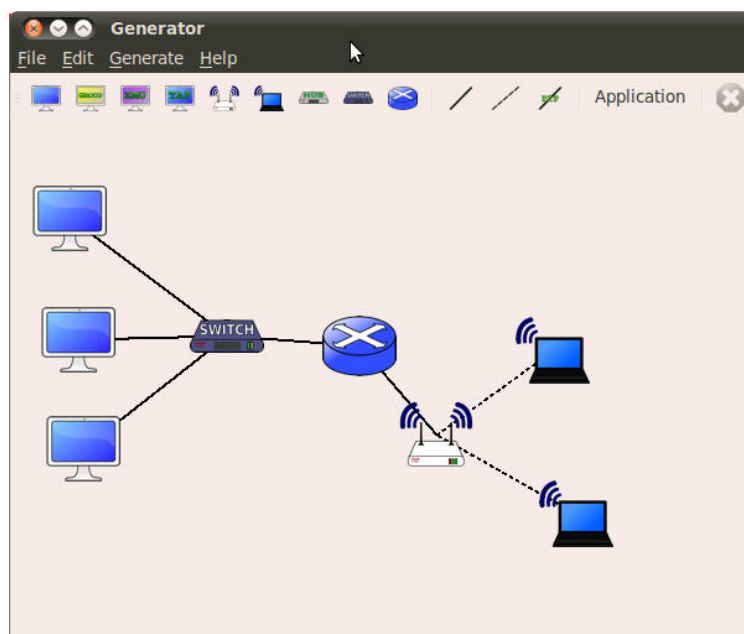
kódom

```
Simulator::Run ();
```

Keďže sa jedná o vizualizér v reálnom čase, nie je príliš vhodný na webové riešenie.

3.2.3.3 Ns3Generator

Jedná sa o vstupný generátor kódu na základe používateľom nakreslenej topológie. Tento program umožňuje generovanie kódu v jazyku Python aj v C++. Podporované sú pevné aj bezdrôtové prvky. Ns3Generator využíva platformu Qt4. Zaujímavou vlastnosťou tohto generátora je možnosť uloženia topológie (a jej opätovné nahranie) vo formáte XML. Táto vlastnosť môže byť využiteľná pri webovom riešení. Rozhranie programu NS3Generator ilustruje Obrázok 15.



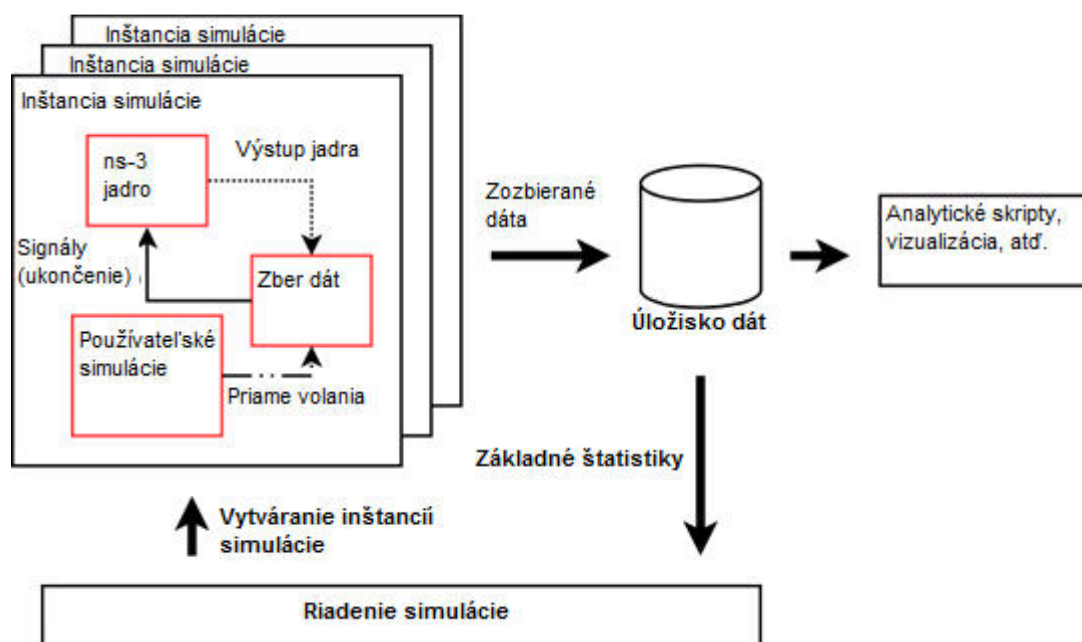
Obrázok 15: Grafické rozhranie programu

3.2.3.4 Statistical Framework for Network Simulation

Nástroj tvorí štatistické rozhranie simulátora ns-3. Cieľom tohto nástroja je poskytnúť funkcie na záznam a základné spracovanie štatistík simulácie a umožniť riadenie simulácie za behu (napr. opakovanie simulácií, ukončenie za určitých podmienok). Keďže nie je potrebné generovať dlhé výstupné súbory a pracovať s nimi, prejavuje sa použitie tohto nástroja zvýšením výkonnosti systému pri simuláciách.

Toto prostredie má byť integrované s existujúcim systémom výstupných súborov. Cieľom je umožniť používateľom používať štatistické prostredie aj bez znalosti základného systému výstupných súborov.

Koncepciu tohto nástroja je znázorňuje Obrázok 16. Experiment môže prebiehať vo viacerých inštanciách (či už sériovo alebo paralelne), ktoré sú spúšťané riadiacim skriptom s prípadnými zmenami parametrov. Dáta sú ukladané na použitie ďalšími nástrojmi na kreslenie grafov alebo štatistickú analýzu.



Obrázok 16: Schéma štatistického systému (24)

Statistical Framework for Network Simulation,
http://www.nsnam.org/wiki/index.php/Statistical_Framework_for_Network_Simulation, máj 2008

Statistical Framework je len v štádiu vývoja a ešte neposkytuje všetky potrebné funkcie, jedná sa však o potenciálne silný nástroj. V súčasnosti sú k dispozícii tieto funkcie:

- Jadro a dva dátové zberače: počítadlo a štatistiky min/max/priemer/suma
- Rozšírenie na prácu s časmi a paketmi
- Výstup formátovaného textu pre program omnetpp
- Databázový výstup na sqlite3
- Metadáta na opis a prácu s jednotlivými pokusmi v rámci experimentov

Vyvíjané sú prioritne tieto ďalšie moduly:

- On-line štatistický kód
- Funkcie v dátových zberačoch na ukončenie simulácií (pri rôznych podmienkach)

- Dátové zberače na zber vzoriek a výstupy v rôznych formátoch

3.2.4 Zhodnotenie ns-3

V tejto časti sú zhrnuté výsledky predchádzajúcich podkapitol. Celkovo bol simulátor ns-3 zhodnotený ako veľmi zaujímavý nástroj do budúcnosti. Momentálne však nepodporuje mnoho funkcií, alebo tieto funkcie nie sú dostatočne implementované. Ďalšou slabinou je aj nedostatočná validnosť jeho výstupov, a veľký počet chýb v momentálne prítomných implementáciách sieťových prístupov. Pre vyššie uvedené dôvody nebude zrejme vhodný pre účely projektu(25)(20).

Výhody

- Vývoj s dôrazom na nové bezdrôtové technológie (802.11, 802.16, GSM, GPRS,...)
- Škálovateľnosť
- Podpora spolupráce simulátora s reálnou sieťou
- Podpora bežných formátov výstupov (.pcap)
- Jednoduchšie používateľské rozhranie
- Natívna podpora 64 bitových a multiprocessorových strojov
- Beh reálneho kódu v rámci ns-3 (reálne aplikácie, protokolové zásobníky)
- Natívna podpora štatistického vyhodnotenia simulácií

Nevýhody

- Stále vo vývoji
- Obmedzené množstvo modulov v porovnaní s ns-2
- Vizualizačný nástroj nie je integrovaný a ani dokončený
- Nedostatočná validnosť sieťových modelov
- Nástroj pre štatistické vyhodnotenie nie je dokončený

Je treba podotknúť, že koncom roka 2010 má byť dostupná nová verzia ns-3 simulátora, ktorá má pridať mimo iného aj podporu LTE, energetický model pre Wi-Fi zariadenia a model pohybu vo VANET sieťach (siete MANET, v ktorých sú uzly automobily). Do budúcnosti sú plánované aj mnohé iné bezdrôtové technológie. Napríklad GSM, GPRS, rôzne verzie 802.11, prístupové metódy CDMA a TDMA (26) (27).

Zoznam podporovaných služieb, protokolov a štandardov podporovaných priamo v jadre simulátora ns-3 ilustruje Tabuľka 2.

Tabuľka 2: Zoznam protokolov podporovaných v jadre ns-3

Aplikačná vrstva	Synchrónne a asynchrónne sokety, paketové sokety, zapnutie/vypnutie aplikácie.
Transportná vrstva	TCP, UDP.
Sieťová vrstva	IPv4, IPv6, ICMP globálne statické smerovanie, multicast. MANET: AODV, OLSR.
Linková vrstva	PointToPoint MAC: CSMA, 802.11e, 802.11n s kontrolou prenosových rýchlostí, 802.16 WiMAX.
Fyzická vrstva	802.11a, 802.11b, 802.11p, 802.11s (mesh), Modely straty signálu, Modely pohybu staníc, Energetický model, Wifi Athstats 5/10Mhz prenosové kanály.

3.3 Zhodnotenie analýzy simulátorov

Analyzovali sme simulátory ns-2 a ns-3. Oba simulátory majú svoje výhody aj nevýhody. Simulátor ns-3, ako novší simulátor, poskytuje väčšie možnosti na vývoj nových nástrojov v porovnaní so simulátorom ns-2. Na druhej strane ns-2 je stabilný simulátor a s jeho vývojom sa pokračuje a tak aj pre tento simulátor má zmysel vytvoriť nové alebo upraviť existujúce podporné programy. Obidva simulátory vyžadujú špeciálne vstupné súbory (natívne neobsahujú grafické rozhranie na editovanie parametrov simulácie), teda skripty v jazyku OTcl (ns-2) alebo Python (ns-3), resp. kód v C++. Oba simulátory poskytujú štruktúrované výstupy (vo formáte .pcap alebo textovej forme), pričom v oboch je možné nastaviť, ktoré údaje sa budú do týchto súborov zaznamenávať. Simulátory sú teda v týchto kritériách podobné, zásadný rozdiel je však v podpore simulovaných objektov a doplnkových programov

Porovnanie podpory protokolov v oboch simulátoroch ilustruje Obrázok 17.

Vrstva	Protokoly podporované v ns-2	Protokoly podporované v ns-3
Aplikačná vrstva	Ping, vat, telnet, FTP, multicast FTP, HTTP, vybrané generátory sieťovej premávky.	Synchrónne a asynchrónne sokety, paketové sokety, zapnutie/vypnutie aplikácie.
Transportná vrstva	TCP (viacero variantov), UDP, SCTP, XCP, TFRC, RAP, RTP. Multicast: PGM, SRM, RLM, PLM.	TCP, UDP.
Sieťová vrstva	IP, Mobile IP, IPinIP, Distance vector smerovanie, Link state smerovanie, Smerovanie na báze zdroja. Multicast: SRM, všeobecný centralizovaný MANET: AODV, DSR, DSDV, TORA, IMEP.	IPv4, IPv6, ICMP globálne statické smerovanie. Multicast: statické smerovanie. MANET: AODV, OLSR.
Linková vrstva	ARP, HDLC, GAF, MPLS, LDP, Diffserv. Správa radov: DropTail, RED, RIO, WFQ, SRR, REM, PQ, VQ. MAC: CSMA, 802.11b, 802.15.4, Aloha	PointToPoint MAC: CSMA, 802.11e, 802.11n s kontrolou prenosových rýchlostí, 802.16 WiMAX.
Fyzická vrstva	Energetický model, Všesmerová anténa, Satelitný opakovač a iné.	802.11a, 802.11b, 802.11p, 802.11s (mesh), Modely straty signálu, Modely pohybu staníc, Energetický model, Wifi Athstats 5/10Mhz prenosové kanály.
Podporné prostriedky	Generátory náhodných čísel, monitorovanie, podpora pre matematické spracovanie, animácie (nam), modely chýb, testovacie scenáre.	Generátory náhodných čísel, monitorovanie, testy jednotiek, modely chýb, vizualizátor mobility.

Obrázok 17: Porovnanie podpory protokolov v jadre ns-2 a ns-3 (28)

Simulátor ns-3 v súčasnosti ešte nepodporuje všetko to, čo simulátor ns-2. V najbližšej dobe sú plánované ďalšie verzie tohto simulátora s doimplementovanými ďalšími funkciami a podporou ďalších protokolov, avšak stavať náš projekt na budúcich verziách nie je vhodné riešenie.

Podobne je to aj s podporou vizualizačných nástrojov – pre ns-3 ich nie je tak veľa ako pre ns-2, a niektoré sú ešte len vo fáze vývoja. Napríklad *Statistical framework for network simulations* bude v budúcnosti silným nástrojom, v súčasnosti však neposkytuje dostatok funkcií. Na základe vykonanej analýzy súčasného stavu a po konzultácii s vedúcim projektu možno konštatovať, že vytvárať podporné programy pre súčasnú verziu ns-3 by bolo samoučelné, keďže onedlho by sa stali zastaranými či zbytočnými (napr. po dokončení Statistical Frameworku). Je lepšie zamerať sa na plne-funkčný a stabilný ns-2 simulátor.

4 Špecifikácia

V projekte sa snažíme vytvoriť komplexné riešenie pre simuláciu počítačovej siete, ktoré by bolo jednoduché na používanie a malo by čo najlepšiu dostupnosť pre široké spektrum používateľov. Projekt by mohol napríklad dopomôcť k zlepšeniu výučby mnohých sieťových predmetov a prispel by k lepšiemu pochopeniu počítačových sietí.

V našom projekte chceme vytvoriť komplexné riešenie pre simuláciu počítačovej siete založenej na simulátore ns-2. Simulátor síce ponúka simuláciu používateľom zadanej počítačovej siete, avšak vstup vyžaduje dobrú znalosť skriptovacieho jazyka TCL. Používateľ je nútený naučiť sa a napísať si vlastné TCL skripty, ktoré potom poskytne simulátoru ako vstup. Taktiež výstup simulátora je náročnejší na čítanie a vyžaduje si znalosť jeho štruktúry, aby používateľ rýchlo vyčítal informácie, ktoré ho zaujímajú. Zameriame sa preto predovšetkým na spracovanie vstupov a výstupov simulátora.

5 Návrh

Navrhované riešenie je aplikácia umiestnená na linuxovom serveri. Rozhraním pre používateľa bude webová lokalita, na ktorej budú dostupné funkcie ako vytvorenie topológie zadaním počtu zariadení v sieti, ako sú prepínače, smerovače a koncové zariadenia. V ponuke budú aj vopred definované jednoduché a náročnejšie topológie. Zadanú topológiu po skončení simulácie používateľ uvidí v súhrnnom zobrazení aj s výsledkami. Zadávané topológie a výsledky simulácií budú uchovávané na serveri pre neskoršie využitie, prípadne s dovoľením používateľov umiestňované do databázy topológií, aby mohli byť poskytnuté aj pre iných používateľov. Výstupy simulátora budú spracovávané tak, aby boli jednoducho čitateľné pre používateľa a vedel z nich rýchlo a jednoducho vyčítať informácie o zadanej sieti, ktoré ho najviac zaujímajú. Tieto informácie, respektíve výsledky simulácie budú zobrazované na unikátnych URL na danej webovej lokalite, aby boli ľahko prístupné. Výsledky sa zobrazia pri vizualizácii simulovanej siete.

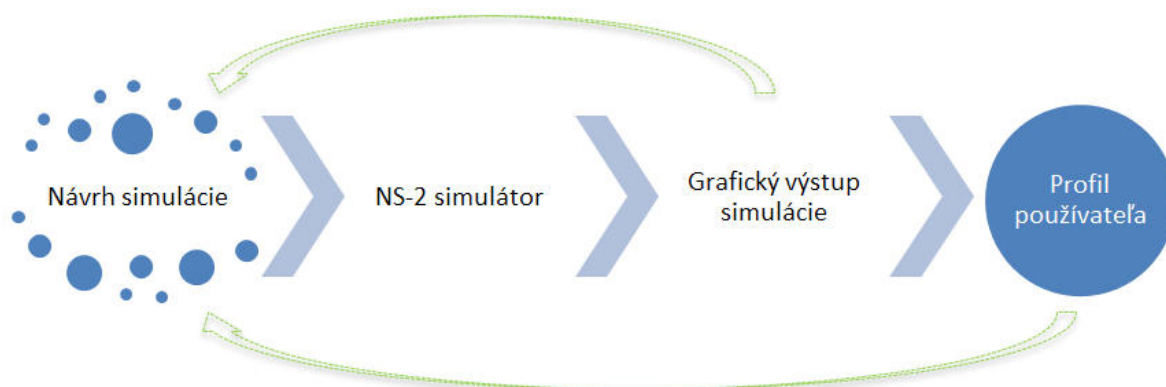
Aplikáciu sme sa rozhodli navrhnúť ako klient-server aplikáciu. Dôvodom takéhoto riešenia je, že niektorí používatelia môžu mať výkonnejší hardvér než akým je vybavený server. Simulácia bude prebiehať lokálne na počítači používateľa, čo spôsobí jej urýchlenie. Výstup simulácie potom používateľ iba odošle na server, ktorý ho následne spracuje a výsledky zobrazí na webovej lokalite. V prípade že používateľ nemá výkonnejší hardvér ako poskytuje server, môže simuláciu spustiť priamo na serveri. V tomto prípade však nemusí čakať a sledovať server kedy simulácia skončí, pretože aplikácia bude využívať e-mailový server, aby upozornila používateľa, že jeho simulácia skončila a taktiež kde môže nájsť jej výsledky.

Výhodou nášho návrhu je vysoká dostupnosť. Potrebný je iba Internet a webový prehliadač. Simulácie môžu prebiehať či už na serveri, alebo lokálne, podľa výberu používateľa. Celkové riešenie sprehľadňuje prácu pri simuláciách správania sa počítačových sietí a uľahčuje vytváranie vstupov a čitateľnosť výstupov. Výhodou taktiež je, že používateľ nemusí čakať na skončenie simulácie pri počítači, keďže čas každej simulácie je rôzny. Po skončení simulácie aplikácia používateľa upovedomí o skončení ním zadanej simulácie.

Nevýhodou takéhoto riešenia je, že pri veľkom množstve simulácií na serveri bude spracovávanie výsledkov časovo náročnejšie.

V projekte budú vo veľkej miere využívané najmä skriptovacie jazyky. Simulátor je kompatibilný s operačným systémom Linux a preto bude aplikácia projektovaná na linuxové webové servery. Preferovať budeme webový server Apache 2, avšak ak to bude potrebné, nebude problém vytvoriť aplikáciu aj pre iný webový server. Požiadavky na hardvér nie sú ničím výnimočné. Simulátor ns-2 však beží iba pod operačným systémom Linux a preto je potrebné nakonfigurovať linuxový server. Softvérové požiadavky taktiež nie sú špeciálne. Naše riešenie budeme vytvárať pomocou bežne dostupných nástrojov na operačnom systéme Linux. Časové požiadavky závisia od náročnosti simulácie a od výkonnosti hardvéru, na ktorom bude aplikácia umiestnená.

V nasledujúcom texte sú opísané typy používateľov, ich práva a jednotlivé komponenty systému. Popíšeme si funkcionality komponentov a zadefinujeme vstupy a výstupy. V závere kapitoly si uvedieme scenáre použitia. Zjednodušenú architektúru riešenia ilustruje Obrázok 18.



Obrázok 18: Architektúra riešenia

5.1 Používatelia

V systéme sa budú nachádzať 3 typy používateľov. Používatelia budú rozdelení na základe ich práv.

Typy používateľov:

- neprihlásený používateľ
- prihlásený používateľ
- administrátor

Neprihlásený používateľ má nasledujúce práva:

- zaregistrovať sa
- prezerať výsledky skončených simulácií
- porovnávať vybrané simulácie

Prihlásený používateľ po registrácii a prihlásení do systému má nasledujúce práva:

- vytvoriť vstupný súbor typu *.tcl
- vytvoriť pomocou autogenerátora súbory pre náhodný pohyb uzlov a náhodný tok dát medzi zariadeniami
- vytvoriť vlastné súbory pre náhodný pohyb zariadení a náhodný tok dát
- spustiť simuláciu na serveri
- zvoliť si parametre, ktoré sa rozhodol sledovať v danej simulácii
- vybrať si vizualizáciu simulácie pomocou obrázkov
- vybrať si vizualizáciu simulácie pomocou videa
- zvoliť si typy grafov, v ktorých budú vykreslené zvolené parametre zo simulácie
- po skončení simulácie zmeniť iba niektoré parametre na sledovanie v simulácii
- porovnať si výsledky simulácie s výsledkami iných simulácií
- uložiť si vstupný súbor typu *.tcl
- uložiť si výstupný súbor simulácie
- uložiť si výsledky simulácie v podobe obrázkov, grafov a/alebo videa
- nahrať na server svoj vlastný vstupný súbor typu *.tcl
- nahrať na server svoj vlastný výstupný súbor simulácie pre zobrazenie výsledkov pomocou obrázkov, grafov a/alebo videa

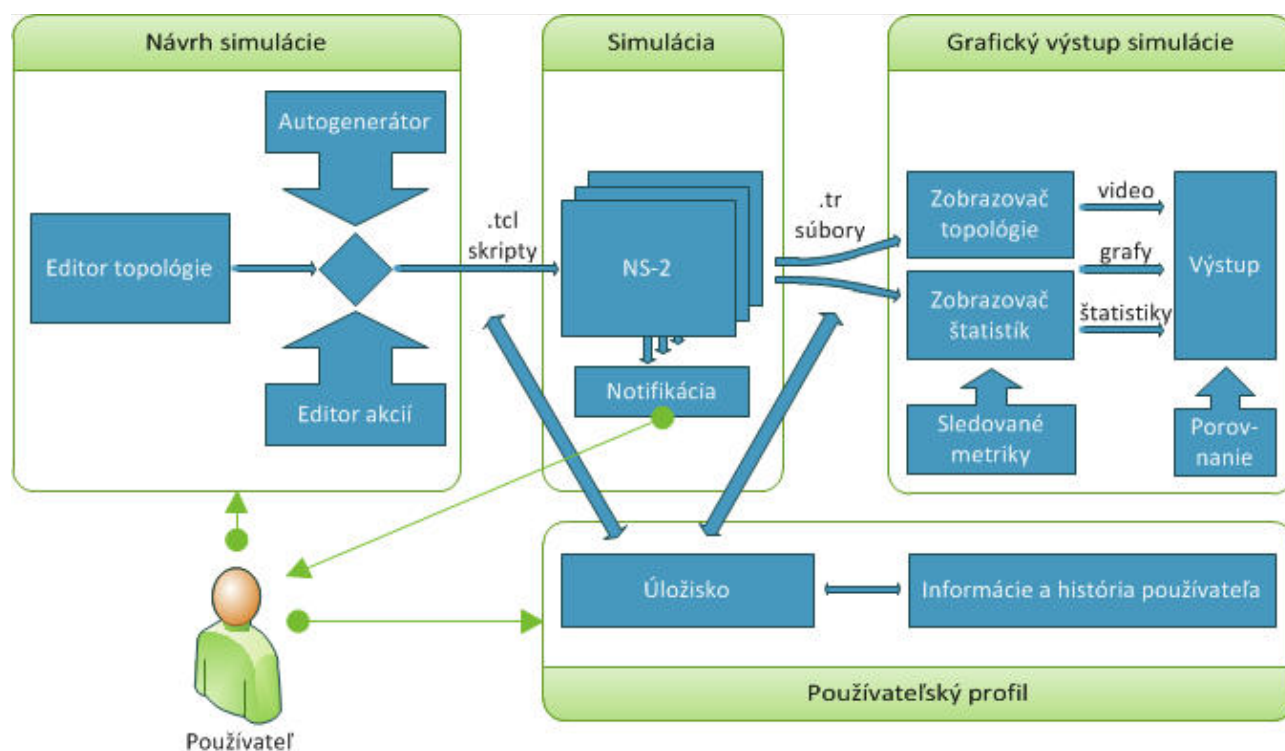
- vyžiadať si zaslanie e-mailu zo systému po skončení simulácie
- spustiť počas jedného prihlásenia naraz 3 simulácie
- pridať poznámky k výsledkom simulácie

Administrátor má v systéme všetky doteraz spomenuté práva a niektoré ďalšie:

- meniť už vygenerované vstupné súbory
- upravovať už zadané parametre pre sledovanie v simulácií
- spustiť neobmedzený počet simulácií
- vytvárať, meniť a upravovať databázy pre ukladanie jednotlivých súborov
- pridávať, meniť a odstraňovať používateľov

5.2 Popis komponentov

V tejto kapitole si popíšeme jednotlivé komponenty podieľajúce sa na celkovom riešení (Obrázok 19). Uvedieme funkcionality každého komponentu, zdefinujeme vstupy a výstupy.



