

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

## **Tím č. 2**

Bc. Martin Čechvala

Bc. Ivana Hucková

Bc. Jakub Obetko

Bc. Richard Roštecký

Bc. Juraj Šubín

Bc. Viktor Šulák

# **Simulácia bezdrôtových sietí**

**Analýza, Špecifikácia, Návrh, Prototyp**

Vedúci tímu: Ing. Peter Magula

november 2012

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Simulácia v bezdrôtových sieťach</b>	<b>2</b>
<b>3</b>	<b>Analýza predchádzajúceho projektu</b>	<b>4</b>
3.1	Analýza riešenia . . . . .	4
3.2	Použitie systému . . . . .	6
<b>4</b>	<b>Analýza ns-2</b>	<b>8</b>
4.1	Sieťový simulátor ns-2 . . . . .	8
4.2	Architektúra ns-2 . . . . .	8
4.3	Podpora bezdrôtových sietí v ns-2 . . . . .	9
4.3.1	LL . . . . .	10
4.3.2	ARP . . . . .	10
4.3.3	IFq . . . . .	10
4.3.4	MAC . . . . .	10
4.3.5	netIF . . . . .	11
4.3.6	Antenna . . . . .	11
4.3.7	Agent . . . . .	11
4.3.8	RTAgent . . . . .	11
4.3.9	Channel . . . . .	11
4.4	Podpora satelitnej komunikácie v ns-2 . . . . .	12
4.5	Podpora WiMAX v ns-2 . . . . .	13
4.6	Podpora Bluetooth v ns-2 . . . . .	13
4.7	Analýza Tcl skriptu . . . . .	13
4.7.1	Konfigurácia drôtovej siete . . . . .	14
4.7.2	Konfigurácia bezdrôtovej siete . . . . .	15
4.8	Formát výstupného súboru . . . . .	19
4.9	Podporné nástroje . . . . .	22
4.9.1	Nam: Network Animator . . . . .	22
4.9.2	Trace graph . . . . .	23
4.9.3	jTrana . . . . .	24
4.9.4	gnuplot . . . . .	25
4.9.5	drawnetwork . . . . .	26
4.9.6	XGraph . . . . .	27
4.9.7	iNSpect . . . . .	28
<b>5</b>	<b>Analýza ns-3</b>	<b>29</b>
5.1	Sieťový simulátor ns-3 . . . . .	29
5.2	Podpora bezdrôtových protokolov . . . . .	31
5.2.1	IEEE 802.11 (Wi-Fi) . . . . .	31

5.2.2	IEEE 802.16 (WiMAX) . . . . .	33
5.2.3	LTE . . . . .	34
5.2.4	VANET . . . . .	34
5.2.5	Podporné modely na simulovanie bezdrôtových sietí . . . . .	34
5.3	Vstupy a výstupy programu . . . . .	35
5.3.1	Vstupy . . . . .	35
5.3.2	Výstupy . . . . .	38
5.4	Podporné nástroje pre ns-3 . . . . .	39
5.4.1	Inet: Internet Topology Generator . . . . .	39
5.4.2	TraceMetrics . . . . .	40
5.4.3	FlowMonitor . . . . .	44
5.4.4	NS3Generator . . . . .	47
5.4.5	NetAnim . . . . .	48
<b>6</b>	<b>Zhodnotenie analýzy</b>	<b>51</b>
<b>7</b>	<b>Špecifikácia</b>	<b>53</b>
<b>8</b>	<b>Návrh riešenia</b>	<b>54</b>
8.1	Editor topológie . . . . .	54
8.2	Generátor skriptu . . . . .	55
8.3	Skript . . . . .	55
8.4	Knižnica . . . . .	55
8.5	Parametrizátor . . . . .	55
8.6	Simulátor ns-3 . . . . .	56
8.7	Flow Monitor . . . . .	56
8.8	awk/sed/gnuplot . . . . .	56
<b>9</b>	<b>Prototyp</b>	<b>57</b>
9.1	Cieľ prototypovania . . . . .	57
9.2	Grafické používateľské rozhranie . . . . .	57
9.3	Technická realizácia GUI . . . . .	58
9.4	Technická realizácia Parametrizátora . . . . .	59
9.5	Výsledky simulácie . . . . .	60
9.6	Simulačný skript . . . . .	60
	<b>Literatúra</b>	<b>62</b>
	<b>Príloha A: Dokumentácia k riadeniu projektu</b>	<b>65</b>

## Zoznam obrázkov

1	Obr. 3.1: Architektúra aplikácie . . . . .	4
2	Obr. 3.2: Prípady použitia systému . . . . .	7
3	Obr. 4.1: Základná architektúra ns-2 . . . . .	9
4	Obr. 4.2: Štruktúra MobileNode a jeho komponentov . . . . .	12
5	Obr. 4.3: Štruktúra BT zásobníka UCBT . . . . .	13
6	Obr. 4.4: Okno programu Nam s ukázkou výstupu . . . . .	23
7	Obr. 4.5: Okno programu Trace graph s ukázkou výstupu . . . . .	24
8	Obr. 4.6: Okno programu jTrana s ukázkou výstupu . . . . .	25
9	Obr. 4.7: Okno programu jTrana s ukázkou výstupu . . . . .	25
10	Obr. 4.8: Ukážka výstupu programu gnuplot . . . . .	26
11	Obr. 4.9: Ukážka výstupu programu gnuplot . . . . .	26
12	Obr. 4.10: Ukážka výstupu programu drawnetwork . . . . .	27
13	Obr. 4.11: Okno programu XGraph s ukázkou výstupu . . . . .	27
14	Obr. 4.12: Ukážka výstupu programu iNSpect . . . . .	28
15	Obr. 5.1: Príklad komunikácie vo WifiNetDevice modeli . . . . .	29
16	Obr. 5.2 - Vrstvy modelov v ns-3 . . . . .	30
17	Obr. 5.3: Základný model siete v ns-3 . . . . .	31
18	Obr. 5.4: Príklad komunikácie vo WifiNetDevice modeli . . . . .	33
19	Obr. 5.5: Okno programu TraceMetrics, záložka <i>Simulation</i> . . . . .	41
20	Obr. 5.6: Okno programu TraceMetrics, záložka <i>Nodes</i> . . . . .	42
21	Obr. 5.7: Okno programu TraceMetrics, záložka <i>Throughput/</i> <i>Goodput</i> . . . . .	42
22	Obr. 5.8: Okno programu TraceMetrics, záložka <i>Little's Result</i> . . . . .	43
23	Obr. 5.9: Okno programu TraceMetrics, záložka <i>Streams</i> . . . . .	43
24	Obr. 5.10: Okno programu TraceMetrics, ukážka detailného zobrazenia toku . . . . .	44
25	Obr. 5.11: Architektúra rozhrania FlowMonitor . . . . .	45
26	Obr. 5.12: Histogramy získané pomocou FlowMonitor . . . . .	47
27	Obr. 5.13 - Rozhranie programu NS3Generator . . . . .	48
28	Obr. 5.14: Štatistické zobrazenie rámcov . . . . .	49
29	Obr. 5.15: Ukažka vizualizácie pomocou NetAnim . . . . .	50
30	Obr. 8.1: Blokový návrh riešenia . . . . .	54
31	Obr. 9.1: Hlavné okno prototypu s popisom . . . . .	57
32	Obr. 9.2: Okno <i>Communication</i> s popisom . . . . .	58
33	Obr. 9.3: Výsledky simulácie . . . . .	60

# 1 Úvod

Tento dokument predstavuje projektovú dokumentáciu k projektu Simulácia bezdrôtových sietí v rámci predmetu Tímový projekt.

V úvode dokumentu je podrobne analyzované riešenie predchádzajúceho tímu, ktorý riešil rovnaký projekt v minulosti. V rámci tejto analýzy je podrobne opísaný implementovaný systém, jeho komponenty a funkcionality, možnosti použitia.

Dokument ďalej obsahuje podrobnú analýzu popredných sieťových simulátorov Network Simulator 2 a Network Simulator 3 (ns-2 a ns-3). V rámci analýzy sú detailne opísané konkrétne funkcie simulátorov, ich architektúra, vlastnosti a použitie. Pre každý simulátor je zdokumentovaná podpora jednotlivých častí a protokolov bezdrôtových sietí. Taktiež sú pre každý simulátor zdokumentované podporné nástroje, ktoré je možné použiť pre zjednodušenie práce. Každý z nástrojov je detailne opísaný. Pre každý simulátor sú taktiež analyzované vstupné a výstupné súbory.

Ďalšou časťou dokumentu je špecifikácia a návrh riešenia. V rámci špecifikácie sú opísané funkcie, ktoré budú naším projektom dodané vybranému simulátoru, ich vstupy a výstupy. V návrhu riešenia sú tieto funkcie detailne špecifikované spolu s grafickým znázornením štruktúry navrhovaného systému.

V závere dokumentu sa nachádza zhodnotenie analýzy, ktoré opisuje jednotlivé dosiahnuté výsledky.

## 2 Simulácia v bezdrôtových sieťach

Počítačové siete tvoria podstatnú časť dnešného sveta a bez ich existencie si život pravdepodobne ani nevieme predstaviť. Ulahčujú nám prácu a vzdelávanie, umožňujú komunikáciu bez ohľadu na vzdialenosť, sú využívané vo všetkých oblastiach života. Pripojení chcú byť všeci, vždy a všade.

Dôležitú časť počítačových sietí tvoria bezdrôtové siete a práve nimi sa budeme v rámci tohto projektu zaoberať. Pripojenie bez potreby dodatočnej kabeľáže alebo zariadení je pohodlné a preto aj veľmi obľúbené. Využíva sa na školách a pracoviskách, v kaviarňach, reštauráciách, na otvorených priestranstvách v mestách alebo na letiskách. Vzhľadom na množstvo využití bezdrôtových sietí je potrebné, aby bola každá sieť prístupná množstvu používateľov a ich potrebám.

Na zabezpečenie funkčnosti siete je potrebné detailne navrhnuť každý jej komponent a analyzovať všetky možné problémy. Aby sa predišlo implementácii zle navrhutej siete, využíva sa simulácia ako prostriedok na overenie správneho fungovania siete ako aj hĺbkovú analýzu možností siete.

Simulácia vo všeobecnosti predstavuje imitáciu správania sa konkrétneho cieľa, v našom prípade bezdrôtovej siete, v reálnom čase. Poskytuje pohľad na výsledné správanie sa siete za určitých podmienok, čím dáva možnosť opraviť chyby v návrhu alebo vylepšiť niektoré jeho časti. Simuláciou si taktiež vieme overiť správne nastavenia parametrov siete, jej pokrytie a možnosti pripojenia. Veľkou výhodou simulácie je možnosť jej opakovania s rôznymi vstupmi alebo nastaveniami, pričom čas behu simulácie si používateľ môže zvoliť. Týmto procesom sa používateľovi otvára možnosť skúmania všetkých prvkov v sieti ako aj všetkých udalostí, ktoré v sieti nastanú. Možnosť nastavenia granularity času umožňuje postupnú analýzu udalostí, ktoré by si pri reálnej prevádzke používateľ nemusel ani všimnúť [1].

Pre úspešnosť simulácie bezdrôtovej siete sú potrebné vstupné dáta - topológia siete, využité protokoly a tok dát v sieti. Možnými dodatočnými vstupmi sú informácie o pohybe jednotlivých uzlov. Na základe týchto vstupných údajov sa v priebehu simulácie vytvorí model zadanej siete a časový priebeh udalostí, ktoré v tomto modeli nastanú. Výstupom simulácie môže byť vyjadrenie stavu siete v jednotlivých krokoch v grafickom alebo textovom tvare. Ďalším výstupom simulácie môže byť zaznamenaný časový sled udalostí, ktoré v jednotlivých krokoch simulácie v sieti nastali. Pri testovaní mobility používateľov môže byť výstupom simulácie záznam siete vo forme videa. Zo všetkých typov výstupov je vhodné vytvárať štatistiky, ktoré reprezentujú všeobecné správanie sa siete za daných podmienok.

Simulácia počítačových sietí má množstvo výhod, kvôli ktorým je často využívaným nástrojom pri návrhu siete. Hlavnou výhodou simulácie je jej virtuálnosť. Pre testovanie alebo návrh novej siete nemusí používateľ mať k

dispozícii všetky jej hardvérové súčasti, stačí mu jeden program - simulátor. Keďže celá simulácia prebieha v podstate na jednom stroji, nie sú potrebné ani priestory, chladenie alebo elektrické napájanie pre každé zariadenie, ktoré by boli nevyhnutnou súčasťou prevádzkovania reálnej siete. Ďalšou výhodou simulácie je možnosť jej opakovania - používateľ môže skúmať detaily priebehu simulácie veľakrát. Taktiež môže meniť vstupy a nastavenia siete a granularitu času, aby zachytil všetky možné udalosti v sieti. Veľkou výhodou simulácie je aj fakt, že používateľ sa nemusí obávať poškodenia alebo zničenia zariadenia pri testovaní hraničných nastavení, keďže celá simulácia prebieha softvérovo. Jednou z výhod môže byť aj vzdelávacia funkcia simulácií - môžu byť použité na praktické aplikovanie teoretických vedomostí študentov a následné sledovanie správania sa siete.

V súčasnosti je dostupných niekoľko softvérových riešení poskytujúcich simulácie bezdrôtových sietí. Medzi najpoužívanejšie špičkové simulátory patria ns-2 a ns-3, ktorými sa budeme zaoberať v ďalších častiach práce. Našu pozornosť zameriame najmä na simulátor ns-3, pre ktorý vytvoríme príjemné rozhranie pre vstupy aj výstupy simulácie.

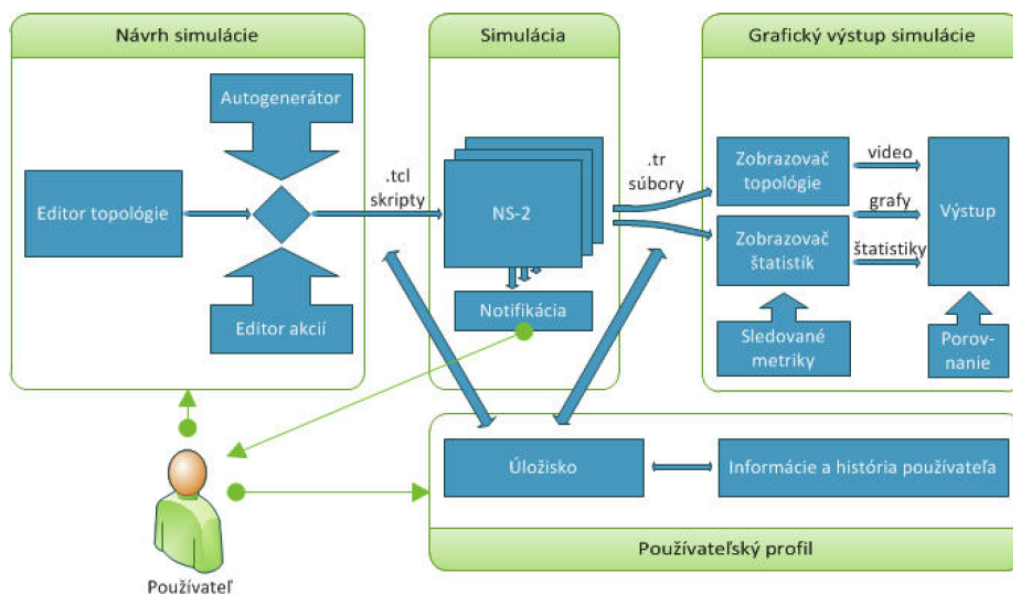
## 3 Analýza predchádzajúceho projektu

### 3.1 Analýza riešenia

Riešenie predchádzajúceho tímu pozostáva z aplikácie umiestnenej na linuxovom serveri. Rozhraním pre používateľa by mala byť webová lokalita, na ktorej sú dostupné všetky funkcie systému. Vzhľadom na to, že sme sa počas analýzy tohto riešenia nedostali k spomínanej stránke, nemali sme možnosť overiť existenciu a funkčnosť implementovaného riešenia. K dispozícii sme mali iba dokumentáciu k projektu, zdrojové kódy a skripty a na základe týchto materiálov bola postavená naša analýza.

Aplikácia bola navrhnutá ako klient-server aplikácia. Výhodou tohto riešenia je možnosť priebehu simulácie lokálne na počítači používateľa (ak má výkonnejší hardvér ako server). Po skončení simulácie server spracuje výsledky a pomocou e-mailu upozorní používateľa.

Aplikácia pozostáva zo štyroch hlavných komponentov, ktorých vzájomné vzťahy sú zobrazené na obrázku Obr. 3.1.



Obr. 3.1: Architektúra aplikácie

Pre prístup ku všetkým funkciám aplikácie je nutné, aby sa používateľ prihlásil. Po prihlásení má k dispozícii nasledujúce možnosti:

- Editor topológie:
  - návrh vlastnej topológie simulovanej siete,
  - vstup: opisné parametre siete,
  - výstup: \*.tcl skript.



- Autogenerátor:
  - vygeneruje súbory náhodného pohybu mobilných zariadení a náhodný tok dát medzi týmito zariadeniami,
  - 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 Editoru topológie.
  
- Editor akcií:
  - vlastné nastavenie pohybu zariadení v sieti a toku dát medzi nimi,
  - vstup: súradnice miest,po ktorých sa budú zariadenia pohybovať,
  - výstup: súbor, ktorý sa vloží do výstupného súboru Editoru topológie.
  
- ns-2:
  - vykonanie simulácie,
  - vstup: \*.tcl skript,
  - výstup: \*.tr súbor s výsledkami.
  
- Notifikácia:
  - notifikácia používateľa na zvolenú e-mailovú adresu.
  
- Zobrazovač topológie:
  - analyzuje výstupný súbor simulácie a zobrazí topológiu, pohyb zariadení,
  - vstup: \*.tr súbor,
  - výstup: video.
  
- Zobrazovač štatistík:
  - analyzuje vstupný súbor a vypočíta hodnoty, ktoré vykreslí do grafu,
  - vstup: \*.tr súbor,
  - výstup: graf alebo štatistická tabuľka.

- Porovnanie:
  - Zmení zobrazenie výsledkov do módu, v ktorom sa môžu porovnať výsledky dvoch simulácií.
- Úložisko:
  - databáza jednotlivých vstupných a výstupných súborov a výsledkov simulácií.
- Informácie a história používateľa:
  - informácie o používateľovi a jeho činnosti v systéme.

## 3.2 Použitie systému

Na obrázku Obr. 3.2 sú znázornené prípady použitia systému. Po registrácii a prihlásení môže používateľ vytvoriť simuláciu (\*.tcl skript) - pomocou editoru, autogenerátoru alebo priamo nahratím z disku lokálneho počítača. Systém následne skontroluje správnosť skriptu (či ide 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 skript prejde všetkými kontrolami, je spustený v simulátore. Po skončení simulácie sa automaticky generujú štatistické dáta - napr. náhľad topológie, videozáznam pohybu uzlov a všeobecné sieťové štatistiky (priepustnosť, oneskorenie, kolísanie oneskorenia).



Obr. 3.2: Prípady použitia systému

Riešenie využíva podporné nástroje ns-2 scenario generator (NSG) ako editor topológie na vytvorenie simulácie a iNSpect na vykresľovanie obrázkov topológie a videí. Na štatistické spracovanie výstupných dát simulácie si predchádzajúci tím vytvoril vlastné skripty - počítadlo paketov, stratovosť paketov, jednosmerné oneskorenie, jitter, priepustnosť, normalizovanú smerovaciu záťaž - na vytvorenie štatistík po skončení simulácie.

## 4 Analýza ns-2

### 4.1 Sieťový simulátor ns-2

Simulátor ns-2 (Network simulator 2) je v súčasnosti jeden z najznámejších a najpoužívanejších simulátorov počítačových sietí v oblasti výskumu a výučby. Je to udalosťou riadený (event-driven) simulačný nástroj, ktorý poskytuje široké možnosti nastavenia simulácie, topológie a dokáže vierohodne simulovať dynamické správanie sa počítačovej siete. Dokáže simulovať funkcie pevných aj bezdrôtových sietí a protokolov (napr. smerovacie protokoly, TCP, UDP, atď.). Testovanie pomocou simulácií nielen v ns-2 má mnohé výhody oproti testovaniu v reálnych podmienkach, ako napríklad:

- lacný spôsob testovania, nevyžaduje sa takmer žiadne technické vybavenie ani zariadenia,
- jednoduché simulovanie komplexných scenárov,
- rýchlejšia dostupnosť výsledkov,
- plná kontrola nad podmienkami testovania, nevplyvajú žiadne náhodné nepredvídateľné faktory.

Nevýhodou je, že modelovanie reálnych systémov v simulátore môže byť veľmi komplexnou záležitosťou. Vďaka svojej flexibilita a modulárnemu princípu získal obrovskú popularitu v oblasti výskumu počítačových sietí už pri svojom zrode v roku 1989. Od tej doby je ns-2 neustále vyvíjaný a jeho funkcie sa rozširovali. Od roku 1995 bol vývoj ns podporovaný agentúrou DARPA cez VINT (Virtual InterNetwork Testbed) [2]. Simulátor ns-2 je voľne dostupný simulátor v súčasnosti na stránke [3]. Posledná verzia ns-2.35 bola vydaná 4. novembra 2011. Vývoj simulátora je teda ešte stále aktívny, aj keď už v roku 2006 paralelne začal vývoj novšieho simulátora ns-3.

### 4.2 Architektúra ns-2

Simulátor je založený na dvoch jazykoch:

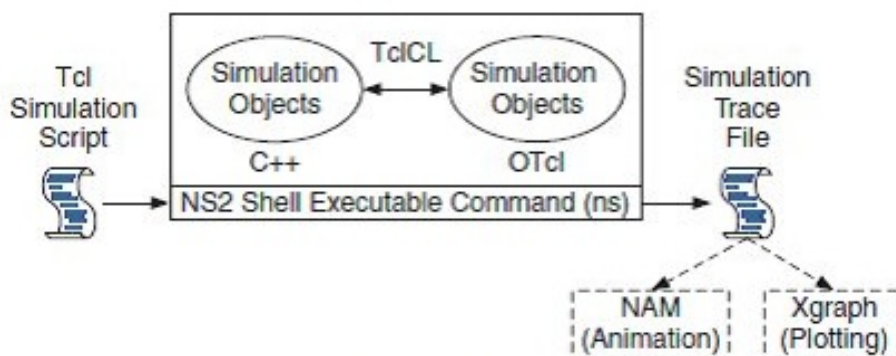
- C++,
- Object-oriented Tool Command Language – OTcl.