Obrázok 19: Podrobná schéma riešenia

Editor topológie

- *funkcionalita*: Umožňuje používateľovi navrhnuť si vlastnú topológiu simulovanej siete
- *vstup*: Opisné parametre simulovanej siete
- *výstup*: *.tcl súbor (skript) s uvedenými parametrami siete

Autogenerátor

- *funkcionalita*: Umožňuje vygenerovať súbory náhodného pohybu mobilných zariadení a náhodný tok dát medzi týmito zariadeniami v sieti
- *vstup1*: typ protokolu (tcp/cbr), počet zariadení, počiatok, počet spojení, rýchlosť
- *vstup2*: počet zariadení, prestávka v sekundách, max. rýchlosť, trvanie simulácie, hraničné rozmery topológie
- *výstup*: súbor, ktorý sa následne vloží do výstupného súboru **Editora topológie**

Editor akcií

- *funkcionalita*: Umožňuje používateľovi vytvoriť vlastné nastavenia pohybu zariadení v sieti a toku dát medzi nimi
- *vstup*: súradnice miest, po ktorých sa budú jednotlivé zariadenia pohybovať. Definovanie modelu komunikácie.
- *výstup*: súbor, ktorý sa následne vloží do výstupného súboru **Editora topológie**

Tieto 3 komponenty vytvárajú spolu „**Návrh simulácie**“, ktorý je jeden zo 4 hlavných častí systému.

Simulátor NS-2

- *funkcionalita*: Vykonanie simulácie mobilnej bezdrôtovej počítačovej siete so zvolenými parametrami.
- *vstup*: *.tcl súbor (skript), výstupný súbor **Návrhu simulácie**
- *výstup*: *.tr súbor s výsledkami simulácie

Notifikácia

- *funkcionalita*: Po zvolení používateľom je na danú e-mailovú adresu odoslaný e-mail o ukončení simulácie
- *vstup*: e-mailová adresa používateľa, informácia zo systému o ukončení simulácie
- *výstup*: notifikačný e-mail o ukončení simulácie

Tento komponent tvorí jadro systému. Vykonáva samotnú simuláciu po prevzatí výstupného súboru z „**Návrhu simulácie**“ na svoj vstup. Jeho výstupný súbor slúži ako vstup pre „**Grafický výstup simulácie**“. Komponentom Notifikácia je používateľ upozornený o ukončení simulácie. Tento komponent nijako neovplyvňuje samotný priebeh simulácie.

Merané metriky

- *funkcionalita*: Definovanie metrík, ktoré používateľ chce sledovať v danej simulácii
- *vstup*: metriky, ktoré budú sledované v simulácii
- *výstup*: vstup pre jednotlivé zobrazovače

Zobrazovač topológie

- *funkcionalita*: Analyzuje výstupný súbor simulácie. Zobrazuje topológiu a simuluje pohyb zariadení a komunikáciu medzi nimi.
- *vstup*: *.tr súbor, výstupný súbor simulácie
- *výstup*: výstupný súbor vo video formáte

Zobrazovač štatistík

- *funkcionalita*: Analyzuje vstupný súbor a vyrába potrebné dáta pre parametre, ktoré si používateľ zadal na sledovanie. Následne vykreslí graf z daných dát.
- *vstup*: *.tr súbor, výstupný súbor simulácie
- *výstup*: graf nameraných hodnôt, prípadne štatistická tabuľka

Porovnanie

- *funkcionalita*: Na želanie používateľa zmení zobrazovanie výsledkov simulácie do módu, kde sa môžu porovnávať výsledky 2 simulácií
- *vstup*: výsledky vybraných simulácií na porovnanie
- *výstup*: porovnanie výsledkov zvolených simulácií

Vyššie uvedené 4 komponenty tvoria 3 zo 4 základných častí systému a to „**Grafický výstup simulácie**“. Vstupom tejto časti je výstup simulácie spracovaný zobrazovačmi do podoby zadanej používateľom.

Úložisko

- *funkcionalita*: Umožňuje používateľovi ukladať do databázy jednotlivé vstupné a výstupné súbory a taktiež výsledky simulácie
- *vstup*: vstupné, výstupné súbory a výsledky simulácie
- *výstup*: uložené súbory a dáta simulácií v databáze

Informácie a história používateľa

- *funkcionalita*: Uložené informácie o používateľovi a jeho činnosti so systémom. Informácie o simuláciách vytvorených daným používateľom. Poznámky k výsledkom jednotlivých simulácií.
- *vstup*: informácie o používateľovi, poznámky k výsledkom, história činnosti
- *výstup*: uložené informácie, poznámky a história činnosti daného používateľa

Spoločne tvoria tieto 2 komponenty poslednú hlavnú časť systému s názvom „**Používateľský profil**“. Slúži najmä na uchovanie informácií o jednotlivých používateľoch, súboroch a výsledkoch simulácií. Taktiež uchováva históriu činnosti používateľov v systéme

5.3 Návrh pozorovaných parametrov QoS

V tejto kapitole sme identifikovali niekoľko parametrov QoS, ktorých pozorovanie v bezdrôtových sieťach by mohlo byť nejakým spôsobom zaujímavé. Význam jednotlivých parametrov je bližšie opísaný nižšie. Vývoj niektorých parametrov počas doby simulácie bude graficky znázornený. Pre iné parametre budú vypočítané len ich priemerné hodnoty, tak ako to znázorňuje nasledujúca tabuľka :

Pozorovaný parameter	Jednotka	Osobitný výpočet pre každý tok	Priemerná hodnota	Aktuálne hodnoty (pre graf)
Priepustnosť	Mb/s	Áno	Áno	Áno
'Packet delivery fraction'	[%]	Áno	Áno	Nie
Množstvo stratených paketov	[Počet]; [B]	Áno	Áno	Áno
Stratovosť paketov	[%]	Áno	Nie	Nie
Jednosmerné oneskorenie	[ms]	Áno	Áno	Áno
Kolísanie oneskorenia	[ms]	Áno	Áno	Áno
Réžia smerovania	[B]	Nie	Áno	Áno
Normalizovaná réžia smerovania	[%]	Nie	Áno	Áno

Tabuľka 3: Zoznam pozorovaných parametrov QoS

Priepustnosť – Tento parameter bude vyhodnotený osobitne pre každý dátový tok. Hodnota priepustnosti dátového toku je daná množstvom dát prijatých na strane prijímača za daný čas.

Packet delivery fraction – Tento parameter udáva percentuálny podiel počtu prijatých paketov koncovou aplikáciou z celkového počtu odoslaných paketov.

Množstvo stratených paketov – počet paketov, ktoré neboli doručené koncovej aplikácií.

Stratovosť paketov – Tento parameter udáva percentuálny podiel zahodených paketov z celkového počtu odoslaných paketov.

Jednosmerné oneskorenie – Časový interval medzi odoslaním paketu a jeho prijatím koncovou aplikáciou.

Kolísanie oneskorenia–Kolísanie oneskorenia je rozdiel oneskorení dvoch po sebe odoslaných paketov patriacich do jedného dátového toku.

Réžia smerovania–Tento parameter vyjadruje množstvo smerovacích informácií odoslaných do siete za jednotku času.

Normalizovaná réžia smerovania – Tento parameter vyjadruje podiel komunikácie súvisiacej so smerovaním voči efektívnej komunikácií v sieti.

5.4 Návrh skriptov pre vyhodnotenie pozorovaných parametrov

Parametre kvality služieb uvedené v predchádzajúcej kapitole budú získané z výstupného súboru ns-2 simulátora. V tejto kapitole uvádzame postup, akým budú tieto parametre z výstupného súboru získané. Za týmto účelom je potrebné navrhnuť a napísať niekoľko skriptov, ktoré vytiahnu z textového súboru požadované údaje v potrebnej forme. Do úvahy budeme pritom brať len výstup typu newtrace, ktorý je určený pre simuláciu bezdrôtových sietí.

5.4.1 Priepustnosť dátových tokov

Priepustnosť bude meraná pre všetky dátové toky. Pre každý tok sa na koncovom uzly sčíta veľkosť všetkých prijatých paketov (príznak **-II** [í el]) patriacich do jedného dátového toku a vydelené sa časom, za ktorý boli tieto pakety prijaté. Tok je jednoznačne určený príznakmi **-Is** (zdrojová IP adresa a port) a **Id** (cieľová IP adresa a port). Koncové uzly sú označené príznakmi **AGT**, pričom označenie **s** na začiatku je vysielateľ a **r** prijímač. Výsledná hodnota bude udaná v jednotke Mb/s. Od časového intervalu, v ktorom budú jednotlivé hodnoty počítané bude závisieť jemnosť výsledného grafu. Skript bude do súboru zaznamenávať dvojice hodnôt vo formáte: [simulačný_čas, priemerná_priepustnosť].

Simulačný čas je daný príznakom **-t**.

Priemerná priepustnosť za daný časový interval t_0 sa vypočíta podľa nasledujúceho vzorca (okamžitá priepustnosť sa vypočítavať nedá) :

$$\frac{\sum \text{Veľkosti_paketov_za_čas_}t_0}{t_0} \times \frac{8}{1000000}$$

Priemerná priepustnosť počas celej doby simulácie sa vypočíta podľa rovnakého kľúča, s tým, že namiesto času t_0 bude dosadený čas celej simulácie. Výstupný súbor bude potom obsahovať iba jedinú hodnotu.

5.4.2 Počet doručených paketov voči počtu odoslaných

Tento parameter hovorí o tom aké percento odoslaných paketov bolo včas doručených do cieľa. Vypočíta sa jednoducho ako pomer počtu odoslaných paketov a prijatých paketov koncovou aplikáciou. Uvedený parameter je potrebné vypočítať pre každý dátový tok osobitne. Jednotlivé dátové toky treba rozlišovať podľa nastavenia **-Is** a **-Id** príznakov. Stratené pakety označené **d** pri tomto parametri neberieme do úvahy.

Vysielateľ je daný príznakmi **AGT + s**

Prijímač je daný príznakmi **AGT + r**

5.4.3 Množstvo stratených paketov

Uvedený parameter je potrebné vypočítať pre každý dátový tok osobitne. Jednotlivé dátové toky treba rozlišovať podľa nastavenia dvojice **-Is** a **-Id**. Počet stratených paketov označených ako **d** je potrebné počítať pre každý časový interval t_0 , aby bolo možné z vypočítaných hodnôt vykresliť graf. Skript bude vypočítané hodnoty zaznamenávať do súboru vo formáte :

[simulačný_čas; počet_stratených_paketov; veľkosť stratených paketov]

Simulačný čas je daný príznakom **-t**.

Počet stratených paketov dostaneme ako súčet záznamov označených **d**, za čas t_0 .

Veľkosť stratených paketov za daný časový interval t_0 sa vypočíta ako suma veľkosti (príznak **-II**) všetkých stratených paketov (**d**), za daný čas. Veľkosť paketov je udaná v bajtoch.

Priemerná stratovosť počas celej doby simulácie sa vypočíta podľa rovnakého kľúča, s tým, že namiesto času t_0 bude dosadený čas celej simulácie. Výstupný súbor bude potom obsahovať iba jedinú hodnotu.

5.4.4 Stratovosť paketov

Stratovosť paketov je percentuálna hodnota, ktorá udáva pomer zahodených paketov k počtu odoslaných paketov. Uvedený parameter je potrebné vypočítať pre každý dátový tok osobitne. Jednotlivé dátové toky treba rozlišovať podľa nastavenia dvojice **-Is** a **-Id**. Stratené pakety sú označené **d**, odoslané ako **AGT +s**. Tento parameter sa vypočíta len ako statická hodnota pre celý priebeh simulácie.

5.4.5 Jednosmerné oneskorenie

Oneskorenie je čas medzi vyslaním paketu a jeho prijatím. Uvedený parameter je potrebné vypočítať pre každý dátový tok osobitne. Jednotlivé dátové toky treba rozlišovať podľa nastavenia dvojice **-Is** a **-Id**. Hodnoty oneskorenia budú zaznamenávané pre každý odoslaný paket patriaci do dátového toku. Udalosť odoslania paketu je daná príznakmi **AGT + s** a prijatie paketu **AGT + r**.

Skript bude vypočítané hodnoty zaznamenávať do súboru vo formáte :

[simulačný_čas; jednosmerné_oneskorenie]

Priemerná hodnota jednosmerného oneskorenia pre dátový tok počas celej simulácie sa vypočíta ako suma jednosmerných oneskorení zaznamenaných v súbore vydelená počtom záznamov.

5.4.6 Kolísanie oneskorenia

Kolísanie oneskorenia je rozdiel oneskorení dvoch po sebe odoslaných paketov patriacich do jedného dátového toku. Jednotlivé dátové toky treba rozlišovať podľa nastavenia dvojice **-Is** a **-Id**. Hodnoty kolísania oneskorení sa vypočítajú ako absolútna hodnota rozdielu každých dvoch po sebe nasledujúcich jednosmerných oneskorení zaznamenaných v súbore s jednosmernými oneskoreniami. Hodnoty kolísania oneskorenia treba pre potreby grafického spracovania zaznamenať do súboru v tvare: [simulačný_čas; kolísanie_oneskorenia]

5.4.7 Réžia smerovania

Réžia smerovania vyjadruje množstvo smerovacích informácií odoslaných do siete za jednotku času. Tento parameter bude meraný pre celú sieť a zaznamenávané budú jej priemerné hodnoty počas časových intervalov t_0 . Hodnoty budú ukladané do súboru podobne ako pri predošlých parametroch určených na grafické vyhodnotenie.

Pakety odoslané smerovacím protokolom sú označené príznakmi **RTR + s + špecifická značka**.

Špecifická značka je unikátnym označením pre každý smerovací protokol :

AODV - AODV
DSDV - message
DSR - DSR
TORA - IMEP

5.4.8 Normalizovaná réžia smerovania

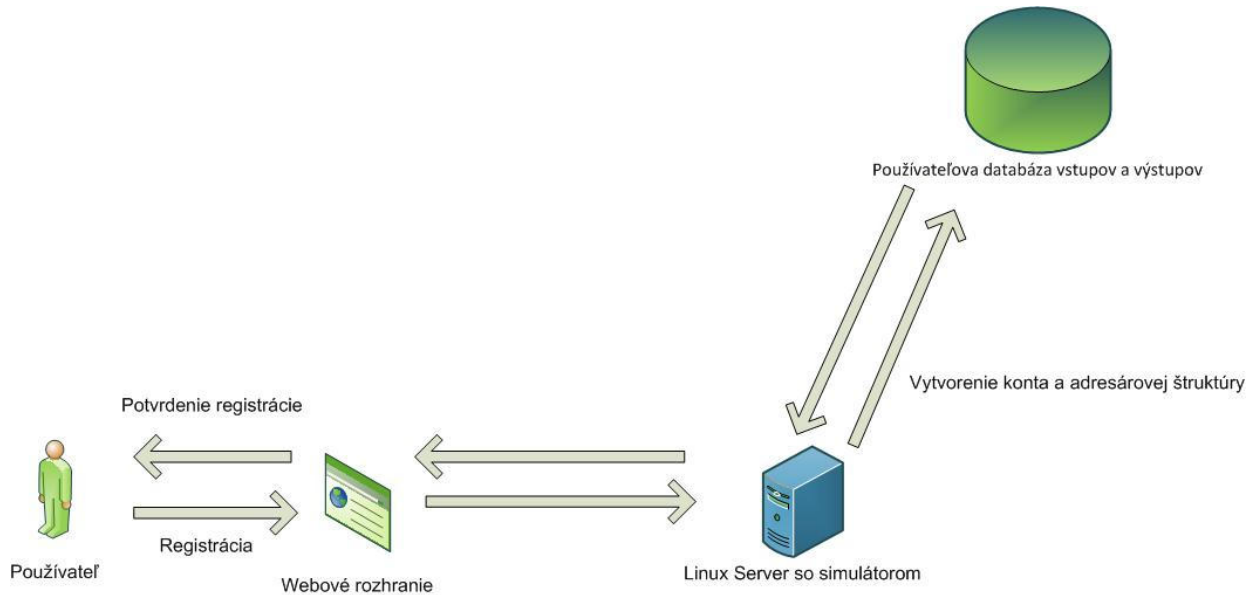
Normalizovaná záťaž spojená so smerovaním vyjadruje podiel komunikácie súvisiacej so smerovaním voči efektívnej komunikácii v sieti. Hodnota tohto parametra sa teda vypočíta ako pomer veľkosti všetkých odoslaných paketov súvisiacich so smerovaním a veľkosti paketov nesúcich aplikačné dáta. Tento parameter bude meraný pre celú sieť a zaznamenávané budú jej priemerné hodnoty počas časových intervalov t_0 . Hodnoty budú ukladané do súboru podobne ako pri predošlých parametroch určených na grafické vyhodnotenie.

Odoslané pakety patriace aplikáciám sú označené príznakmi **AGT + s**.

Pakety odoslané smerovacím protokolom sú označené príznakmi **RTR + s + špecifická značka**

6 Funkcionality podporované prototypom

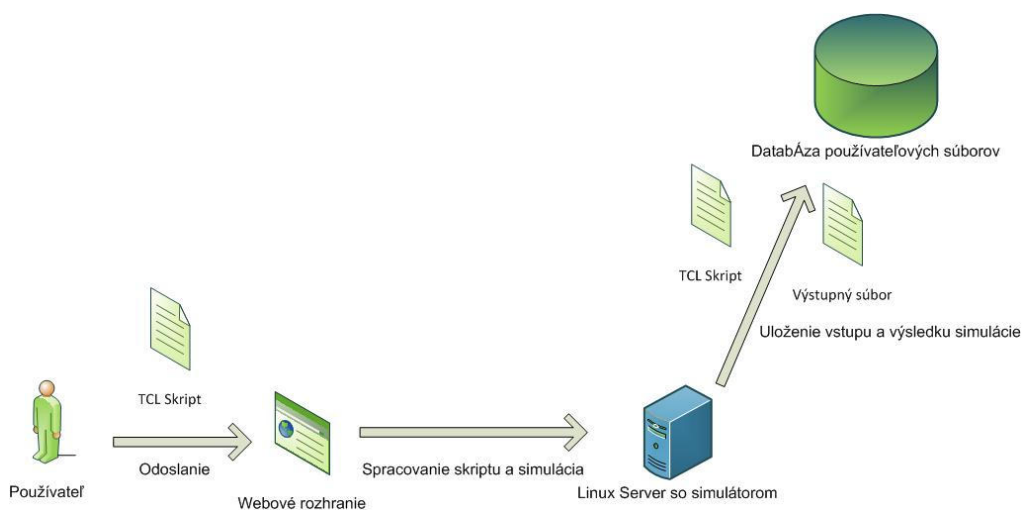
V nasledujúcej kapitole sú formou diagramov priblížené funkcie implementované v prototypu systému. Konkrétne ide o nasledujúce funkcie:



Obrázok 20: Registrácia a prihlásenie používateľa

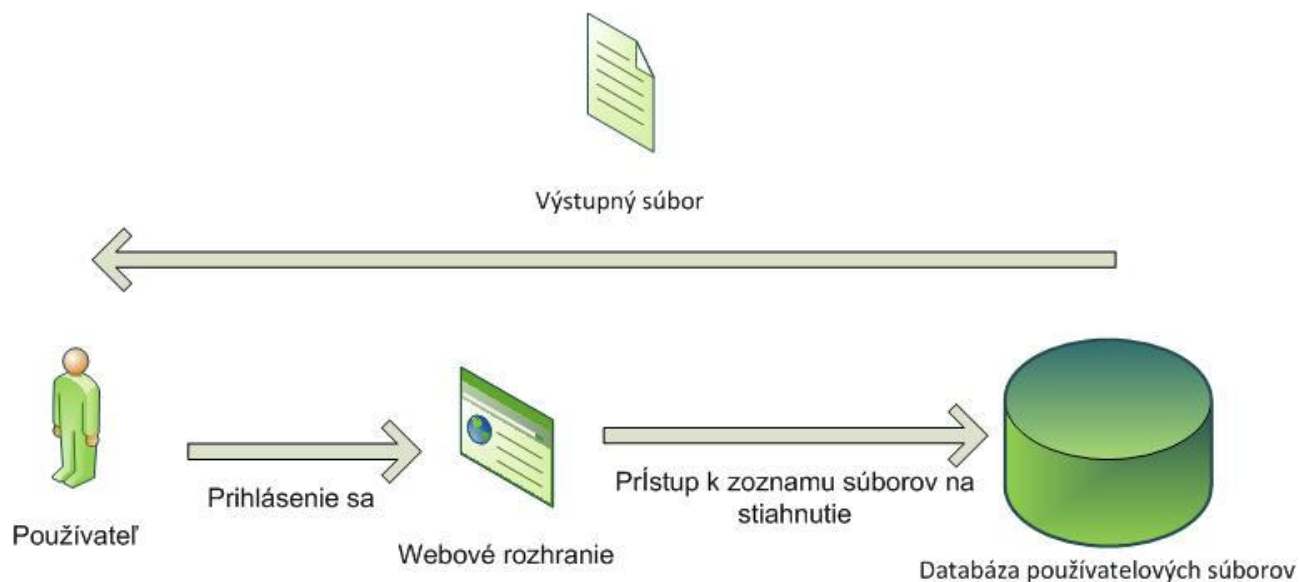
Na využívanie nášho systému je potrebné, aby mal používateľ v systéme vytvorené konto. Používateľ jednoducho vyplní údaje o sebe, najmä e-mailovú adresu, na ktorú mu budú chodiť notifikácie ohľadne dokončených simulácií. E-mailová adresa taktiež slúži ako prihlasovacie meno do systému. Po vytvorení konta sa používateľ môže do systému prihlásiť a pracovať so systémom.

V nasledujúcich situáciách predpokladáme, že používateľ je registrovaný a prihlásený v systéme.



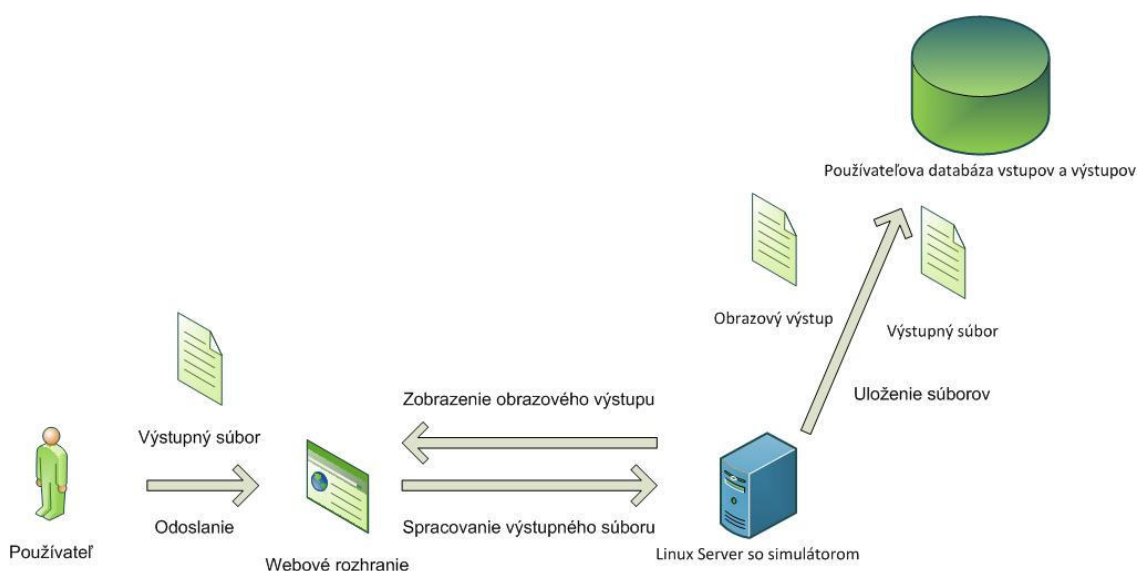
Obrázok 21: Zaslание simulácie na spracovanie

Po prihlásení do systému môže používateľ odoslať svoj test do systému. Test je následne kontrolovaný pre zabezpečenie správnej syntaxe. Ak prejde syntaktickou kontrolou, je spustený a výsledok simulácie a jej vstup je uložený do databázy k používateľovmu profilu. Používateľ si môže následne tieto simulácie znovu spustiť, editovať, poprípade stiahnuť zdrojový alebo výstupný súbor.



Obrázok 22: Stiahnutie výstupného súboru (trace file) zo servera

Ako už bolo spomenuté v predchádzajúcej situácii, používateľ si môže stiahnuť výstupný súbor. Systém túto funkciu umožňuje cez webové rozhranie. Účelom tejto funkcie je následná analýza výstupného súboru na používateľovom počítači.

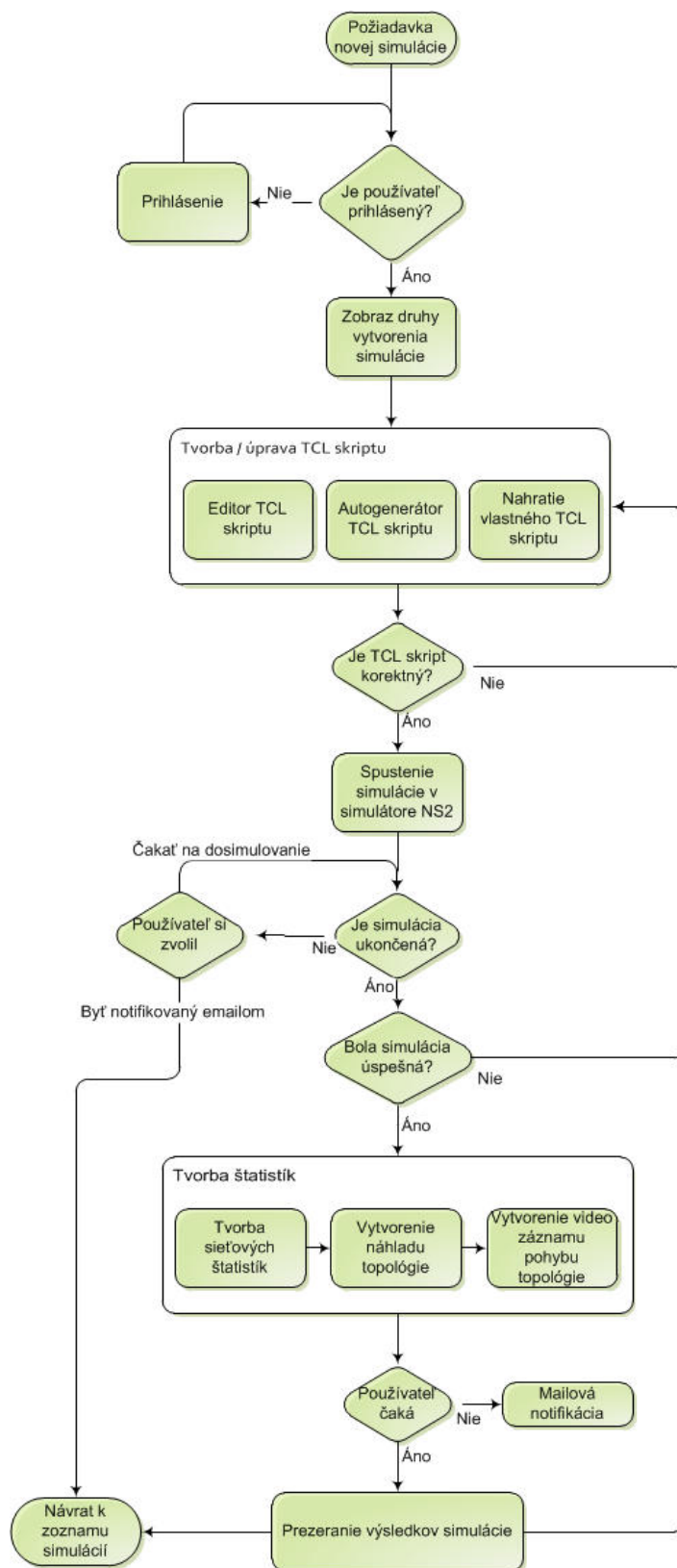


Obrázok 23: Zaslanie výstupného súboru na server za účelom vytvorenia grafického výstupu

Samozrejme používateľ sa môže rozhodnúť pre analýzu súboru v našom systéme. Systém obsahuje sadu skriptov, ktoré vizualizujú niektoré parametre siete. Používateľ jednoducho odošle výstupný súbor na server (alebo ho vyberie z databázy, pokiaľ sa jedná o výstupný súbor zo simulácie v systéme) a ten ho následne spracuje. Výstupný súbor aj obrazový výstup sú uložené na server do profilu používateľa.

7 Funkcionality podporované finálnou verzou

V nasledujúcej kapitole sú formálnymi prostriedkami opísané funkcionality podporované finálnou verzou systému.



Obrázok 24: Flow diagram systému

Obrázok 24 znázorňuje flow diagram pre implementovaný systém. Na používanie systému musí byť používateľ prihlásený. Následne môže vytvoriť simuláciu. Simulácia je definovaná TCL skriptom, ktorý systém umožňuje vytvoriť buď pomocou editoru, autogenerátoru, alebo priamo nahratím z disku lokálneho počítača. Systém ďalej kontroluje správnosť TCL skriptu, a to najmä:

- či sa jedná o bezdrôtovú simuláciu
- či je v skripte správne zadefinované sledovanie parametrov na úrovni MAC, agentov a smerovacích informácií
- či je výstupný súbor zadefinovaný v novom formáte

Ak TCL skript prejde všetkými kontrolami, je spustený v simulátore. Používateľ si môže vybrať, či chce čakať na dokončenie simulácie, alebo radšej uprednostní e-mailovú notifikáciu. Tu musíme podotknúť, že notifikácia je síce implementovaná, ale vzhľadom na obmedzenie zo strany FIIT, nie je možné túto funkcionálnosť použiť.

Po skončení simulácie sa automaticky generujú štatistické dáta. Medzi ne patrí: náhľad topológie, videozáznam pohybu uzlov a všeobecné sieťové štatistiky. Štatistiky sú zobrazené vo forme osobitných grafov pre každý pozorovný parameter kvality služieb, ako priepustnosť, oneskorenie, kolísanie oneskorenia a iné.

Po vygenerovaní týchto údajov môže používateľ dané údaje analyzovať a na základe nich simuláciu upraviť a spustiť znovu.

Obrázok 25 sumarizuje prípady použitia pre implementovaný systém.



Obrázok 25: Prípady použitia systému

8 Overenie správnosti výstupov

Nami navrhnutý systém vyhodnocuje niekoľko vytypovaných parametrov QoS, ktorých pozorovanie v bezdrôtových sieťach je často predmetom výskumu v tejto oblasti. Parametre QoS sú štandardne definované v dokumentoch RFC organizácie IETF. Z týchto definícií vychádzajú aj naše vzorce, na základe ktorých sú pozorované parametre počítané. Algoritmy pre výpočet boli implementované vo vlastnoručne zhotovených skriptoch, ktorých bližšia funkcionálna bola opísaná v kapitole zaoberajúcej sa návrhom systému. Vlastnoručné zhotovenie nových skriptov, nás prinútilo podrobne preštudovať všetky detaily spojené s vyhodnocovaním výstupných súborov ns-2 simulátora. Pri prebratí vyhodnocovacích skriptov z iného projektu by mohlo dôjsť k prehliadnutiu niektorých viac či menej dôležitých aspektov.

Samotný fakt, že nami navrhnuté skripty vychádzajú zo štandardnej definície parametrov QoS je, ak nie zárukou, tak aspoň predpokladom ich správnosti. Výstupy všetkých skriptov boli samozrejme aj riadne otestované. Počas testovania bolo vykonaných viac ako 300 simulácií, pričom výstupy boli podrobne analyzované, pričom sa overovalo, či namerané hodnoty korešpondujú s očakávaniami. Pri testovaní boli odhalené mnohé drobné chyby, ktoré čiastočne zkreslovali výpovednú hodnotu výsledku. Na základe priebehu záverečných testov môžeme skonštatovať, že systém ponúka na výstupe správne hodnoty pozorovaných parametrov QoS, špecifikovaných v návrhu.

9 Záznam o používaní systém

Po dôkladnom návrhu systému nasledovala jeho implementácia. Keďže sme si zvolili webové riešenie, prvým krokom bola implementácia samotného webového rozhrania. Po jeho dokončení a riadnom otestovaní sme sa mohli posunúť na ďalšiu etapu, ktorou bola inštalácia servera a migrácia stránky na tento server. Následne bolo na server nainštalované potrebné softvérové vybavenie (simulátor a jeho podporné moduly) a mohli sme začať z ich integráciou.

Od apríla sme začali systém aktívne používať. Samozrejme najprv testoval každý člen tímu iba svoj kód (skripty na generovanie grafov, modul na generovanie videa a pod.). Neskôr sa tieto moduly začali integrovať so simulátorom a stránkou. Do tejto doby išlo všetko podľa plánu. Až v máji sa začali objavovať problémy s grafickým rozhraním, ktoré vyžaduje modul pre vizualizáciu pohybu uzlov a topológie. Zdlhavý proces predstavovala aj spoločná integrácia skriptov, ktoré vyhodnocujú jednotlivé parametre do jedného veľkého nástroja.

Posledné dni sa systém vystavoval záťažovým testom, keďže všetci členovia tímu na ňom ladili svoje moduly. Musíme skonštatovať, že systém tento nápor zvládol. Pri tomto masívnom používaní boli odhalené faktické chyby vo výpočte parametrov jednotlivých sieťových veličín, ktoré boli promptne odstránené. Pri používaní systému bolo vytvorených a spustených viac ako 300 simulácií, čo dokumentuje snahu tímu o vytvorenie použiteľného a správne fungujúceho systému.

10 Čo sme nestihli

Počas používania a testovania systému sme prišli s mnohými nápadmi a vylepšeniami, ktoré sa nám aj podarilo implementovať. Bohužiaľ išlo iba o doplnenie funkcionality, ktoré bolo implementačne najmenej náročné, ale prinieslo používateľovi väčší komfort. Medzi funkcie, ktoré sa nám implementovať nepodarilo patrí editovanie simulačných skriptov autogenerátorom, ktoré by pravdepodobne vyžadovalo významné zásahy do systému (úprava databázy a pod.).

Taktiež pre prehľadnosť grafov, by bolo vhodné implementovať používateľky nastaviteľnú granularitu grafov. Na tento nedostatok sa však prišlo až v poslednej fáze testovania. Preto táto relatívne jednoduchá funkcionality nie je implementovaná. Príjemným rozšírením systému by bolo tiež fórum, kde by si používatelia mohli vymieňať svoje skúsenosti a rady súvisiace s používaním nášho systému. Ideálne by bolo vytvorenie komunity používateľov.

Na strane simulátora sa nám nepodarilo sfunkčniť smerovanie TORA a DSDV. Toto smerovanie nefunguje v našej verzii simulátora a je to známy problém, ktorý sa snažia autori opraviť. Pri spustení simulácie s daným druhom smerovania dôjde k páde simulátora. Tento pád sa nám podarilo odstrániť, avšak uzly stále nevytvárajú smerovacie tabuľky podľa prijatých smerovacích správ. Na Internete existuje mnoho návodov na opravu tejto chyby, bohužiaľ tie, ktoré sme skúšali nevedeli k zdarnému koncu.

Okrem nedostatkov, ktoré sme identifikovali sami, nás nápadmi obohatili aj kolegovia na študentskej vedeckej konferencii. Medzi ich nápady patrí napríklad spúšťanie simulácií v distribuovanom prostredí. Avšak táto funkcionality je podporovaná až v simulátore ns3. Pre ns2 existuje síce rozšírenie, ktoré však nie je plne kompatibilné s najnovšou verziou simulátora. Teda jeho použitie by si vyžadovalo veľké zmeny kódu simulátora, poprípade celého serverového prostredia

Posledný podnet zo strany kolegov na študentskej vedeckej konferencii, bol návrh vytvorenia LIVE CD, s obrazom nášho systému. Používateľ, ktorý by chcel nasadiť systém v domácich podmienkach, by nemusel postupovať podľa návodu, ale jednoducho by spustil systém z CD. Následne by ho z LIVE CD mohol aj inštalovať.

Keď sa zhrnú všetky vyššie spomínané vylepšenia, vieme si predstaviť pokračovanie nášho tímového projektu v ďalších semestroch. Kostra riešenia je modulárna, preto je do budúcnosti možné simulátor ns-2 vymeniť za novší ns-3 simulátor. S takouto obmenou by sa však spájali početné zmeny v moduloch, ktoré spracúvajú vstupy a výstupy simulátora.

11 Čo sme sa naučili

Prínos projektu pre jednotlivých členov tímu môžeme rozdeliť na dve hlavné stránky:

- sociálnu
- vedomostnú.

Zo sociálnej stránky je to hlavne práca v kolektíve. Tím musel diskutovať o problémoch, ktoré vznikali, pretože tieto problém sa dotýkali viacerých členov. Samotná komunikácia však nestačí, bolo treba tieto problém spoločne riešiť. Pri riešení týchto problémov sa stretnutia vo väčších skupinách ukázali ako veľmi efektívne. Samozrejme na komunikáciu v tíme sa používali aj rôzne elektronické prostriedky (Google groups, ICQ, Skype), avšak tie neboli zďaleka tak efektívne ako osobné stretnutia.

Poslednou sociálnou zručnosťou, ktorú tím získal je rešpekt k vedúcemu. Práve vedúci riadi celý tím, pozná jeho slabé stránky a taktiež silné stránky jeho členov. Preto vie efektívne prideliť úlohy v tíme, poprípade poradiť na koho sa obrátiť s konkrétnym problémom. Nerešpektovanie vedúceho teda vedie k rozporom tíme, zlému rozdeleniu práce, čím v konečnom dôsledku utrpí kvalita výsledného produktu.

Z profesijných zručností môžeme spomenúť hlavne prax v analýze problémov a následne aj schopnosť túto analýzu zdokumentovať. V budúcnosti bude pre nás, ako profesionálov v oblasti informačných technológií veľmi dôležité, aby sme sa dokázali vyjadriť jasne a jednoznačne. V profesijnom živote nás často budú trápiť termíny, a tie nás neobišli ani pri tomto projekte. Ich dodržovanie je dôležité pre pohodu v tíme a zachovanie kvality produktu. Na toto by si mal dať pozor každý tím a je to hlavne zodpovednosť vedúceho, aby dokázal prácu na projekte naplánovať tak, aby bolo dostatočne veľa času na riešenie nečakaných komplikácií a chýb.

Okrem vyššie spomínaných bola pre tím veľkou skúsenosťou aj študentská vedecká konferencia, na ktorú sme sa s našim systémom dostali.

12 Zhodnotenie

V tejto kapitole zhodnotíme výsledok celého projektu vzhľadom na ciele, ktoré sme si stanovili v začiatkoch práce.

Medzi hlavné ciele, ktoré sme si vytýčili patrilo vytvorenie serverového riešenia, ktoré by podporovalo manažovanie simulácií prostredníctvom webstránky. Tento bod sme splnili stopercentne. Vytvorené riešenie poskytuje komfortné prostredie pre simuláciu počítačových sietí prostredníctvom webového rozhrania, kde používateľ podľa svojho rozhodnutia môže na skončenie simulácie počkať, alebo byť o ňom notifikovaný prostredníctvom e-mailu.

Požiadavku na simuláciu bezdrôtových sietí sme taktiež splnili. Riešenie v súčasnosti plne podporuje smerovacie protokoly AODV a DSR. Implementácia podpory protokolov DSDV a TORA je takmer dokončená, problematickým bodom ich integrácie do nášho riešenia však je fakt, že jadro simulátora ns-2 ich stopercentne nepodporuje. Počas práce na projekte sme otestovali množstvo patchov, ktoré mali využívanie DSDV a TORA smerovania v ns-2 simulátore umožniť, žiaden však nefungoval. Jestvuje však množstvo patchov, ktoré sme otestovať nestihli. V prípade rozširovania nášho produktu v budúcnosti je teda veľký predpoklad, že sa správny patch nájde.

Ďalším cieľom bolo vytvorenie skriptov, ktoré by v spolupráci s grafickým programom generovali prehľadné grafy zobrazujúce niektoré základné parametre simulovaných sietí. V súčasnosti naše riešenie generuje pre každý dátový tok grafy stratovosti paketov, priepustnosti dátových tokov, jednosmerného oneskorenia, kolísania oneskorenia, smerovacej záťaže a normalizovanej smerovacej záťaže. Zároveň sa pre každú simuláciu zobrazí počet odoslaných a zahodených paketov pre každý dátový tok. Pozitívom je taktiež vygenerovanie videa zobrazujúceho pohyb a komunikáciu medzi uzlami, ktoré naše riešenie vytvorí pre každú simuláciu.

Hlavnou ideou nášho riešenia bolo zjednodušenie práce s ns-2 simulátorom pre používateľa. Pri pohľade na vytvorené riešenie môžeme skonštatovať, že túto ideu splňa, no nie úplne. Vzhľadom na fakt, že vyhodnocovanie simulácií, ktoré náš produkt poskytuje, je striktne zamerané na bezdrôtové siete, pričom plne podporuje len smerovacie protokoly AODV a DSR, môžeme konštatovať, že oblasť sieťových simulácií, ktoré naše riešenie zjednodušuje, je relatívne úzka. Pre daný typ simulácií však poskytuje rôzne možnosti, ktoré prácu s týmto typom simulácií značne uľahčia.

Výrazným pozitívom je aj fakt, že naše riešenie je relatívne jednoducho rozšíriteľné o podporu iných typov simulácií. Vďaka jeho modulárnosti by s implementáciou ďalších rozšírení, po zoznámení sa so štruktúrou riešenia, nemal mať problém žiaden skúsenejší programátor.

Celkovo hodnotíme výstup nášho projektu ako riešenie splňajúce drvivú väčšinu špecifikovaných požiadaviek, no zároveň obsahujúce aj prídavnú, vopred neplánovanú funkcionálnu.

13 Záver

V tomto dokumente sme sa zaoberali zjednodušením sieťových simulácií pomocou simulátora ns-2. Navrhnuté serverové riešenie, ktoré tvorí základ nášho produktu, poskytuje oproti klientskému riešeniu nespočetné množstvo výhod, ktoré ocenia hlavne neskúsení používatelia sieťových simulácií, ale aj používatelia, ktorí nemajú k dispozícii dostatočne veľkú výpočtovú kapacitu pre realizáciu zložitých simulácií.

Používateľské rozhranie, ktoré je riešené ako jednoduchá a intuitívna webová stránka, plní jeden z cieľov stanovených v úvode práce, a to cieľ priblíženia sieťových simulácií študentom a menej skúseným používateľom. Možnosť zadávania vstupných údajov pre simuláciu buď vo forme TCL skriptu, alebo za pomoci prostriedkov na vytváranie TCL skriptov integrovaných do našej webstránky, zabezpečuje značnú flexibilitu realizácie simulácií. Naše riešenie štandardne poskytuje ako výstup simulácie pestrú paletu graficky znázornených výsledkov, ktorých interpretácia je zrejmalá a jasná aj pre menej sieťovo zdatných používateľov. Zároveň však naše riešenie poskytuje používateľovi možnosť prezerania celého výstupného súboru ns-2 simulátora. Táto funkcionálna umožňuje individuálne spracovanie výstupných súborov simulácie pre zdatnejších používateľov. Všetky tieto faktory spoločne prispievajú k vysokej použiteľnosti nášho produktu, čím sme splnili náš zámer.

Pri riešení tohto projektu sme sa ponorili do tajov tímovej práce so všetkým, čo k tomu patrí. Naučili sme sa akceptovať autority, v tomto prípade zvoleného lídra, čo sa môže v našom ďalšom profesijnom živote ukázať ako výrazná devíza. Práca v tíme nás zároveň naučila používať prostriedky podporujúce kolaboráciu, ako je napríklad Dropbox.

V neposlednom rade je nutné podotknúť, že simulácia počítačových sietí je technická oblasť, ktorá je väčšine členov nášho tímu veľmi blízka, preto bola práca na tomto projekte pre nás nie len obohatením, ale aj potešením. Napriek nespočetným bezsenným nociam, počas ktorých nás tento projekt obral o spánok, musíme skonštatovať, že tieto noci boli investované do niečoho, čo malo zmysel, a na čo sme hrdí.

14 Bibliografia

1. **Institute for Simulation & Training**. <http://www.ist.ucf.edu/background.htm>. [Online] University of Central Florida, Október 2010. [Dátum: 5. November 2010.]
2. **Ns-2 Roadmap**. [Online] [Dátum: 5. November 2010.] <http://nslam.isi.edu/nslam/index.php/Roadmap>.
3. **ISSARIAKUT T, EKRAM H.** *Introduction to Network Simulator NS2*. 2009. ISBN: 978-387-71759-3.
4. **Ns-2 User information**. [Online] [Dátum: 5. November 2010.] http://nslam.isi.edu/nslam/index.php/User_Information.
5. **FALL, K a VARADHAN, K.** *The ns Manual*. 2000. http://nslam.isi.edu/nslam/index.php/NS_manual.
6. **Tutorial for the Network Simulator "ns"**. [Online] [Dátum: 5. November 2010.] <http://www.isi.edu/nslam/ns/tutorial/index.html>.
7. **WANG G, XIA Y, HARRISON, D.** *An NS2 TCP Evaluation Tool: Installation Guide and Tutorial*. 2007.
8. **WILLIAMS T, KELLY C.** *GnuPlot documentation*. 2003. http://www.gnuplot.info/docs_4.4/gnuplot.pdf.
9. **MERRITT, E.** Demo scripts for gnuplot version 4.4. [Online] Apríl 2009. [Dátum: 5. November 2010.] <http://gnuplot.sourceforge.net/demo>.
10. **Xgraph team**. Xgraph documentation & tutorial. [Online] Apríl 2007. [Dátum: 5. November 2010.] <http://www.xgraph.org/>.
11. **BUCHHEIM, T.** Nam: Network Animator. [Online] Júl 2002. [Dátum: 5. November 2010.] <http://www.isi.edu/nslam/nam/>. <http://www.isi.edu/nslam/nam/>.
12. **Toilers team**. *About iNSpect*. 2010. <http://toilers.mines.edu/Public/Code/Nsinspect.html>.
13. **JONES, E.** *802.11 Simulations Using ns2*. 2005. <http://evanjones.ca/ns2.html>.
14. **WEINGÄRTNER, E.** *A performance comparison of recent network simulators*. 2009. <http://portal.acm.org/citation.cfm?id=1817510>.
15. **Ns-3 proposal to NSF: Project summary**. [Online] 22. September 2006. [Dátum: 5. November 2010.] <http://www.nslam.org/docs/proposal/summary.pdf>. <http://www.nslam.org/docs/proposal/summary.pdf>.
16. **Ns-3 overview**. [Online] August 2010. [Dátum: 5. November 2010.] <http://www.nslam.org/docs/ns-3-overview.pdf>. <http://www.nslam.org/docs/ns-3-overview.pdf>.
17. **Ns-3 proposal to NSF: Project description**. [Online] 22. September 2006. [Dátum: 5. November 2010.] <http://www.nslam.org/docs/proposal/project.pdf>.
18. **Ns-3 Software Architecture**. [Online] 16. Október 2007. [Dátum: November. 5 2010.] <http://www.nslam.org/docs/design.pdf>.
19. **Ns-3 Tutorial**. [Online] November 2010. [Dátum: 5. November 2010.] <http://www.nslam.org/docs/tutorial.pdf>.
20. **LACAGE, M.** An ns-3 tutorial. [Online] Apríl 2010. [Dátum: 5. November 2010.] <http://www.nslam.org/tutorials/ns-3-tutorial-tunis-apr09.pdf>.
21. **NetAnim**. [Online] [Dátum: 5. November 2010.] <http://www.nslam.org/wiki/index.php/NetAnim>.
22. **VIRGENSH, R.** Diskusný príspevok. [Online] September 2006. http://groups.google.com/group/ns-3-users/browse_thread/thread/458d5d9de0a7397?pli=1.
23. **PyViz**. [Online] Október 2010. <http://www.nslam.org/wiki/index.php/PyViz>.
24. **Statistical Framework for Network Simulation**. [Online] Máj 2008. [Dátum: 5. November 2010.] http://www.nslam.org/wiki/index.php/Statistical_Framework_for_Network_Simulation.
25. **M, LACAGE**. An ns-3 tutorial. [Online]

26. **CARNEIRO, G.** Ns-3 network simulator, lab brief on ns-3. [Online] 20. April 2010. [Dátum: 5. November 2010.] <http://www.nsnam.org/tutorials/NS-3-LABMEETING-1.pdf>.
27. Ns-3 tutorial. [Online] <http://www.nsnam.org/docs/tutorial.pdf>.
28. **HENDERSON, T a LACAGE, M.** *An end-to-end tout of a simulation*. 2. Marec 2009. <http://www.nsnam.org/workshops/wns3-2009/ns-3-tutorial-part-2.pdf>.

Príloha A

Dokumentácia k riadeniu projektu

1 Riadenie projektu

Nasleduje časť dokumentu, ktorá sa zaoberá riadením projektu Programová podpora pre sieťový simulátor, ktorý vznikol na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Obsahuje ponuku, plán projektu, krátkodobé a dlhodobé úlohy jednotlivých členov tímu, zápisnice zo stretnutí tímu, manažment verzií, konfigurácií a zmien, a preberacie protokoly.

Ponuka
Tímový projekt

Team3

Autori:

Bc. Martin Nagy
Bc. Dávid Oros
Bc. Roman Panenka
Bc. Martin Pirháč
Bc. Hana Severínová
Bc. Martin Svetlák

predmet: Tímový projekt 1
akademický rok: 2010/2011

Predstavenie členov tímu

- **Bc. Martin Nagy:** Absolvent bakalárskeho štúdia na FIIT STU v študijnom programe Počítačové systémy a siete. Momentálne pracuje vo vývojovom stredisku firmy Alcatel – Lucent ako SGSN feature tester. Jeho skúsenosti s testovaním aplikácií na platforme Linux budú veľkým prínosom pre kvalitu projektu. V predchádzajúcej práci získal praktické skúsenosti s konfiguráciou siete a riešením problémov v sieti. V bakalárskej práci sledoval zahľtenie v sieti a okrajovo sa venoval aj rôznym sieťovým simulátorom, čo je tiež dôležitá skúsenosť pre danú tému.
- **Bc. Dávid Oros:** Absolvent bakalárskeho štúdia na FIIT STU v Bratislave. Popri štúdiu získal mnoho skúsenosti s administráciou unix/linux operačných systémov. Taktiež získal vedomosti a skúsenosti s návrhom a konfiguráciou počítačových sietí. Ďalej by sa rád zaoberal serverovými riešeniami a konfiguráciou unix/Linux serverov pre konkrétne serverové aplikácie. Je členom Platon Group (skupina vyvíjajúca voľne šíriteľný softvér pod licenciou GNU/GPL). Má praktické skúsenosti s virtualizačnými nástrojmi.
- **Bc. Roman Panenka:** Absolvent bakalárskeho štúdia na FIIT STU, študijného programu Počítačové systémy a siete. Popri štúdiu rozširoval svoje vedomosti v oblasti teórie a konfigurácie počítačových sietí, ktoré nadobudol počas štúdia na regionálnej sieťovej CISCO akadémii. Tieto vedomosti pretavil do podoby CCNA certifikátu a ďalej ich používa pri vzdialenej podpore zákazníckych sietí spoločnosti HP. Jeho prínosom do tímového projektu by boli aj bohaté skúsenosti s prácou v tíme, ktoré nadobudol pri vypracovávaní úspešného projektu do medzinárodnej súťaže Imagine Cup.
- **Bc. Martin Pirháč:** Absolvent bakalárskeho štúdia na FIIT STU, študijného programu Počítačové systémy a siete. Počas štúdia sa venoval predovšetkým problematike počítačových sietí. Prínosom k projektu by mohla byť znalosť sieťového simulátora ns-2, ktorý bol kľúčovým komponentom v systéme navrhnutom v rámci bakalárskej práce. Práca bola orientovaná na analýzu niektorých prístupov predchádzania zahľtenia a správy radov v počítačových sieťach.
- **Bc. Hana Severínová:** Absolventka bakalárskeho štúdia na FIIT STU v odbore Počítačové systémy a siete. Počas štúdia sa venovala hlavne problematike operačného systému Linux a počítačovým sieťam. Má skúsenosti s programovaním zavádzaných modulov jadra operačného systému Linux, pri čom nadobudla bohaté skúsenosti so skriptovacími jazykmi bash a C-shell. Momentálne popri štúdiu pracuje ako systémový administrátor, zároveň sa zaoberá správou Oracle a MS SQL databáz.
- **Bc. Martin Svetlík:** Absolvent bakalárskeho štúdia na FIIT STU, študijného programu Počítačové systémy a siete. Popri štúdiu pracoval na Inštitúte informatiky a štatistiky, kde získal ďalšie skúsenosti s Unixovými strojmi a štatistickým vyhodnocovaním. Okrem toho má skúsenosti s tvorbou webových stránok.