Zatiaľ čo C++ definuje vnútorný mechanizmus simulačných objektov, OTcl nastavuje simuláciu skladaním a konfiguráciou objektov ako aj plánovaním diskretných udalostí. C++ a OTcl sú vzájomne prepojené cez TclCL. Objekty a premenné vytvorené v OTcl sú cez TclCL mapované do C++. Využitie OTcl má výhodu v tom, že simulačné skripty nie je nutné písať v C++, čo je pomerne komplikovaný programovací jazyk. Simulačný skript napísaný v OTcl je teda vstupom do ns-2. OTcl je jazyk s jednoduchou syntaxou, ktorý poskytuje

jednoduché možnosti integrácie s inými programovacími jazykmi. Jeho hlavné charakteristiky sú:

- umožňuje rýchly vývoj,
- poskytuje grafické rozhranie,
- je kompatibilný s mnohými platformami,
- flexibilný pre integráciu,
- je to skriptovací jazyk.

OTcl umožňuje v ns-2 kompletnú kontrolu nad nastavením simulácie, konfiguráciou topológie, ale aj nastavením náhodných udalostí (napríklad vytvorenie nového TCP toku). OTcl umožňuje používateľovi rýchle zmeny v nastaveniach simulácie alebo konfigurácie a rýchle opätovné spustenie simulácie. Týmto spôsobom je možné analyzovať relatívne veľké množstvo možností konfigurácie za krátky čas. C++ je v ns-2 využitý na manipuláciu s bajtmi, prácu s paketmi a implementáciu algoritmov [4]. Základná architektúra simulátora ns-2 je zobrazená na obrázku Obr 4.1.



Obr. 4.1: Základná architektúra ns-2

### 4.3 Podpora bezdrôtových sietí v ns-2

Základ bezdrôtového modelu v ns-2 pozostáva z mobilného uzla (MobileNode) s prídavnými podpornými funkciami, ktoré umožňujú simuláciu ad-hoc sietí, WLAN, atď. Objekt MobileNode je v C++ odvodený od základného objektu Node, ktorému boli pridané niektoré funkcionality mobilného uzla ako napríklad možnosť pohybovať sa v rámci definovanej topológie, možnosť prijímať a odosielať signál cez bezdrôtové kanály (rádiové vlny), atď. Sieťový zásobník mobilného uzla pozostáva z linkovej vrstvy (LL), ARP modulu pripojeného k LL, prioritného radu rozhrania (IFq), MAC vrstvy (MAC), sieťového rozhrania (netIF) a toto všetko je pripojené ku kanálu [5]. Tieto komponenty sú vytvárané a spájané v OTcl. Jednotlivé komponenty si popíšeme podrobnejšie.

### 4.3.1 LL

Objekt LL je zodpovedný za simulovanie linkových protokolov. Ďalšou dôležitou funkciou linkovej vrstvy je nastavovanie správnych MAC adries v hlavičke rámca. LL používané componentom MobileNode je rovnaké ako LL používané componentom Node. Jediným rozdielom je v ARP module pripojenom k LL, ktorý mapuje MAC adresy rozhraní k známym IP adresám. Štandardne všetky odchádzajúce pakety sú odovzdané do LL smerovacím agentom. LL posúva pakety ďalej do nastaveného radu rozhrania. Všetky prichádzajúce pakety sú odovzdávané LL vrstve od MAC vrstvy, ktorá je následne odovzdaná na node entry point [5].

### 4.3.2 ARP

ARP modul prijíma dotazy od LL. Pokiaľ je známa MAC adresa cieľa, ARP modul ju zapíše do hlavičky rámca. V opačnom prípade sa vyšle ARP broadcast a paket sa zatiaľ dočasne uloží. Pre každú neznámu cieľovú MAC adresu sa vytvorí zásobník pre jediný paket. V prípade, že sú odoslané viaceré pakety do rovnakého cieľa, ARP modul predošlý uložený paket vyhodí. Akonáhle je MAC adresa známa, paket je odovzdaný do radu rozhrania (IFq) [5].

### 4.3.3 IFq

IFq je implementovaný ako prioritný rad, ktorý prioritizuje pakety smerovacích protokolov, ktoré ukladá na začiatok radu. Podporuje funkciu filtrovania všetkých paketov v rade a odstránenia tých so špecifickou cieľovou adresou[5].

### 4.3.4 MAC

MAC vrstva realizuje funkcionality podľa 802.11 štandardu. Od verzie ns-2.33 je k dispozícii niekoľko 802.11 implementácií [5]. Používané sú nasledované modely:

- **802.11 DCF:** Model používa RTS/CTS/DATA/ACK model pre unicast komunikáciu.
- **802.11 s infraštruktúrnymi rozšíreniami:** Rozšírenia zahŕňajú pasívne a aktívne prehľadávanie, autentifikáciu, asociáciu, inter-AP komunikáciu.
- **802.11Ext:** Model obsahuje funkcionality ako štrukturovaný návrh modulov MAC, kumulatívny výpočet SINR, sledovanie strát paketov na fyzickej vrstve a iné.
- **dei80211mr:** Model zahŕňa podporu viacerých prístupov na fyzickej vrstve, simuláciu vysielania na rôznych frekvenciách, atď.

#### 4.3.5 netIF

netIF slúži ako hardvérové sieťové rozhranie mobilného uzla pre prístup ku kanálu. Toto rozhranie simuluje kolízie, propagáciu rádiových vln a prijímanie paketov z kanálu. Rozhranie označí každý vysielaný paket metadátami súvisiacimi s odosielajúcim rozhraním ako napríklad sila signálu, vlnová dĺžka, atď. Tieto metadáta sú používané propagačným modelom na sieťovom rozhraní prijímajúceho uzla, ktorý tak rozlíši, či má daný paket dostatočnú silu signálu na to, aby bol detekovaný a prijatý na danom rozhraní [5].

#### 4.3.6 Antenna

MobileNode používa všesmerovú anténu [5].

#### 4.3.7 Agent

Agent je zodpovedný za generovanie a prijímanie paketov. Reprezentuje aplikačnú a transportnú vrstvu. Podporuje rôzne typy komunikácie ako CBR, FTP, TCP, atď.

#### 4.3.8 RTAgent

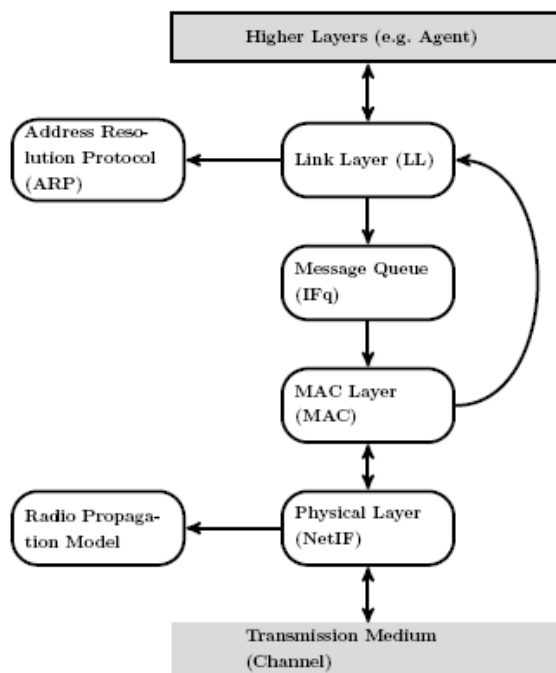
RTAgent je smerovací agent. Simuluje smerovacie protokoly a výpočet najlepších ciest medzi uzlami. Pre ad-hoc siete v súčasnosti ns-2 podporuje nasledovné smerovacie protokoly:

- DSDV,
- DSR,
- TORA,
- AODV,
- PUMA,
- M-DART.

#### 4.3.9 Channel

Objekt Channel simuluje prenos rámca na fyzickej vrstve. Channel je implementovaný ako zdieľané médium s podporou pre súperenie o prístup. Umožňuje MAC vrstve vykonávať carrier sense, súperenie a detekciu kolízií. Ak sa vyskytne kolízia, kanál nastaví kolízny príznak. Kontrolou tohto príznaku MAC implementuje detekciu kolízií. V bezdrôtových sieťach sa ako prenosové médium využívajú rádiové vlny, ktoré sú zdieľaným médium. Nakoľko čas vysielania paketu je daný jeho veľkosťou (počtom bitov) a rýchlosťou modulácie MAC rozhrania, objekt Channel jednoducho nastaví signál vyťaženia po dobu vypočítanú z daných parametrov. Rovnako naplánuje čas doručenia paketu na čas, ktorý získa sčítaním času prenosu a času vysielania paketu [6].

Štruktúra mobilného uzla MobileNode a spomínaných komponentov je znázornená na obrázku Obr. 4.2.



Obr. 4.2: Štruktúra MobileNode a jeho komponentov

## 4.4 Podpora satelitnej komunikácie v ns-2

Presné simulácie satelitných sietí vyžadujú detailné modely charakteristík rádiových frekvencií (interferencia, slabnutie), interakcií protokolov (interakcia zvyškových chýb s kódmi na kontrolu chýb) a druhoradých orbitálnych efektov (precesia, gravitačné anomálie, atď.). Avšak pre simuláciu základných charakteristík satelitnej komunikácie je možné tieto modely vynechať. Simulátor ns-2 umožňuje modelovať tri typy satelitných uzlov:

- geostacionárne,
- negeostacionárne,
- terminálne.

Každý satelitný uzol obsahuje jednu alebo viac satelitných sieťových zásobníkov, ku ktorým sú pripojené kanály na fyzickej vrstve. Štruktúra satelitného uzla je podobná ako štruktúra mobilného uzla, obsahuje komponenty ako LL, netIF, IFq, MAC, propagačný model rádiových vln. Je možné definovať niekoľko typov liniek. Pre vytvorenie spojenia medzi terminálnym zariadením na zemi a satelitom (uplink) je linka označená ako geo alebo polar. Pre vytvorenie spojenia v opačnom smere (downlink) sa využívajú gsl linky [7].

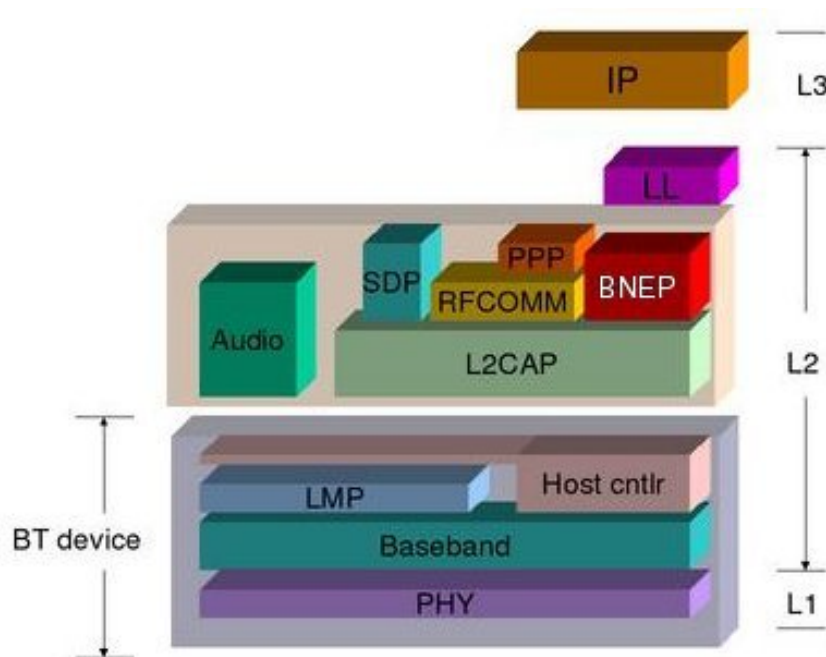


## 4.5 Podpora WiMAX v ns-2

Simulátor ns-2 vo svojej implementácii nepodporuje komunikáciu cez WiMAX protokol. WiMAX modul je ale implementovaný spoločnosťou Computer Networks Laboratory a tento modul je možné stiahnuť a doinštalovať do ns-2 simulátora. Modul implementuje 802.16 štandard, päť typov tokov, mechanizmy pre pridelenie šírky pásma a umožňuje konfiguráciu QoS. Modul podporuje TDD režim a PMP topológiu a využíva bezdrôtové kanály implementované v ns-2. Modul je možné stiahnuť na webovej stránke[8] .

## 4.6 Podpora Bluetooth v ns-2

Pre podporu Bluetooth komunikácie je možné doinštalovať do ns-2 UCBT modul, ktorý simuluje Bluetooth siete do detailu. UCBT implementuje Bluetooth zásobník zahŕňajúci základnú frekvenčnú, LMP, L2CAP a BNEP vrstvy. Štruktúra zásobníka je zobrazená na obrázku Obr. 4.3 [9].



Obr. 4.3: Štruktúra BT zásobníka UCBT

## 4.7 Analýza Tcl skriptu

Tcl skript sa používa pre opis konfigurácie siete, ktorú sa chystáme simulovať v simulátore ns-2. Uvedieme si 2 základné Tcl skripty, jeden pre drôtovú a jeden pre bezdrôtovú sieť o veľkosti dvoch uzlov, aby sme demonštrovali netriviálnosť simulácie bezdrôtových sietí. Netriviálnosť napríklad v tom zmysle, koľko parametrov je nutné nakonfigurovať pre uzol bezdrôtovej siete oproti uzlu drôtovej siete pre spustenie simulácie na základnej úrovni.

### 4.7.1 Konfigurácia drôtovej siete

Základný TCL skript, ktorý slúži na simuláciu topológie obsahujúcej dva uzly prepojené pomocou jednej linky sa vytvára podľa nasledujúcich krokov [12]:

1. Vytvorenie objektu simulátora.

```
set ns [new Simulator]
```

2. Otvorenie súboru (v príklade „out.nam“) slúžiaceho na zápis výstupných dát simulácie vhodných pre program nam (network animator).

```
set nf [open out.nam w]
$ns namtrace-all $nf
```

3. Definovanie ukončovacej procedúry.

```
proc finish {} {
    global ns nf
    $ns flush-trace
    close $nf
    exec nam out.nam &
    exit 0
}
```

4. Vytvorenie uzlov.

```
set n0 [$ns node]
set n1 [$ns node]
```

5. Vytvorenie linky na prepojenie vyššie definovaných uzlov n0 a n1. V príklade sa jedná o duplexnú linku so šírkou kanála 1 Mb, oneskorením 10 ms, a FIFO radom, ktorý v prípade preplnenia zahadzuje prichádzajúce pakety (technika DropTail).

```
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
```

6. Definovanie premávky v topológii. Posielanie dát medzi uzlami je v ns simulátore realizované prostredníctvom tzv. agentov. Dáta sa vždy posielajú medzi dvomi agentmi. V príklade bude zasielať uzol n0 generované dáta uzlu n1 využitím protokolu UDP.

- 6.1 Vytvorenie UDP agenta pre uzol n0.

```
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
```

- 6.2 Vytvorenie CBR (Constant Bit Rate) zdroja dát a jeho priradenie agentovi udp0. Zdroj bude generovať pakety o veľkosti 500 bajtov každých 0,005 sekúnd.

```
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
$cbr0 attach-agent $udp0
```

- 6.3 Vytvorenie agenta pre uzol n1, ktorý bude slúžiť pre zachytávanie paketov (Null).

```
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
```

- 6.4 Prepojenie oboch vytvorených agentov.

```
$ns connect $udp0 $null0
```

- 6.5 Určenie dĺžky činnosti CBR zdroja zadaním času začiatku činnosti a času ukončenia činnosti (vzhľadom na simulačný čas).

```
$ns at 0.5 "$cbr0 start"
$ns at 4.5 "$cbr0 stop"
```

7. Naplánovanie ukončovacej procedúry 5 sekúnd po začiatku simulácie (určenie dĺžky simulačného času).

```
$ns at 5.0 "finish"
```

8. Spustenie simulácie, koniec TCL skriptu.

```
$ns run
```

#### 4.7.2 Konfigurácia bezdrôtovej siete

Základný TCL skript, ktorý slúži na simuláciu jednoduchej bezdrôtovej topológie obsahujúcej dva uzly sa vytvára podľa nasledujúcich krokov [12]:

1. Definovanie premenných, pomocou ktorých sa budú konfigurovať bezdrôtové uzly.

typ kanála (channel type)

```
set val(chan) Channel/WirelessChannel
```

model šírenia bezdrôtového signálu (radio-propagation model)

```
set val(prop) Propagation/TwoRayGround
```

typ sieťového rozhrania (network interface type)

```

set val(netif)    Phy/WirelessPhy
typ MAC vrstvy (MAC type)
set val(mac)      Mac/802_11
typ radu na rozhraní (interface queue type)
set val(ifq)      Queue/DropTail/PriQueue
typ linkovej vrstvy (link layer type)
set val(ll)       LL
model antény (antenna model)
set val(ant)      Antenna/OmniAntenna
maximálny počet paketov v rade na rozhraní (max packet in ifq)
set val(ifqlen)   50
počet bezdrôtových / mobilných uzlov (number of mobilenodes)
set val(nn)       2
smerovací protokol (routing protocol)
set val(rp)       DSDV

```

2. Vytvorenie objektu simulátora.

```
set ns [new Simulator]
```

3. Otvorenie súboru (v príklade „priklad01.tr“) slúžiaceho na zápis výstupných dát simulácie.

```
set tracefd      [open priklad01.tr w]
$ns_ trace-all $tracefd
```

4. Vytvorenie objektu topológie, ktorý zaznamenáva pohyby mobilných uzlov v rámci určených hraníc.

```
set topo         [new Topography]
```

5. Definovanie hraníc topológie. V príklade sa budú mobilné uzly môcť pohybovať po ploche 100x100 metrov.

```
$topo load_flatgrid 100 100
```

6. Vytvorenie GOD (General Operations Director) objektu. GOD objekt má informáciu o celkovom počte uzlov a udržiava tabuľku najkratších ciest medzi jednotlivými uzlami (metrika v tabuľke je počet hopov).

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

7. Vytvorenie uzlov.

- 7.1 Pred samotným vytvorením uzlov je potrebné ich nakonfigurovať. Toto sa vykoná pomocou aplikácie na konfiguráciu uzlov, kde sa určia všetky parametre.

```
$ns_ node-config -adhocRouting $val(rp) \
    -llType $val(ll) \
    -macType $val(mac) \
    -ifqType $val(ifq) \
    -ifqLen $val(ifqlen) \
    -antType $val(ant) \
    -propType $val(prop) \
    -phyType $val(netif) \
    -topoInstance $topo \
    -channelType $val(chan) \
    -agentTrace ON \
    -routerTrace ON \
    -macTrace OFF \
    -movementTrace OFF
```

- 7.2 Vytvorenie dvoch uzlov node\_(0) a node\_(1) vo for cykle pre zjednodušenie. Príkaz random-motion 0 znamená, že nie je povolený náhodný pohyb uzlov.

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

8. Vytvoreným uzlom je potrebné definovať počiatočnú pozíciu po spustení simulácie. Pozícia sa definuje zadaním hodnôt x, y a z.

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

```
$node_(1) set X_ 90.0
$node_(1) set Y_ 85.0
$node_(1) set Z_ 0.0
```

9. Keďže sa nepovolil náhodný pohyb uzlov, definuje sa explicitne. Príkaz obsahuje simulačný čas, v ktorom sa uzol začne pohybovať a súradnicami určený cieľ pohybu.

```
$ns_ at 50.0 "$node_(1) setdest 25.0 20.0 15.0"
$ns_ at 10.0 "$node_(0) setdest 20.0 18.0 1.0"
$ns_ at 100.0 "$node_(1) setdest 90.0 80.0 15.0"
```

10. Definovanie premávky v topológii. V príklade bude zasielať uzol node\_(0) dáta uzlu node\_(1) využitím protokolu TCP.