Motivácia

Všetci členovia nášho tímu sú absolventi bakalárskeho štúdia na FIIT STU, program Počítačové systémy a siete. Všetci členovia tímu už majú aj mnoho skúsenosti s reálnym návrhom a konfiguráciou počítačových sietí, sieťových zariadení a taktiež koncových zariadení. Medzi koncové zariadenia samozrejme patria klientské stanice, ale aj servery.

Pre túto tému sme sa rozhodli, lebo by sme radi naďalej rozvíjali naše skúsenosti v tejto oblasti. Náš projekt by sme radi orientovali aj smerom k študentom. Nami navrhované riešenie by dopomohlo k zlepšeniu výučby najmä predmetu Počítačové siete II. Prispelo by k lepšiemu pochopeniu problematiky počítačových sietí.

V neposlednom rade je pre nás motiváciou vytvoriť komplexné riešenie pre simuláciu počítačovej siete, ktoré by bolo jednoduché na používanie a malo by čo najlepšiu dostupnosť pre široké spektrum používateľov.

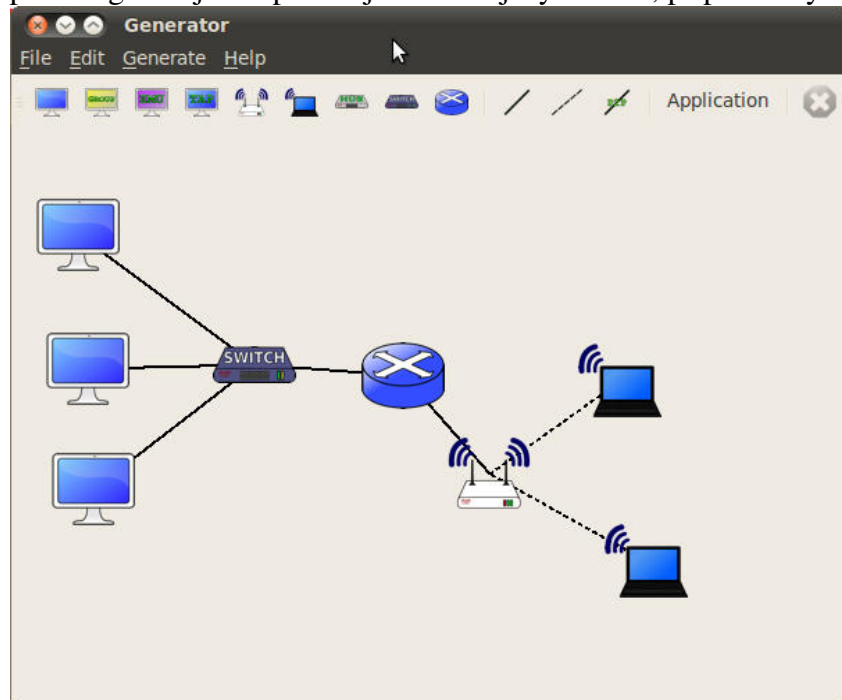
Existujúce riešenia

Pre simulátory ns-2 a ns-3 existuje v súčasnosti viacero rozširujúcich modulov, ktoré umožňujú generovanie simulačných scenárov, ich vizualizáciu, ako aj štatistické vyhodnotenie a grafické zobrazenie výsledkov simulácií. V našom projekte plánujeme analyzovať možnosti existujúcich modulov a niektoré ich funkcie využiť v našom systéme. Niektoré existujúce moduly, ktoré by potencionálne mohli byť využité v našom systéme pre uľahčenie prípravy a vyhodnotenia simulácií sú uvedené nižšie.

Rozširujúce moduly pre simulátor ns-3

Ns3Generator¹

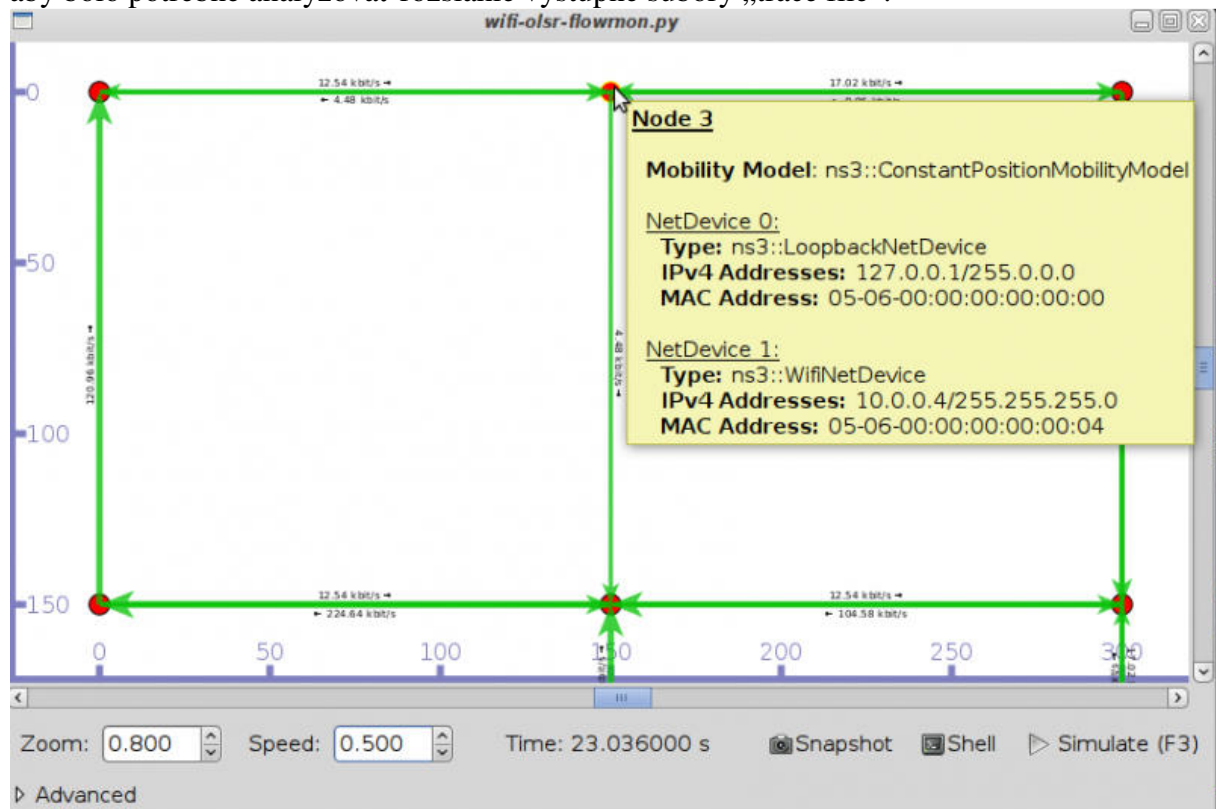
Modul umožňuje prostredníctvom GUI intuitívne navrhnuť topológiu pre simuláciu, pričom generuje zodpovedajúci kód v jazyku C++, prípadne Python.



¹ <http://www.nsnam.org/wiki/index.php/Ns3Generator>

PyViz²

PyViz slúži na vizualizáciu simulácií v reálnom čase. Počas simulácie je teda možné sledovať priebeh simulácie a aktuálny stav jednotlivých sieťových prvkov bez toho, aby bolo potrebné analyzovať rozsiahle výstupné súbory „trace file“.

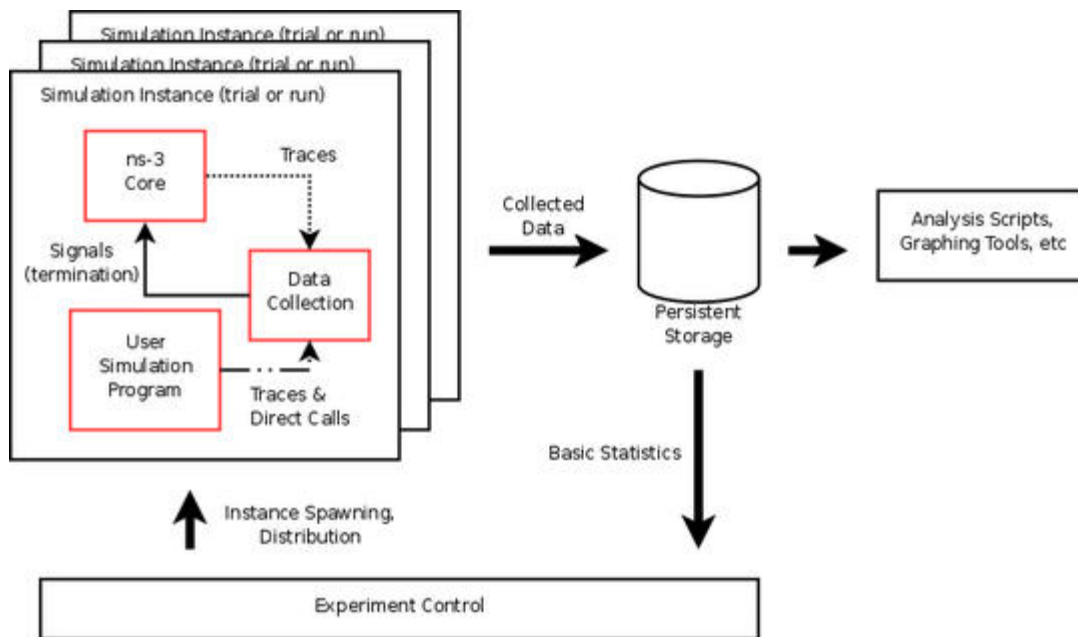


Statistical Framework for Network Simulation³

Tento modul je v súčasnosti vo vývoji. Ako samotný názov napovedá, cieľom tohto modulu je poskytnúť vhodné prostredie pre analýzu a štatistické vyhodnotenie simulácie. Dôraz sa pritom kladie na minimalizáciu potreby narábať s rozsiahlymi výstupnými súbormi „trace file“, čo by malo výrazne zvýšiť efektivitu simulácií a skrátiť dobu ich trvania. Modul by mal tiež umožniť reprodukovateľnosť a opakovateľnosť simulačných scenárov. Chýbať by nemala ani možnosť zastaviť simuláciu pri dosiahnutí určitého stavu. Rovnaké ciele si kladieme aj pri vývoji nášho systému. Konceptuálny návrh systému je zobrazený na nasledujúcom obrázku.

² <http://www.nsnam.org/wiki/index.php/PyViz>

³ http://www.nsnam.org/wiki/index.php/Statistical_Framework_for_Network_Simulation



Rozširujúce moduly pre ns-2

Simulátor ns-2 prešiel od svojho vzniku v roku 1996 rozsiahlym vývojom. Preto existuje pre ns-2 oproti jeho potencionálnemu nástupcovi ns-3 (predstavený v r.2008) oveľa viac rozširujúcich modulov. Cieľom tohto dokumentu však nie je ich podrobná analýza, preto uvádzame len odkaz na ich zoznam:

http://nsnam.isi.edu/nsnam/index.php/Contributed_Code#Support.

Návrh riešenia

V našom projekte chceme vytvoriť komplexné riešenie pre simuláciu počítačovej siete založenej na simulátore NS-2, resp. NS-3. Oba tieto simulátory síce ponúkajú simuláciu používateľom zadanej počítačovej siete, avšak vstup spomínaných simulátorov vyžaduje dobrú znalosť skriptovacieho jazyka TCL. Používateľ je nútený naučiť sa a napísať si vlastné TCL skripty, ktoré potom poskytne simulátoru ako vstup. Taktiež výstup simulátora je náročnejší na čítanie a vyžaduje si znalosť jeho štruktúry, aby používateľ rýchlo vyčítal informácie, ktoré ho zaujímajú. Zameriame sa preto predovšetkým na spracovanie vstupov a výstupov simulátora.

Navrhované riešenie je aplikácia umiestnená na linuxovom serveri. Rozhraním pre používateľa bude webová lokalita, na ktorej budú dostupné funkcie ako vytvorenie topológie zadaním počtu zariadení v sieti, ako sú prepínače, smerovače a koncové zariadenia. V ponuke budú aj vopred definované jednoduché a náročnejšie topológie. Zadanú topológiu po skončení simulácie používateľ uvidí v súhrnnom zobrazení aj s výsledkami. Zadávané topológie a výsledky simulácií budú uchovávané na serveri pre neskoršie využitie, prípadne s dovolením používateľov umiestňované do databázy topológií, aby mohli byť poskytnuté aj pre iných používateľov. Výstupy simulátora budú spracovávané tak, aby boli jednoducho čitateľné pre používateľa a vedel z nich rýchlo a jednoducho vyčítať informácie o zadanej sieti, ktoré ho najviac zaujímajú. Tieto informácie, respektíve výsledky simulácie budú zobrazované na unikátnych URL na danej webovej lokalite, aby boli ľahko prístupné. Výsledky sa zobrazia pri vizualizácii simulovanej siete.

Aplikáciu sme sa rozhodli navrhnuť ako klient-server aplikáciu. Dôvodom takéhoto riešenia je, že niektorí používatelia môžu mať výkonnejší hardvér než akým je vybavený server. Simulácia bude prebiehať lokálne na počítači používateľa, čo spôsobí jej urýchlenie.

Výstup simulácie potom používateľ iba odošle na server, ktorý ho následne spracuje a výsledky zobrazí na webovej lokalite. V prípade že používateľ nemá výkonnejší hardvér ako poskytuje server, môže simuláciu spustiť priamo na serveri. V tomto prípade však nemusí čakať a sledovať server kedy simulácia skončí, pretože aplikácia bude využívať e-mailový server, aby upozornila používateľa, že jeho simulácia skončila a taktiež kde môže nájsť jej výsledky.

Výhodou nášho návrhu je vysoká dostupnosť. Potrebný je iba internet a webový prehliadač. Simulácie môžu prebiehať či už na serveri, alebo lokálne, podľa výberu používateľa. Celkové riešenie sprehladňuje prácu pri simuláciách správania sa počítačových sietí a uľahčuje vytváranie vstupov a čitateľnosť výstupov. Výhodou taktiež je, že používateľ nemusí čakať na skončenie simulácie pri počítači, keďže čas každej simulácie je rôzny. Po skončení simulácie aplikácia používateľa upovedomí o skončení ním zadanej simulácie.

Nevýhodou takéhoto riešenia je, že pri veľkom množstve simulácií na serveri bude spracovávanie výsledkov časovo náročnejšie.

Alternatívne riešenia

Alternatívami k tomto riešeniu budú najmä skriptovacie jazyky a ponuka webových serverov. Simulátor je kompatibilný s operačným systémom Linux a preto aplikácia bude projektovaná na linuxové webové servery. Taktiež je na výber programovací jazyk danej webovej lokality. Preferovať budeme webový server Apache 2, avšak ak to bude potrebné, nebude problém vytvoriť aplikáciu aj pre iný webový server.

Istá alternatíva je dostupná už priamo v riešení, a to kde bude prebiehať simulácia. Tento výber je iba na používateľovi. Výhody a nevýhody oboch budú popísané v dokumentácii k aplikácií.

Požiadavky

Požiadavky na hardvér nie sú ničím výnimočné. Simulátor NS-2/NS-3 však beží iba pod operačným systémom Linux a preto je potrebné nakonfigurovať linuxový server. Softvérové požiadavky taktiež nie sú špeciálne. Naše riešenie budeme vytvárať pomocou bežne dostupných nástrojov na operačnom systéme Linux. Časové požiadavky závisia od náročnosti simulácie a od výkonnosti hardvéru, na ktorom bude aplikácia umiestnená.

Prečo by táto téma mala byť pridelená práve nám

Ponúkame komplexné riešenie, ktoré má vysokú dostupnosť, je orientované na uľahčenie práce používateľa pri používaní sieťových simulátorov. Sprehladňuje spracovávanie výsledkov simulácie a taktiež zjednodušuje vytváranie vstupov pre simulátor. Uceluje pohľad na prácu s počítačovými sieťami.

V našom tíme sú ľudia, ktorí majú prax a skúsenosti v každej oblasti, ktorú naše riešenie zahŕňa, ako je konfigurácia a správa linuxového serveru, návrh a spracovanie vstupov a výstupov pomocou skriptovacích jazykov, vývoj webovej aplikácie ako aj z oblasti bezpečnosti serverových aplikácií. V neposlednom rade máme skúsenosti s návrhom a realizáciou reálnych počítačových sietí.

Priority jednotlivých tém

1. Programová podpora pre sieťový simulátor
2. Vývoj vzorových aplikácií pre IPTV v prostredí Microsoft Mediaroom Framework
3. Diagnostika porúch diskretných udalostných systémov založená na modeloch a aplikácie v informatike
4. Prostredie pre návrh digitálnych systémov (Digital System Designer)
5. Simulátor komunikácie v počítačovej sieti
6. Využitie vlastností HP Tablet PC na inováciu vyučovania predmetu Testovateľnosť digitálnych systémov
7. Návrh vnoreného systému (Embedded Development)
8. Využitie vlastností HP Tablet PC na inováciu vyučovania
9. Distribuovaný odkladací priestor virtuálnej pamäte

Príloha 1 – Sumárny rozvrh členov tímu a návrh času stretnutí

		7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	
		7:50	8:50	9:50	10:50	11:50	12:50	13:50	14:50	15:50	16:50	17:50	18:50	19:50	20:50	
Pondelok	Martin Nagy										TP1	Preferovaný čas stretnutí				
	Dávid Oros										TP1					
	Roman Panenka										TP1					
	Martin Pirháč										TP1					
	Hana Severínová										TP1					
	Martin Svetlák			TK					ZKGRA		TP1					
Utorok	Martin Nagy	KOD											BMIS	BMIS		
	Dávid Oros															
	Roman Panenka															
	Martin Pirháč	KOD														
	Hana Severínová												BMIS	BMIS		
	Martin Svetlák									BPS						
Streda	Martin Nagy		BKS		KSS		KSS				KOD					
	Dávid Oros				KSS		KSS									
	Roman Panenka		BKS		KSS		KSS						BPS			
	Martin Pirháč		BKS		KSS		KSS				KOD		BPS			
	Hana Severínová	BKS			KSS		KSS						BPS			
	Martin Svetlák				BKS											
Štvrtok	Martin Nagy	BPS	BKS								APS		VSPI			
	Dávid Oros	BPS	BKS								APS		VSPI			
	Roman Panenka	BPS	BKS								APS		VSPI			
	Martin Pirháč	BPS	BKS								APS		VSPI			
	Hana Severínová	BPS	BKS								APS		VSPI			
	Martin Svetlák	BPS	BKS					ZKGRA			APS		VSPI			
Piatok	Martin Nagy															
	Dávid Oros															
	Roman Panenka															
	Martin Pirháč															
	Hana Severínová															
	Martin Svetlák														Doplňujúce pedagogické štúdium	

Plán projektu

V tabuľke 1 je znázornený plán vypracovávania projektu. Tento plán bol priebežne aktualizovaný vzhľadom na aktuálny stav riešenia projektu. Spätným pohľadom na tento plán môžeme konštatovať, že bol z väčšej časti dodržaný a splnený.

Dátum	Činnosť
September 2010	Príprava ponuky
4.10.2010	Pridelenie úloh v tíme, zriadenie projektovej dokumentácie
11.10.2010	Úvodné analýzy simulátorov ns-2 a ns-3
18.10.2010	Hĺbková analýza simulátora ns-2, zadefinovanie rozhodovacích kritérií pre výber simulátora
4.11.2010	Hĺbková analýza simulátora ns-3, porovnanie simulátorov ns-2 a ns-3, konečný výber simulátora
8.11.2010	Kompletizácia časti analýza a ucelenie časti návrh a špecifikácia prvej verzie dokumentu.
November 2010	Vytváranie podrobného návrhu riešenia
November 2010	Spojzdenie servera, na ktorom pobeží výsledné riešenie.
December 2010	Vytvorenie skriptov na spracovanie výstupných súborov simulátora ns-2.
December 2010	Vytvorenie prototypu riešenia
Február 2011	Analýza existujúcich prostriedkov na vyhodnocovanie výstupných súborov simulátora.
Február 2011	Príprava stránky na integráciu so simulátorom.
Marec 2011	Inštalácia potrebných programov na server.
Marec 2011	Vytvorenie príspevku na študentskú vedeckú konferenciu IIT.SRC.
Marec 2011	Analýza štruktúry výstupných súborov simulátora pre rôzne smerovacie protokoly.
Apríl 2011	Vytvorenie posteru na IIT.SRC.
Apríl 2011	Realizácia vyhodnocovacích skriptov.
Apríl 2011	Prepojenie webstránky so skriptami.
Apríl 2011	Inštalácia grafických programov na server.
Máj 2011	Odladenie riešenia a doimplementovanie chýbajúcich častí.
Máj 2011	Dohotovenie dokumentácie.

Tabuľka 1. Plán projektu

Úlohy členov tímu

Nasleduje zoznam krátkodobých a dlhodobých úloh jednotlivých členov tímu. Dlhodobé úlohy budú mať členovia tímu až do skončenia práce na projekte, krátkodobé úlohy sú jednorazové.

Bc. Martin Nagy

- dlhodobé úlohy:
 - riadenie tímu
- krátkodobé úlohy:
 - analýza simulátora ns-3 – časť úvod a inštalačné manuály
 - časť dokumentu Návrh
 - formálna úprava dokumentu
 - analýza funkcionality TCP Evaluation Tool a iNSpect
 - návrh vyhodnocovacích skriptov
 -

Bc. Dávid Oros

- krátkodobé úlohy:
 - analýza ns-2 – podpora bezdrôtových sietí, TCL
 - špecifikácia riešenia
 - inštalácia a nastavenie programového vybavenia na server
 - implementácia vyhodnocovacích skriptov

Bc. Roman Panenka

- dlhodobé úlohy:
 - správa tímovej webstránky
- krátkodobé úlohy:
 - analýza simulátora ns-2 – časť možnosti vizualizácie topológie, podporné nástroje tretích strán
 - špecifikácia riešenia

Bc. Martin Pirháč

- dlhodobé úlohy:
 - kontrolovanie projektových termínov
- krátkodobé úlohy:
 - prezentácia o simulátore ns-2
 - analýza ns-2 – ns-2 prehľad, výhody oproti ns-3
 - finálne úpravy dokumentu
 - základné scenáre simulácií
 - návrh vyhodnocovacích skriptov

Bc. Hana Severínová

- dlhodobé úlohy:
 - vytváranie zápisníc zo stretnutí
- krátkodobé úlohy:
 - analýza ns-3 – vstupné a výstupné súbory simulátora ns-3, podpora bezdrôtových sietí
 - úvod a riadiaca dokumentácia projektu
 - vytvorenie posteru na IIT.SRC
 - implementácia vyhodnocovacích skriptov

Bc. Martin Svetlák

- krátkodobé úlohy:
 - analýza ns-3 – programová podpora
 - zhodnotenie a záver dokumentu
 - návrh grafickej časti riešenia - GnuPlot, iNSpect
 - implementácia skriptov realizujúcich grafické vyhodnotenie výsledkov simulácie

Podiel autorov na produkte

Vzhľadom na fakt, že jednotliví členovia tímu mali na projekte rôzne úlohy na rôznej úrovni náročnosti, je relatívne problematické určiť percentuálny podiel jednotlivých autorov na výslednom produkte. Keďže však každá úloha, ktorú tím realizoval či už jednotlivo, alebo skupinovo bola pre výsledné riešenie kritická, môžeme percentuálny podiel každého člena odhadnúť ako hruba rovnocenný.

Kvôli dodržaniu formálnej stránky dokumentu uvádzame odhadnutý percentuálny podiel v tabuľke 2.

Meno	Percentuálny podiel práce na projekte
Martin Nagy	16,67%
Dávid Oros	16,67%
Roman Panenka	16,67%
Martin Pirháč	16,67%
Hana Severínová	16,67%
Martin Svetlík	16,67%

Tab.2 - Percentuálny podiel autorov na výslednom produkte

Zápisnica zo stretnutia Tímu 3

Stretnutie č.1

Dátum: 4.10.2010
Čas: 16:00
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlík

Náplň stretnutia :

- Stanovenie termínu pravidelných stretnutí tímu na pondelok 17:00 – 20:00 hod
- Pridelenie hesiel do softvérového laboratória
- Určenie niektorých funkcií v tíme :
 - Riadenie tímu: Bc. Martin Nagy
 - Správa webu: Bc. Roman Panenka
 - Zápis stretnutí : Bc. Hana Severínová
- Ciele do najbližšieho stretnutia :
 - Zriadenie e-mailovej skupiny pre tím a jeho vedúceho
 - Návrh dizajnu tímového webu
 - Nainštalovanie simulátora ns-3 (v3.9)

Zápisnica zo stretnutia Tímu 3

Stretnutie č.2

Dátum: 11.10.2010
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová. Bc. Martin Svetlík

Kontrola plnenia úloh stanovených na poslednom stretnutí - e-mailová skupina pre tím vytvorená

Náplň stretnutia

- kontrolór deadlineov na stránke prof.Bielikovej a Ing. Hudeca – Bc. Martin Pirháč
- načrtnutie plánu postupu prác na projekte
- určenie obsahu jednotlivých častí projektu
- zadefinovanie rozhodovacích kritérií pre výber simulátoru: súčasný stav daného simulátora, zhodnotenie funkcionality, čo poskytujú, programovací jazyk, inštalácia, hardvérové požiadavky, dostupné moduly, výstup a iné

Ciele do najbližšieho stretnutia

- predstavenie webstránky – Bc. Roman Panenka
- prednáška Úvod do skriptovacieho jazyka TCL a simulátora ns-2 – Bc. Martin Pirháč
- inštalácia ns-2 aj ns-3
- porovnanie simulátorov ns-2 a ns-3 z vybraných hľadísk
- porovnanie ns-2 a ns-3, potom hrubá analýza a návrh.