- 10.1 Vytvorenie TCP agenta pre uzol node\_(0).

```
set tcp [new Agent/TCP]
$tcp set class_ 2
$ns_ attach-agent $node_(0) $tcp
```

- 10.2 Vytvorenie FTP aplikácie pre generovanie dát a jej priradenie uzlu node\_(0).

```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
```

- 10.3 Vytvorenie agenta pre uzol node\_(1), ktorý bude slúžiť na zachytávanie paketov.

```
set sink [new Agent/TCPSink]
$ns_ attach-agent $node_(1) $sink
```

- 10.4 Prepojenie oboch vytvorených agentov.

```
$ns_ connect $tcp $sink
```

- 10.5 Určenie začiatku činnosti FTP aplikácie.

```
$ns_ at 10.0 "$ftp start"
```

11. Pred ukončením simulácie je potrebné zresetovať vnútorné sieťové komponenty uzlov.

```
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 100.0 "$node_($i) reset";
}
```

12. Naplánovanie ukončovacej procedúry stop bezprostredne po resete uzlov. Procedúra stop zabezpečí správne uloženie súboru s výstupnými dátami simulácie. V príklade simulácia trvala 100 sekúnd. Príkaz ns\_halt zabezpečí, že nebudú posielané ďalšie dáta.

```

proc stop {} {
    global ns_ tracefd
    close $tracefd
}

$ns_ at 100.0001 "stop"
$ns_ halt"

```

13. Spustenie simulácie, koniec TCL skriptu.

```
$ns run
```

## 4.8 Formát výstupného súboru

Výstupný súbor obsahujúci všetky simulačné dáta sa nazýva trace file. Pre výstupný súbor obsahujúci simulačné dáta bezdrôtovej siete bol vyvinutý nový formát, ktorý sa aktivuje príkazom `$ns use-newtrace` v Tcl skripte. Výstup je potom formátovaný nasledovne [13]:

```

s -t 0.267662078 -Hs 0 -Hd -1 -Ni 0 -Nx 5.00 -Ny 2.00 -Nz 0.00 -Ne
-1.000000 -Nl RTR -Nw --- -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It
message -Il 32 -If 0 -Ii 0 -Iv 32

```

Ako vidieť na príklade vyššie, výstup možno rozdeliť do niekoľkých „tagových skupín“:

### 1. Typ udalosti

**s** – poslanie paketu (send)

**r** – prijatie paketu (receive)

**d** – zahodenie paketu (drop)

**f** – preposlanie paketu (forward)

### 2. Všeobecný tag

**-t** <time> - čas

**-t** <global settings> - globálne nastavenia

### 3. Charakteristiky uzla

**-Ni** <node id> - identifikátor uzla

**-Nx** <node's x-coordinate> - x-ová súradnica uzla

**-Ny** <node's y-coordinate> y-ová súradnica uzla

**-Nz** <node's z-coordinate> z-ová súradnica uzla

- Ne** <node energy level> - energetická úroveň uzla
- Nl** <trace level, napr. AGT, RTR, MAC> - úroveň zachytávania
- Nw** <reason for the event> - príčina udalosti

Parameter „príčina udalosti“ obsahuje skratku dôvodu zahodenia paketu. Vysvetlivky jednotlivých skratiek:

- END** DROP\_END\_OF\_SIMULATION - koniec simulácie
- COL** DROP\_MAC\_COLLISION - kolízia na MAC vrstve
- DUP** DROP\_MAC\_DUPLICATE - kolízia MAC adries
- ERR** DROP\_MAC\_PACKET\_ERROR - chybný paket
- RET** DROP\_MAC\_RETRY\_COUNT\_EXCEEDED - prekročený max počet pokusov na opätovné preposlanie
- STA** DROP\_MAC\_INVALID\_STATE - neplatný stav
- BSY** DROP\_MAC\_BUSY - uzol zaneprázdnený
- NRTE** DROP\_RTR\_NO\_ROUTE - nie je platná cesta
- 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 - neplatná správa typu callback
- IFQ** DROP\_IFQ\_QFULL - plný rad
- ARP** DROP\_IFQ\_ARP\_FULL - zahodené protokolom ARP
- OUT** DROP\_OUTSIDE\_SUBNET - napr. smerovacie správy z inej domény

#### 4. Informácie o pakete na sieťovej vrstve

- Is** <source address.source port number> - zdrojová adresa.zdrojový port
- Id** <dest address.dest port number> - cieľová adresa.cieľový port
- It** <packet type> - typ paketu
- Il** <packet size> - veľkosť paketu
- If** <flow id> - identifikátor toku
- Ii** <unique id> - unikátny identifikátor
- Iv** <tll value> - hodnota TTL

#### 5. Informácie o ďalšom hope

- Hs** <id for this node> - identifikátor uzla (ďalšieho hopu)



**-Hd** <id for next hop towards the destination> - identifikátor uzla pre ďalší uzol vzhľadom na cieľ

6. Informácie o pakete na linkovej vrstve

**-Ma** <duration> - čas trvania

**-Md** <dst's ethernet address> - cieľová MAC adresa

**-Ms** <src's ethernet address> - zdrojová MAC adresa

**-Mt** <ethernet type> - typ rámca

7. Informácie o pakete na aplikačnej vrstve

**-P arp** – informácie o ARP (Address Resolution Protocol) protokole:

**-Po** <ARP Request/Reply> - operačný kód (dotaz/odpoveď)

**-Pm** <src mac address> - zdrojová MAC adresa

**-Ps** <src address> - zdrojová IP adresa

**-Pa** <dst mac address> - cieľová MAC adresa

**-Pd** <dst address> - cieľová IP adresa

**-P dsr** – informácie o ad-hoc smerovacom protokole DSR (Dynamic source routing):

**-Pn** <how many nodes traversed> - počet prejdenných uzlov

**-Pq** <routing request flag> - príznak smerovacieho dotazu

**-Pi** <route request sequence number> - sekvenčné číslo smerovacieho dotazu

**-Pp** <routing reply flag> - príznak odpovede na smerovací dotaz

**-Pl** <reply length> - veľkosť odpovede

**-Pe** <src of the source routing->dst of the source routing> -zdroj a cieľ zdrojového smerovania

**-Pw** <error report flag> chybový príznak

**-Pm** <number of errors> - počet chýb

**-Pc** <report to whom> - adresát oznámenia chýb

**-Pb** <link error from linka->linkb> - chyba linky pri prenose z A do B

**-P cbr** - informácie o CBR (Constant bit rate) aplikácii:

**-Pi** <sequence number> - sekvenčné číslo

**-Pf** <how many times this paket was forwarded> - koľkokrát bol paket preposlaný

**-Po** <optimal number of forwards> - optimálny počet preposlaní

**-P tcp** - informácie o TCP toku:

**-Ps** <seq number> - sekvenčné číslo

**-Pa** <ack number> - číslo potvrdenia

**-Pf** <how many times this pkt was forwarded> - koľkokrát bol paket preposlaný

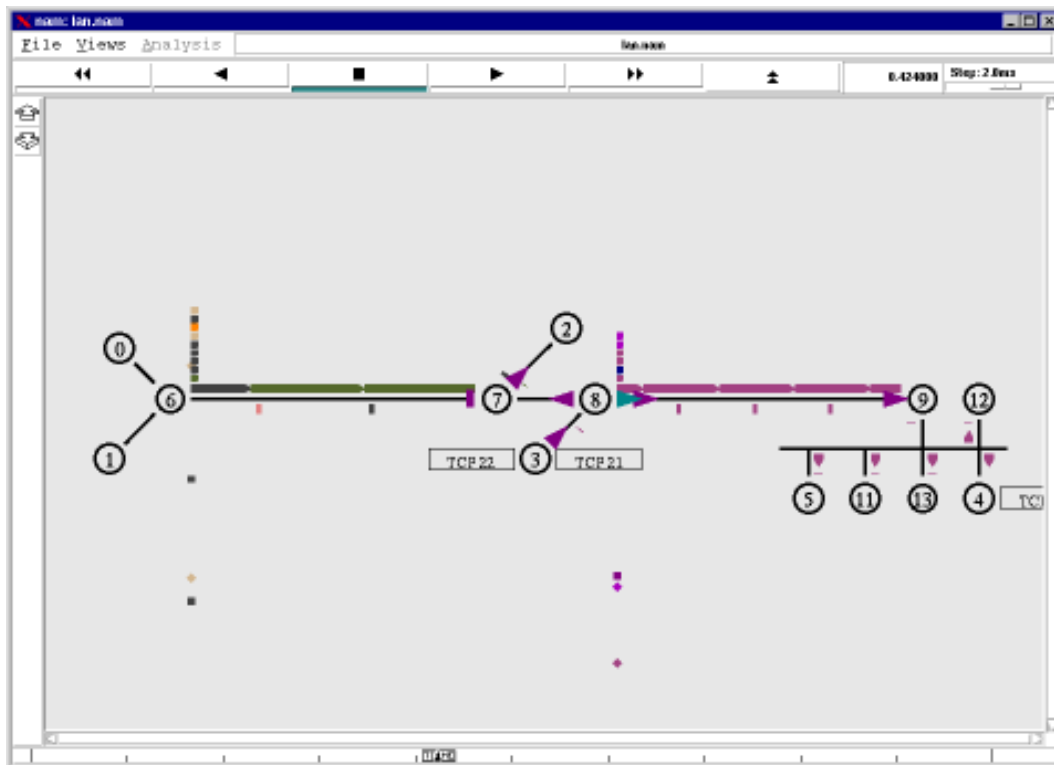
**-Po** <optimal number of forwards> - optimálny počet preposlaní

## 4.9 Podporné nástroje

Nakoľko simulátor ns-2 má len konzolové rozhranie, vzniklo množstvo podporných nástrojov pre grafické vytváranie vstupov pre simuláciu alebo interpretovanie výstupov simulácie. Tieto nástroje slúžia na vytvorenie používateľsky prívetivejšieho prostredia pre simulátor. Niečo podobné bude aj našou úlohou, pokiaľ zvolíme pre implementáciu použitie simulátora ns-2. Potom by pripadalo do úvahy zakomponovať do riešenia už existujúce nástroje tretích strán, preto niektoré z nich teraz popíšeme.

### 4.9.1 Nam: Network Animator

Nam je animačný nástroj založený na Tcl/Tk. Slúži na animovanie toku rámcov zo súboru zachytených simulačných udalostí a aj zachytených udalostí reálnej prevádzky. Podporuje zobrazenie topológie siete, animácie na úrovni toku paketov a niektoré nástroje pre analýzu dát. Základné prvky animácie sú uzol, linka, rad, paket a agent. Vstupom pre tento program je najčastejšie výstupný simulačný súbor simulátora ns-2, no dá sa použiť aj súbor so zachytenou reálnou prevádzkou (napr. programu Tcpdump). Najnovšia verzia programu 1.15 bola vydaná v roku 2011. Okno programu je zobrazené na obrázku Obr. 4.4. [13]



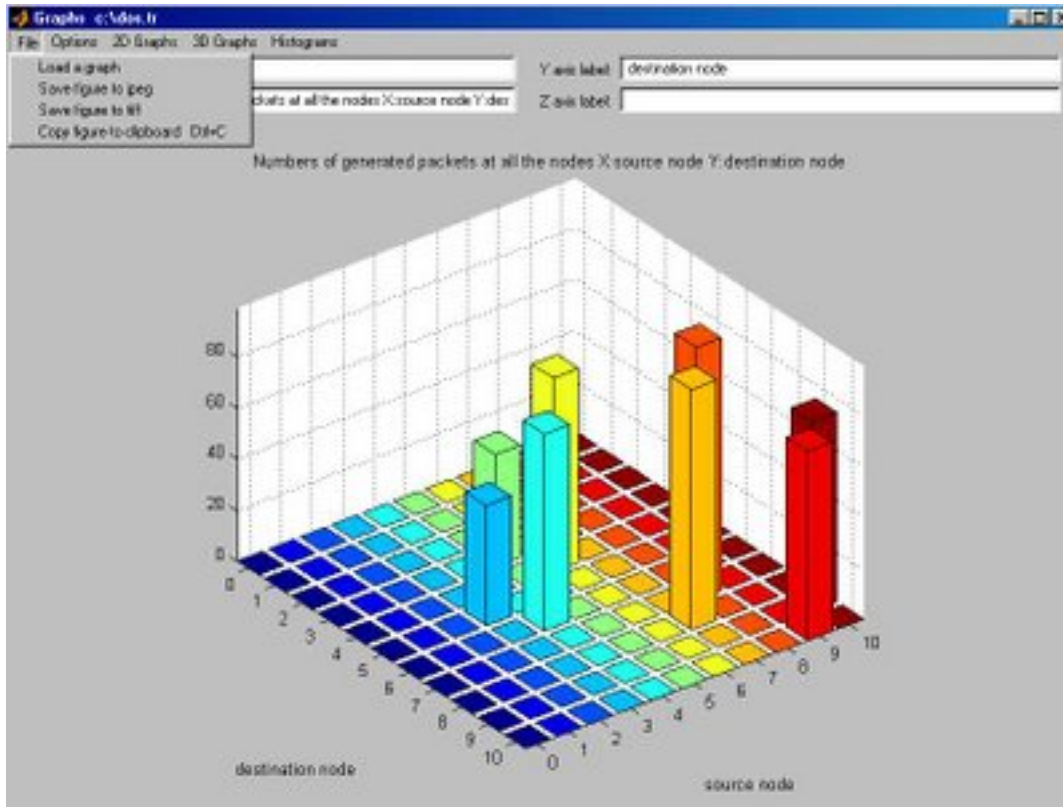
Obr. 4.4: Okno programu Nam s ukázkou výstupu [14]

#### 4.9.2 Trace graph

Program Trace graph je analyzátor sieťových súborov so zachytenými udalosťami / paketmi, ktorý bol vyvinutý pre spracovanie výstupných simulačných súborov programu ns-2. Podporuje všetky formáty výstupných simulačných súborov ns-2 (sieťové, satelitné, bezdrôtové...). Pre fungovanie programu je požadovaný Matlab verzie 6.1 a viac. Takisto pre optimalizovanie rýchlosti parsovania výstupných simulačných súborov programu ns-2 sa odporúča program Trace converter. Najnovšia verzia programu 2.05 bola vydaná v roku 2007. [15] Program môže zobrazovať:

- 238 2D grafov,
- 12 3D grafov,
- oneskorenia, časy spracovania, časy okruhov, grafy priepustnosti, štatistiky,
- grafy a štatistiky celej siete, linky alebo uzla,
- informácie o x, y, z súradniciach: minimum, priemer, maximum, štandardná odchýlka, medián.

Všetky výsledky z programu sa dajú uložiť do textových súborov, grafy sa dajú uložiť ako jpeg alebo tiff obrázky. Program síce poskytuje zaujímavé funkcie a grafické výstupy (obrázok Obr. 4.5), no jeho prípadné použitie by bolo pomerne komplikované.

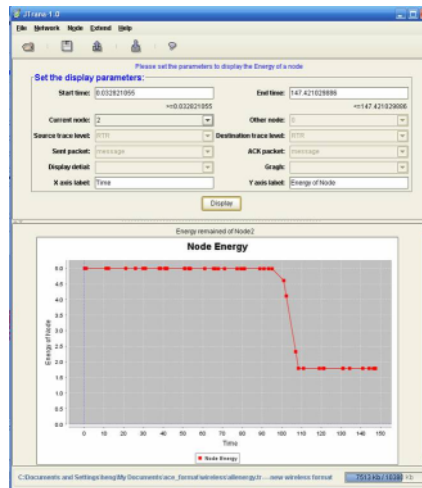


Obr. 4.5: Okno programu Trace graph s ukázkou výstupu [15]

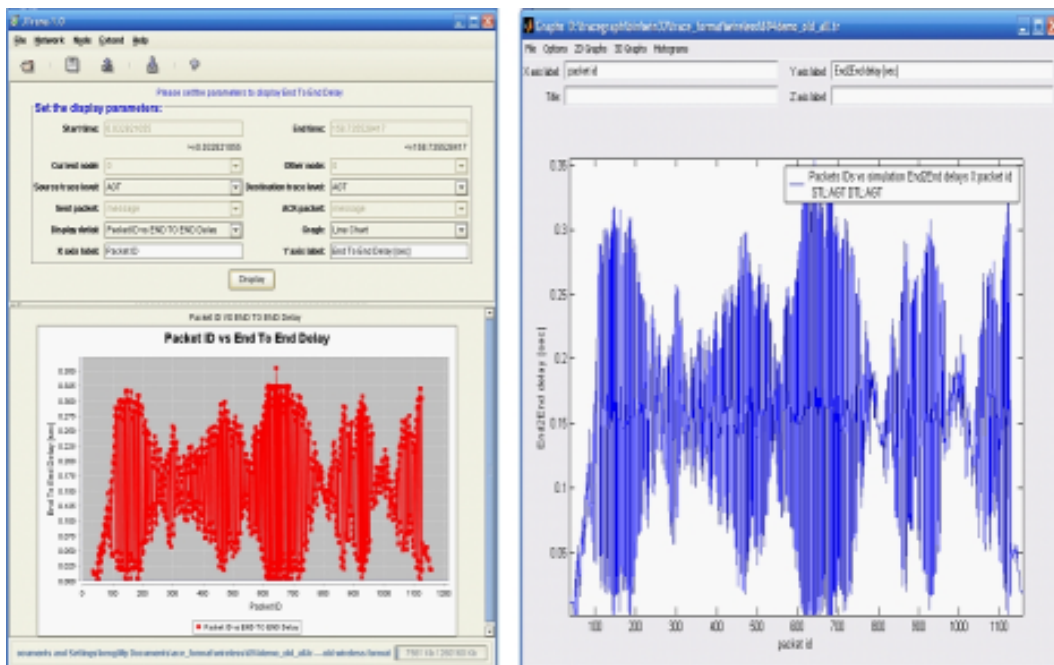
### 4.9.3 jTrana

jTrana je analyzátor výstupných simulačných súborov programu ns-2 výhradne pre bezdrôtové siete. Vstupom môže byť výstupný simulačný súbor bezdrôtovej siete programu ns-2 v starom alebo novom formáte. Pre fungovanie programu je požadovaná databáza MySQL a prostredie Java JDK 1.4 a vyššie, čo opäť značne komplikuje prípadné využitie v projekte. Výstupom programu je graf alebo dáta v textovej podobe. Okno programu je zobrazené na obrázku Obr. 4.6 a Obr. 4.7. Program podporuje [16]:

- funkcie pre celú sieť: všeobecný prehľad, zostatok energie celej siete, štatistiky paketov,
- funkcie pre jednotlivé uzly: informácie o zvolenom uzle, identifikátory paketov, priepustnosť, sekvenčné číslo, oneskorenie, RTT, pohyb,
- špeciálne funkcie: napr. získanie dát z databázy pomocou nejakého skriptu a získanie výsledkov.



Obr. 4.6: Okno programu jTrana s ukázkou výstupu [16]



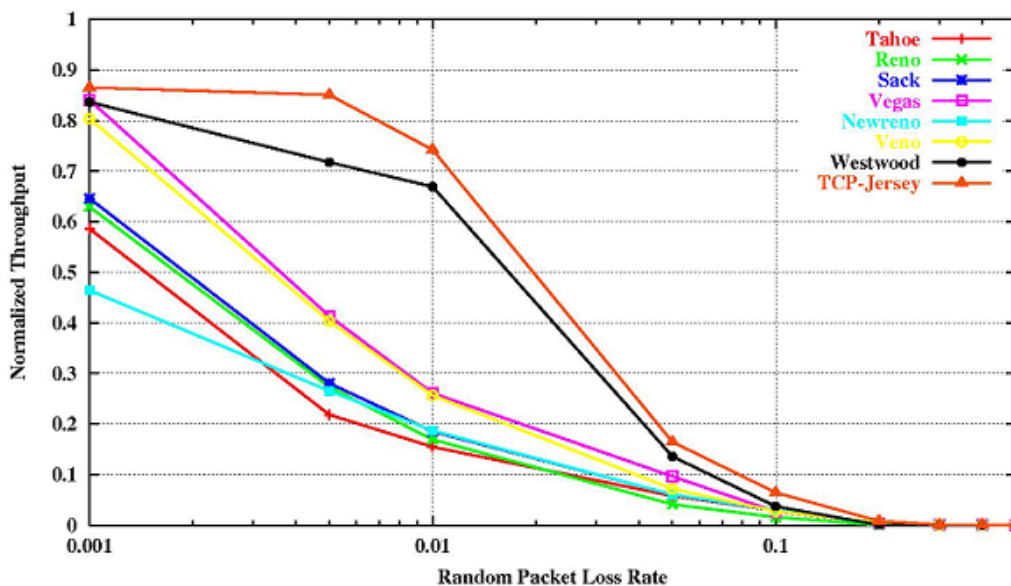
Obr. 4.7: Okno programu jTrana s ukázkou výstupu [16]

#### 4.9.4 gnuplot

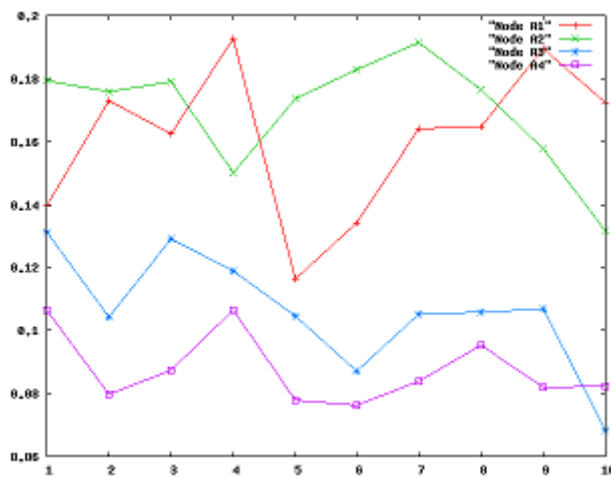
Program gnuplot je nástroj na tvorbu grafov. Môže byť využitý na vykresľovanie grafov funkcií a jednotlivých bodov v 2D alebo 3D zobrazení a rôznych formátoch. Jeho primárnou úlohou je teda grafické zobrazenie skúmaných dát. Najnovšia verzia programu 4.6.1 bola vydaná v roku 2012. Príklad výstupu je zobrazený na obrázkoch Obr. 4.8 a Obr. 4.9. Program ponúka [17]:

- vykresľovanie dvojrozmerných funkcií a bodov vo viacerých štýloch (body, čiary, boxy),
- vykresľovanie trojrozmerných bodov a povrchov vo viacerých štýloch (obrysy, mriežka),
- algebraické počítanie vo formáte čísla integer, float,

- používateľsky definované funkcie a klávesové skratky,
- multiplatformovosť, veľa formátov výstupných súborov.



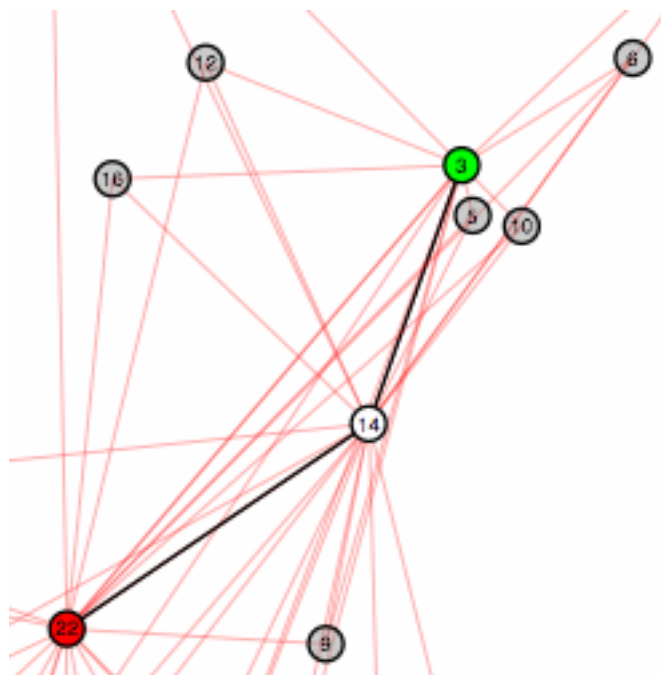
Obr. 4.8: Ukážka výstupu programu gnuplot [17]



Obr. 4.9: Ukážka výstupu programu gnuplot [17]

#### 4.9.5 drawnetwork

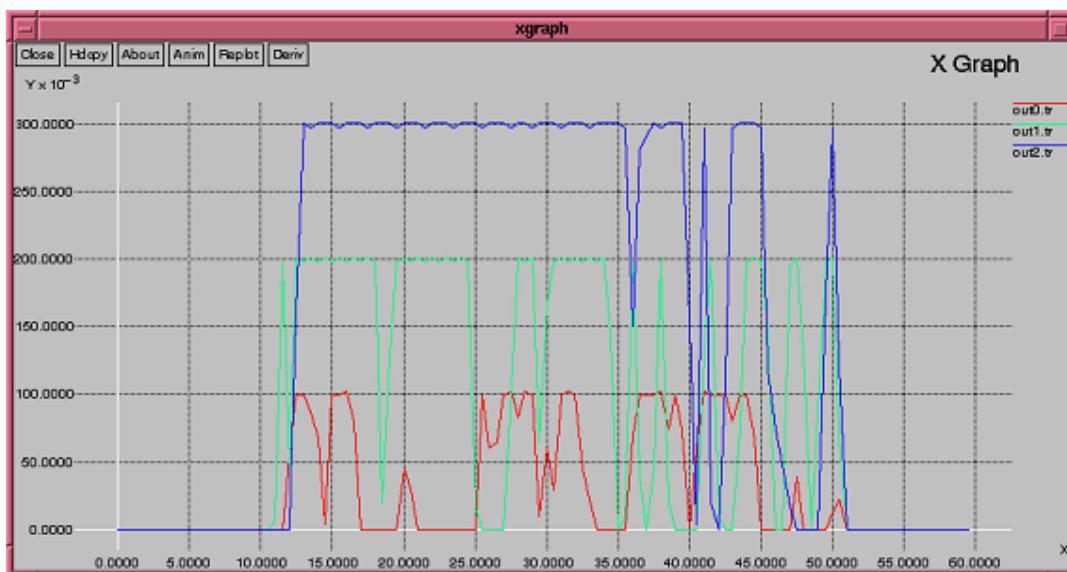
Program drawnetwork je schopný vykresľovať topológiu bezdrôtovej siete. Jediné dostupné informácie sú zdrojový kód, rok vydania 2004, a že program bol vytvorený v grafickom prostredí GnuStep. Výstup programu drawnetwork je zobrazený na obrázku Obr. 4.10. [18]



Obr. 4.10: Ukážka výstupu programu drawnetwork [18]

#### 4.9.6 XGraph

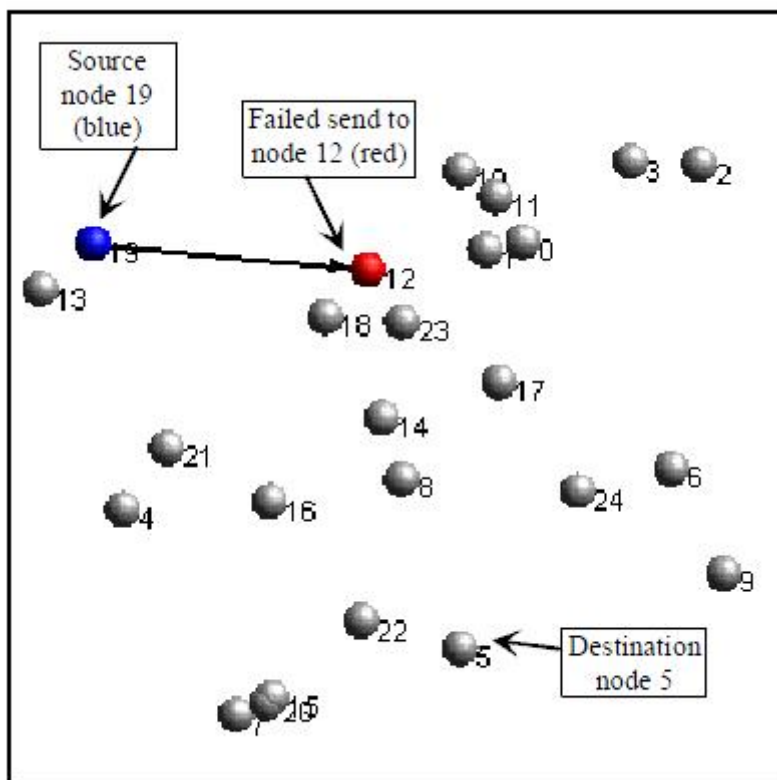
Program XGraph slúži na vykresľovanie grafov. Dáta môže program čítať zo súboru alebo môžu byť zadané štandardným vstupom. Program je schopný zobrazíť až 64 nezávislých skupín dát (funkcií) použitím rozličných farieb alebo štýlov čiar pre každú funkciu. Zobrazený graf obsahuje názov, označenie osí, mriežku grafu aj s označením hodnôt. Program môže byť do veľkej miery parametrizovaný, akceptuje veľké množstvo nastavení, ktoré môžu byť zadané v príkazovom riadku alebo zdrojovom súbore programu. Okno programu (obrázok Obr. 4.11) obsahuje aj interaktívne tlačidlá Hardcopy, Close a About. [19]



Obr. 4.11: Okno programu XGraph s ukážkou výstupu [12]

### 4.9.7 iNSpect

iNSpect je program, ktorý bol vyvinutý za účelom priamej vizualizácie a analýzy simulácií bezdrôtových sietí, pričom vstupom programu je výstupný simulačný súbor bezdrôtovej siete programu ns-2. Jeho hlavným zameraním je vizuálna prezentácia pohybu uzlov a dátových transakcií bezdrôtových sietí. Môže byť použitý prakticky z hociktorým simulátorom, ktorý poskytne výstup v požadovanom formáte. Program dokáže napríklad animovať MANET sieť bez toho, aby musel spustiť program ns-2 (číta dáta zo súboru pohybov) a tým sa značne znižujú aj nároky na beh tohto programu. Program umožňuje používateľovi spustiť simuláciu (pohybovať sa po časovej osi), pozastaviť ju, prejsť na zadaný simulačný čas alebo voliť si medzi jednoduchým zobrazením a zobrazením s podrobnejším prehľadom. Takisto poskytuje prostriedky pre tvorbu screenshotov, dokonca aj filmov. Príklad výstupu programu iNSpect je zobrazený na obrázku Obr. 4.12. [20]



Obr. 4.12: Ukážka výstupu programu iNSpect [20]



## 5 Analýza ns-3

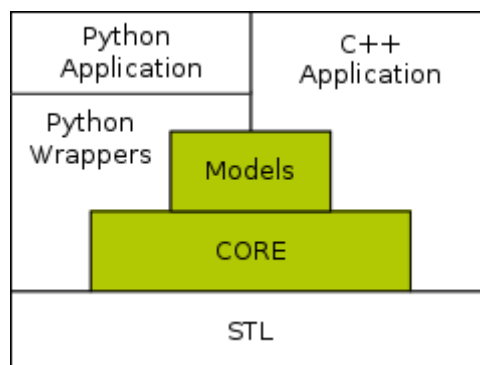
### 5.1 Sieťový simulátor ns-3

Simulátor ns-3 je sieťový simulátor, zameraný predovšetkým na výskum a vzdelávanie. Je to voľne šíriteľný softvér pod GNU GPLv2 licenciou a je voľne dostupný. Cieľom ns-3 je vyvinúť preferované, otvorené simulačné prostredie. Simulátor ns-3 nie je pokračovanie simulátoru ns-2. Je to čisto nový simulátor. Simulátor ns-3 má aktualizácie každé tri mesiace. Každá nová verzia obsahuje nové vylepšenia, opravené chyby a aktualizovanú dokumentáciu.

Simulátor ns-3 má jedny z najnižších pamäťových nárokov medzi sieťovými simulátormi a taktiež patrí medzi najlepšie v procesorovom zaťažení.

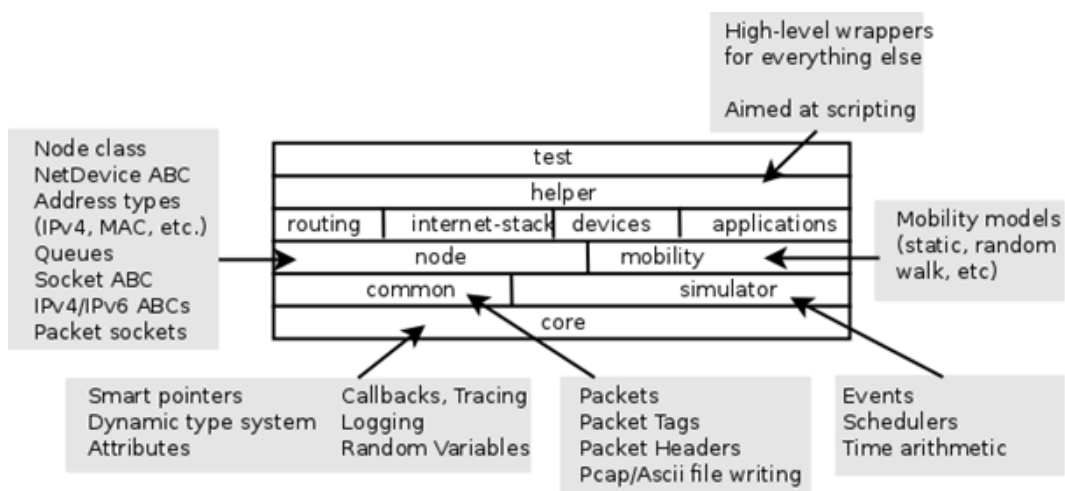
Projekt simulátora ns-3 sa zaviazal k budovaniu pevného simulačného jadra, ktoré je dobre zdokumentované, ľahko použiteľné a odladiteľné. Simulácia podporuje okrem IP sietí aj iné typy. Väčšina používateľov sa ale zameriava na bezdrôtové komunikácie, ktoré zahŕňajú Wi-Fi, WiMAX, LTE alebo sa zameriavajú na smerovacie protokoly ako OLSR a AODV. Simulátor ns-3 podporuje aj „real-time“ plánovač, ktorým je možné vytvárať prípady použitia pre interakciu s reálnymi systémami. Užívateľia môžu vysielat a prijímať pomocou ns-3 vygenerované pakety na skutočných sieťových zariadeniach. Simulátor ns-3 taktiež môže slúžiť ako softvérové príslušenstvo nahrádzajúce linku medzi dvomi virtuálnymi zariadeniami.

Simulátor tvorí knižnica C++, ktorá poskytuje sadu sieťových simulačných modelov implementovaných ako C++ objekty zabalené prostredníctvom jazyka Pythone. Používatelia môžu pracovať s knižnicou prostredníctvom písania aplikácií v jazyku C++ alebo Pythone, ktorými nastavujú a konfigurujú parametre simulácie. Príklad komunikácie vo WifiNetDevice modeli je zobrazený na obrázku Obr. 5.1.



Obr. 5.1: Príklad komunikácie vo WifiNetDevice modeli [38]

Knižnica simulátora ns-3 je zabalená do Pythonu vďaka knižnici Pybindgen. Pybingen je modul pre Python, ktorý je zameraný na generovanie C/C++ kódu. Simulátor ns-3 má viaceré výhody oproti iným sieťovým simulátorom. Simulátor ns-3 kladie prioritu na C++ a Python. Mnoho iných simulátorov má rôzne vlastné modelovacie jazyky na opis správania a toku programu. V ns-3 používatelia môžu využiť plnohodnotne C++ aj Python. Simulátor ns-3 je vybavený nízkoúrovňovým API, ktoré umožňuje pokročilým používateľom pružnosť pri konfigurácii rôznymi spôsobmi. Najvrchnejšia vrstva sa nazýva „pomocná“ vrstva, ktorá poskytuje ľahké použitie funkcií so základným správaním. Používatelia môžu pracovať naraz aj s najvrchnejšou vrstvou, pomocnou a zároveň aj s najspodnejšou vrstvou API (obrázok Obr 5.2).

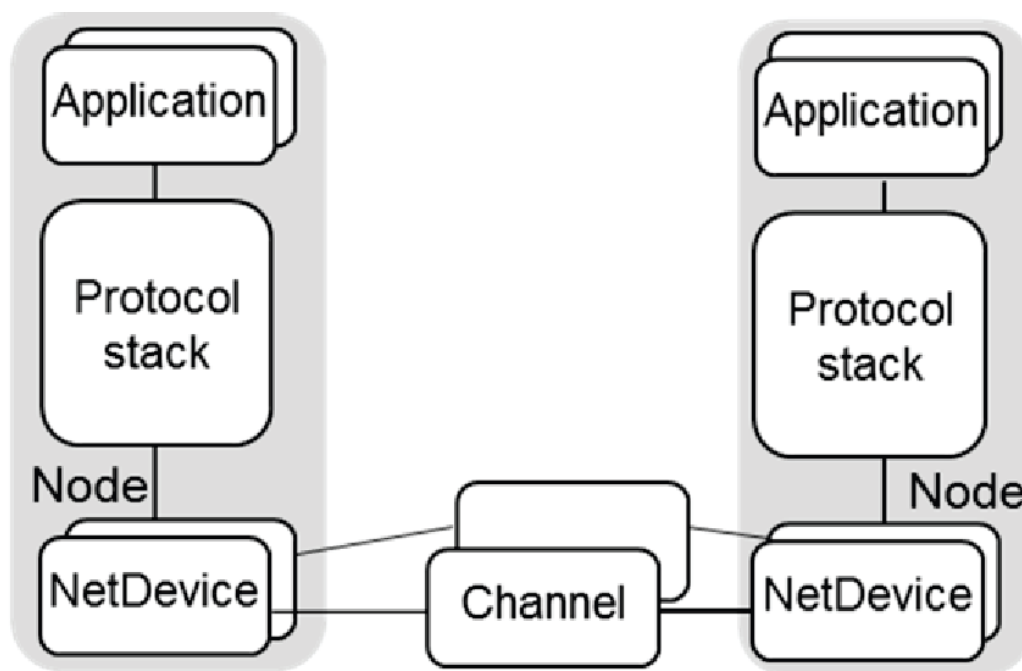


Obr. 5.2 - Vrstvy modelov v ns-3 [37]

Simulačný dizajn pri ns-3 je orientovaný na prípady použitia, ktoré umožňujú simulátoru interakciu s reálnym svetom. Pakety ns-3 sú interne uchovávané v zásobníku, podobne ako pri reálnych systémoch a sú pripravené na serializáciu a následne môžu byť odoslané na reálne sieťové zariadenie. Uzly sú navrhnuté tak, aby korešpondovali s linuxovou architektúrou. Kód je takto lepšie použiteľný a jednoduchšie sa robia porovnania s reálnymi systémami. Simulátor ns-3 je vybavený integrovaným atribútovo založeným informačným systémom pre správu parametrov simulácií. Všetky parametre sa dajú nastavovať z príkazového riadku. Nedostatok ns-3 je, že nemá vlastné vývojové prostredie pre konfiguráciu, ladenie, spúšťanie a vizualizovanie simulácie, ako je to pri iných simulátoroch alebo platformách. Typický pracovný postup pre ns-3 je prioritne prevádzaný cez príkazový riadok. Niektorí vývojári používajú ako vývojové prostredie Eclipse. [26, 27, 28, 32, 34, 37, 38]

## 5.2 Podpora bezdrôtových protokolov

Simulátor ns-3 podporuje viacero druhov bezdrôtových prenosov. Niektoré sú ešte vo vývoji a niektoré už plne funkčné. Objekty, ktoré simulujeme sa vytvárajú skladaním z modelov. Všetky modely sú dôkladne zdokumentované v **doxygen** dokumentácii. Každé zariadenie má svoju aplikačnú vrstvu, protokolový zásobník a sieťový uzol (obrázok Obr. 5.3). Sieťové uzly sú vzájomne poprepájané kanálmi znázorňujúcimi fyzické prenosové médiá.[27]



Obr. 5.3: Základný model siete v ns-3 [33]

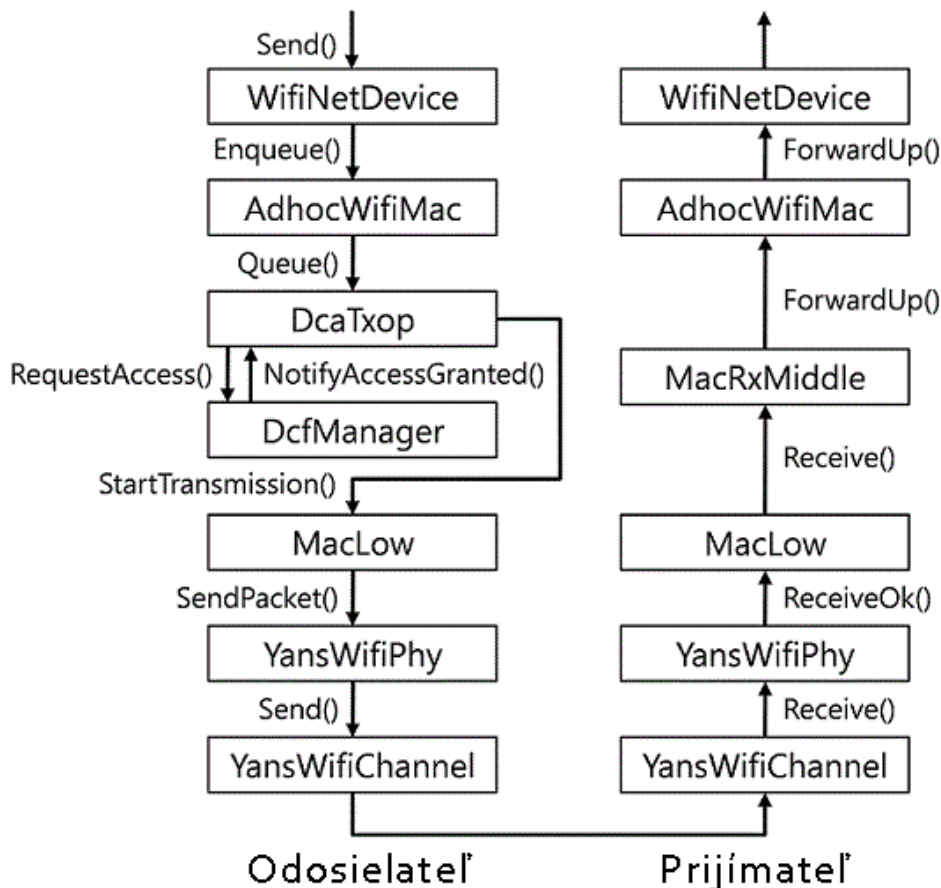
### 5.2.1 IEEE 802.11 (Wi-Fi)

Simulátor ns-3 podporuje vytváranie sietí oboch najpoužívanejších typov Ad-Hoc aj Infrastructure. Bezdrôtové zariadenia sa tvoria modelom WifiNetDevice. Ten sa delí na nasledovné úrovne:

- **PHY** – fyzická vrstva prenosu. Na tejto vrstve sú implementované prenosové štandardy IEEE 802.11a/b/g a pripravuje sa aj implementácia štandardu IEEE 802.11n. Takisto sú na tejto vrstve aj modely straty signálu - Nakagami, Rayleigh, Friis, LogDistance, FixedRss, náhodný a dva modely oneskorenia – založený na vzdialenosti a náhodný.
- **Nižšia MAC vrstva** – používa rôzne algoritmy na kontrolu prenosovej rýchlosti, tie sú však radené ako samostatná úroveň WifiNetDevice modelu. Nižšia MAC vrstva je rozdelená na tri menšie časti:
  - **MacLow** - stará sa o správy typu RTS (Request to Send - žiadosť o vysielanie), CTS (Clear to Send - linka pripravená na vysielanie),

- DATA (dátový rámec) a ACK (Acknowledgement - potvrdzovací rámec),
- **DcfManager** - implementuje funkcie DCF (Distributed Coordinated Function - distribuovaná koordinačná funkcia) a EDCA (Enhanced distributed channel access - rozšírený distribuovaný prístup ku kanálom, súčasť štandardu IEEE 802.11e). Tieto štandardy zabezpečujú koordináciu a v prípade EDCA aj prioritizovanie časovo citlivých vysielaných dát,
  - **DcaTxop / EdcaTxopN** - obsahuje zásobník rámcov a stará sa o ich fragmentáciu a znovuposielanie. Na tejto vrstve by mal byť v čoskoro implementovaný aj model 802.11ext z ns-2.[24, 30, 31, 33]
- **Vyššia MAC vrstva** – táto vrstva môže byť reprezentovaná jedným z troch modelov:
    - **AP** – prístupový bod,
    - **STA** – stanica, ktorá nie je AP,
    - **STA in IBSS** – zariadenie, ktoré nie je AP a patrí do Independent Basic Service Set siete, nazývanej aj ad-hoc.
  - **Algoritmy na kontrolu prenosovej rýchlosti** – slúžia len nižšej MAC vrstve

Modelu WifiNetDevice sa odovzdávajú modely prvých troch úrovní priamo a posledná cez ďalší, globálny model. Táto komunikácia je zobrazená na obrázku Obr. 5.4.[24]



Obr. 5.4: Príklad komunikácie vo WifiNetDevice modeli [24]

### 5.2.2 IEEE 802.16 (WiMAX)

Technológia WiMAX má v ns-3 dva základné modely – `BaseStationNetDevice` (základňová stanica) a `SubscriberStationNetDevice` (klientská stanica). Ako pri ostatných technológiách prenosu v ns-3, aj táto je rozdelená na viaceré modely.

- **Prenosový kanál** je v tomto prípade reprezentovaný modelom `WimaxChannel` alebo `SimpleOFDMWimaxChannel`, ktorý ho rozširuje. V prípade, že zariadenie vyšle rámec, kanál vyráta oneskorenie a stratu signálu.
- **Fyzický model** má v prípade WiMAX základnú úlohu dopraviť zhuk rámecov z kanálu na MAC vrstvu a naopak. Taktiež vyráta hodnotu šumu a podľa toho zisťuje, či je schopný správne dekódovať dáta.
- **MAC model** sa delí na podvrstvu konvergenencie (CS - convergence sublayer), spoločnú MAC podvrstvu (MAC-CPS - MAC Common Part Sublayer) a fyzickú vrstvu (PHY - physical layer). CS podvrstva je zodpovedná za prijímanie rámcov z vyšších vrstiev, priradovanie rámcov k správnym komunikáciám. Na správne zaradenie prichádzajúcich rámcov

používa triedu IP klasifikátor. Spoločná MAC podvrstva vykonáva základnú MAC adresáciu a implementuje režim Point-to-Multipoint (jeden uzol komunikuje s viacerými). Tento režim využívajú základňové stanice na spravovanie klientských staníc. MAC model tiež zabezpečuje službu QoS (Quality of Service) na prioritizovanie doručovania rámcov podľa harmonogramu doručovania. Tento harmonogram slúži na zabezpečenie prenosu dát od klientov s minimálnym množstvom kolízií. [23]

### 5.2.3 LTE

Vývoj simulácie inteligentných sietí LTE v simulátore ns-3 prebieha v rámci projektu LENA (LTE-EPC Network Simulator). V máji 2012 sa LENA oficiálne pričlenil k ns-3. Cieľom projektu je umožniť vývojárom LTE zariadení overiť ich kompatibilitu s už existujúcimi zariadeniami. Aj napriek tomu, že je už mnoho vecí implementovaných, viacero základných vecí ešte nie je úplne dokončených. Preto táto technológia zatiaľ nie je vhodná na použitie v našom projekte. [30]

### 5.2.4 VANET

Ďalšia zaujímavá technológia v štádiu vývoja pre ns-3. Jedná sa o automobilovú ad-hoc sieť, kde sa automobily na maximálnu vzdialenosť 300m spájajú do siete a vzájomne komunikujú pre zvýšenie bezpečnosti cestnej premávky. V súčasnosti prebieha neverejný vývoj modelu pre ns-3 a tak štádium dokončenosti nie je známe. Táto technológia preto taktiež nie je vhodná na použitie v našom projekte. [30]

### 5.2.5 Podporné modely na simulovanie bezdrôtových sietí

#### Anténa

Tento model môže byť použitý pri simulácii viacerých typov bezdrôtových sietí. Slúži na usmernenie signálu. Štandardne, ak nie je aplikovaný, tak sa bezdrôtové zariadenia správajú ako by mali všesmerovú anténu.