Zápisnica zo stretnutia Tímu 3

Stretnutie č.3

Dátum: 18.10.2010

Čas: 17:30

Miesto: Softvérové štúdio

Účast': Ing. Peter Magula, Bc. Martin Nagy (meškanie 1/2 hodiny), Bc. Dávid Oros,
Bc. Roman Panenka, Bc. Martin Pirháč, Bc. Hana Severínová. Bc. Martin

Svetlík

Kontrola plnenia úloh stanovených na poslednom stretnutí

- webstránka vytvorená
- prednáška odprezentovaná

Náplň stretnutia

- prezentácia sieťového simulátoru ns-2 a TCL skriptov
- stanoviť náhradný termín stretnutia v siedmom týždni – štvrtok 18:00
- v šiestom týždni: porovnanie simulátorov ns-2 a ns-3 v dvoch tímoch – ktorý simulátor je lepší?
- vybraný typ sietí: bezdrôtové (802.11, Bluetooth,...)
- rozdelenie do dvoch tímov:
 1. analýza ns-2: Dávid Oros, Martin Pirháč, Martin Svetlík
 2. analýza ns-3: Martin Nagy, Roman Panenka, Hana Severínová

Ciele do najbližšieho stretnutia

- zistiť veľkosť identifikátoru vo výstupnom súbore ns-2 simulátora
- premyslieť vizualizáciu topológie zo zdrojového súboru
- prezentácia o ns-3
- popísať ns-2, jeho vlastnosti, architektúra, na základné porovnanie
- analýza existujúcich riešení
- podľa zadania do tímu naštudovať simulátor so zameraním na bezdrôtové siete
- do siedmeho týždňa: analýza ns-2 a ns-3, definitívny výber simulátora, definitívny výber typu sietí, veličiny pre vybraný typ sietí

Zápisnica zo stretnutia Tímu 3

Stretnutie č.3.1

Dátum: 27.10.2010

Účast': Bc. Dávid Oros, Bc. Roman Panenka, Bc. Martin Pirháč

Ciele do najbližšieho stretnutia

- Dávid Oros:
 - podpora bezdrôtových sietí
 - TCL
- Martin Pirháč:
 - ns-2 overview, výhody oproti ns-3
 - zistiť veľkosť identifikátoru vo výstupnom súbore ns-2 simulátora
- Roman Panenka:
 - možnosti vizualizácie topológie
 - podporné nástroje tretích strán
- Termín odovzdania: 31.10.2010

Zápisnica zo stretnutia Tímu 3

Stretnutie č.3.2

Dátum: 2.11.2010

Miesto: blok D

Účasť: Bc. Martin Nagy, Bc. Hana Severínová, Bc. Martin Svetlík

☐ **Ciele do najbližšieho stretnutia**

- Martin Nagy:
 - úvod do simulátora ns-3
 - inštalačné manuály ns-3
- Hana Severínová:
 - vstupné a výstupné súbory simulátora ns-3
 - podpora bezdrôtových technológií
- Martin Svetlík:
 - programová podpora pre ns-3

Zápisnica zo stretnutia Tímu 3

Stretnutie č.4

Dátum: 4.11.2010

Čas: 18:00

Miesto: blok C, tretie poschodie

Účast': Ing. Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová

Kontrola plnenia úloh stanovených na poslednom stretnutí

- diskusia ns-2 verzus ns-3
- výber ns-2, pretože ns-3 má veľa chýb a ešte nie je veľmi doimplementované, nedá sa spoľahnúť

že v ďalších vydaniach bude doimplementované všetko to, čo je naplánované

Ciele do najbližšieho stretnutia

- dopísať úvod – prečo je vhodné simulovať (lebo je náročné a drahé robiť si laboratórium keď treba veľa hardvéru, ...) - Hana Severínová
- analýza toho, aké riešenia už pre ns-2 existujú
- vlastnosti ns-2, hĺbková analýza simulátora – čo vie ponúknuť
- všetky obrázky prepísať do slovenčiny
- preveriť, či na TPcup treba pracovať podľa nejakej metodiky – Martin Nagy
- premyslieť budúcnosť nášho projektu, kam chceme smerovať
- do 8.11. hotová predbežná analýza
- zistiť, či sa na konci semestra dokladá CD, či treba mať funkčný prototyp, alebo stačia ukážky – Martin Pirháč
- hĺbková analýza ns-2 – Dávid Oros, Roman Panenka, Martin Pirháč
- zhodnotenie analýzy – Martin Svetlík
- formálne úpravy a manažment okolo dokumentu, spájanie čiastkových dokumentov – Martin Nagy

Zápisnica zo stretnutia Tímu 3

Stretnutie č.5

Dátum: 8.11.2010

Čas: 18:00

Miesto: Softvérové štúdio

Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlík

Náplň stretnutia

- Kontrola a finalizácia obsahu dokumentu
- Návrh obsahu špecifikácie
- Návrh obsahu časti Návrh – vytvorenie blokového diagramu systému
- Rozhodnutie, či projekt prihlásiť do TPcup-u a IITSRC – Tpcup nie, IITSRC áno

Ciele do najbližšieho stretnutia

- Odovzdanie dokumentácie v podateľni
- Vytlačiť, vyplniť a dať potvrdiť preberací protokol
- Doplniť zápisnice, plán projektu a úvod k časti riadiaca dokumentácia
- Doplniť krátkodobé a dlhodobé úlohy jednotlivých členov tímu
- Do zhodnotenia analýzy dopísať čo chceme urobiť
- Do návrhu doplniť blokový diagram
- Do špecifikácie: špecifikácia funkcií, priority, typy používateľov a priradiť im funkcie
- Zadefinovať aj obmedzenia systému

Zápisnica zo stretnutia Tímu 3

Stretnutie č.6

Dátum: 15.11.2010

Čas: 18:00

Miesto: Softvérové štúdio

Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka, Bc. Martin Svetlík

Náplň stretnutia

- Odovzdanie práce P. Magulovi
- Oboznámenie sa s prácou Tímu 8
- Dohodnutie funkcií, ktoré sa budú implementovať do prototypu
 - Inštalácia operačného systému a jadra simulátora
 - Registrácie
 - Notifikačné maily
 - Uživatelské rozhranie
 - Uploadovanie TCL skriptu
 - Simulácie cez web
 - Stiahnutie tracefile
 - Jednoduché grafy

Ciele do najbližšieho stretnutia

- Posudok práce Tímu 8 (zodpovedný: M. Svetlík)
- Dodať viac bibliografie do finálneho dokumentu
- Lepšie citácie v analýze
- Rozšíriť hrubý návrh na finálny návrh
- Dohodnúť s Ing. Steinmüllerom server – Linux, Internet, SSH, port 80, 20GB HDD,
- 1GB RAM (P. Magula)

Zápisnica zo stretnutia Tímu 3

Stretnutie č.7

Dátum: 22.11.2010

Čas: 18:00

Miesto: Softvérové štúdio

Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlík

Náplň stretnutia

- Diskusia o posudku pre oponentský tím
- Diskusia o posudku oponentského tímu
- Rozprava o stave vyjednaní servera

Ciele do najbližšieho stretnutia

- Dokončenie posudku pre oponentský tím
- Objednanie a vyzdvihnutie serveru u p. Lacka
- Predbežná prezentácia webového rozhrania riešenia
- Práca na dokumentácii

Zápisnica zo stretnutia Tímu 3

Stretnutie č.8

Dátum: 29.11.2010

Čas: 18:00

Miesto: Softvérové štúdio

Účast': Ing. Peter Magula, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlík

Náplň stretnutia

- prebratie servera
- uskladnenie servera
- prezentácia používateľského rozhrania

Ciele do najbližšieho stretnutia

- Martin Pirháč – vyprodukovať posudok na posudok
- Roman Panenka – vytvorenie registrácie používateľov
- Martin Svetlík – reakcia na posudok
- Hana Severínová - vytvorenie ukázkového skriptu
- Dávid Oros – inštalácia serveru a inštalácia príručka celého systému
- Martin Pirháč, Martin Nagy - dokumentácia
- prezentácia oponentskému tímu 13.12. pri odovzdávaní – dohodnúť s Grellnethom
- dohodnúť, či bude prezentácia na notebooku alebo projektore

Zápisnica zo stretnutia Tímu 3

Stretnutie č.9

Dátum: 28.2.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlák

Náplň stretnutia

- Voľba parametrov, ktoré sa budú vyhodnocovať:
 - priepustnosť
 - packetloss
- parametre sa budú vyhodnocovať napríklad vzhľadom na smerovací protokol, mobility model,...

Ciele do najbližšieho stretnutia

- Základné scenáre vstupov – Martin Pirháč, Martin Svetlák, Hana Severínová = motivácia
- Rozdiely medzi výstupnými súbormi drôtových a bezdrôtových simulácií, zistiť čo všetkodokáže TCP Evaluation Tool a iNSpect – Martin Nagy
- Aktualizovať webstránku – Roman Panenka
- Nainštalovať ns-2 na server – Dávid Oros
- Podrobnejší opis funkcionality riešenia

Zápisnica zo stretnutia Tímu 3

Stretnutie č.10

Dátum: 14.3.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlák

Náplň stretnutia

- Analýza využiteľnosti iNSpectu ako zdroja animácií a vizualizácií topológií
- Konzultácia problémov vzniknutých pri vytváraní ukážkových tcl skriptov
- Rozprava o vypracovanom dokumente motivácia

Ciele do najbližšieho stretnutia

- Do dokumentu popísať jednotlivé parametre trace súboru
- Vyskúšať skripty na serveri
- Otestovať prepojenie webstránky so simulátorom
- Vytvoriť/upraviť existujúce prostriedky na vytváranie štatistík z výstupných súborov
- Zistiť aké existujú ďalšie prostriedky na spracovávanie výstupov simulátora
- Porovnať výstupné súbory pre rôzne smerovacie protokoly

Zápisnica zo stretnutia Tímu 3

Stretnutie č.11

Dátum: 21.3.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirhác, Bc. Martin Svetlák

Náplň stretnutia

- Diskusia o príspevku na IITSrc
- Návrh tímových tričiek k príležitosti prezentácie projektu
- Diskusia o problémoch s ns2 pri simulácii rôznych bezdrôtových sietí
- Diskusia o problémoch s nástrojmi pre ns2

Ciele do najbližšieho stretnutia

- Úprava príspevku na IITSrc –Martin Svetlák, Hana Severínová
- Návrh posteru pre IITSrc – Hana Severínová, Martin Svetlák
- Pokračovanie v dlhodobých úlohách ohľadne systému a nástrojov pre ns2-
MartinNagy, Dávid Oros, Roman Panenka, Martin Pirhác

Zápisnica zo stretnutia Tímu 3

Stretnutie č.12

Dátum: 28.3.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlák

Náplň stretnutia

- Diskusia o upravenom príspevku na IITSrc
- Diskusia o problémoch s iNSpectom

Ciele do najbližšieho stretnutia

- Práca na pridelných úlohách
- Práca na dokumentácii
- Práca na webstránke

Zápisnica zo stretnutia Tímu 3

Stretnutie č.13

Dátum: 4.4.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Hana Severínová

Náplň stretnutia

- Rozprava o doteraz dosiahnutom progrese
- Analýza zostávajúcich úloh
- Pridelenie prác na ďalšie obdobie

Ciele do najbližšieho stretnutia

- Martin Nagy a Martin Pirháč - analýza trace súborov v zmysle návrhu vyhodnocovacích skriptov
- Hana Severínová a Dávid Oros - realizácia vyhodnocovacích skriptov
- Roman Panenka - kompletizácia dát na webstránke, zdokonalenie funkcionalitywebstránky
- Martin Svetlák - návrh a realizácia zakomponovania grafických programov do riešenia,
- GnuPlot, iNSpect

Zápisnica zo stretnutia Tímu 3

Stretnutie č.14

Dátum: 11.4.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová

Náplň stretnutia

- Rozprava o nájdených awk skriptoch
- Analýza zistení o štruktúre trace fileov
- Prezentácia webstránky

Ciele do najbližšieho stretnutia

- Martin Pirháč - ako spojzdníť smerovanie tora na ns2 simulátore
- Martin Nagy - ísť nakupovať do Auparku
- Dávid Oros - rozbehať toru na serveri
- Hana Severínová, Dávid Oros - verifikácia a modifikácia existujúcich awk skriptov
- Martin Pirháč - verifikácia skriptov a jej zdokumentovanie
- Martin Svetlík - verifikácia iNSpectu
- Zaviesť pojem priepustnosť uzla

Zápisnica zo stretnutia Tímu 3

Stretnutie č.15

Dátum: 18.4.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlík

Náplň stretnutia

- Analýza výstupných tr súborov
- Návrh prepojenia grafických výstupov so skriptami generujúcimi vlastnosti simulácie

Ciele do najbližšieho stretnutia

- Dokončenie aplikačnej časti
- Hana Severínová, Dávid Oros - implementácia skriptov
- Hana Severínová - návrh a realizácia posteru
- Martin Pirháč, Martin Nagy - dohotovenie analýzy tr súborov
- Martin Svetlík - analýza a návrh výstupných grafov
- Roman Panenka - objednanie tlače posteru

Zápisnica zo stretnutia Tímu 3

Stretnutie č.16

Dátum: 29.4.2011
Čas: 12:00
Miesto: Internát Mladosť
Účast': Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka, Bc. Hana Severínová,
Bc. Martin Svetlák

Náplň stretnutia

- Zjednotenie skriptov
- Prepojenie web-stránky so skriptami a simulátorom
- Opravovanie chýb na stránke
- Inštalácia iNSpectu-u na server a jeho prepojenie so skriptami
- Analýza generátora skriptov

Ciele do najbližšieho stretnutia

- Analýza generátora skriptov
- Opravenie chýb vo webovom rozhraní
- Riadiaca dokumentácia
- Používateľské príručky
- Príprava na IIT.SRC

Zápisnica zo stretnutia Tímu 3

Stretnutie č.17

Dátum: 2.5.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlák

Náplň stretnutia

- Analýza a odľad'ovanie naimplementovaného riešenia
- Rozbehanie iNSpect-u na serveri
- Diskusia o dokumentácii

Ciele do najbližšieho stretnutia

- Peter Magula - poslat zdrojový kód DSDV a TCL skript kde funguje mobilita
- Informovať používateľa čo všetko má splniť pri nahrávaní TCL skriptu
- V skriptoch doplniť ako prvý bod bod (x,0)
- Doplniť adaptívnu granularitu grafov (AGG) v skriptoch - dĺžka simulácie/200
- Pripraviť e-mailovú notifikáciu a počkať na informácie ohľadne nej
- Doriešiť znázorňovanie mobility
- Dávid Oros, Hana Severínová - autogenerátor
- Dorobiť dokumentáciu
- Roman Panenka - dorobiť Captchu

Zápisnica zo stretnutia Tímu 3

Stretnutie č.18

Dátum: 9.5.2011
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová

Náplň stretnutia

- Zhodnotenie problémov pri riešení
- Návrh mechnizmov dorábania dokumentácie
- Riešenie problémov s vizualizáciou simulácie a grafmi

Ciele do najbližšieho stretnutia

- Preverí sa a následne vykoná pridanie okienka a tlačidla, ktore by prekreslili graf z granularitou na osi x podľa vôle používateľa
- Martin Pirháč - extrahovanie teoretického textu z dokumentu Ako napísať skripty
- Dokončenie používateľskej príručky
- Dokončenie riadiacej dokumentácie
- Preveriť akým spôsobom sa dá upraviť granularita tak, aby boli v prípade krátkej simulácie grafy postačujúce

Manažment verzií, konfigurácií a zmien

V tejto časti dokumentácie sa budeme venovať manažmentu projektu. Obsahuje informácie o použitých prostriedkoch, pomocou ktorých sa dosahuje manažment verzií, konfigurácií a zmien.

Dropbox

Dropbox je systém na zdieľanie súborov medzi viacerými používateľmi, založený na synchronizácii súborov. Priečink, ktorý pomocou systému Dropbox spolu zdieľajú viacerí používatelia, má u každého z nich rovnaký obsah.

Systém Dropbox používame pri vypracovávaní tohto projektu na zdieľanie dokumentov, programov a iných typov súborov potrebných na vypracovanie projektu.

Google Groups

Pri práci na tomto projekte prebieha komunikácia cez emailovú komunikáciu pomocou Google Groups. Poslaním e-mailu na adresu skupiny vytvorenej na stránke groups.google.com sa táto pošle všetkým členom tejto skupiny.

Microsoft Word 2007

Písanie dokumentácie projektu prebieha pomocou Microsoft Word 2007. Tento kancelársky program poskytuje možnosť komentovať úseky textu, pričom sa zaznamená autor komentáru a dátum. Tento program poskytuje taktiež možnosť vidieť zmeny v dokumente, pričom sa dá vidieť typ zmeny, aj meno používateľa, ktorý zmenu vykonal.

Obe tieto funkcie sú veľmi vhodné pri kolektívnom vypracovávaní dokumentu, kvôli ich verziovaciemu a kolaboratívne charakteru.

Posudok projektovej dokumentácie k projektu tímu 8

1 Úvod

Tento dokument predstavuje posudok k projektovej dokumentácii projektu *Simulátor komunikácie v počítačovej sieti* od kolektívu autorov z tímu č. 8 v rámci predmetu *Tímový projekt I.* pod vedením Ing. I. Grellnetha, PhD. Posudok je členený na dve časti, v prvej je projekt posúdený po vecnej stránke, v druhej časti po formálnej.

2 Posudok z vecného hľadiska

Predložená dokumentácia k projektu projekt je obsahovo členená na 6 kapitol prvej úrovne (0-5), kapitola 0 je úvodom, kapitola 4 je záverom, kapitola 5 obsahuje zoznam zdrojov, ostatné kapitoly sa zaoberajú analýzou, špecifikáciou a návrhom. Neštandardné je poradie kapitol, keď kapitola *Špecifikácia* je pred kapitolou *Analýza*. Nakoľko sa jedná o pokračovanie projektov z predošlých rokov, bolo by viac než vhodné najprv analyzovať stav, v akom sa projekt nachádza, až potom špecifikovať, čím do projektu prispeje tento tím.

V kapitole 2.1 (*Analýza aktuálneho stavu projektu VLAB*) je opísaný len pokus o inštaláciu VLABu. Chýba analýza funkcií, ktoré do systému implementovali tímy v predchádzajúcich rokoch, popis vstupov a výstupov systému, používateľského rozhrania a analýza dátových štruktúr použitých v týchto projektoch (v texte sa viackrát spomína databáza – čo obsahuje táto databáza?).

V práci chýba analýza nových možností, ktoré sú uvedené v špecifikácii a návrhu, napríklad protokolu IPv6, chat skriptov (spomenutých v kap. 3.5).

V kapitole 2.6 (*Využitie RAM IOSmi*) je spomenuté dopočítanie *idlePC* hodnôt pre niektoré IOSy. Ako sa tieto hodnoty počítajú?

Časti textu v kapitole 3 (*Návrh*) sú vhodné skôr do analýzy, resp. dokumentácie k riadeniu projektu. Podpora IPv6, ktorá sa javí ako jedna z nosných častí projektu, je aj v tejto kapitole opísaná len pár riadkami. Na druhej strane návrh modulu monitorujúceho vyťaženie CPU a RAM je rozpracovaný na dosť podrobnej úrovni.

Kapitola 3.4 má rovnaký text ako kapitola 2.1.

Viackrát sú v posudzovanej práci spomenuté konkrétne implementačné kroky, ktoré tím už podnikol v rámci implementácie (napr. kap. 2.4.2, kde sa spomína vytvorenie *konfigurácie UML jadra*), tieto kroky však nie sú ďalej zdokumentované (t.j. chýba napr. popis uvedenej konfigurácie). Tieto kroky odporúčame v ďalšej fáze projektu zdokumentovať.

3 Posudok z formálneho hľadiska

V texte je často vynechaná diakritika a často sa v ňom vyskytujú gramatické chyby a preklepy – „predpripravené“ (kap. 1.4), „niesú“ (kap. 2.1), „relatívne“ (kap. 2.3.3), „nadvezujú“ (kap. 3), „inhalačných príručiek“ (kap. 3.4) a iné. Slovo „lab“, ktoré je v práci mnohokrát použité, v slovenčine neexistuje a nebolo vysvetlené v kap. 0.3. Skratka UML bola definovaná ako „Unified Modeling Language“, no je používané vo význame „User-Mode Linux jadro“. Na popis niektorých činností boli použité slangové slová (napr. kap. 1.2.2, kde sa o inštalícii hovorí slovami „rozbiehať“, „nahadzovať systém“, kap. 2.6.1: „zabiť programy“). Predmet „ps2“ (str. 2) neexistuje, jedná sa zrejme o predmet PSIP.

V obsahu je chybný odkaz na kapitolu 4. Skratka IPv6 je v zozname skratiek dvakrát.

Zoznam použitých zdrojov nie je písaný v jednotnom typografickom formáte, a zdroje nachádzajúce sa na Internete nie sú uvedené vo formáte ISO STN 690. V práci je značná časť informácií prebraná z iných zdrojov, no nie sú uvedené odkazy na tieto zdroje. Napríklad sa jedná o kap. 1.3.1, 2.3.2. Zdroj www.zive.sk, citovaný pod číslom [1], nie je seriózny zdroj, odporúčame informáciu overiť.

4 Posudok dokumentácie k riadeniu projektu

Dokumentácia k riadeniu projektu obsahuje všetky potrebné náležitosti. Chýba popis tabuľky na 13. strane tejto dokumentácie.

5 Zhodnotenie

Z projektu je vidieť odvedenú prácu študentov, no nie všetky kroky sú dostatočne zdokumentované a predložená dokumentácia obsahuje viaceré chyby po formálnej aj obsahovej stránke. Projekt hodnotíme stupňom dobrý.

Vyjadrenie sa k posudku

Po dôkladnom prečítaní a analyzovaní jednotlivých pripomienok, ktoré nám boli vytýkané v posudku k práci, ktorý vyhotovil tím č. 8 sme sa rozhodli na zapracovaní nasledovných zmien.

Kapitola „Úvod“

Do úvodnej časti dokumentu sme zapracovali zoznam skratiek, ktorý obsahuje zoznam všetkých skratiek použitých v práci. Skratky boli pôvodne vysvetlené priamo v texte, čo nemusí byť v niektorých prípadoch veľmi praktické, keďže nie všetci čitatelia budú čítať celý dokument. Zoznam skratiek umiestnený v úvodnej kapitole dokumentu je tiež vhodný v prípade, že čitateľ zabudne význam niektorej skratky. V tomto prípade čitateľ nemusí hľadať vysvetlenie v predchádzajúcom texte, ale nájde ho v spomínanom zozname. Táto výhrada k práci bola oprávnená a jej doplnenie bude mať prínos k čitateľnosti a ľahšiemu pochopeniu práce.

Kapitola „Simulácia v počítačových sieťach“

Tím 8 navrhol logické rozdelenie tejto časti na dve podkapitoly :

- Simulácie
- Simulácie počítačových sietí

Takéto rozdelenie by určite nebolo na škodu, avšak vzhľadom na rozsah tejto kapitoly ho nepovažujeme za nutné. Táto kapitola je totiž zamýšľaná len ako úvod do problematiky simulácií, pričom jej účelom bolo poukázať na výhody simulácie a dôvody, prečo je takýto prístup vhodný práve v oblasti počítačových sietí. Keďže táto kapitola nepatrí medzi ťažiskové rozhodli sme sa ďalej ju nečleniť na podkapitoly, aby jej čitateľ neprikladal zbytočne veľkú dôležitosť.

Kapitola „Simulátory počítačových sietí“

Pri analýze simulátora ns-2 nám bola vyčítaná horšia čitateľnosť a pochopiteľnosť vstupného tcl skriptu a výstupného súboru (trace file). Táto kapitola v žiadnom prípade neslúži ako manuál k simulátoru ns-2. Cieľom bolo hlavne analyzovať možnosti simulátora v oblasti simulácie bezdrôtových sietí a priblížiť formát jeho štandardného vstupu a výstupu. Konfigurácia simulátora prostredníctvom skriptu bola uvedená len ako názorná ukážka fungovania simulátora, keďže mnohí čitatelia majú značne skreslenú predstavu o ovládaní a fungovaní tohto simulátora. Formát výstupného súboru bol uvedený len na ilustráciu a demonštráciu širokých možností simulátora v tejto oblasti. Jeho konkrétna podoba závisí od konkrétnej simulácie, preto nebol uvedený vzorový výstup, aby sme čitateľa nezavádzali.