#### Mobilita

Umožňuje získať aktuálnu pozíciu v priestore, získať vzdialenosť od zadaného bodu (uzla), zadať uzlu konštantnú rýchlosť alebo konštantné zrýchlenie, povinné body, a taktiež mu prikázať náhodný pohyb po ploche. Využíva karteziánsku súradnicovú sústavu na zápis polohy v priestore.

#### Rušenie

Ďalší model spomedzi vyvíjaných. Simuluje rušenie bezdrôtového signálu a je určený hlavne na štúdium možností jeho zmiernenia. Je to zaujímavá možnosť, keďže v dnešnej dobe bezdrôtové siete veľmi často zápasia s rôznymi rušeniami znižujúcimi ich funkčnosť. Tento model je ešte stále vo vývoji, ale má

verejne dostupné viaceré metódy. Preto by eventuálne mohol byť zaujímavým rozšírením nášho projektu.[30, 27]

## 5.3 Vstupy a výstupy programu

### 5.3.1 Vstupy

Vstupom do simulátora ns-3 sú topológie napísané v jazyku C++ alebo Python. Tieto topológie tvoria vykonateľný program. Jednotlivé modely simulátoru ns-3 sú volané prostredníctvom hlavičkových súborov. Modelom sa priradujú parametre pri inicializácii. Tým sa definuje ich správanie.[24, 31]

#### C++

Nasledujúca ukážka nám ukáže, ako napísať v C++ jednoduchú topológiu, kde jeden bezdrôtový prístupový bod vysiela druhému UDP datagramy. Príklad vstupnej topológie [24]:

```
// hlavičky programu ns-3 pridávajúce programu funkcionalitu
#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;
int main (int argc, char *argv[])
{
    // vytvorenie troch sieťových uzlov
    NodeContainer wifiNodes;
    wifiNodes.Create (3);
    YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default ();
    *YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default();
    *wifiPhy.SetChannel (wifiChannel.Create ());
    *WifiHelper wifi = WifiHelper::Default ();
    *NqosWifiMacHelper wifiMac = NqosWifiMacHelper::Default ();
    *wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager");
    *wifiMac.SetType("ns3::AdhocWifiMac");
    *NetDeviceContainer wifiDevices = wifi.Install(WifiPhy,wifiMac,wifiNodes);

    // nastavenie statických pozícií
    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::ConstantPositionMobilityModel");
mobility.Install (wifiNodes);

// vytvorenie IP siete
InternetStackHelper internet;
internet.Install (wifiNodes); Ipv4AddressHelper address;
address.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer wifiInterfaces;
wifiInterfaces = address.Assign( wifiDevices );

// vytvorenie UDP generátoru
UdpEchoServerHelper echoServer (9);
ApplicationContainer serverApps = echoServer.Install(wifiNodes.Get(0));
serverApps.Start (Seconds (1.0));
serverApps.Stop (Seconds (10.0));
UdpEchoClientHelper echoClient (wifiInterfaces.GetAddress (0), 9);
echoClient.SetAttribute ("MaxPackets", UIntegerValue (2));
echoClient.SetAttribute ("Interval", TimeValue (Seconds (1.)));
echoClient.SetAttribute ("PacketSize", UIntegerValue (1024));
ApplicationContainer clientApps = echoClient.Install(wifiNodes.Get(2));
clientApps.Start( Seconds (2.0) );
clientApps.Stop( Seconds (10.0) );

// naplnenie smerovacích tabuliek údajmi
Ipv4GlobalRoutingHelper::PopulateRoutingTables ();

// spustenie samotnej simulácie
Simulator::Stop (Seconds (5.05));
Simulator::Run ();
Simulator::Destroy ();
return 0;
}
```

## Python

Ako sme spomínali v kapitole 3.1, vstupné Python skripty sú pomocou knižnice Pybindgen prekladané z jazyka Python do jazyka C++. Tým pádom je výstup programu rovnaký pri oboch spôsoboch zápisu simulácie. Príklad vstupnej topológie:



```
// hlavičky programu ns-3 pridávajúce programu funkcionality
import ns.applications
import ns.core
import ns.internet
import ns.network
import ns.point_to_point

ns.core.LogComponentEnable("UdpEchoClientApplication", ns.core.LOG_LEVEL_INFO)
ns.core.LogComponentEnable("UdpEchoServerApplication", ns.core.LOG_LEVEL_INFO)

// vytvorenie sieťových uzlov
nodes = ns.network.NodeContainer()
nodes.Create(2)

// sieťové spojenie medzi uzlami
pointToPoint = ns.point_to_point.PointToPointHelper()
pointToPoint.SetDeviceAttribute("DataRate", ns.core.StringValue("5Mbps"))
pointToPoint.SetChannelAttribute("Delay", ns.core.StringValue("2ms"))

devices = pointToPoint.Install(nodes)

stack = ns.internet.InternetStackHelper()
stack.Install(nodes)

// nastavenie sieťovej adresácie
address = ns.internet.Ipv4AddressHelper()
address.SetBase(ns.network.Ipv4Address("10.1.1.0"), ns.network.Ipv4Mask("255.255.255.0"))

interfaces = address.Assign (devices);

// vytvorenie aplikačnej vrstvy
echoServer = ns.applications.UdpEchoServerHelper(9)

serverApps = echoServer.Install(nodes.Get(1))
serverApps.Start(ns.core.Seconds(1.0))
serverApps.Stop(ns.core.Seconds(10.0))

echoClient = ns.applications.UdpEchoClientHelper(interfaces.GetAddress(1), 9)
echoClient.SetAttribute("MaxPackets", ns.core.UintegerValue(1))
echoClient.SetAttribute("Interval", ns.core.TimeValue(ns.core.Seconds (1.0)))
echoClient.SetAttribute("PacketSize", ns.core.UintegerValue(1024))

clientApps = echoClient.Install(nodes.Get(0))
clientApps.Start(ns.core.Seconds(2.0))
clientApps.Stop(ns.core.Seconds(10.0))
```

```
//spustenie samotnej simulácie
ns.core.Simulator.Run()
ns.core.Simulator.Destroy()
```

### Kompilácia vstupnej topológie [34]:

```
./waf --shell
python /path/to/your/example/my-script.py
```

alebo

```
./waf -run path-to-topology
```

### 5.3.2 Výstupy

Výstup programu môže byť smerovaný buď do súboru, alebo do konzolového okna. Výstup do súboru podporuje na rozdiel od ns-2 formát .pcap, ktorý sa dá otvoriť v sieťovom analyzátoe Wireshark (Ethereal). Priamo podporuje aj výstupný formát .tr.[26]

#### ASCII Trace .tc

Formát podobný výstupu z ns-2 simulátora opísanému v kapitole 2.8 . Pri každej udalosti používa rozlišovacie značky „+“ (pridanie rámcu do zásobníku), „-“ (odobratie rámcu zo zásobníku), „d“ (zahodenie rámcu) alebo „r“ (prijatie rámcu na sieťovom uzle). Ďalej sa v tomto výpise nachádzajú detaily samotného rámcu ako napríklad použité protokoly, IP adresy alebo aplikačné porty. Syntax potrebná na výstupe do súboru vo formáte .tr[36]:

```
AsciiTraceHelper ascii;
pointToPoint.EnableAsciiAll(ascii.CreateFileStream("outputFile.tr"));
```

Príklad výstupného súboru vo formáte .tr [35]:

```
r
2.25732
/NodeList/1/DeviceList/0/$ns3::PointToPointNetDevice/MacRx
  ns3::Ipv4Header (
    tos 0x0 ttl 64 id 0 protocol 17 offset 0 flags [none]
    length: 1052 10.1.1.1 > 10.1.1.2)
  ns3::UdpHeader (
    length: 1032 49153 > 9)
  Payload (size=1024)
```

#### Wireshark .pcap

Formát .pcap je natívnym formátom v súčasnosti najpoužívanejšieho sieťového analyzátoe WireShark. Tento formát sa dá využiť na následné zobrazenie

rámcov a ich manuálnu analýzu v prípade sledovania špecifickej komunikácie. V našom prípade vieme použiť konzolový analyzátor TShark na vytvorenie grafov. Dá sa v ňom sledovať napríklad vyťaženie média (v našom prípade jednotlivých kanálov) alebo sledovať a zobrazovať jednotlivé komunikácie.

Potrebná syntax na výstupe do súboru vo formáte .pcap[25, 35]:

```
pointToPoint.EnablePcapAll ("outputFile");
```

Príklad výstupného súboru vo formáte .pcap [35]:

```
2.257324 IP 10.1.1.1.49153 > 10.1.1.2.9: UDP, length 1024
```

```
2.257324 IP 10.1.1.2.9 > 10.1.1.1.49153: UDP, length 1024
```

## XML

XML nie je natívny formát výstupu ns-3. Je umožnený modelom AnimationInterface, presnejšie jeho metódou SetXMLOutput(). Výstupný súbor oddeľuje topológiu, rámce na káblovej sieti a rámce na bezdrôtovej sieti. Potrebná syntax na výstupe do súboru vo formáte .xml [29, 27]:

```
#include "ns3/netanim-module.h"
std::string animFile = "output.xml" ;
AnimationInterface anim (animFile);
```

## 5.4 Podporné nástroje pre ns-3

### 5.4.1 Inet: Internet Topology Generator

#### Popis

Inet je voľne šíriteľný program vyvíjaný na Univerzite Michigan. Je schopný generovať topológie na úrovni autonómnych systémov. Súčasťou vygenerovanej topológie sú len informácie o konektivite uzlov, nenachádzajú sa tu informácie o oneskoreniach, šírkach pásiem jednotlivých liniek a podobne. Vygenerovaná topológia má charakteristiku reálnej internetovej topológie podobného rozsahu z obdobia 1997-2000 a neskôr. Program je určený na generovanie topológií s minimálnym počtom uzlov 3037. [21]

#### Použitie

Syntax príkazu:

```
inet -n N [-d k] [-p n] [-s sd] [-f of]
```

Parametre:

**-n** <N> - počet uzlov v topológii

**-d** <k> - podiel uzlov prvého stupňa (predvolené je 0,3)

**-p** <n> - veľkosť plochy topológie (predvolené je 10000)

**-s** <sd> - inicializačná hodnota generátora náhodných čísiel (predvolené je 0)

**-f** <of> - názov výstupného ladiaceho súboru (predvolené je *stderr*)

Príklady použitia:

1. Vygenerovanie topológie o veľkosti 6000 uzlov s použitím predvolených hodnôt, výstup bude uložený v súbore s názvom *Inet.6000*

```
inet -n 6000 > Inet.6000
```

2. Vygenerovanie topológie s parametrami:

- počet uzlov 7000,
- podiel uzlov prvého stupňa 25 percent,
- veľkosť plochy 15000 x 15000 bodov,
- inicializačná hodnota generátora náhodných čísiel 16,
- názov výstupného ladiaceho súboru *debug*,
- výstup programu uložený v súbore *Inet.7000*.

```
inet -n 7000 -d .25 -p 15000 -s 16 -f debug > Inet.7000
```

Formát výstupného súboru vyzerá takto:

```
nodes links
...
id x y
...
id1 id2 weight
```

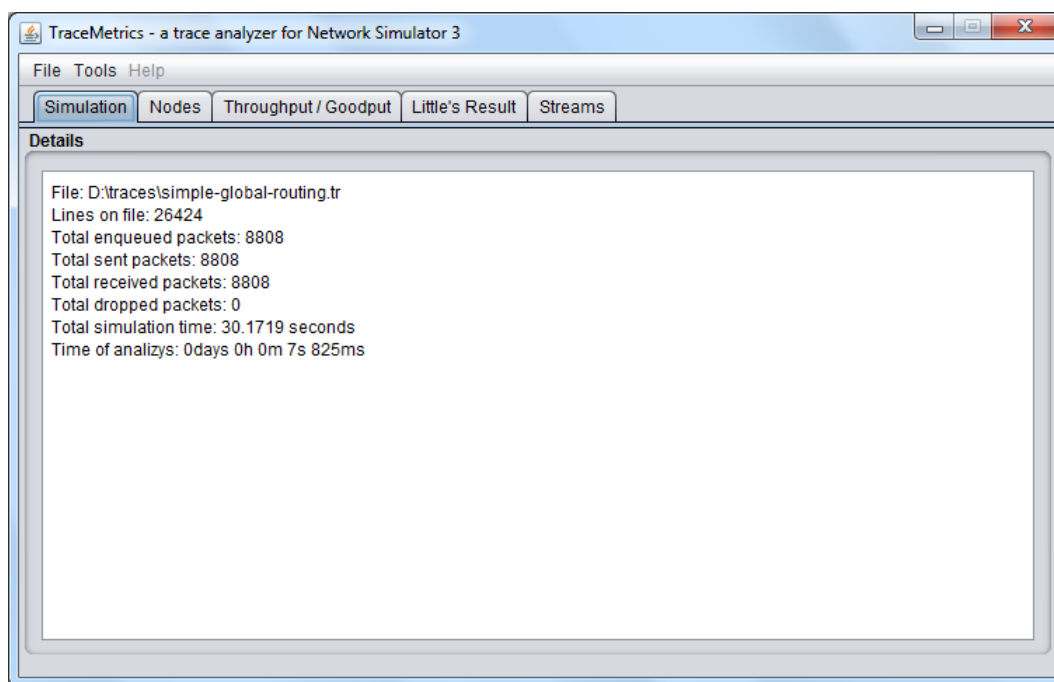
Prvý riadok obsahuje dve čísla. Hodnota *nodes* predstavuje celkový počet uzlov v topológii, hodnota *links* predstavuje celkový počet liniek. V ďalšej časti sa nachádza umiestnenie jednotlivých uzlov, každý riadok obsahuje identifikátor uzla (*id*) a jeho x-ovú a y-ovú pozičnú súradnicu (*x y*). V poslednej časti sú definované linky topológie. Každá linka je definovaná identifikátormi uzlov, ktoré prepájajú (*id1 id2*) a parametrom váha linky (*weight*).

### 5.4.2 TraceMetrics

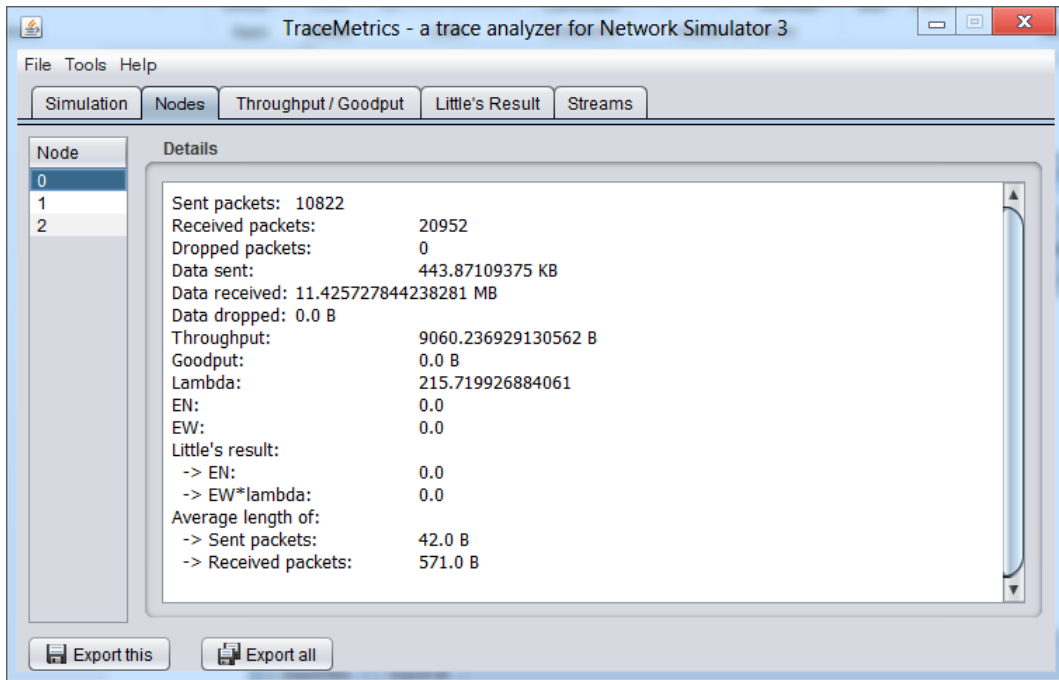
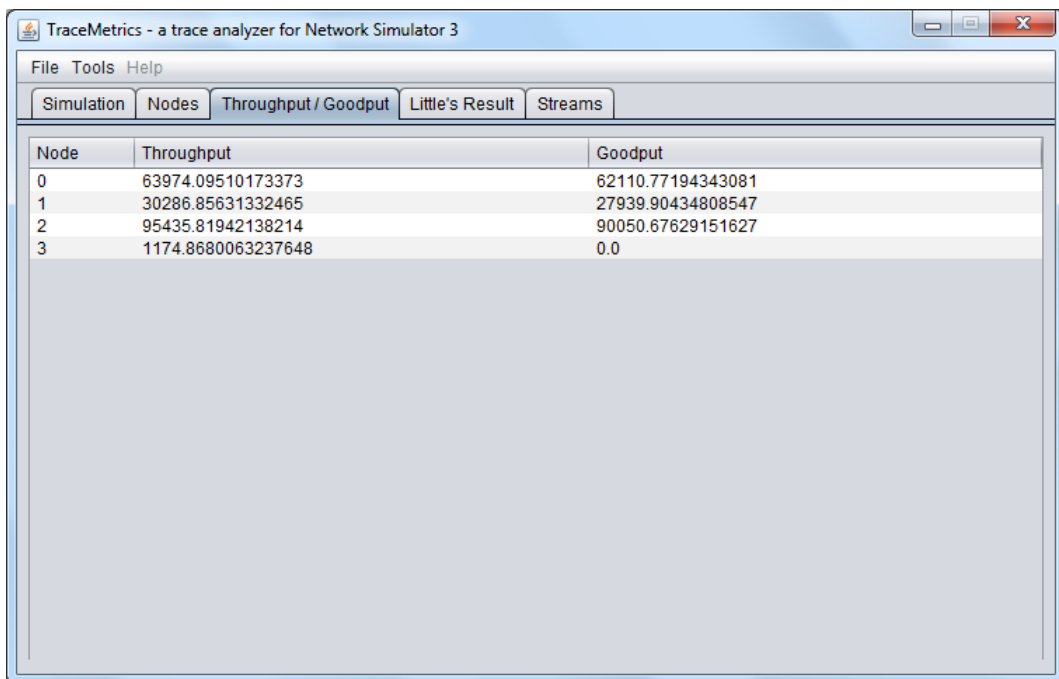
TraceMetrics je analyzátor výstupných simulačných súborov programu ns-3. Jedná sa o voľne šíriteľný program s otvoreným zdrojovým kódom (opensource) vyvíjaný v jazyku Java. Program poskytuje nasledovné funkcie [22]:

- všeobecné informácie o analyzovanej simulácii ako napr. cesta k výstupnému simulačnému súboru programu ns-3, počet riadkov tohto súboru, počítadlá paketov, simulačný čas (obrázok Obr. 5.5),

- detailné informácie o každom uzle ako napr. počet prijatých, odoslaných a zahodených paketov, priepustnosť, priemerné hodnoty niektorých sledovaných parametrov (obrázok Obr. 5.6),
- porovnanie celkovej a relatívnej priepustnosti jednotlivých uzlov (obrázok Obr. 5.7),
- porovnanie priemernej dĺžky radu paketov, priemerného času pobytu paketov v uzle a ďalších parametrov jednotlivých uzlov (obrázok Obr. 5.8),
- detailné informácie o TCP a UDP tokoch v simulácii až na úroveň analýzy jednotlivých paketov každého toku (obrázky Obr. 5.9 a Obr. 5.10),
- možnosť vygenerovania skriptu pre program gnuplot na grafické zobrazenie vybraných informácií,
- možnosť exportovať vybrané informácie (ako napr. podrobné informácie o uzloch alebo informácie o TCP a UDP tokoch).