Podrobnejšia analýza výstupného súboru bude predmetom ďalšej časti práce, v ktorej bude tento výstup spracovaný do používateľsky čitateľnejšej podoby (grafy, tabuľky, animácie), čo je nakoniec aj cieľom našej práce. V tejto časti bude tiež spresnený výber grafického vizualizéra na základe ich dôkladnejšieho otestovania. Výber vizualizéra nebol súčasťou analýzy aj z dôvodu, že ešte nebol jasný výber simulátora na ktorý sa bude naša práca orientovať.

Po formálnej stránke nám bola vytknutá len nejednoznačnosť použitia skratiek ns2, ns-2 a NS2. Oficiálne zaužívanou skratkou pre simulátor je ns-2, ostatné variácie boli spravidla použité len ako súčasť názvu niektorého z podporných nástrojov. Túto nejednoznačnosť pre istotu objasníme v zozname skratiek v úvodnej kapitole.

K analýze simulátora ns-3 boli ako nedostatky uvedené absencia obrázku výstupu .pcap v prostredí Wireshark a tabuľky, v ktorej sú zosumarizované podporované protokoly v jadre ns-3 podobne ako to bolo v kapitole o ns-2. Tiež boli odhalené protokoly podporované v ns-3, ktoré však chýbajú na Obrázku 17.

Tabuľku do kapitoly zaoberajúcu sa analýzou simulátora ns-3 bude samozrejme potrebné doplniť. Takisto doplníme aj chýbajúce protokoly v Obrázku 17. Pre zaradenie obrázka z prostredia Wireshark, ktorý by demonštroval formát výstupu typu .pcap však nevidíme objektívny dôvod.

Kapitola „Špecifikácia a hrubý návrh riešenia“

V tejto časti nám neboli vytknuté žiadne vážnejšie nedostatky s výnimkou chýbajúceho návrhu výberu konkrétnych komponentov. Ako sme vyššie spomenuli, konkrétny výber vizualizéra a iných komponentov bude bližšie opísaný ďalšej časti práce, v ktorej bude prehĺbený súčasný hrubý návrh.

Dokumentácia k riadeniu projektu

V tejto časti práce bolo poukázané na absenciu rozdelenia autorov jednotlivých častí dokumentácie. Ako bolo správne poznamenané, úlohy jednotlivých členov tímu sú jasne dané a dokumentované v zápisniciach zo stretnutí.

Dokumentácia k riadeniu projektu tvorí samostatnú časť práce v podobe prílohy. Nie je nám známe ako lepšie by sme mohli túto dokumentáciu oddeliť od zvyšku práce, keďže práca má byť zviazaná ako jedna kniha.

Zhodnotenie

Okrem vyššie uvedených zmien, ktoré nám boli vytknuté tímom č. 8, boli v práci aktualizované nefunkčné odkazy a opravené drobné gramatické a štylistické chyby. Celkovo hodnotíme posudok od tímu č. 8 pozitívne. Boli v ňom obsiahnuté viaceré cenné rady a odporúčania ako aj odhalené niektoré závažnejšie nedostatky, ktoré bolo potrebné do práce dodatočne zapracovať. Niektoré výhrady však boli podľa nášho názoru irelevantné a po ich zvážení sme sa rozhodli, že nebudú do dokumentu zapracované.

Posudok prototypu tímu 8

Predložená dokumentácia je logicky členená do viacerých kapitol, obsahuje 6 strán textu a priložené elektronické médium.

Médium obsahuje testovacie skripty, vstupy a výstupy testovania. Keďže nám bola odovzdaná iba časť dokumentácie k prototypu, očakávali by sme prítomnosť aspoň elektronickej kópie celkového dokumentu na médiu. Dokumentácia k prototypu neobsahuje slovník a sú v nej použité pojmy a skratky, ktoré boli vysvetlené v prvotnej dokumentácii, teda na zlepšenie prehľadu by bolo naozaj vhodnejšie odovzdať dokumentáciu celú alebo ju pripojiť na elektronickom médiu.

Práca začína úvodom do témy testovania žiakov a vysvetľuje motiváciu pre tvorbu takéhoto systému. Je nutné oceniť dva prístupy k testovaniu nakonfigurovaných topológií – funkčný a parsovací. Parsovací prístup je vhodný pre kontrolu funkcionalít, ktoré majú byť presne prítomné v konfiguračnom súbore alebo v kontrolnom výpise smerovača. Funkčné slúži na kontrolu napríklad konektivity medzi dvoma podsieťami (nezáleží na tom, či je konektivita vytvorená pomocou statického alebo dynamického smerovania). Funkcia prototypu je vysvetlená na jednoduchom príklade topológie

Systém po skontrolovaní topológie vypíše bodové hodnotenie za dané úlohy. V budúcnosti by sme radi uvítali aj zobrazovanie dôvodu, prečo boli body strhnuté poprípade pridelené. Podľa nášho názoru, toto prinesie väčšiu prehľadnosť do systému hodnotenia a taktiež tu bude možné oddeliť náhodné preklepy (chybné jedno číslo v IP adrese) od faktických chýb (žiak naozaj nevedel ako niečo nakonfigurovať). Celkovo by táto funkcionalita priniesla väčší prehľad do bodového hodnotenia.

Prototyp nám bol riadne odprezentovaný, prínos študentov tímu 8 je v pridaní testovacieho modulu. Taktiež netreba zabudnúť na problémy, ktoré mali s uvedením VLABu do opätovného chodu.

V dokumentácii aj v prezentácii prototypu chýbal návrh systému a opis hlavných funkčných blokov pomocou štandardných nástrojov (napr. UML).

Nakoniec by sme radi vyzdvihli možnosť práce viacerých študentov na jednom zadaní. Kladne hodnotíme tiež prítomnosť vývojových diagramov pre funkčné testovanie a bodové hodnotenie podľa štandardu UML. Chybičkou krásy v dokumentácii sú drobné gramatické a formálne chyby, nie je ich však príliš veľa a nepôsobia rušivo. Taktiež význam obsahu bez uvedených čísel na samotných stranách je otázný.

Celkovo hodnotíme prototyp známku “viac ako dobrý“.

Príloha B
Inšalačná príručka

1 Inštalačná príručka

Pre simulátor ns-2 sme zvolili prostredie operačného systému Linux, konkrétne distribúciu Ubuntu. V nasledujúcich kapitolách a podkapitolách si uvedieme nastavenie, ktoré je potrebné vykonať pre beh NS2IT na serveri.

1.1 Operačný systém

Operačný systém nainštalujeme aj s grafickým rozhraním, ktoré je potrebné kvôli programu Inspect. Následne je potrebné nainštalovať nasledujúce balíky a závislosti (ďalšie balíky, ktoré sú potrebné – závislosti rieši inštalátor). Použijeme nasledovný príkaz:

- `apt-get install build-essential autoconf automake libxmu-dev g++-4.3 apache2 php5 mysql-serverGtkGLextfre GLUT3 freeglut3-dbg freeglut3-devxhostx11-xserver-utilsgtkpodxinitxauthpostfixmencoder xdm mailx xvfb`
- **stiahneme NS2:**
`wget http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.34/ns-allinone-2.34.tar.gz`
- **rozbalíme:** `tar -zxvf ns-allinone-2.34.tar.gz`
- **zmeníme v súbore** `'/ns-allinone-2.34/otcl-1.13/Makefile.in'` **premennú** `CC` z `@CC@` na `gcc-4.3` (`CC= gcc-4.3`)
- `cd ns-allinone-2.34` a **spustíme inštaláciu** príkazom `./install`
- **po úspešnej inštalácii** musíme nastaviť správne cesty pre NS2, dosiahneme to príkazom:
`export PATH=$PATH+/root/ns-allinone-2.34/bin:/root/ns-allinone-2.34/tcl8.4.18/unix:/root/ns-allinone-2.34/tk8.4.18/unix`
- **presunieme sa do priečinku ns-2.34** pomocou príkazu `cd ns-2.34` a **spustíme validačné** testy príkazom: `./validate`

Ak prebehlo všetko vporiadku, uvidíme na poslednom riadku správu: „`validate overall report: all tests passed`“

Pre úplnosť uvádzame aj konfiguračný súbor pre inštalačný manažér balíkov APT

„`/etc/apt/sources.list`“:

```
root@team3:~# cat /etc/apt/sources.list | grep -v ^# | grep -v "^$"
deb http://sk.archive.ubuntu.com/ubuntu/ maverick main restricted
deb-src http://sk.archive.ubuntu.com/ubuntu/ maverick main restricted
deb http://sk.archive.ubuntu.com/ubuntu/ maverick-updates main restricted
deb-src http://sk.archive.ubuntu.com/ubuntu/ maverick-updates main restricted
deb http://sk.archive.ubuntu.com/ubuntu/ maverick universe
deb-src http://sk.archive.ubuntu.com/ubuntu/ maverick universe
deb http://sk.archive.ubuntu.com/ubuntu/ maverick-updates universe
deb-src http://sk.archive.ubuntu.com/ubuntu/ maverick-updates universe
deb http://sk.archive.ubuntu.com/ubuntu/ maverick multiverse
deb-src http://sk.archive.ubuntu.com/ubuntu/ maverick multiverse
deb http://sk.archive.ubuntu.com/ubuntu/ maverick-updates multiverse
deb-src http://sk.archive.ubuntu.com/ubuntu/ maverick-updates multiverse
deb http://security.ubuntu.com/ubuntu maverick-security main restricted
deb-src http://security.ubuntu.com/ubuntu maverick-security main restricted
deb http://security.ubuntu.com/ubuntu maverick-security universe
deb-src http://security.ubuntu.com/ubuntu maverick-security universe
deb http://security.ubuntu.com/ubuntu maverick-security multiverse
deb-src http://security.ubuntu.com/ubuntu maverick-security multiverse
root@team3:~#
```

1.2 Inštalácia a konfigurácia LAMP

Skratkou LAMP označujeme Linux Apache Mysql Php prepojenie komponentov. Slúži na prevádzkovanie webovej stránky naprogramovanej v jazyku PHP a prácou s dátami uloženými v MySQL databáze.

Inštaláciu potrebných balíkov sme vyriešili v podkapitole 1.1 – príprava operačného systému a potrebných balíkov. Boli to balíky apache2, php5 a mysql-server. Balíkov je potrebných viac, avšak inštalátor tieto závislosti rieši automaticky a všetko potrebné nainštaluje. Ak budú potrebné špecifické balíčky, budú v tejto príručke uvedené v neskoršej kapitole, resp. podkapitole.

1.2.1 APACHE2

Webový server Apache2 nie je potrebné špeciálne konfigurovať. Je potrebné iba vytvoriť VirtualHost pre potreby danej webovej stránky. To znamená, aby DocumentRoot v danom VirtualHoste ukazoval na miesto kde sú uložené na serveri zdrojové kódy web stránky. Príklad VirtualHostu je uvedený v prílohe.

1.2.2 MySQL

Databázový server tak isto postačuje v základnej konfigurácii. Je však potrebné vytvoriť databázu, s ktorou web stránka spolupracuje pri práci s dátami. V prvom je potrebné sa prihlásiť na príkazový riadok pre MySQL príkazom: `mysql -u root -p` a po vyzvaní zadať heslo pre používateľa „root“, ktoré sa zvolilo pri inštalácii MySQL serveru.

Následne je potrebné vytvoriť databázu pre web stránku. V MySQL príkazovom riadku zadáme príkaz: `create database <názov databázy>;` To nám vytvorí prázdnu databázu. Aby s touto databázou mohla pracovať web stránka, musíme vytvoriť používateľa, ktorý bude mať plné práva pre prácu s danou databázou. V príkazovom riadku MySQL zadáme príkaz: `grant all privileges on <názov databázy>.* to '<login>'@'localhost' identified by '<heslo>';` následne musíme spustiť príkazom `flush privileges;` znovu načítanie všetkých práv, aby boli tieto nami novo pridané hneď aktuálne.

Na vytvorenie štruktúry databázy použijeme SQL skript napísaný v súbore „database.sql“, ktorý je súčasťou prílohy. Skript použijeme nasledovným príkazom: `mysql -u login -p názov databázy < database.sql`

Po vyzvaní zadáme heslo a skript vytvorí potrebné tabuľky v databáze pre našu web stránku.

1.2.3 PHP

Inštaláciu PHP balíkov a potrebných súčastí zabezpečuje inštalátor. Náš projekt využíva PHP funkciu „exec“, ktorá vykonáva príkazy zadané na web stránke na serveri. Preto je potrebné vypnúť SAFE MODE v konfigurácii PHP. To dosiahneme modifikáciou konfiguračného súboru pomocou editora. Konfiguračný súbor sa nachádza na serveri `/etc/php5/apache2/php.ini` – V tomto súbore lokalizujeme riadok `safe_mode`

= on a zmeníme on na off. Následne je potrebné reštartovať apache2, aby si načítal aktuálnu konfiguráciu PHP. To dosiahneme príkazom: `/etc/init.d/apache2 restart`

1.3 Záplata ns-2.34 pre TORA

Pre prácu so smerovacím protokolom TORA potrebujeme upraviť ns-2 nasledovným spôsobom. Všetky úpravy je potrebné robiť pod používateľom „root“.

- `su -` zmeníme aktuálneho používateľa na „root“
`cd ../ns-2.34/tora/-` vojdem do priečinka s NS2
`mv tora.h tora.old.h mv tora.cc tora.old.cc` – odzaložujeme aktuálne súbory
- `wget http://wpage.unina.it/marcello.caleffi/ns2/tora.cc`
`wget http://wpage.unina.it/marcello.caleffi/ns2/tora.h`
vyššie uvedenými príkazmi stiahneme nové súbory
- `vim ../ns-2.34/imep/imep.cc`– editorom upravíme súbor imep.cc nasledovne:

riadok `rexmitTimer.start(rexat - CURRENT_TIME);` zmeníme na:

```
if (rexat-CURRENT_TIME<0.000001)
    rexmitTimer.start(0.000001);
else
    rexmitTimer.start(rexat - CURRENT_TIME);
```

- `cd ../ns-2.34/` a spustíme `make` a následne `make install`– týmito príkazmi znova prekompilujeme NS2, aby zmeny boli účinné
- v tom istom priečinku spustíme `./validate`– uistíme sa, že vykonané zmeny neobmedzili funkčnosť NS2 simulátora

1.4 Inštalácia Inspectu

Pre inštaláciu nášho upraveného Inspectu je potrebné nahráť na server náš balík priložený k celkovému produktu NS2IT. Balík je potrebné na serveri rozbaľiť a v vojsť do novovzniknutého priečinku Inspect, kde je potrebné vykonať príkaz `./install`.

1.5 Príprava mailového serveru

Na serveri je potrebné nakonfigurovať e-mailový server, ktorý dokáže posielať správy. Nie je však potrebné konfigurovať server pre príjem pošty. Pre tento účel využijeme popredný linuxový mailserverový balík s názvom Postfix.

Program postfix má 2 konfiguračné súbory umiestnené v priečinku `/etc/postfix/` s názvami `main.cf` a `master.cf`. Konfiguračný súbor `master.cf` nie je potrebné nijako upravovať. Konfiguračný súbor `main.cf` musí mať približne nastavenia nasledovné:

```
mydomain = <názov domény na ktorej je prevádzkovaný server>
myorigin = $mydomain
```