Obr. 5.5: Okno programu TraceMetrics, záložka *Simulation* [22]

Obr. 5.6: Okno programu TraceMetrics, záložka *Nodes* [22]Obr. 5.7: Okno programu TraceMetrics, záložka *Throughput/Goodput* [22]

Node	Lambda	E[W]	E[N]	E[W] * Lambda
0	62.110771943430805	0.0	0.0	0.0
1	55.87980869617094	0.0024142348754408317	0.13490698298748047	0.13490698298725776
2	145.9636284092152	0.06184400544959394	9.026975430781645	9.02697543078201
3	27.973047769613448	0.0	0.0	0.0

Obr. 5.8: Okno programu TraceMetrics, záložka *Little's Result* [22]

Stream	IPs	Ports	Number of packets	Average Delay	Delay variance	Average PDV	PDV variance	Average IPDV	IPDV variance
Stream 0	10.1.1.2 --> 10.1.1.1	49153 --> 50000	10476	0.006574556128293023	1.6743029689409555E-8	0.004234556128294001	1.6741431462058523E-8	4.162291169453117E-7	6.020867326114095E-9
Stream 1	10.1.2.2 --> 10.1.2.1	49153 --> 50000	10476	0.006574556128293023	1.6743029689409555E-8				

Obr. 5.9: Okno programu TraceMetrics, záložka *Streams* [22]

Sequence number	ACK number	Delay	PDV	IPDV
0	0	0.02880000000000038	0.001500000000000568	L
1	1	0.027300000000000324	0.0	-0.001500000000000568
1	1	0.03160000000000096	0.004300000000000637	0.004300000000000637
537	1	0.02929999999999216	0.001999999999988916	-0.0023000000000001745
1073	1	0.02929999999999216	0.001999999999988916	0.0
1609	1	0.03639999999998656	0.00909999999998332	0.00709999999999944
2145	1	0.03639999999998656	0.00909999999998332	0.0
2681	1	0.03650000000000002	0.00919999999999875	1.000000000015433E-4
3217	1	0.02929999999999216	0.001999999999988916	-0.007200000000000983
3753	1	0.03749999999999645	0.0101999999999932	0.008200000000000043
4289	1	0.0373999999999988	0.01009999999999554	-9.99999999976694E-5
4825	1	0.0367999999999995	0.00949999999999176	-6.00000000000378E-4
5361	1	0.0367999999999995	0.00949999999999176	0.0
5897	1	0.0367999999999995	0.00949999999999176	0.0
6433	1	0.0413999999999944	0.01409999999999113	0.00459999999999375
6969	1	0.0413999999999944	0.01409999999999113	0.0
7505	1	0.0413999999999944	0.01409999999999113	0.0
8041	1	0.02930000000000092	0.002000000000000668	-0.012099999999998445
8577	1	0.03749999999999645	0.0101999999999932	0.008199999999998653
9113	1	0.0373999999999988	0.01009999999999554	-9.99999999976694E-5
9649	1	0.04210000000000136	0.014800000000001035	0.0047000000000001481
10185	1	0.04210000000000136	0.014800000000001035	0.0
10721	1	0.0421999999999935	0.01489999999999025	9.999999999799059E-5
11257	1	0.0420999999999958	0.01479999999999258	-9.99999999976694E-5
11793	1	0.05030000000000001	0.02299999999999687	0.008200000000000043

Obr. 5.10: Okno programu TraceMetrics, ukážka detailného zobrazenia toku [22]

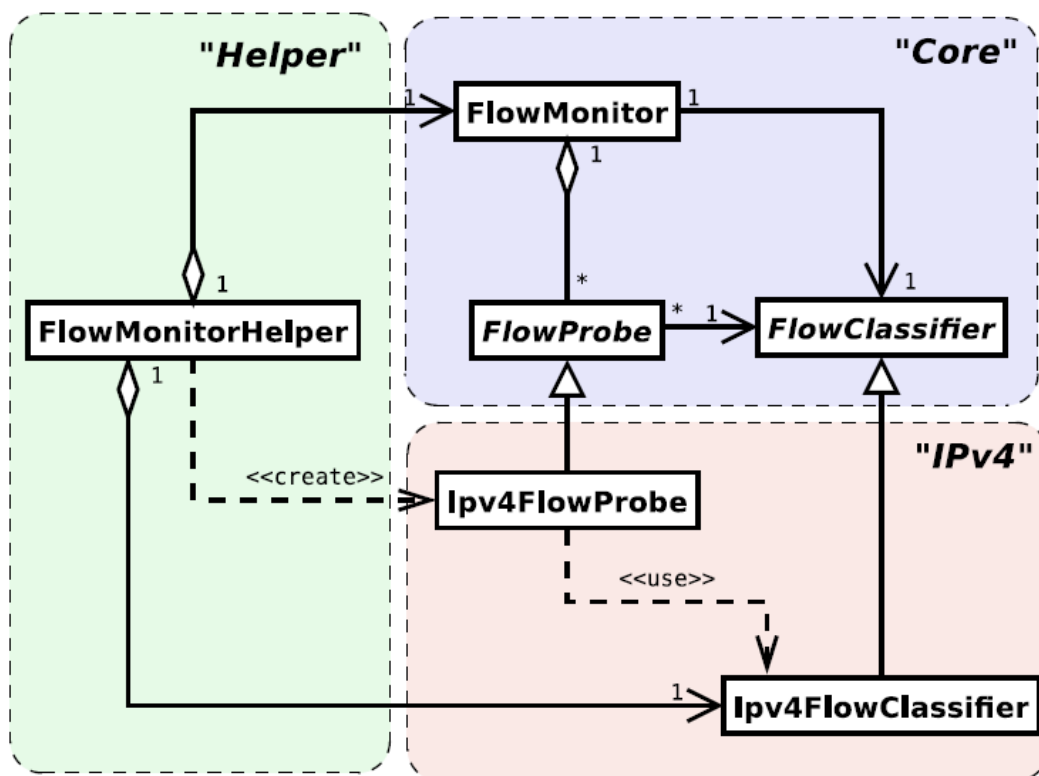
### 5.4.3 FlowMonitor

FlowMonitor je monitorovacie rozhranie sietí simulovaných v ns-3. Toto rozhranie slúži na sledovanie jednotlivých tokov v simulácii a zo zachytených údajov je schopné vytvoriť štatistiky sledovaním veľkého množstva parametrov (oneskorenie, jitter, straty, atď.). Výhody rozhrania FlowMonitor sú [10]:

- **jednoduchosť použitia:** aktivácia len zopár riadkami v skripte simulácie, automatická detekcia tokov, minimálna konfigurácia,
- **vyvážené množstvo zachytených údajov:** správne množstvo zachytených údajov pre vytvorenie korektných štatistík,
- **výstupné súbory sú jednoducho spracovateľné:** výstup môže byť uložený do XML súboru alebo SQL databázy,
- **rozšíriteľné:** umožňuje vytvoriť rozšírenia pre monitorovanie iných typov tokov,
- **minimalizácia vyťaženia CPU a pamäte.**

Architektúra rozhrania je organizovaná do troch tried: Helper, Core, IPv4. Táto štruktúra je zobrazená na obrázku Obr. 5.11.





Obr. 5.11: Architektúra rozhrania FlowMonitor [10]

### Použitie

FlowMonitor sa aktivuje niekoľkými jednoduchými príkazmi v skripte simulácie. Príklad aktivácie rozhrania je nasledovný [10]:

```
flowmon_helper = ns3.FlowMonitorHelper( )
monitor = flowmon_helper.InstallAll( )
monitor.SetAttribute( "DelayBinWidth", ns3.DoubleValue(0.001))
monitor.SetAttribute( "JitterBinWidth", ns3.DoubleValue(0.001))
monitor.SetAttribute( "PacketSizeBinWidth", ns3.DoubleValue(20))
ns3.Simulator.Run( )
monitor.SerializeToXmlFile("results.xml", True, True)
```

Kód vykonáva nasledujúce kroky:

1. Vytvorí sa FlowMonitorHelper objekt.
2. Zavolá sa metóda InstallAll na tento objekt. Výsledkom je vytvorenie monitoru tokov, ktorý je nastavený na monitorovanie IPv4 na všetkých uzloch v simulácii.
3. Nakonfigurovanie niekoľkých parametrov výsledného histogramu.
4. Spustenie simulácie.
5. Zápis výsledkov do XML súboru *results.xml*. Druhý parameter funkcie SerializeToXmlFile indikuje, či chceme uložiť aj histogram, a tretí parameter označuje, či chceme uložiť aj per-probe štatistiky.

Pre vykreslenie grafov je nutné napísať skript, ktorý prečíta údaje z XML a vytvorí histogram. Tento program je možné vytvoriť napríklad v jazyku Python. Príklad takéhoto skriptu je nasledovný [10]:

```

et=ElementTree.parse(sys.argv[1])
bitrates=[]
losses=[]
delays=[]
for flow in et.findall("FlowStats/Flow"):
    #filter out OLSR
    for tpl in et.findall("Ipv4FlowClassifier/Flow"):
        if tpl.get('flowId')==flow.get('flowId'):
            break
        if tpl.get("destinationPort")== '698':
            continue

losses.append(int(flow.get('lostPackets')))

rxPackets=int(flow.get('rxPackets'))
if rxPackets==0:
    bitrates.append(0)
else:
    t0=long(flow.get('timeFirstRxPacket')[:-2])
    t1=long(flow.get('timeLastRxPacket')[:-2])
    duration=(t1-t0)*1e-9
    bitrates.append(8*long(flow.get('rxBytes'))/duration*1e-3)
    delays.append(float(flow.get('delaySum')[:-2])*1e-9/rxPackets)

pylab.subplot(311)
pylab.hist(bitrates,bins=40)
pylab.xlabel("Flowbitrate(bit/s)")
pylab.ylabel("Numberofflows")

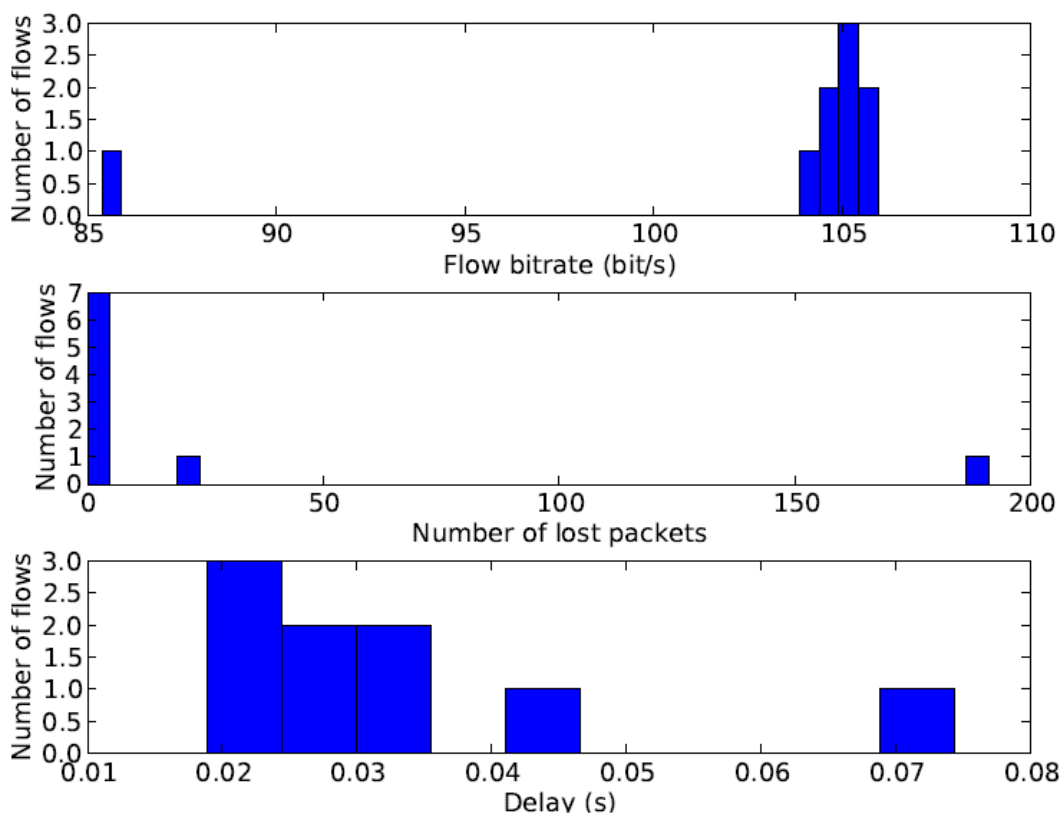
pylab.subplot(312)
pylab.hist(losses,bins=40)
pylab.xlabel("Numberoflostpackets")
pylab.ylabel("Numberofflows")

pylab.subplot(313)
pylab.hist(delays,bins=10)
pylab.xlabel("Delay(s)")
pylab.ylabel("Numberofflows")

pylab.subplotsadjust(hspace=0.4)
pylab.savefig("results.pdf")

```

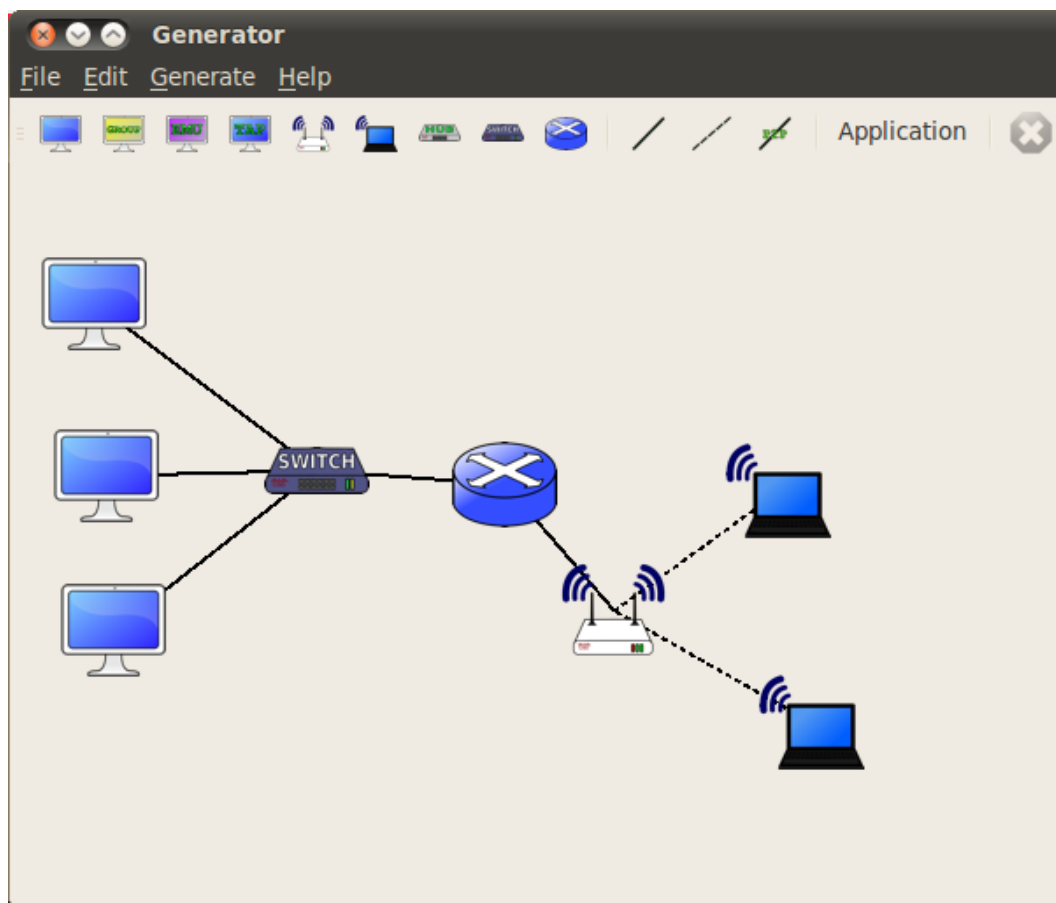
Grafy získané podobným skriptom sú zobrazené na obrázku Obr 5.12.



Obr. 5.12: Histogramy získané pomocou FlowMonitor [10]

#### 5.4.4 NS3Generator

NS3Generator je program zameraný na poskytnutie rýchlej a efektívnej metódy vytvorenia ns-3 topológie. Poskytuje intuitívne GUI, pomocou ktorého je možné pridávať uzly, linky, aplikácie. Program využíva platformu Qt4 a podporuje pevné aj bezdrôtové prvky. Výstupom je C++ simulačný skript, ktorý sa použije ako vstup pri sputení simulácie. Vytvorenú topológiu je možné uložiť do XML súboru. Rozhranie tohto simulátora je zobrazené na obrázku Obr. 5.13 [11].



Obr. 5.13 - Rozhranie programu NS3Generator [11]

#### 5.4.5 NetAnim

NetAnim je vizualizačná nadstavba nad ns-3. Je postavený na multiplatformovom vývojovom balíku Qt 4 s grafickým balíkom QGraphics. Nástroj NetAnim je momentálne vo svojej tretej verzii. NetAnim je použiteľný na drôtové aj bezdrôtové siete. Umožňuje vytvárať štatistiky prenesených rámcov (obrázok Obr. 5.14) aj s filtrovaním. Taktiež vie vizualizovať dráhy pohyblivých uzlov (obrázok Obr. 5.15). Z hľadiska vizualizácie je podobný zameraniu nášho projektu.

	Tx Time	From Node Id	To Node Id	
1	2.5e-05	0	5	Wifi MGT_BEACON FromDS: 0 toDS: 0 DA: ff:ff:ff:ff:ff:ff
2	2.5e-05	0	6	Wifi MGT_BEACON FromDS: 0 toDS: 0 DA: ff:ff:ff:ff:ff:ff
3	2.5e-05	0	7	Wifi MGT_BEACON FromDS: 0 toDS: 0 DA: ff:ff:ff:ff:ff:ff
4	0.000167033	5	6	Wifi MGT_ASSOCIATION_REQUEST FromDS: 0 toDS: 0 DA: ff:ff:ff:ff:ff:ff
5	0.000167033	5	7	Wifi MGT_ASSOCIATION_REQUEST FromDS: 0 toDS: 0 DA: ff:ff:ff:ff:ff:ff
6	0.000167033	5	0	Wifi MGT_ASSOCIATION_REQUEST FromDS: 0 toDS: 0 DA: ff:ff:ff:ff:ff:ff
7	0.000279066	0	5	Wifi CTL_ACK RA:00:00:00:00:00:07
8	0.000279066	0	6	Wifi CTL_ACK RA:00:00:00:00:00:07
9	0.000279066	0	7	Wifi CTL_ACK RA:00:00:00:00:00:07
10	0.000402103	6	5	Wifi MGT_ASSOCIATION_REQUEST FromDS: 0 toDS: 0 DA: ff:ff:ff:ff:ff:ff
11	0.000402103	6	0	Wifi MGT_ASSOCIATION_REQUEST FromDS: 0 toDS: 0 DA: ff:ff:ff:ff:ff:ff
12	0.00051414	0	5	Wifi CTL_ACK RA:00:00:00:00:00:08
13	0.00051414	0	6	Wifi CTL_ACK RA:00:00:00:00:00:08
14	0.00051414	0	7	Wifi CTL_ACK RA:00:00:00:00:00:08

Obr. 5.14: Štatistické zobrazenie rámcov [29]

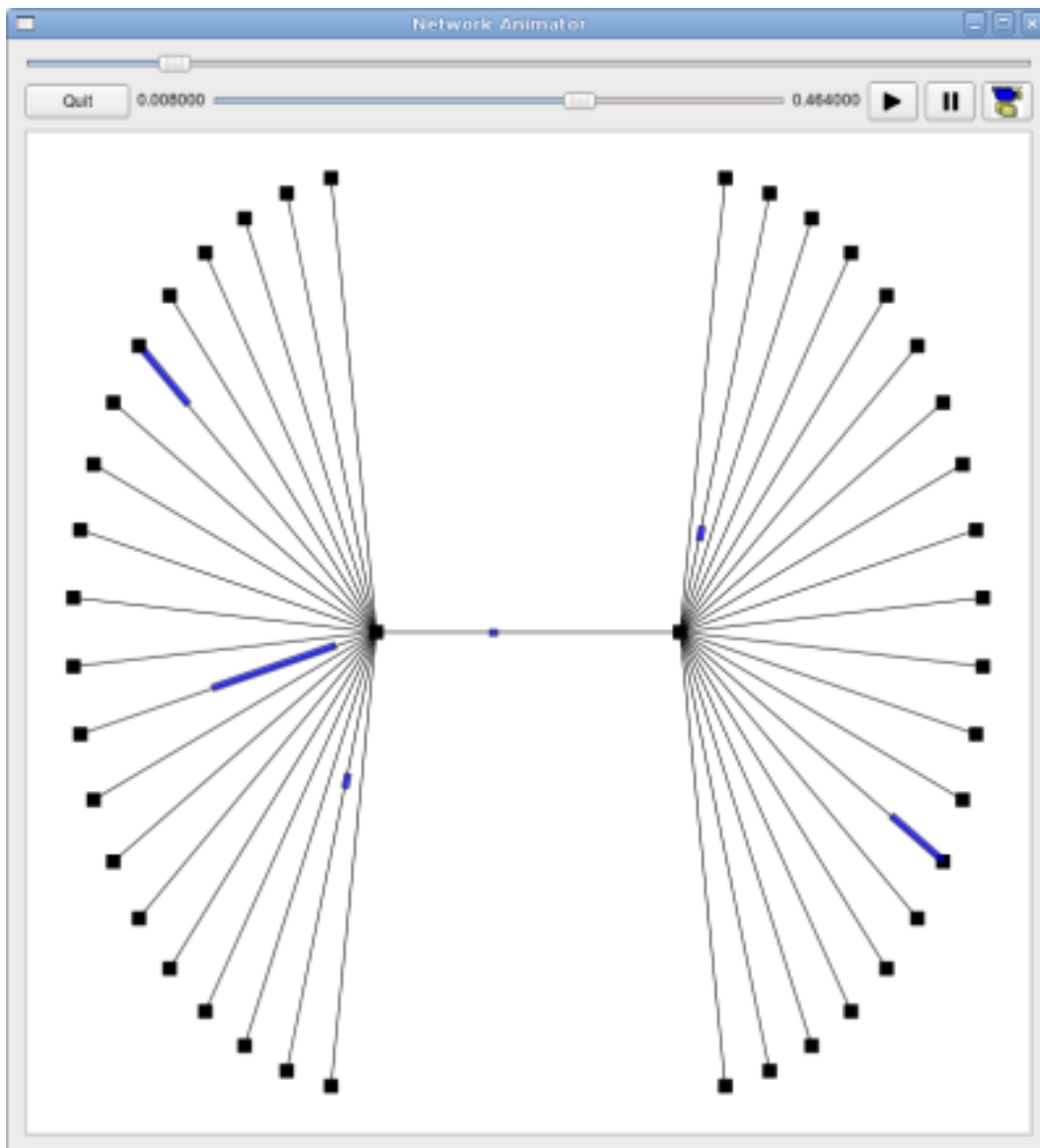
V prvom kroku tento nástroj spustí zvolenú simuláciu a spracuje výstupy do XML podoby. Tie následne vizualizuje. XML výstupy vytvára za použitia modelu ns3::AnimationInterface, ktorý je priamo integrovaný v ns-3. Príklad priradenia AnimationInterface potrebného na vytvorenie XML formátovaných výstupov:

```
AnimationInterface anim;
if (port > 0) {
    anim.SetServerPort (port);
}
else if (!animFile.empty ()) {
    anim.SetOutputFile (animFile);
}
anim.StartAnimation ();
```

#### Štruktúra XML súboru:

1. Topológia
  - Uzly
  - Linky
2. Rámce prenesené drôtovými spojeniami
3. Rámce prenesené bezdrôtovými spojeniami

V druhom kroku sú zozbierané informácie zobrazené s rôznymi možnosťami nastavenia rýchlosti prehrávania a s možnosťou zastavenia prehrávania. [29]



Obr. 5.15: Ukažka vizualizácie pomocou NetAnim [29]

XML výstupy používané v tomto nástroji by sme mohli využiť aj v našom programe, ak by sme sa tak rozhodli. Iné jeho časti nie sú pre nás príliš zaujímavé z iného ako inšpiračného hľadiska.