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
```

```
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

readme_directory = no

alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases

inet_interfaces = loopback-only
local_transport = error:local
```

Veľmi dôležité sú posledné 2 riadky v konfiguračnom súbore, ktoré hovoria o tom, že e-maily budú iba posielané, ale nebudú žiadne prijaté.

Otestovať nastavenia servera môžeme príkazom `mailx -S subjekt prijemnca@domena.tld< sprava.txt`

Príloha C
Používateľská príručka

Ako bolo v dokumentácii spomínané, náš systém je webovou aplikáciou. Na prácu s ňou je potrebné použiť internetový prehliadač a zadať do poľa adresy 147.175.159.188.

2 Registrácia a prihlásenie

Po zadaní spomínanej IP adresy do prehliadača sa zobrazí hlavná stránka (obrázok 1).



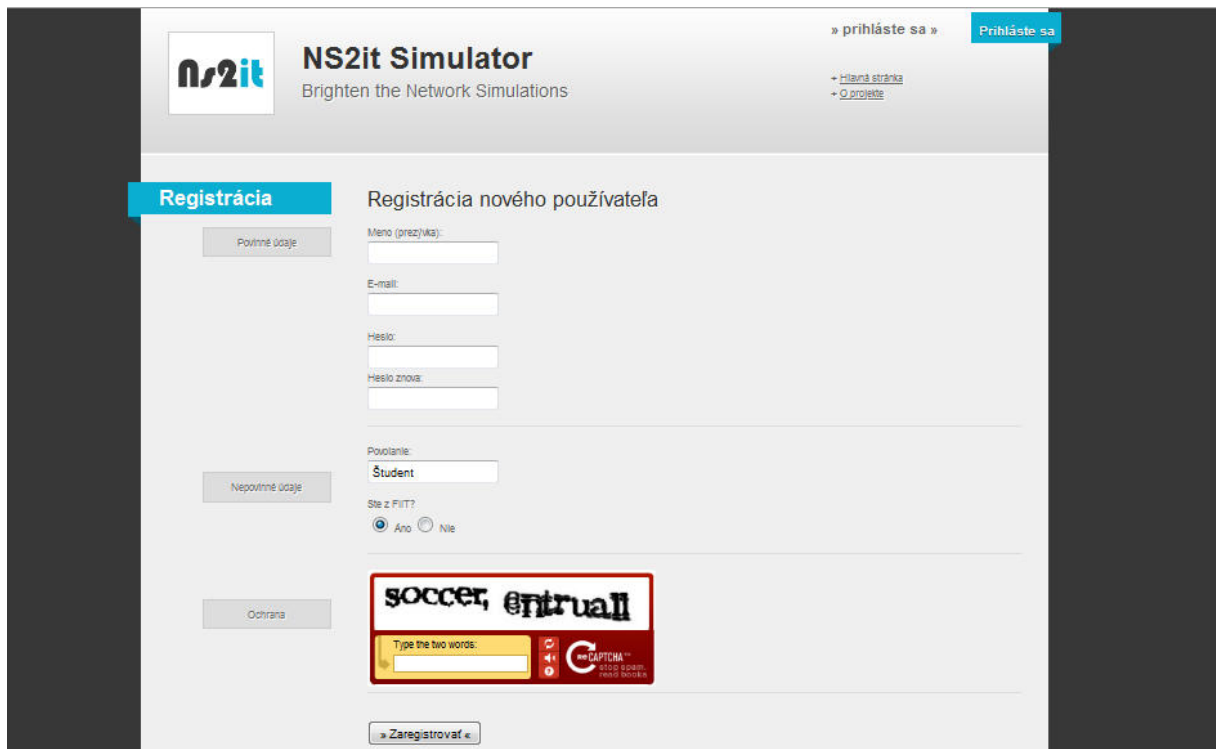
Obrázok 1: Hlavná stránka systému

Ak je používateľ už registrovaný a má prihlasovacie údaje, stačí sa prihlásiť pomocou modrého okna v pravom hornom rohu (obrázok 2). Ak používateľ nemá prihlasovacie údaje, je potrebné sa registrovať.



Obrázok 2: Prihlasovací formulár

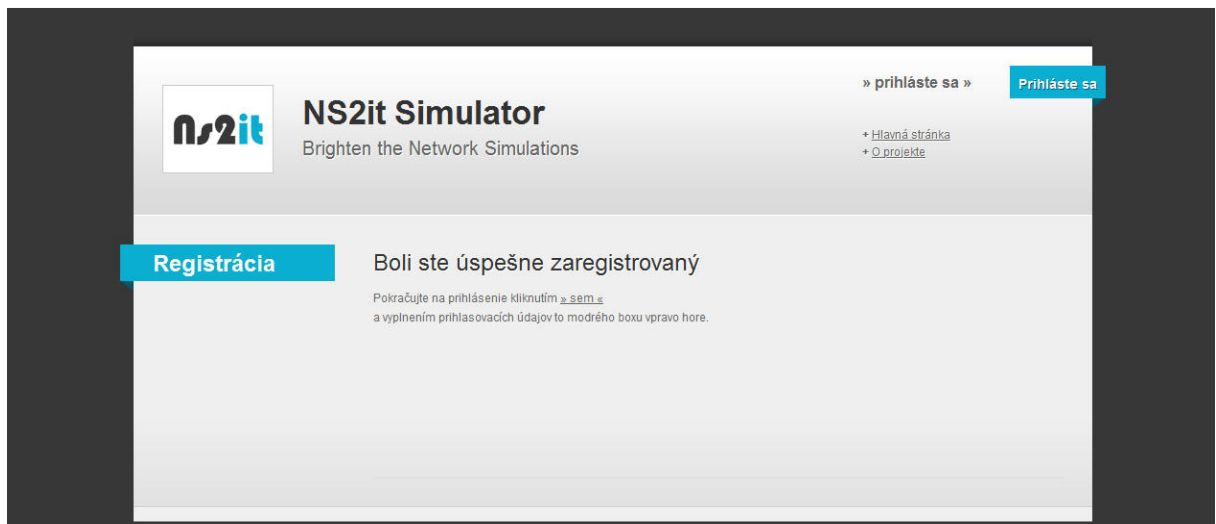
Pri registrácii je potrebné zadať meno (bude slúžiť ako prihlasovacie meno), e-mail (slúži na notifikáciu pomocou e-mailu), heslo. Nakoniec je potrebné prepísať bezpečnostný reťazec, ktorý slúži na ochranu proti internetovým robotom (obrázok 3).



The screenshot shows the registration page of the NS2it Simulator. The header includes the NS2it logo, the text "NS2it Simulator" and "Brighten the Network Simulations", and navigation links for "prihláste sa" and "Prihláste sa". Below the header, there is a "Registrácia" tab and a "Registrácia nového používateľa" section. The form is divided into three parts: "Povinné údaje" (Meno (prezývka), E-mail, Heslo, Heslo znova), "Nepovinné údaje" (Povolanie, Sta z FIIT?), and "Ochrana" (a CAPTCHA challenge with the words "soccer" and "entrual"). A "Zaregistrovať" button is at the bottom.

Obrázok 3: Registračný formulár

Po úspešnej registrácii sa zobrazí nasledovná potvrdzujúca stránka (obrázok 4). Ďalej sa pokračuje prihlásením sa do systému.



The screenshot shows the confirmation page after registration. The header is identical to the registration page. The "Registrácia" tab is active. The main content area displays "Boli ste úspešne zaregistrovaný" and instructions: "Pokračujte na prihlásenie kliknutím » [sem](#) « a vyplnením prihlasovacích údajov to modrého boxu vpravo hore." There is a large empty box for the login details.

Obrázok 4: Potvrdenie registrácie

Po úspešnom prihlásení sa používateľ nachádza na hlavnej stránke (profil používateľa). V hornej časti obrazovky sa nachádza menu a v dolnej časti obrazovky sa nachádzajú simulácie, ktoré patria prihlásenému používateľovi (*Tvoje simulácie*). Ako novo registrovaný používateľ by mal byť tento zoznam prázdny, pre úplnosť sme sem však pridali jednu testovaciu simuláciu (obrázok 5). Po kliknutí na danú simuláciu sa otvoria podrobnosti o simulácii (podrobnosti v kapitole 2).



Obrázok 5: Profil používateľa

3 Simulovanie

Na spustenie simulácie je nutné najprv napísať simulačný skript v jazyku TCL. Pre vytvorenie novej simulácie pokračujeme kliknutím na tlačidlo nová simulácia (obrázok 5). Keďže sa náš systém snaží tvorbu a vyhodnocovanie skriptov čo najviac zjednodušiť, má používateľ na výber tri možnosti tvorby skriptov (obrázok 6):

- autogenerátor
- vlastná simulácia z editora
- nahrať súbor so simuláciou.

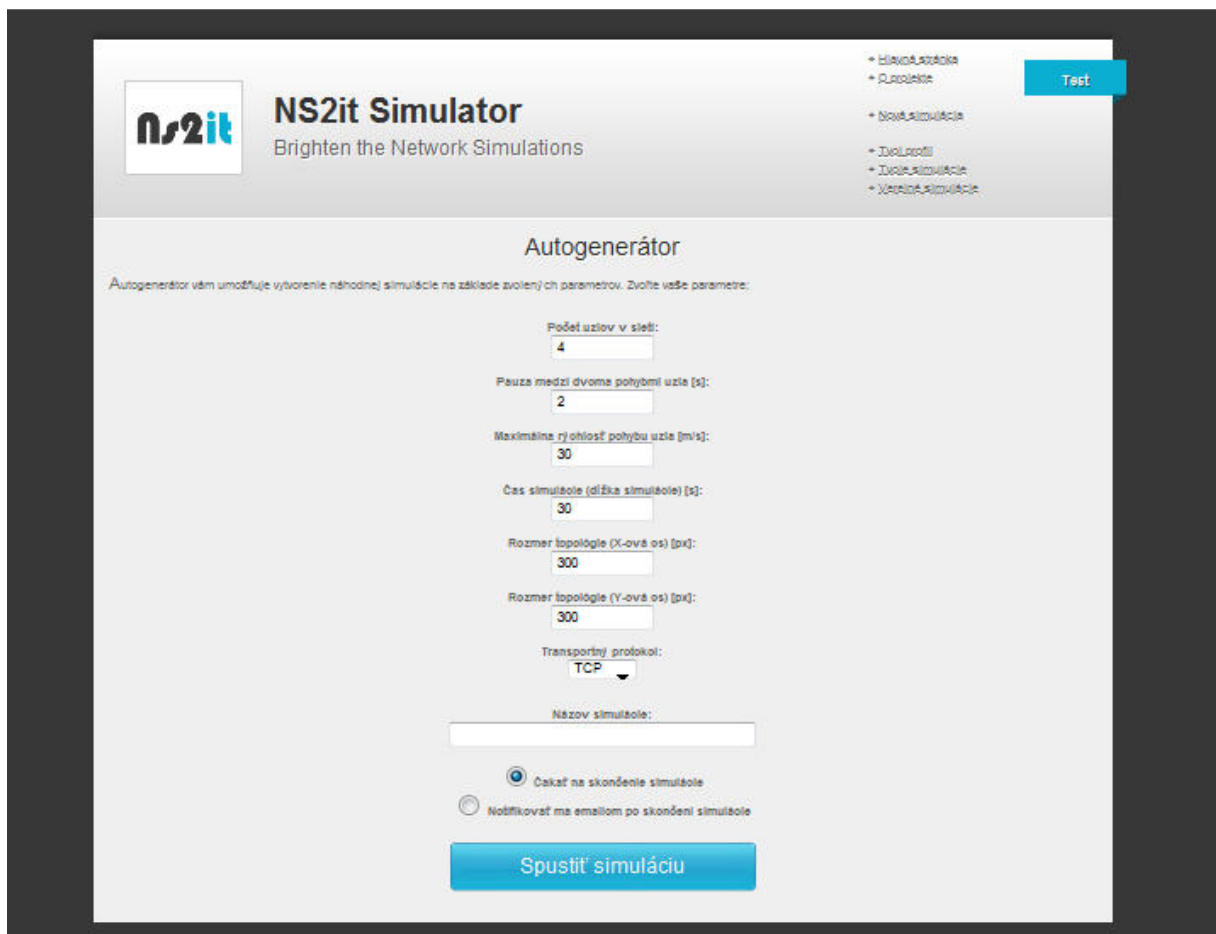
Autogenerátor je funkcionálna pre začínajúcich používateľov, ktorí práve sa práve začínajú zaoberať problematikou sietí. Systém vygeneruje simuláciu podľa zvolených kritérií:

- Počet uzlov v sieti
- Čas medzi dvoma pohybmi uzla
- Maximálna rýchlosť pohybu uzla
- Dĺžka simulácie
- Rozmer topológie (x-ová os)
- Rozmer topológie (y-ová os)
- Transportný protokol používaný uzlami



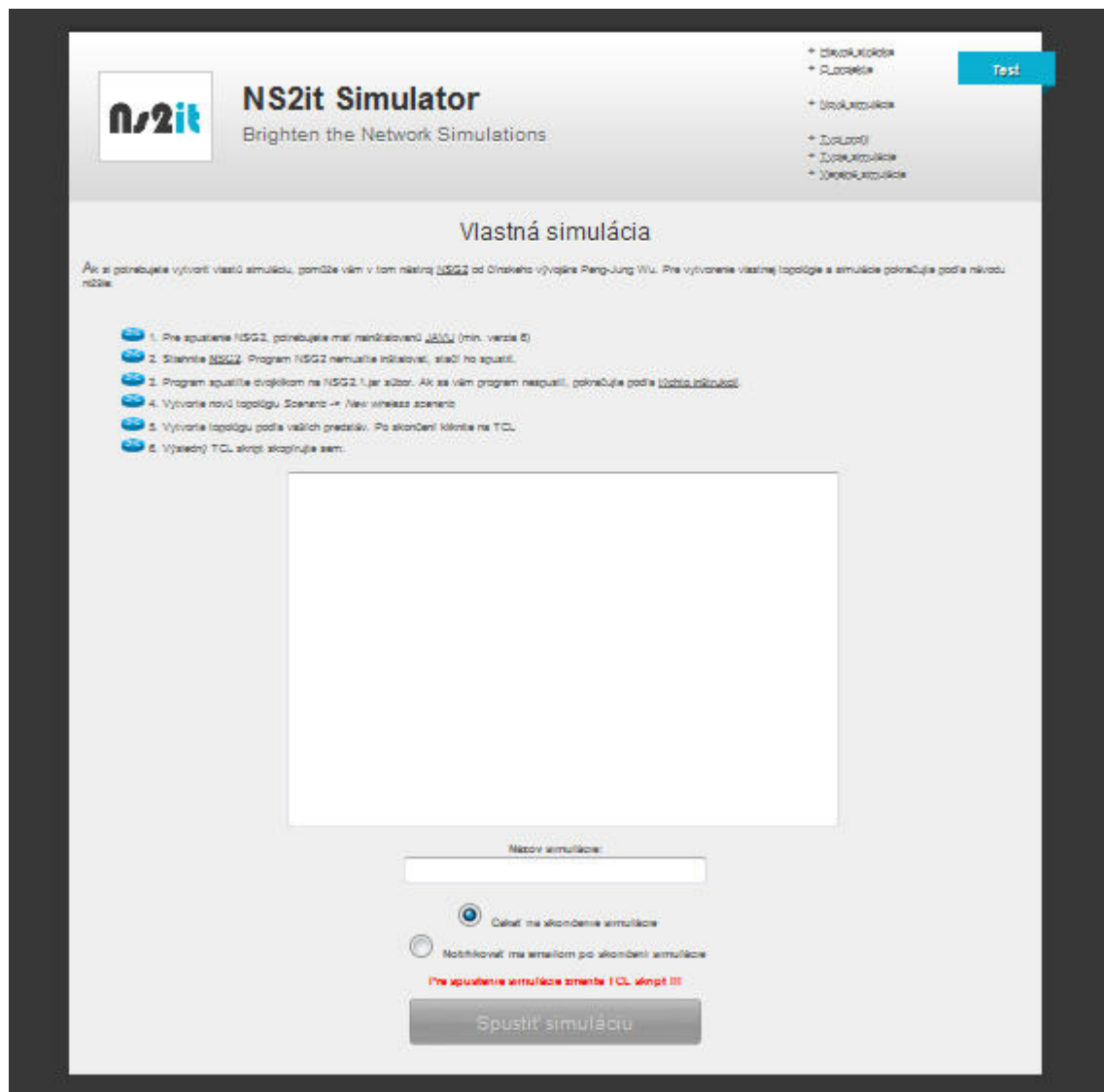
Obrázok 6: Možnosti tvorby simulácie

Používateľ sa takto jednoducho oboznámi s princípmi platnými v sieťach (obrázok 7).

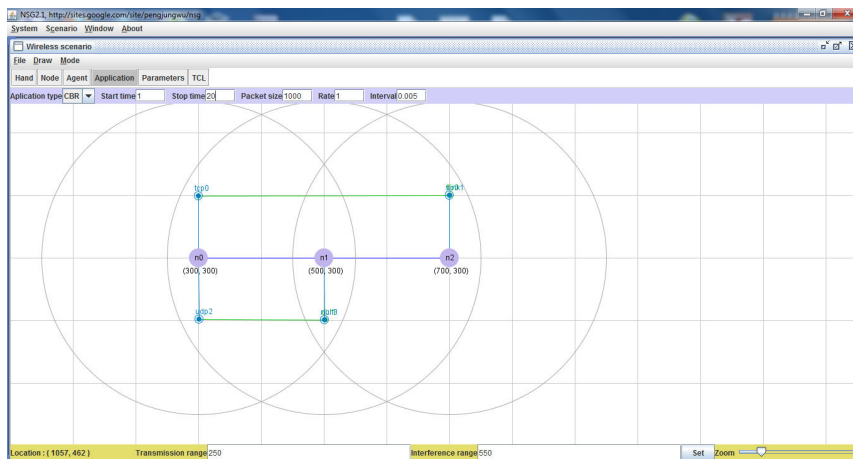


Obrázok 7: Autogenerátor

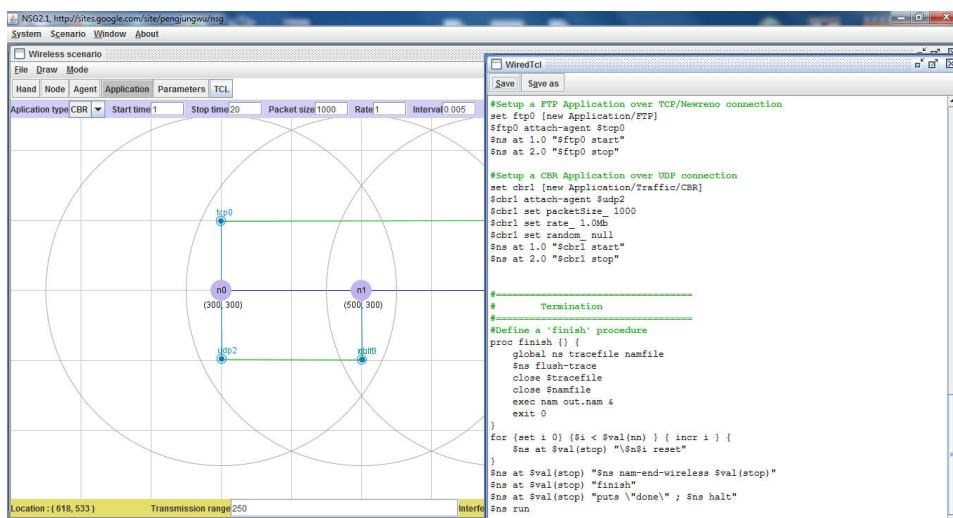
Druhou možnosťou je vytvorenie simulácie v editore (obrázok 8). Ako editor je použitý NS2 scenario generator (NSG). Tento program vyžaduje podporu jazyka Java vo verzii minimálne 6. Program sa dá stiahnuť po kliknutí na odkaz na stránke systému. Tvorba simulačného scenára je veľmi jednoduchá a intuitívna (obrázok. 9). Keď po vytvorení scenára používateľ klikne na tlačidlo TCL, kde sa mu zobrazí scenár simulácie v jazyku TCL (obrázok 10).



Obrázok 8: Tvorba simulácie v editore

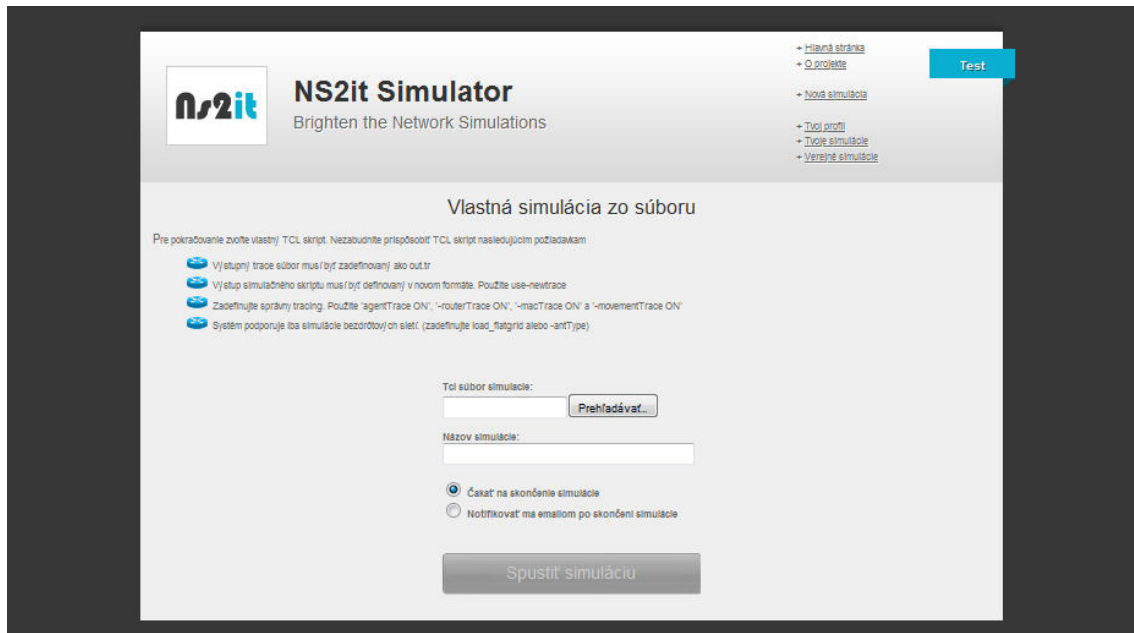


Obrázok 9: Tvorenie scenára v NSG



Obrázok 10: TCL skript vygenerovaný NSG

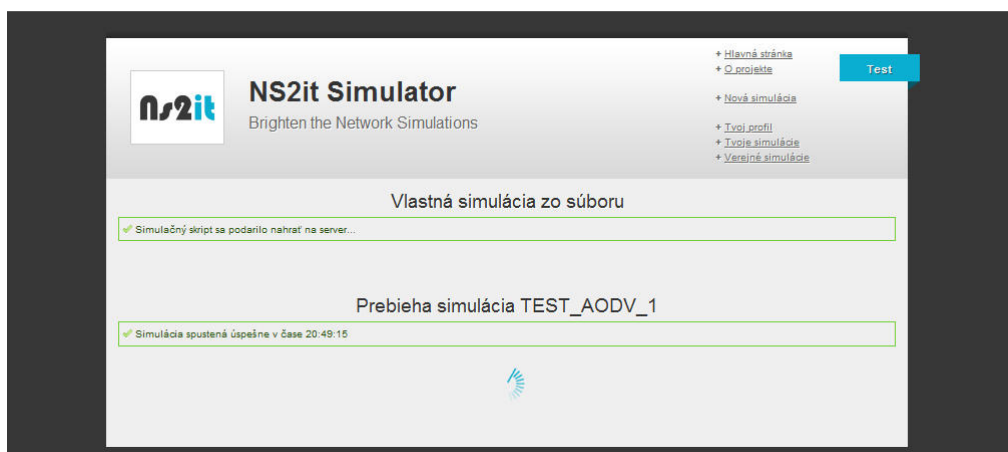
Tretou možnosťou je nahranie vlastného TCL skriptu z lokálneho počítača (obrázok 11).



Obrázok 11: Vloženie vlastného skriptu

Po nahrať alebo vytvorení skriptu systém kontroluje správnosť skriptu a začne skript vykonávať. Používateľ môže počkať kým sa skript vykoná, alebo si zvolí cestu mailovej notifikácie. Kvôli obmedzeniam zo strany FIIT, je mailová notifikácia blokována a teda nefunkčná.

Počas vykonávania simulácie sa zobrazuje používateľovi kontrolná obrazovka (obrázok 12). Používateľ nemusí stránku znovu načítavať. Po ukončení simulácie je notifikovaný potvrdzujúcou obrazovkou (obrázok 13).



Obrázok 12: Obrazovka pri vykonávaní simulácie

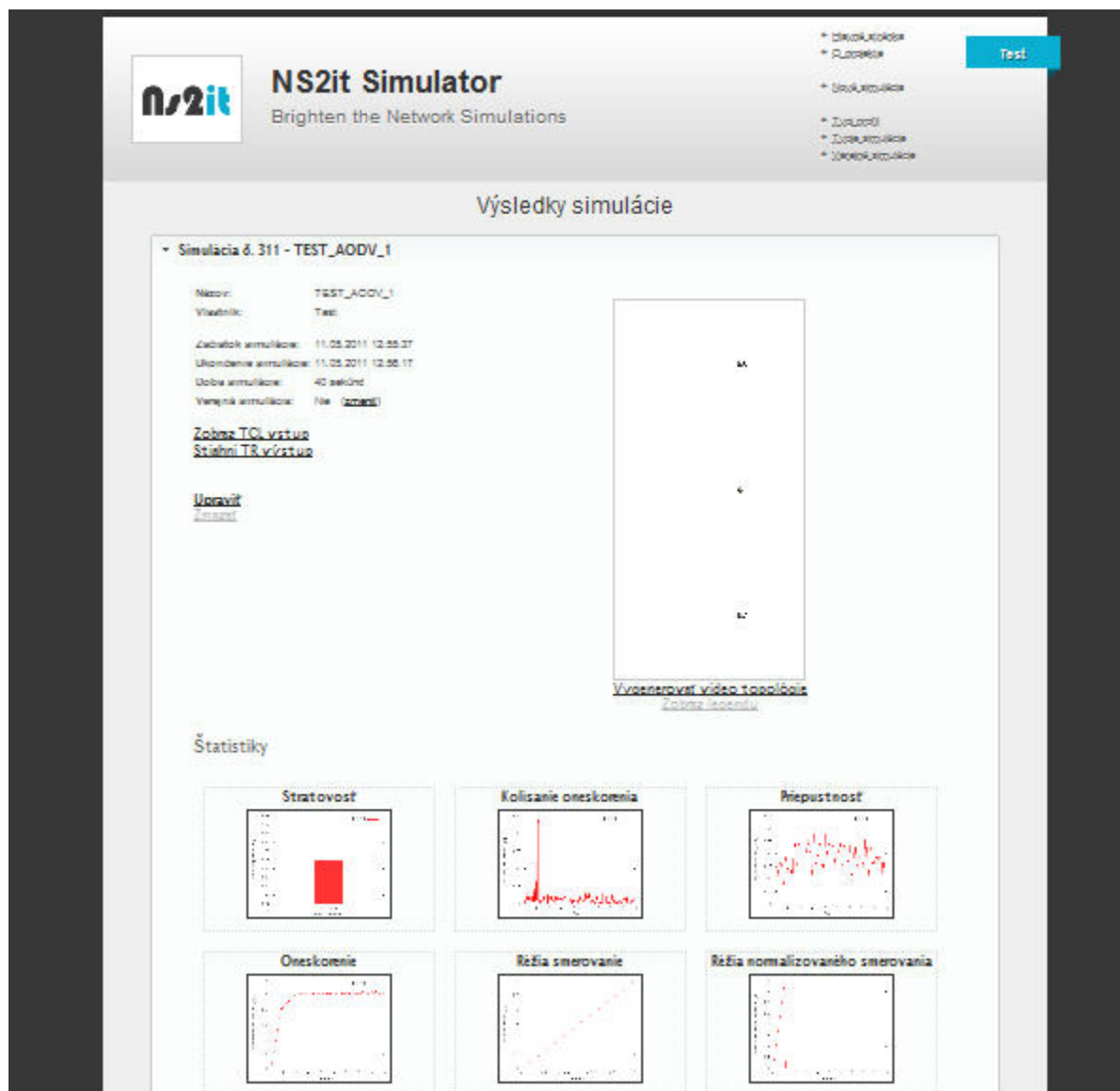


Obrázok 13: Dokončenie simulácie

Používateľ pokračuje ďalej kliknutím na modré tlačidlo „pokračuj na výsledky simulácie“. Po kliknutí sa mu zobrazí okno so štatistikami (obrázok 14).

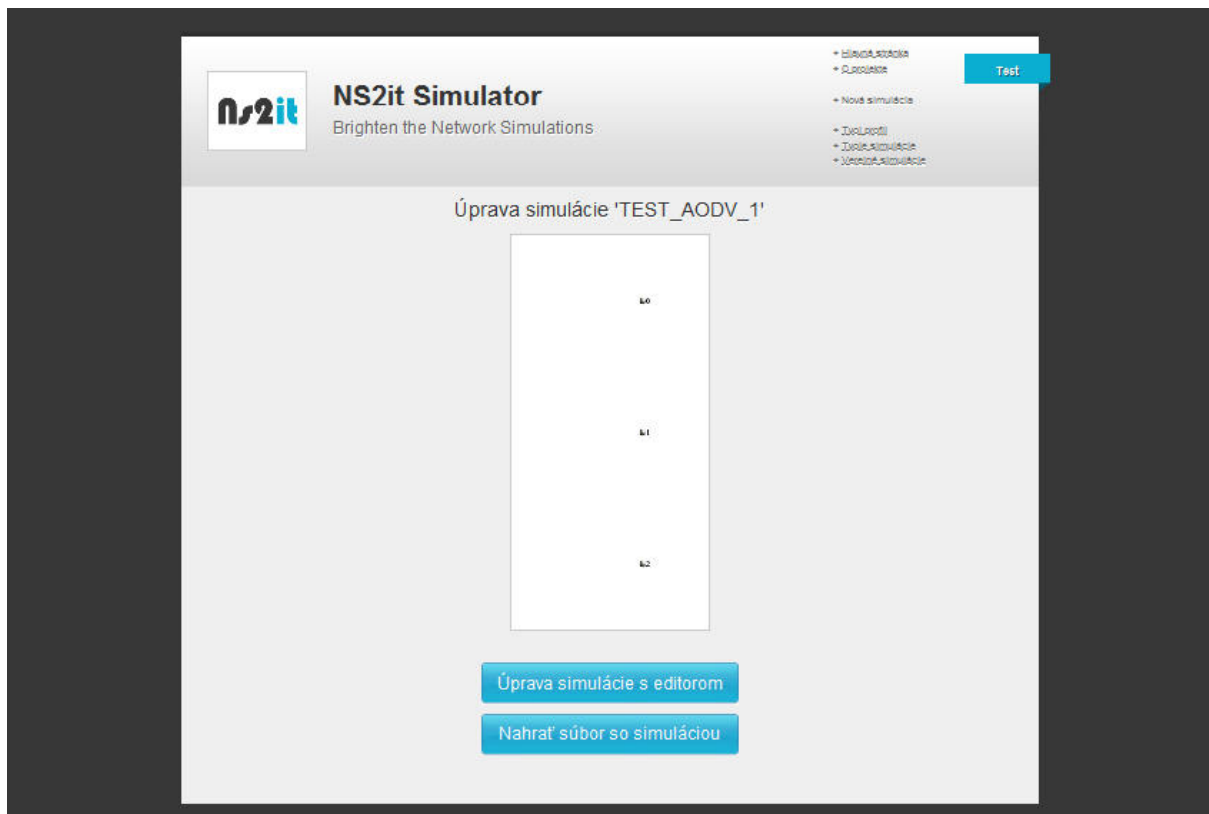
Na tejto stránke si dokáže používateľ stiahnuť vstupný TCL skript a aj výstupný tr.out súbor. Nachádza sa tu tiež obrázok topológie, spolu so štatistikami v prehľadných grafoch. Najväčšou devízou systému je možnosť vizualizácie simulácie. Na tento účel slúži video, ktoré sa vygeneruje počas simulácie. Uzly v sieti sú zobrazené kruhmi. Tieto kruhy v závislosti od svojej aktivity (prijatie paketu, odoslanie paketu atď.) menia svoju farbu. Detaily ohľadom videa sú uvedené v legende.

Ak používateľ chce svoju simuláciu zvejniť, klikne na odkaz „zmeniť“. Zoznam verejných simulácií je možno nájsť v hornom menu



Obrázok 14: Výsledok simulácie

Ak používateľ dokončil analýzu simulácie, má možnosť túto simuláciu upraviť. Úprava je možná dvoma spôsobmi a sice pomocou editor alebo nahratím súboru z lokálneho disku (obrázok 15). Postup je od tohto bodu obdobný, ako pri vytváraní novej simulácie



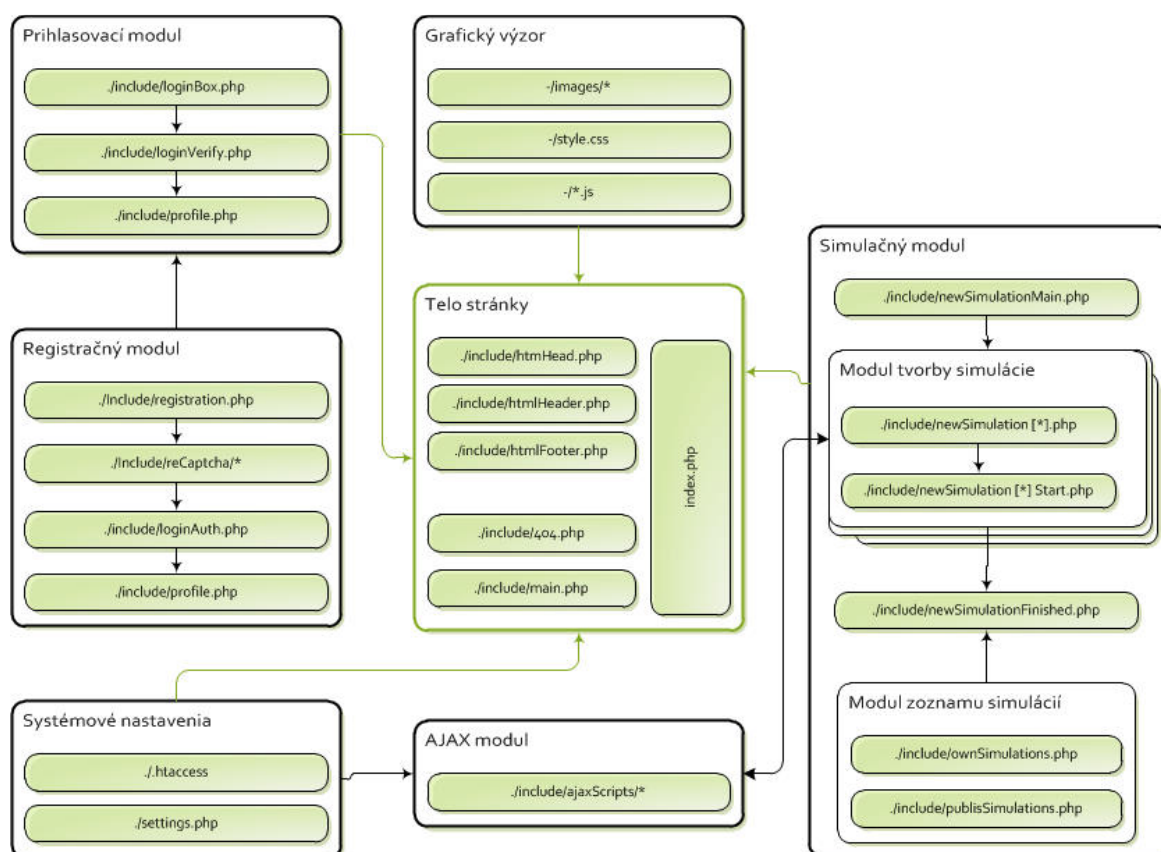
Obrázok 15: Úprava simulácie

Príloha D
Technická dokumentácia

1 Architektúra aplikácie

Webová aplikácia je uložená na webovom serveri Apache. Používatelia prístupujú k aplikácii prostredníctvom internetového prehliadača so zapnutou podporou JavaScriptu. Aplikácia je dynamická XHTML 1.0 Transitional webová stránka generovaná skriptovacím jazykom PHP na strane servera. Tento prístup nám umožnil vytvoriť atraktívnu dynamickú web stránku prístupnú na množstve platformách.

Architektúra aplikácie bol už od začiatku pripravovaný s požiadavkami na škálovateľnosť a bezpečnosť. Rozdelenie jednotlivých aplikačných skriptov do architektonických modulov je znázornené na obrázku (obrázok 1). Každý z modulov je pred spustením skontrolovaný, či sa spúšťa z nami definovaného prostredia (a nie z prostredia potenciálneho útočníka) a či je používateľ autentifikovaný a autorizovaný pre prístup k modulu.



Obrázok 1: Architektúra aplikácie

Hlavnou časťou aplikácie je telo stránky definované v skripte `index.php`. Do skriptu sa vkladajú dôležité systémové nastavenia (pripojenie k MySQL databáze, systémové premenné, globálne funkcie, funkcie zabezpečujúce korektný prístup k databáze, niektoré nastavenia webového servera) a súbory zabezpečujúce vzhľad webovej stránky (kaskádové štýly, obrázky, animačné javaskripty).

Správu a prihlasovanie nových používateľov má na starosti prihlasovací modul spolu z registračným modulom. Tento modul autentifikuje alebo najskôr zaregistruje používateľa pred prístupom k simulačnému modulu. Pre zamedzenie prihlasovania fiktívnych používateľov (internetových robotov) prechádzajú používatelia pri prihlasovaní kontrolou reCaptcha.

1.1 Simulačný modul

Simulačný modul vykonáva všetku funkcionality týkajúcu sa tvorby a úpravy simulácií. Pri komunikácii z príkazovým riadkom servera a pri vykonávaní vzdialených funkcií komunikuje s ostatnými skriptami prostredníctvom technológie AJAX. Tieto funkcie sú uložené v AJAXovom module. Pred vykonaním AJAX modulu sa vždy kontroluje, či je volaný naším modulom a či je kontaktovaný AJAXom, nie priamo z internetového prehliadača.

Simulačný modul obsahuje moduly tvorby simulácie (tvorba autogenerátorom, tvorba editorom, tvorba vlastným skriptom ...) a moduly zoznamu simulácií, ktoré zobrazujú existujúce simulácie podľa vlastných filtrov.

Spustenie simulácie nahratého TCL skriptu sa vykoná PHP skriptom komunikujúcim s príkazovým riadkom servera (Obrázok 2).