## 6 Zhodnotenie analýzy

V analýze sme hľadali odpovede na tri hlavné otázky:

- Budeme rozširovať systém našich predchodcov?
- Budeme vytvárať nanovo vlastný systém ?
- Ktorý sieťový simulátor použijeme ?

Naša analýza pozostáva z troch hlavných častí. Každá jedna časť analýzy je veľmi dôležitá pri vytváraní nášho návrhu systému.

Prvá časť spočívala v analýze predchádzajúceho projektu, v ktorej sme sa ho snažili čo najdetailnejšie analyzovať, aby sme získali potrebné znalosti o funkcionalite celého systému. Predošlý systém je postavený na sieťovom simulátore ns-2. Ide o internetovú aplikáciu bežiacu na linuxovom serveri. Jedná sa o klient-server aplikáciu, ktorej výhoda je možnosť priebehu simulácie na lokálnom počítači používateľa. Aplikácia sa skladá z troch hlavných častí: návrh simulácie, samotná simulácia a grafický výstup simulácie. Navyše ale obsahuje aj používateľský profil.

Predchádzajúci tím implementoval veľké množstvo funkcionality od editora topológie, kde si používateľ môže sám vytvoriť a nastaviť parametre siete, cez editor akcií, kde si používateľ sám nastaví pohyb zariadení v priestore a prípadne tok dát medzi zariadeniami, až po kvalitné výstupy formou grafov, štatistík alebo video simulácie. Po analýze tohto systému sme zhodnotili, že projekt je kvalitne spracovaný po každej stránke a jeho možné vylepšenia by neboli hodné tímového projektu. Vzhľadom na to, že sieťový simulátor ns-3 nie je priamym pokračovaním ns-2, prerábanie tohto systému na použitie ns-3 by znamenalo, ako vo svojej dokumentácii spomínal aj predchádzajúci tím, veľký zásah do samotného jadra ich systému. Inak povedané, takmer nový projekt. Po analýze predchádzajúceho projektu sme sa rozhodli, že budeme vytvárať nový projekt. Z pôvodného projektu samozrejme použijeme niektoré vhodné skripty, ktoré by boli použiteľné aj v našom projekte.

V druhej a tretej časti analýzy sme sa zamerali na analýzu sieťových simulátorov ns-2 a ns-3. Aj keď predchádzajúci tím urobil tieto analýzy, je potrebné ich urobiť ešte raz vzhľadom na to, že ich robili zhruba pred dvomi rokmi. Je to dôležité hlavne kvôli simulátoru ns-3, ktorý mal v tej dobe slabú podporu, veľa chýb a aj malú komunitu ľudí okolo seba, čo bol aj dôvod, prečo predchádzajúci tím zvolil simulátor ns-2 pre projekt. Na základe týchto analýz sme

získali prehľad o rozdieloch medzi samotnými simulátormi ns-2 a ns-3. Sieťový simulátor ns-2 je po mnohých rokoch jeho existencie pomerne rozšírený a stále vychádzajú jeho aktualizácie. Je to stabilný a používaný sieťový simulátor, ktorého vstupy vieme vytvárať za pomoci jazyka C++ a Tcl skriptov. Výstup zo simulácie je iba jedným spôsobom, a to tzv. trace súborom obsahujúcim všetky dáta. Pre ns-2 existuje veľké množstvo rozširujúcich modulov, počnúc rôznymi editormi, rozšíreniami pre simuláciu rôznych typov sietí, až po rôzne nástroje spracúvajúce výstup. Stav simulátora ns-3 sa začal výrazne zlepšovať a v tejto dobe je to plne použiteľný simulátor s trojmesačnými aktualizáciami a slušnou komunitou vývojárov. Oproti ns-2 má viaceré výhody. Jeho vstupy je možné tiež vytvárať pomocou objektovo orientovaného jazyka C++, ale ako druhý spôsob je použiteľný aj jazyk Python. Projekt ns-3 sa zaviazal k budovaniu pevného simulačného jadra, ktoré je dobre zdokumentované. Simulátor ns-3 má jedny z najmenších pamäťových nárokov medzi sieťovými simulátormi. Má v sebe implementovanú podporu bezdrôtových sietí, čo je pre nás výhodou. Simulátor ns-3 dokáže priamo pracovať aj s reálnymi zariadeniami. Jeho výstupy sú obohatené oproti ns-2 o súbory typu .pcap pre známy sieťový analyzátor Wireshark a prostredníctvom jedného zo svojich modelov aj o štruktúrovaný XML formát. Rozširujúcich modulov nie je také množstvo ako pri ns-2, ale nepovažujeme to za až takú veľkú nevýhodu.

Z našej analýzy sme dospeli k záveru, že budeme vytvárať nový projekt a bude založený na novšom sieťovom simulátore ns-3.



## 7 Špecifikácia

Výstupom projektu bude jednoduché používateľské prostredie pre prácu so sieťovým simulátorom ns-3, ktoré bude vhodné aj pre ľudí bez skúseností so sieťovými simulátormi.

Prostredie bude realizované ako webová aplikácia s intuitívnym grafickým rozhraním a jednoduchým ovládaním. Používateľ si bude môcť zdefinovať simuláciu buď priamo skriptom v jazyku Python/C++, alebo prostredníctvom grafického editora topológie.

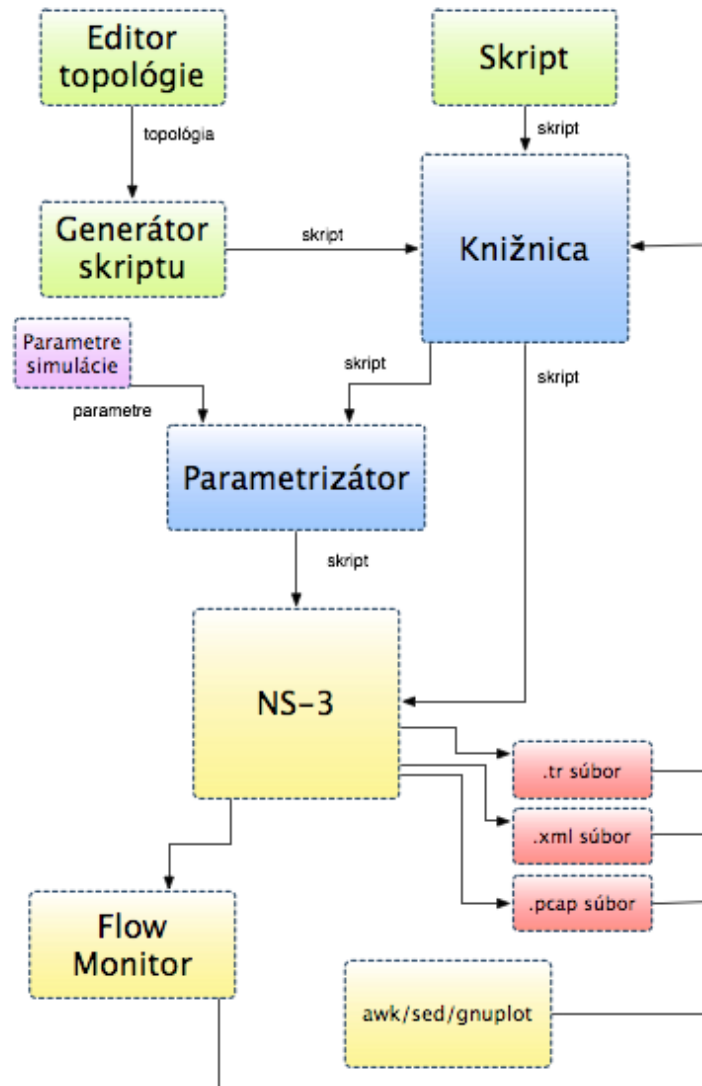
Ďalšou možnosťou vytvorenia topológie bude použitie parametrizovaného autogenerátora, ktorý automaticky navrhne topológiu a vytvorí k nej prislúchajúci skript. Vstupom autogenerátora budú hodnoty ako napríklad počet uzlov, rýchlosť uzlov, typ siete, parametre mobility, smerovací protokol.

Používateľ si bude taktiež môcť uložiť svoju topológiu do knižnice. Vďaka tomu si ostatní používatelia a aj sám autor budú môcť neskôr zobrazit topológiu alebo výsledky simulácie. Používatelia si budú môcť zdieľať medzi sebou jednotlivé knižnice.

Výstupom aplikácie bude štatistické zhodnotenie vykonanej simulácie spolu s grafickou reprezentáciou výsledkov vo forme grafov, tabuliek a manažérskych zhrnutí. Výsledky si bude môcť používateľ uložiť do knižnice pre neskoršie použitie alebo zdieľanie s inými používateľmi.

## 8 Návrh riešenia

Na obrázku Obr. 8.1 je zobrazený blokový návrh celého riešenia.



Obr. 8.1: Blokový návrh riešenia

### 8.1 Editor topológie

Editor topológie bude grafická webová aplikácia, pravdepodobne v jazyku JavaScript, prípadne Java alebo Flash. Používateľ bude môcť v nej nakresliť topológiu použitím rôznych posuvných objektov, podobne ako napríklad v Microsoft Visio. Medzi týmito objektami bude môcť kresliť drôtové / bezdrôtové spojenia, nastavovať im parametre a prípadne vektorovo programovať ich pohyb.

Výstup z editora topológie bude topológia (reprezentovaná ako neorientovaný ohodnotený graf), ktorá bude vstupom do generátora skriptu.

## 8.2 Generátor skriptu

Bude softvér, ktorý zoberie na vstupe topológiu reprezentovanú grafom, jej parametre, vektory pohybu, atď. a vytvorí z nej spustiteľný skript pre ns-3. Ešte nie je rozhodnuté, či sa bude jednať o C++ program alebo Python skript. Toto ale nie je pre aktuálne potreby nevyhnutné.

Generátor interpretuje topológiu a vygeneruje skript. Výstup generátora bude práve tento skript.

## 8.3 Skript

Používateľovi bude umožnené aj priamo vložiť do systému hotový skript (aj parametrizovateľný). Tento skript bude uložený do knižnice, aby mohol byť neskôr použitý.

## 8.4 Knižnica

Vstupom do knižnice je:

- skript - výstup z generátora skriptu alebo priamo skript vložený používateľom do systému ručne,
- .tr, .xml, .pcap súbor - tieto súbory sú výstupy simulácie a budú uchovávané v knižnici,
- výsledky testu - graficky spracované dáta vo forme grafov, štatistické tabuľky, merania,
- flow monitor - výstup z flow monitoru (ak sa ho rozhodneme použiť) bude tiež uložený v knižnici.

Knižnica bude organizovaná dvojúrovňovo a to tak, že používateľ si bude môcť vytvoriť „adresár“ a v rámci neho uchovávať topológie. Adresáre sú vhodný spôsob, ako pridať do knižnice organizačný prvok.

Taktiež bude tieto adresáre možné zdieľať alebo sprístupňovať ostatným používateľom, čím podporíme kolaboráciu medzi používateľmi. Samostatné topológie sa tiež budú môcť zdieľať.

Všetky dáta (samotná topológia, skripty, parametre, grafy, tabuľky, výstupy, ostatné súbory) sa budú držať pri objekte topológie. Keď používateľ vykoná meranie tak toto meranie sa uloží k topológii, ktorá bola vstupom pre meranie.

Výstupom knižnice budú všetky tieto súbory na stiahnutie. Navyše výstupom budú aj jednotlivé grafy a štatistiky.

Ďalším výstupom bude skript do ns-3.

## 8.5 Parametrizátor

Blok parametrizátor reprezentuje logiku parametrizovania skriptov. Parametrizovaný skript je taký, v ktorom sú použité nejaké vopred definované

postupnosti znakov, ktoré sú pri parametrizovaní nahradené konkrétnymi hodnotami.

Tieto hodnoty (parametre) budú môcť byť zadávané priamo z webovského rozhrania a nie zdĺhavým písaním do kódu.

Parametrizovateľný skript bude pred vstupom do knižnice podrobený parametrizačnému analyzátoru, ktorý zistí, či sa v skripte nachádzajú nejaké parametre a ak áno, tak ich používateľovi pri spúšťaní simulácie ponúkne na vyplnenie, následne ich doplní do skriptu.

Vstup parametrizátora bude parametrizovateľný skript. Jeho výstup bude hotový skript.

## 8.6 Simulátor ns-3

NS-3 je blok samotného simulátora. Jeho vstupy sú skript a prepínače. Jeho výstupy sú .tr súbor popisujúci namerané dáta pri simulácii, .xml súbor a .pcap súbor s paketmi vygenerovanými počas simulácie.

Blok NS-3 môže byť prepojený s Flow Monitorom, toto prepojenie zvažime.

## 8.7 Flow Monitor

Flow Monitorom dokážeme merať oneskorenie, jitter a straty priamo a nemusíme zložito analyzovať výstupy simulátora ns-3. Flow Monitor dokáže exportovať výstupy do XML súboru alebo SQL databázy. Jeho použitie zvažime pri implementácii.

## 8.8 awk/sed/gnuplot

Toto sú pomocné programy, ktorými budeme spracovávať výsledky zo simulátora ns-3 a Flow Monitoru. Ich použitie bude silne závisieť od výstupov a štatistík, ktoré bude systém produkovať.

## 9 Prototyp

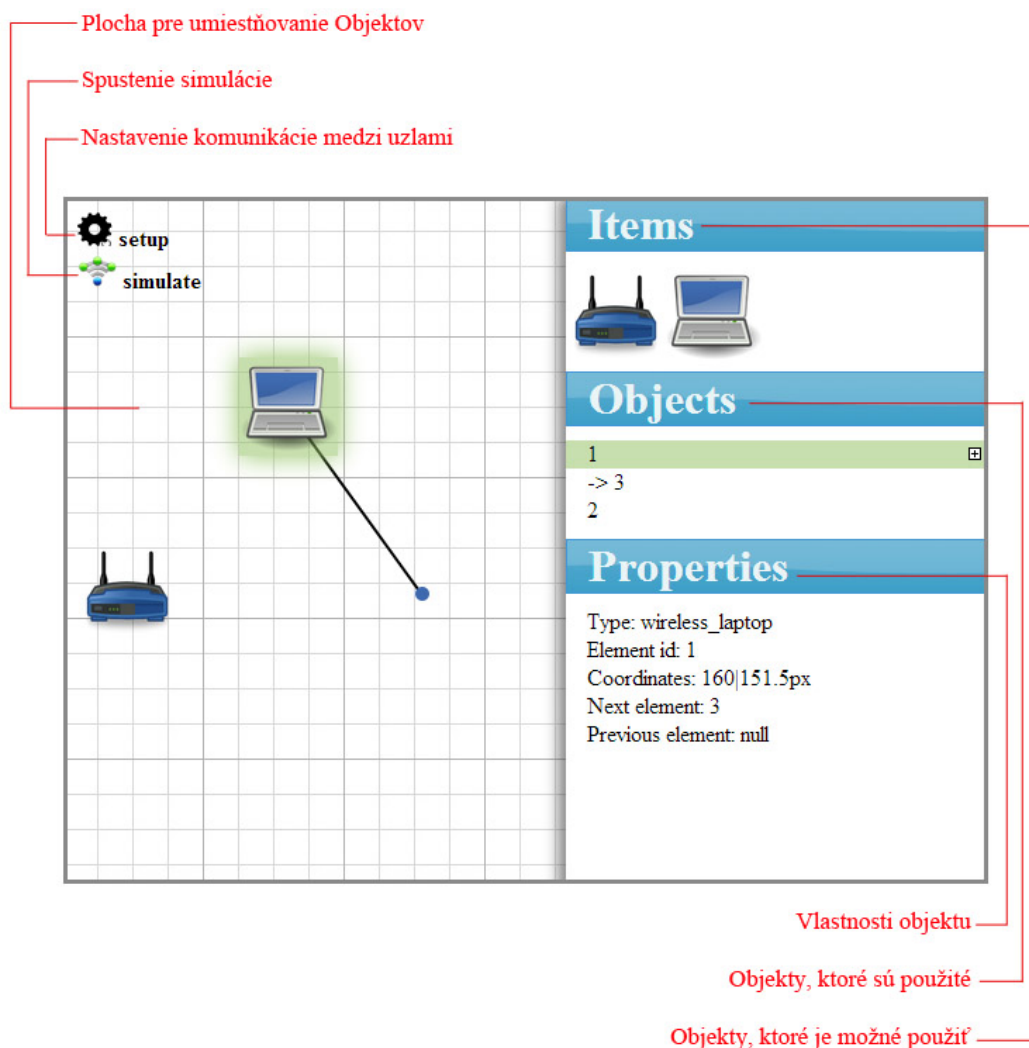
### 9.1 Cieľ prototypovania

Naším cieľom pri vytváraní prototypu bolo spojazdnenie servera s nainštalovaným sieťovým simulátorom ns-3 a podpornými nástrojmi a skriptami pre prácu s ním. Chceli sme implementovať editor topológie s čiastočnou funkcionalitou a parametrizátor na strane vstupov. Na strane výstupov bolo cieľom poskytnutie grafu z vybraných informácií pomocou programu gnuplot a taktiež poskytnutie výstupných súborov simulácie používateľovi na stiahnutie.

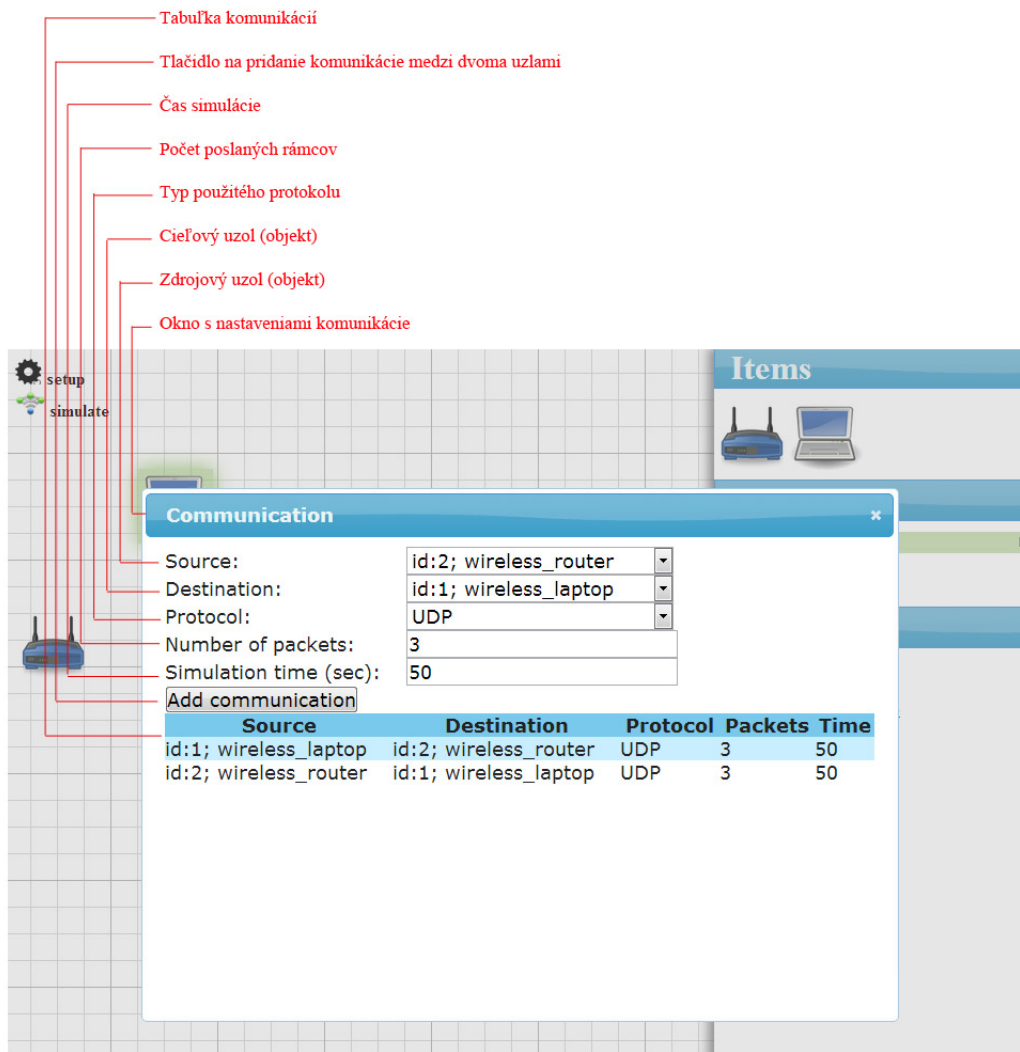
V nasledujúcich kapitolách sú popísané dosiahnuté výsledky pri implementácii prototypu.

### 9.2 Grafické používateľské rozhranie

Grafické používateľské rozhranie prototypu je spolu s jeho popisom zobrazené na obrázkoch Obr. 9.1 a Obr. 9.2.



Obr. 9.1: Hlavné okno prototypu s popisom

Obr. 9.2: Okno *Communication* s popisom

### 9.3 Technická realizácia GUI

Grafické používateľské rozhranie používa do značnej miery vývojové nástroje z balíkov jQuery a jQuery UI, čo sú robustné balíky obsahujúce veľké množstvo funkcií a nástrojov písaných v skriptovacom jazyku JavaScript. jQuery obsahuje hlavne funkcie na všeobecnú prácu s elementmi na stránke. Naopak jQuery UI je zamerané na zjednodušenie tvorby animovaných prvkov na stránke a zabezpečuje ich spätnú kompatibilitu aj so staršími prehliadačmi.

JavaScript je jazyk vykonávaný na strane klienta a tým pádom vytváranie topológie prebieha kompletne v prehliadači používateľa bez účasti serveru.