```
$serverPath= "/var/www".substr($simulation['path'],1);
$tclPath = $serverPath. "simulacia".$simulation['id'].".tcl";

$command = "cd ".$serverPath." && /usr/local/ns-allinone-2.34/ns-2.34/ns ".
           $tclPath." 2> error ; touch OK ; cd / & ";
$exec = exec($command);
```

Obrázok 2: PHP skript vykonávajúci simuláciu NS2 simulátorom

Po úspešnej simulácii sa vykoná generovanie výstupných sieťových štatistík, výstupných grafov, obrázku a videa topológie (Obrázok 3).

```

if ($mobileNetwork)
{
    //@@@@@@@@@@@@@@@@@@@@
    // generovanie statistik a grafov
    //@@@@@@@@@@@@@@@@@@@@
    $strPath = $serverPath. "out.tr";
    $command = "cd /data/funkcne_skripty/ ; ./monumentalny_skript.sh ".$strPath." ".
                $serverPathNoSlash." 2> ".$serverPathNoSlash."/errorTr ; cd ";
    exec($command);

    //@@@@@@@@@@@@@@@@@@@@
    //generovanie topologie
    //@@@@@@@@@@@@@@@@@@@@
    $command = " cd ".$serverPath." ; xvfb-run /data/Inspect/ns2it_inspect.sh -p
                -tf out.tr -o topology.png -tcl ".$tclPathRelative." 2> errorInspect";
    echo $command;
    $exec = exec($command);
    warning("Inspect errors: <br />".nl2br($exec));
}

$query="UPDATE simulation SET sim_finished='".time()."' WHERE id=".$sim_id." LIMIT 1";
mysql_query($query);

```

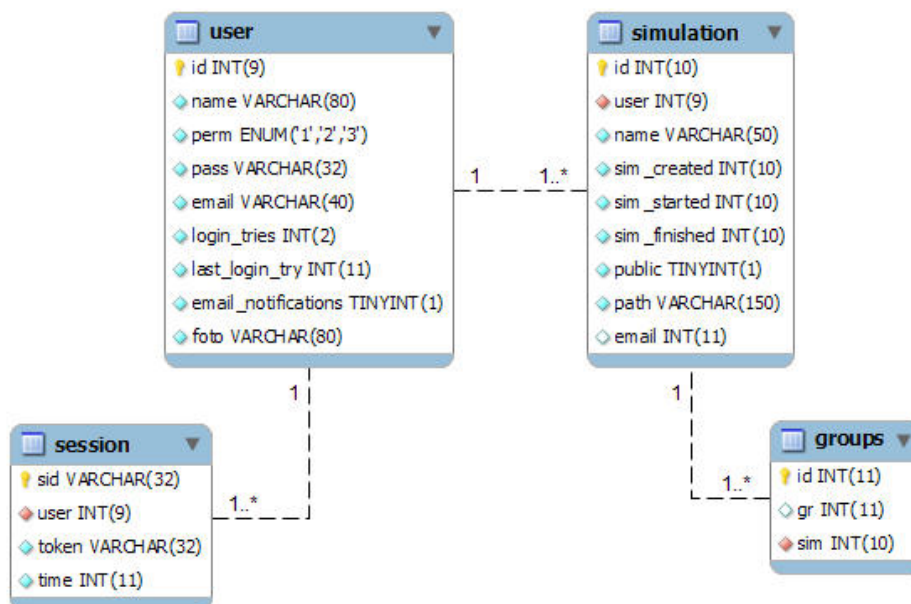
Obrázok 3 - PHP skript generujúci sieťové štatistiky a topológiu

2 Databáza údajov

Ako databázu údajov sme použili MySQL databázu typu InnoDB. Pre úplné fungovanie webovej aplikácie sme vytvorili štyri databázové tabuľky user, simulation, group a session. Vzťahy medzi tabuľkami sú vytvorené cudzími kľúčmi a ich integritu zabezpečujú automatické operácie ON_UPDATE a ON_DELETE. K databáze sa prístupuje prostredníctvom PHP knižníc z jednotlivých modulov web aplikácie.

2.1 Fyzický model databázy

Fyzický model je tvorený štyrmi entitami user, simulation, group a session. Typy stĺpcov boli vhodne vybrané s ohľadom na úložné miesto databázového servera a na potreby aplikačných modulov. Schéma entít a vzájomných vzťahov je znázornená na obrázku (Obrázok 4)



Obrázok 4: Fyzický databázový model

2.2 Opis databázových entít

2.2.1 User

Základnou entitou navrhnutého modelu údajov je entita *User*, predstavujúca zaregistrovaného používateľa. Entita ukladá všetky potrebné informácie o používateľoch. Používateľ sa registruje prostredníctvom registračného formulára vo webovej aplikácii. Následne sa môže pod svojím prihlasovacím menom a heslom prihlásiť, čo je nutná vstupná podmienka pri prístupovaní k väčšine častí nášho systému.

Opis jednotlivých polí je nasledovný:

- *Id* – unikátny identifikátor používateľa, táto hodnota sa automaticky zvyšuje je primárnym kľúčom entity
- *Name* – používateľské meno používateľa nutné pri prihlasovaní
- *Perm* – používateľský stupeň, súčasný systém označuje všetkých používateľov hodnotou 1. V budúcnosti je možné pridať administrátorov s hodnotou 2.
- *Pass* – používateľské heslo nutné pri prihlasovaní. Heslo je šifrované MD5 nevratnou šifrou.
- *Email* – email používateľa, kde bude informovaný o skončených simuláciách a novinkách systému.
- *Login_tries* – počet neúspešných prihlásení
- *Last_login_try* – čas posledného neúspešného prihlásenia. Keď presiahne počet neúspešných prihlásení za posledných 10 minút hodnotu 10, používateľ bude na hodinu zablokovaný.
- *Email_notifications* – povolenie notifikácií emailom
- *Foto* – používateľská fotka

2.2.2 Session

Entita *Session* obsahuje úspešné používateľské prihlasovacie relácie. Je používaná prihlasovacím modulom. Okrem hlavnej funkcie uchovávanía zoznamu prihlásených používateľov, plní funkciu informovania o dlhej neaktivite používateľa, čo vedie k jeho automatickému odhláseniu. Medzi entitami *User* a *Session* existuje relácia typu 1:n.

Opis jednotlivých polí je nasledovný:

- *sid* – unikátny identifikátor používateľskej relácie, táto hodnota sa automaticky zvyšuje je primárnym kľúčom entity
- *user* – používateľ, ktorý je pod touto id relácie prihlásený. Cudzí kľúč stĺpca *User.id*.
ON_UPDATE akcia – CASCADE
ON_DELETE akcia - CASCADE
- *token* - token relácie
- *time* – čas poslednej používateľovej aktivity v tejto relácii. Po dlhej neaktivite je relácia ukončená a používateľ automaticky odhlásený.

2.2.3 Simulation

Entita *Simulation* obsahuje informácie o všetkých simuláciách vytvorených používateľmi. Vzniká, mení sa a zaniká interakciou v simulačnom module. Niektoré informácie ako napr. grafy sieťových štatistík simulácie a video záznam pohybu uzlov simulácie sú uložené v súborov systéme. Každá simulácia má unikátny priečink. Medzi entitami *Simulation* a *User* existuje relácia typu 1:n.

Opis jednotlivých polí je nasledovný:

- *id* – unikátny identifikátor simulácie, táto hodnota sa automaticky zvyšuje je primárnym kľúčom entity
- *user* – používateľ, ktorý simuláciu vytvoril. Cudzí kľúč stĺpca *User.id*.
ON_UPDATE akcia – CASCADE
ON_DELETE akcia - RESTRICT
- *name* – názov simulácie definovaný používateľom. Ak používateľ názov neurčil, tak je odvodený z id simulácie.
- *sim_created* – čas vytvorenia simulácie
- *sim_started* – čas začiatku simulácie simulátorom ns-2
- *sim_finished* - čas ukončenia simulácie simulátorom ns-2
- *public* – ak je nastavené na 0, tak simulácia nie je verejná a je viditeľná iba pre používateľa, ktorý ju vytvoril. Ak je hodnota 1, tak je simulácia verejná.
- *path* – cesta k unikátnemu priečinku v súborovom systéme.
- *email* - emailová adresa na ktorú chce byť používateľ notifikovaný po ukončení simulácie simulátorom NS2.

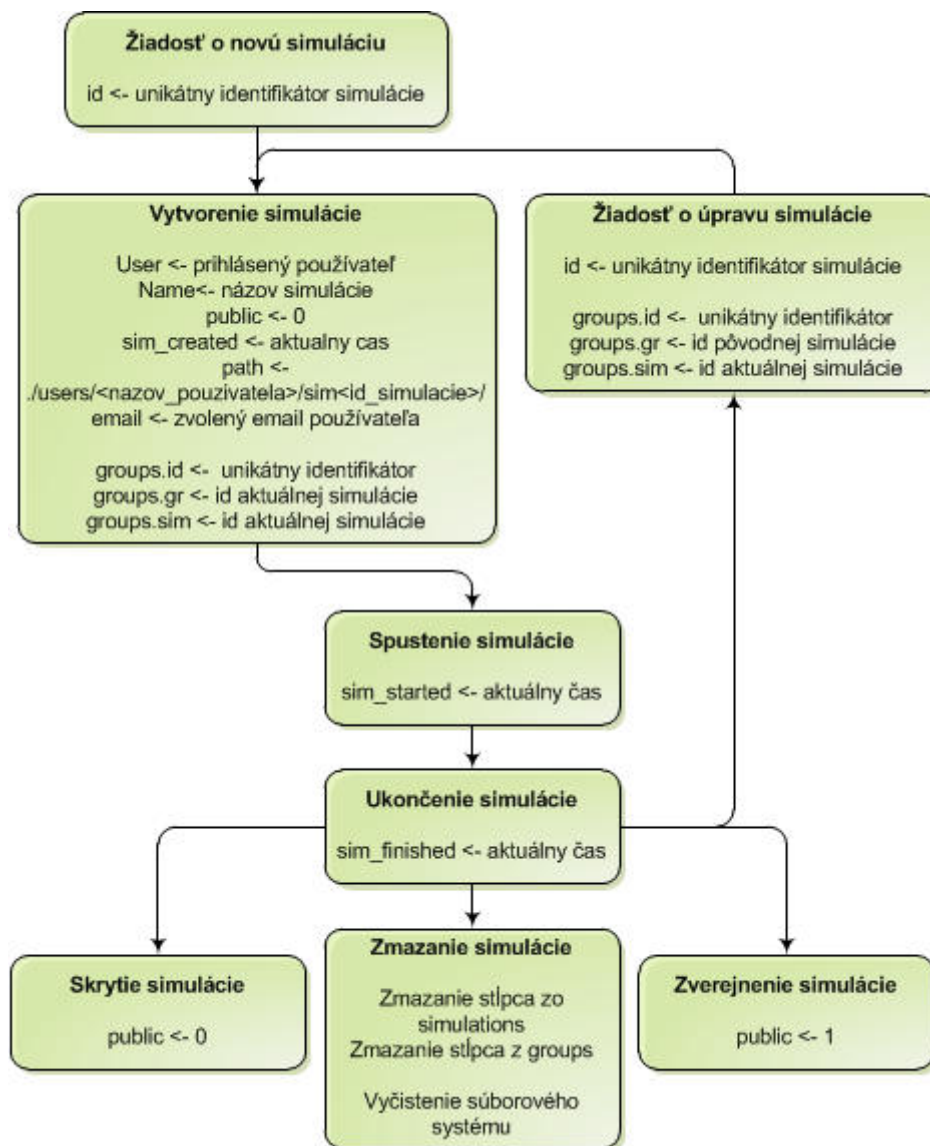
2.2.4 Group

Entita *Group* obsahuje informácie o vytvorených skupinách simulácií. Skupina simulácie vzniká v simulačnom module vždy po vytvorení novej simulácie a je rozširovaná pri každej úprave simulácie. Pre lepšie znázornenie interakcie simulačného modulu a entít

Group a Simulation slúži obrázok 5. Medzi entitami *Simulation* a *Group* existuje relácia typu n:1.

Opis jednotlivých polí je nasledovný:

- *id* – unikátny identifikátor skupiny simulácií, táto hodnota sa automaticky zvyšuje je primárnym kľúčom entity
- *gr* - číslo skupiny simulácií. Je rovnaké prvou simuláciou skupiny.
- *Sim* – identifikátor simulácie patriacej do skupiny. Cudzí kľúč stĺpca *User.id*.
ON_UPDATE akcia – CASCADE
ON_DELETE akcia – SET NULL



Obrázok 5: Interakcia simulačného modulu a databázových entít

3 Inspect

Na vykresľovanie obrázkov topológie a videí bol použitý program iNSpect vo verzii 3.5. Tento softvér funguje jednoduchým spôsobom, na vstupe číta trace file a vykresľuje priebeh simulácie (pohyb uzlov a komunikáciu) na obrazovku. iNSpect poskytuje funkcie zachytenia aktuálneho obrázku na obrazovke a jeho uloženia vo formáte png. Funkcia zachytávania videa je v iNSpecte riešená opakovaným volaním funkcie zachytenia obrázku a následným spustením programu mencoder, ktorý konvertuje sériu zachytených obrázkov na video.

Keďže je však iNSpect ovládaný cez GUI, boli potrebné zmeny pre potreby jeho integrácie do nášho riešenia. Okrem toho sa pri testovaní vyskytli problémy pri niektorých trace-filoch, preto boli modifikované aj ďalšie funkcie.

3.1 Zmeny v iNSpecte

V nasledujúcom texte sú uvedené zmeny v jednotlivých triedach iNSpectu. V zdrojovom kóde sú tieto zmeny označené komentárom “//team3”

eventBuilder – trieda, slúžiaca na spracovanie vstupných súborov a vytvorenie zobrazovaných udalostí z nich

v súbore *eventBuilder.h* pribudli protected premenné

- gint numNodes; (počet uzlov v simulácii)
- bool* defined_start; (ukazovateľ na pole, kde budú uložené informácie, či už bola definovaná počiatočná pozícia uzla)

v súbore *eventBuilder.cpp*

- vo funkcii void eventBuilder::readNSTrace() bola pridaná alokácia premennej defined_start.
- Na základe premennej defined_start sa potom vo funkcii int eventBuilder::parseTraceMobilityLine(string mobilityLine) určí, či už bola určená začiatočná pozícia uzla, ktorého pohyb predstavuje daný riadok v trace-file, a ak nie, bude za začiatočnú pozíciu uzla považovaná začiatočná pozícia v tomto pohybe.

iNSpect – hlavná trieda spúšťajúca program

v súbore *iNSpect.cpp*

- zmenená kontrola počtu argumentov, keďže v našom riešení sa preberá viac parametrov ako argumenty pri spúšťaní programu

vizManager – trieda zodpovedná za vytvorenie inštancií ostatných tried podľa zadaných parametrov.

v súbore *vizManager.cpp*

- boli modifikované kontroly parametrov podľa nášho formátu.

- Okrem toho, keďže spúšťame program príkazom, bez použitia ovládačov používateľského rozhrania, spúšťa táto trieda po vytvorení objektov funkciu na zachytenie obrázku alebo sekvencie obrázkov.

VizProperties – trieda zodpovedná za generovanie obrázkov a videa a taktiež obsahujúca parametre vizualizácie. Táto trieda bola najviac zmenená.

v súbore *VizProperties.h* pribudli deklarácie premenných

- double endTime;
- char* videoName;
- char* pngFileName;
- gboolean captureModeVideo;
- int picsMade;

a funkcií

- void setCapturing(double sttime)
- void makePic(char* filename, double capttime);
- void makeVid(char* filename, double starttime, double endtime, int frames);

v súbore *vizProperties.cpp*

- vo funkcii void vizProperties::screenCapture(char* savePath) bolo zmenené vytváranie názvov súborov. V prípade videa sú obrázky ukladané do dočasného adresára so špecifikovaným názvom (<názov>), následne je vytvorené video s názvom <názov>.avi
- V prípade obrázku je uložený s názvom <názov> do aktuálneho adresára
- Sekvencia obrázkov za účelom vytvárania videa bola pôvodne číslovaná podľa simulačného času vo formáte sekunda.stotiny.png. V prípade celej sekundy bol však obrázok očíslovaný sekunda.png, a pri spracovaní mencoderom do videa bol zaradený na zlé miesto. Táto chyba bola odstránená a aj pri celej sekunde sa čísluje sekunda.00000.png
- pridaná funkcia void vizProperties::setCapturing(double sttime) zapína nastavujúca simulačný čas a spúšťajúca zachytávanie obrázkov
- pridaná funkcia void makePic(char* filename, double capttime) – funkcia nastavuje potrebné parametre pre zachytenie obrázku a volá funkciu screenCapture
- pridaná funkcia void makeVid(char* filename, double starttime, double endtime, int frames) nastavuje potrebné parametre pre zachytenie obrázku a volá funkciu screenCapture

3.2 Vstupné parametre iNSpectu

Modifikovaný iNSpect pracuje v dvoch režimoch – obrázkovom a v režime videa.

v obrázkovom režime sa spúšťa príkazom

```
iNSpect -t <config><trace_file> -p <output_file><time>
```

kde <config> je cesta ku konfiguračnému súboru (vysvetlené neskôr), <trace_file> je cesta k trace súboru na spracovanie, <output_file> je meno výstupného súboru (obrázka), <time> je čas, v ktorom sa má urobiť snímka.

V režime videa sa iNSpect spúšťa príkazom

```
iNSpect -t <config><trace_file> -a
<temp><start_time><end_time><granularity>
```

kde <config> a <trace_file> majú rovnaký význam ako pri obrázkovom režime, <temp> špecifikuje názov podadresára pre dočasné súbory, <start_time> a <end_time> špecifikujú časový rozsah zachytávania videa a <granularity> označuje, ako často má iNSpect zachytávať snímky do videa (koľkokrát za sekundu).

Konfiguračný súbor, ktorý sa zadáva parametrom <config> obsahuje konfiguráciu vykresľovania, napríklad farby uzlov s rôznou aktivitou, veľkosti uzlov, a podobné parametre, ktoré sú v našom riešení vždy rovnaké. Okrem toho sú v konfiguračnom súbore aj základné údaje o simulácii, konkrétne počet uzlov, rozmery simulovanej topológie, a dĺžka simulácie.

3.3 Predpríprava údajov pre iNSpect

Na prípravu parametrov pre iNSpect a jeho spúšťanie slúži script `ns2it_inspect.sh`. Tento script umožňuje komplexnú manipuláciu s parametrami iNSpectu, keďže ich umožňuje získavať z rôznych zdrojov.

Skript umožňuje nastaviť parametre manuálne, alebo spracuje `.tcl` súbor a získava z neho požadované parametre – rozmery topológie, dĺžku simulácie a počet uzlov a pre nešpecifikované parametre nastaví predvolené hodnoty. Jednotlivé prepínače, ich význam a predvolené hodnoty sú uvedené v tabuľke 1.

Prepínač	Význam	Predvolená hodnota
-p / -v	Režim obrázku alebo režim videa. Povolený iba jeden z nich súčasne.	-
-tf	Adresa trace-filu. Povinné	-
-o	Názov výstupného obrázku.	-
-tcl	Adresa <code>.tcl</code> súboru	-
-time	Čas, v ktorom sa má zachytiť obrázok, resp interval, v ktorom sa má nahrávať video. Môže byť až po parametric <code>-p</code> alebo <code>-v</code> , aby bol známy mód (a teda počet súvisiacich časov)	Pri <code>-p</code> : - Pri <code>-v</code> : "0 <dĺžka_sim>"
-tmp	Pri mode videa prefix názvu dočasných súborov	"tmp"
-g	Granularita zachytávania videa	-*
-dim	Rozmery topológie	-*
-n	Počet uzlov	-*
-e	Dĺžka simulácie v sekundách	-*

Tabuľka 1: Parametre skriptu `ns2it_inspect.sh`

* - v prípade špecifikovania `-tcl` parametra sú použité hodnoty z `tcl` súboru

V rámci skriptu `ns2it_inspect.sh` sú vykonané tieto funkcie:

- Kontrola súborov – skontroluje sa, či existuje trace-file a tcl skript.
- Z tcl skriptu sa načítajú začiatočné pozície uzlov do trace-file. Toto je potrebné pre prípad stojacich uzlov, keďže iNSpect číta informácie o umiestnení uzlov z tzv. movementTraces v trace-file, ktoré špecifikujú pohyb vo formáte
M <číslo_uzla> (<x1, y1, z1>), (<x2, y2, z2>), <rýchlosť>, kde prvé tri súradnice sú súradnice začiatku pohybu a druhá trojica určuje cieľ pohybu. V prípade stojaceho uzla však tieto informácie nie sú v súbore, a teda iNSpect nevie ani, kde sa uzol nachádza. Preto sa začiatočné súradnice pridávajú do trace-filu ako pohyb na mieste s rýchlosťou 0.
- Vygeneruje sa konfiguračný súbor iNSpectu na základe zadaných parametrov alebo tcl skriptu. Podstatné parametre sú

```
numNodes <počet_uzlov>
endTime <dĺžka_simulácie>
maxX <rozmer_topógie>
maxY <druhý_rozmer_topológie>
```

Ostatné parametre sú nechané na predvolených hodnotách

- Spustí sa iNSpect v požadovanom móde
- Pri video móde sa zo sekvencie obrázkov vytvorí video programom mencoder

4 Skripty na spracovanie dát

Súčasťou riešenia je sada skriptov vykonávajúcich štatistické spracovanie dát. Tieto skripty sú uložené v adresári `/data/funkcne_skripty`

- `flow_counter.sh` – počíta odoslané pakety a stratené pakety pre každý tok.
- `jednosmerne_oneskorenie.sh` – skript rátajúci oneskorenia pre jednotlivé toky.
- `jitter.sh` – skript počíta rozdiely oneskorení paketov – tzv. jitter pri jednotlivých tokoch.
- `normalized_routing_load.sh` – skript počíta normalizovanú smerovaciu záťaž
- `stratovost.sh` – skript počíta stratovosti pre jednotlivé toky
- `throughput.sh` – skript počíta priepustnosť pre jednotlivé toky
- `plot_bar.sh` – skript nastavuje parametre vykresľovania stĺpcových grafov a dátové vstupy pre gnuplot
- `plot_line.sh` – skript nastavuje parametre vykresľovania čiarových grafov a dátové vstupy pre gnuplot
- `monumentalny_skript.sh` – tento skript je zavolaný po vykonaní simulácie. Spúšťa skripty pre spracovanie jednotlivých parametrov a následne spúšťa skripty na vykreslenie grafov pre tieto parametre.

Príloha E
Elektronické médium

Obsah adresárov elektronického média:

- web_staticky
 - Predstavuje webovú prezentáciu nášho tímu
- web_system
 - Nachádzajú sa tu PHP skripty všetkých modulov systému a súbory tvoriace používateľské rozhranie systému
- skripty_shell
 - Adresár obsahuje skripty, ktoré spracúvajú výstupný súbor zo simulátora, ďalej skripty generujúce grafy a modul na generovanie videa a obrázku topológie
- skripty_simulacie
 - Tu sa nachádzajú ukážkové simulačné scenáre v jazyku TCL