Pomocou jQuery UI sa vykonáva najdôležitejšia funkcia grafického rozhrania – pridávanie nových uzlov do topológie a ich umiestnenie na požadované miesto. Taktiež umiestňovanie bodov označujúcich cestu mobilného zariadenia. Vytváranie čiar zobrazujúcich túto cestu prebieha v jQuery doplnku nazvanom jQuery SVG.

Modálne okno, v ktorom sa dajú vytvárať komunikácie medzi jednotlivými

sietovými uzlami je tiež vytvorené za použitia jQuery UI.

Vizuálna štylizácia jednotlivých elementov na stránke prebieha webovým štandardom CSS.

Odosielanie dát na spracovanie prebieha asynchrónne a používa metódu \$.ajax, ktorej odovzdávame požadované nastavenia v objekte, ktorý je jej parametrom.

Údaje o sietových uzloch a komunikáciách sú uložené v samostatných objektoch reprezentujúcich serializované pole, ale pre potreby odosielania na server sú spojené dokopy a odoslané tak. Vďaka tomu sú po odoslaní na server obsiahnuté v globálnej premennej \$\_POST reprezentovanej v našom prípade polom polí.

## 9.4 Technická realizácia Parametrizátora

Po prijatí požiadavky webovým serverom sa spustí skript sim.php na web serveri. Tento skript najprv premapuje čísla jednotlivých uzlov na čísla uzlov ako im rozumie ns-3. V GUI môžu byť uzly pridávané v rôznom poradí, pričom každý uzol dostane sekvenčné číslo, ktoré sa nikdy neopakuje. Sietový simulátor ns-3 vyžaduje číslovať uzly od 0 po N-1, kde N je počet uzlov.

Následne pre každú definovanú komunikáciu spustíme simuláciu v ns-3 podľa vopred pripraveného skriptu, ktorý parametrizujeme zadanými parametrami. Medzi zadané parametre patrí: zdrojový uzol, cieľový uzol, počet uzlov v sieti, počet paketov na odoslanie, čas simulácie.

Prvých 30 sekúnd simulácie konverguje OLSR, preto odporúčame zadávať čas väčší ako 30 sekúnd. Pakety sa posielajú najčastejšie raz za sekundu. V prípade, že používateľ zadá viac paketov ako čas behu simulácie (mimo prvých 30 sekúnd), skript automaticky zmenší počet paketov tak, aby sa poslal práve 1 za sekundu. Inak sa pakety posielajú rovnomerne tak, aby sa za trvanie simulácie poslali všetky.

Sietový simulátor ns-3 zapíše svoje výstupy do dočasného dopredu vytvoreného adresára v systémovej adresári /tmp. Potom sa programom gnuplot skonverguje výstupný graf *FlowVSThroughput.plt* do formátu png. Všetky výstupné súbory sa takto zabalia do jedného ZIP archívu. Tieto akcie sa udejú pre každú komunikáciu zvlášť.

Nakoniec sa všetky výstupné dáta všetkých komunikácií zabalia do jedného kompletného ZIP archívu.

Po simulácii skript na serveri vráti AJAX-ovej funkcii, ktoré ho volala, úspech. Toto spôsobí presmerovanie používateľa na stránku s výsledkami simulácie.

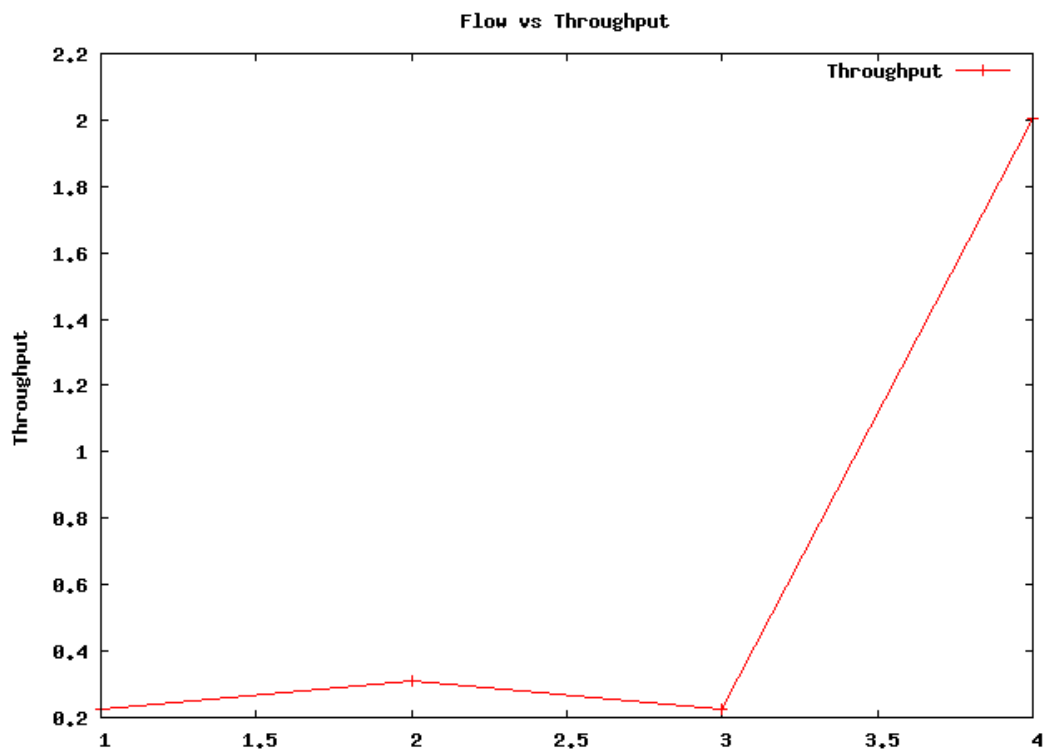
## 9.5 Výsledky simulácie

Obr. 9.3 zobrazuje vzhľad stránky s výsledkami. Používateľ si v hornej časti môže stiahnuť archív celej simulácie. Potom nasledujú grafy a archívy jednotlivých simulácií.

[Stiahnuť archív celej simulácie](#)

### Simulácia #1

#### Graf



Obr. 9.3: Výsledky simulácie

## 9.6 Simulačný skript

Prototyp využíva pri simulácii modulárnu, upravenú verziu príkladového skriptu pre simuláciu Wi-Fi adhoc sietí. Skript je napísaný v programovacom jazyku C++, ktorý je jedným z dvoch možností pre programovanie vstupného programu do ns-3. Druhou možnosťou je napísať skript v jazyku Python.

Skript vytvára topológiu uzlov usporiadanú do mriežky. Mriežka má staticky nastavenú šírku na 5 uzlov, takže počet uzlov v jednom riadku mriežky je vždy najviac 5. Nakoľko je v skripte použitý *ConstantPositionMobilityModel*, uzly sú statické, nehýbu sa počas simulácie, ale ostávajú na danej pozícii v mriežke. Vzdialenosť medzi uzlami je nastaviteľná, prednastavená je na 500 metrov. Uzly v mriežke musia byť od seba vzdialené maximálne 500 metrov, pri vyššej



vzdialenosti už nie sú schopné medzi sebou komunikovať.

V simulácii sa používa smerovací protokol OLSR pre adhoc siete. Protokol OLSR zabezpečuje komunikáciu aj vzdialených uzlov, ktoré sa vzájomne nevidia tak, že sa použijú iné uzly v topológii, ktoré sú dostupné po ceste správy. Ak teda komunikujú dva takto vzdialené uzly, správa je preposielaná viacerými uzlami v topológii.

Keďže je v topológii použitý protokol OLSR, je potrebné na začiatku simulácie vyčleniť čas na konvergenciu smerovania v topológii. Tento čas sme vyčlenili na 30 sekúnd, počas ktorých si uzly vytvárajú smerovacie tabuľky. Až po 30 sekundách začne prebiehať výmena UDP správ medzi špecifikovanými uzlami. Počas jednej simulácie si môžu správy vymeniť len dva uzly, teda špecifikuje sa jeden zdrojový a jeden cieľový uzol. Počet uzlov v topológii je nastaviteľný. Komunikačný tok môže mať rôzny počet paketov. Komunikácia prebieha cez protokol IPv4 na sieťovej vrstve a protokol UDP na transportnej vrstve RM OSI.

Ako už bolo spomenuté, skript má niekoľko používateľom nastaviteľných parametrov:

- zdrojový uzol (sourceNode) – ktorý uzol bude zdrojom UDP toku,
- cieľový uzol (sinkNode) – ktorý uzol bude cieľom UDP toku,
- počet uzlov (numNodes) – koľko uzlov sa nachádza v topológii,
- počet paketov UDP toku (numPackets) – koľko paketov si dané dva uzly vymenia,
- čas simulácie (simTime) – ako dlho bude simulácia trvať.

Počas simulácie sa sledujú sieťové rozhrania jednotlivých uzlov, pričom záznam o odchytených paketoch na rozhraní každého uzla sa ukladá do samostatného .pcap a .tr súboru. Tieto súbory sú po ukončení simulácie poskytnuté ako výstup pre analýzu simulačných výsledkov. Výstupom simulácie je aj každé dve sekundy aktualizovaný zoznam smerovacích tabuliek jednotlivých uzlov. Tento zoznam sa nachádza v súbore s príponou .routes.

V simulačnom skripte bol použitý aj podporný nástroj FlowMonitor. Počas simulácie sleduje a vypočítava bajtovú priepustnosť na jednotlivých uzloch. Tieto informácie si ukladá do .plt a XML súboru, ktoré sú taktiež výstupom po ukončení simulácie. XML súbor ma príponu .flowmon. Z tohto súboru sa následne generuje graf pomocou nástroja gnuplot.

Skript bude v budúcnosti poskytovať oveľa širšie možnosti nastavenia jednotlivých uzlov a protokolov, napríklad nastavenie pozície uzla, definovanie smeru pohybu, nastavenie rôznych smerovacích protokolov ako aj nastavenie toho, aké parametre má sledovať nástroj FlowMonitor a mnohé iné nastavenia. Skript bude v budúcnosti umožňovať komunikáciu medzi viacerými uzlami súčasne.

## Literatúra

- [1] Hughes, J. 2009. *Network Simulation Introduction*.  
<http://www.openextra.co.uk/articles/network-simulation>
- [2] Issariyakul, T., Hossain, E. 2009. *Introduction to Network Simulator NS2*.  
<http://kpath-routing-in-ns2.googlecode.com/files/Introduction%20to%20Network%20Simulator%20NS2.pdf>
- [3] NS Manual. *The Network Simulator: Building Ns*.  
<http://www.isi.edu/nsnam/ns/ns-build.html>
- [4] Ezreik, A., Gheryani, A. 2012. *Design and Simulation of Wireless Network using NS-2*.  
<http://psrcentre.org/images/extraimages/412630.pdf>
- [5] NS Manual. *16.1.3 Network Components in a mobilenode*.  
<http://www.isi.edu/nsnam/ns/doc/node173.html>
- [6] NS Manual. *14.3 Channel Class*.  
<http://www.isi.edu/nsnam/ns/doc/node145.html>
- [7] Grega, M. 2005. *Possibility of using network simulator ns-2 for modeling satellite networks*.  
<http://www.aei.tuke.sk/pdf/2005-04/Grega.pdf>
- [8] CNL. *WiMAX module for the ns-2 simulator*.  
[http://www.lrc.ic.unicamp.br/wimax\\_ns2/](http://www.lrc.ic.unicamp.br/wimax_ns2/)
- [9] UCBT. *Bluetooth extension for NS2 at the University of Cincinnati*.  
<http://www.cs.uc.edu/~cdmc/ucbt/>
- [10] Carneiro, G. 2005. *FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3)*.  
<http://paginas.fe.up.pt/~mricardo/doc/conferences/nstools2009/flowmon-paper.pdf>
- [11] NS3Generator. *NS3Generator*.  
<http://www.nsnam.org/wiki/index.php/Ns3Generator>
- [12] Greis, M. *Tutorial for the Network Simulator "ns"*.  
<http://www.isi.edu/nsnam/ns/tutorial/index.html>
- [13] Henderson, T. 2011. *The ns Manual*.  
<http://www.isi.edu/nsnam/ns/doc/node1.html>
- [14] <http://barcampbangalore.org/bcb/bcb12/network-simulator>

- [15] NS-2 Blog. 2008. *Trace graph - Network Simulator NS-2 trace files analyser*.  
<http://ns-2.blogspot.sk/2008/10/trace-graph-network-simulator-ns-2.html>
- [16] Qian, H., Fang, W. *JTrana: A Java-based NS2 Wireless Trace Analyzer*.  
<https://sites.google.com/site/ns2trana/>
- [17] Williams, T., Kelley, C. *gnuplot*. <http://www.gnuplot.info/>
- [18] Jones, E. 2007. *Tools for ns2 Simulations*.  
<http://www.evanjones.ca/software/ns2tools.html>
- [19] Tahiliani, M. *XGraph Utility of NS-2*.  
[http://mohit.ueuo.com/NS-2\\_Tutorials/XGraph%20Utility%20of%20NS%20-%202.pdf](http://mohit.ueuo.com/NS-2_Tutorials/XGraph%20Utility%20of%20NS%20-%202.pdf)
- [20] Kurkowski, S., Camp, T., Colagrosso, M. 2006. *A Visualization and Analysis Tool for Wireless Simulations: iNSpect*.  
[http://scholar.google.sk/scholar\\_url?hl=sk&q=http://citeseerx.ist.psu.edu/viewdoc/download%3Fdoi%3D10.1.1.71.2453%26rep%3Drep1%26type%3Dpdf&sa=X&scisig=AAGBfm2gKK-VMHUXDIxRmYb5uYZvVz5jtw&oi=scholar&ei=Y36OULymFofotQbau4CQCA&ved=0CB0QgAMoADAA](http://scholar.google.sk/scholar_url?hl=sk&q=http://citeseerx.ist.psu.edu/viewdoc/download%3Fdoi%3D10.1.1.71.2453%26rep%3Drep1%26type%3Dpdf&sa=X&scisig=AAGBfm2gKK-VMHUXDIxRmYb5uYZvVz5jtw&oi=scholar&ei=Y36OULymFofotQbau4CQCA&ved=0CB0QgAMoADAA)
- [21] Jin, Ch., Chen, Q., Jamin, S. *Inet: Internet Topology Generator*.  
<http://topology.eecs.umich.edu/inet/inet-2.0.pdf>
- [22] TraceMetrics. *TraceMetrics - A trace file analyzer for Network Simulator 3*.  
<http://www.tracemetrics.net>
- [23] Thomas, Ch. 2011. *NS-3 simulation of WiMAX networks*.  
<http://cse.wustl.edu/Research/Lists/Technical%20Reports/Attachments/954/NS-3%20Simulation%20of%20WiMAX%20Networks.pdf>
- [24] Junseok, K. 2011. *WIFI on NS-3*.  
[http://www2.engr.arizona.edu/~junseok/ns3\\_wifi.htm](http://www2.engr.arizona.edu/~junseok/ns3_wifi.htm)
- [25] Cochard-Labbé, O. 2010. *TCP flow analysis with tshark, tcptrace and xplot*.  
<http://blog.cochard.me/2010/08/tcp-flow-analysis-with-tshark-tcptrace.html>
- [26] Henderson, T. 2008. *ns-3 tutorial*.  
<http://www.slideshare.net/vinotd/tutorial-ns-3tutorialslides>

- [27] ns-3 project. 2012. *ns-3 Model Library*.  
[http://www.nsnam.org/docs/release/3.15/models/  
ns-3-model-library.pdf](http://www.nsnam.org/docs/release/3.15/models/ns-3-model-library.pdf)
- [28] Information & Telecommunication Technology Center, The University of Kansas *Models and Tools for ns-3 Network Simulator*.  
[https://wiki.ittc.ku.edu/resilinet/Models\\_and\\_Tools\\_for\\_  
ns-3\\_Network\\_Simulator](https://wiki.ittc.ku.edu/resilinet/Models_and_Tools_for_ns-3_Network_Simulator)
- [29] nsNam. *NetAnim*.  
<http://www.nsnam.org/wiki/index.php/NetAnim>
- [30] nsNam. *Current Developement - 802.11 Model Extensions*.  
[http://www.nsnam.org/wiki/index.php/Current\\_Development#802.  
11\\_model\\_extensions](http://www.nsnam.org/wiki/index.php/Current_Development#802.11_model_extensions)
- [31] Cetinkaya, E., Sterbenz J. 2011. *Wireless Networking Simulation with ns-3*.  
[http://www.ittc.ku.edu/~jjpgs/courses/mwnets/  
lecture-lab-wireless-ns3-display.pdf](http://www.ittc.ku.edu/~jjpgs/courses/mwnets/lecture-lab-wireless-ns3-display.pdf)
- [32] ns-3 backup. 2009. *Building a Wireless Network Topology*.  
[http://ns-regression.ee.washington.edu:8010/html/docs/  
release/3.6/tutorial/tutorial\\_26.html](http://ns-regression.ee.washington.edu:8010/html/docs/release/3.6/tutorial/tutorial_26.html)
- [33] Mielczarek I., Stelter A. 2011. *Simulation of WLAN using ns-3*.  
[http://www.pwt.et.put.poznan.pl/PWT\\_2011/PWT%202011\\_5476.pdf](http://www.pwt.et.put.poznan.pl/PWT_2011/PWT%202011_5476.pdf)
- [34] Carneiro, G. 2009. *NS-3 Tutorial*.  
[http://rtcm.inescn.pt/fileadmin/rtcm/WorkShop\\_13\\_Fev\\_09/ppt3.  
pdf](http://rtcm.inescn.pt/fileadmin/rtcm/WorkShop_13_Fev_09/ppt3.pdf)
- [35] nsNAM. 2010. *Using the Tracing System* .  
[http://www.nsnam.org/docs/release/3.9/tutorial/tutorial\\_23.  
html](http://www.nsnam.org/docs/release/3.9/tutorial/tutorial_23.html)
- [36] Severínová, H., Nagy, M., Oros, D., Panenka, R., Pirháč, M., Svetlík, M. 2010. *Programová podpora pre sieťový simulátor*.  
[http://labss2.fiit.stuba.sk/TeamProject/2010/team03pss/ine/  
projektova\\_dokumentacia1.pdf](http://labss2.fiit.stuba.sk/TeamProject/2010/team03pss/ine/projektova_dokumentacia1.pdf)
- [37] nsNAM. 2012. *ns-3 Manual*.  
[http://www.nsnam.org/docs/release/3.15/tutorial/  
ns-3-tutorial.pdf](http://www.nsnam.org/docs/release/3.15/tutorial/ns-3-tutorial.pdf)
- [38] nsNAM. 2012. *ns-3 Key technologies*.  
<http://www.nsnam.org/overview/key-technologies/>

Príloha A  
**Dokumentácia k riadeniu projektu**

## Úvod

V tejto časti dokumentu sa nachádza dokumentácia k riadeniu projektu s názvom Simulácia bezdrôtových sietí, ktorý vznikol a je realizovaný na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Obsah je nasledovný:

1. Ponuka - dokument, ktorým sa tím uchádzal o pridelenie tejto témy
2. Plán projektu - postupnosť jednotlivých prác na projekte s určením ich termínov
3. Úlohy členov tímu - krátkodobé a dlhodobé úlohy jednotlivých členov tímu
4. Zápisnice zo stretnutí
5. Štandardy kódovania
6. Posudok analýzy a hrubého návrhu tímu č. 7
7. Posudok našej analýzy a hrubého návrhu vypracovaný tímom č. 7
8. Vyjadrenie sa k posudku
9. Manažment verzií, konfigurácií a zmien - použitý verzionovací systém a pod.
10. Preberacie protokoly

**Slovenská technická univerzita v Bratislave**  
**Fakulta informatiky a informačných technológií**

**Martin Čechvala**  
**Ivana Hucková**  
**Jakub Obetko**  
**Richard Roštecký**  
**Juraj Šubín**  
**Viktor Šulák**

**Tím č. 2**

**Ponuka**

Vedúci tímu: Ing. Peter Magula

september 2012

## 1 Tím

- **Bc. Čechvala Martin** - absolvent bakalárskeho štúdia na STU FIIT. Má silné znalosti operačného systému GNU/Linux, viacerých programovacích jazykov (C, PHP, Python, Perl, Pascal, QT, Java, x86 asm, TCL), vývojových prostredí (MS-VS, Lazarus, Eclipse) a sieťových tém. S Linuxom pracuje vyše 7 rokov (Gentoo, Debian, Ubuntu) a s BSD posledný rok. Počas strednej školy bol správcom školskej siete. Posledné takmer 4 roky pracuje ako PHP/C programátor, sieťový technik a správca systémov na báze Unix. Popri hlavnej práci pracuje pre viacero aj ISP firiem, implementuje hlavne VoIP siete. V poslednom čase sa zameriava práve na sieťové technológie, je držiteľom certifikátu CCNP a Cisco inštruktor na STU FIIT, často využíva emulátor GNS3. Príspevok k projektu bude hlavne: koncepčný návrh, implementácia systému v PHP a podporného softvéru, sieťové otázky.
- **Bc. Richard Roštecký** - titul bakalár získal na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave, v študijnom programe počítačové a komunikačné systémy a siete. V rámci bakalárskeho štúdia nadobudol znalosti v rôznych oblastiach informatiky. Počnúc návrhom informačných systémov až po ich programovanie, testovanie. Nadobudol schopnosti programovania v jazykoch JAVA, C, C++, C#, PHP a písania dotazov pre databázy MySQL, SQLite. Inžinierske štúdium pokračuje na rovnakom odbore a škole ako bakalárske. Počas štúdia pracuje na vlastných projektoch internetových stránok, grafických návrhoch reklamy vytvorenej pomocou grafického softvéru Adobe Photoshop a pracoval pre firmu zaoberajúcu sa tvorbou internetových stránok kde si priblížil redakčný systém Drupal, databázu PostgreSQL spolu s prácou v tíme synchronizovanú pomocou SVN. Pri bakalárskej práci sa naučil vyvíjať aplikácie pre smartfóny Samsung, konkrétne platformu Bada. Aplikácia bola certifikovaná spoločnosťou Samsung. Vzhľadom na sieťové technológie je študentom CISCO network academy, kde doposiaľ urobil tri semestre zo štyroch v CCNA certifikácii. Jeho prínosom pre tím bude jeho kreatívne myslenie.
- **Bc. Jakub Obetko** - absolvent bakalárskeho štúdia na FIIT STU, odbor PKSS. Počas štúdia sa zameriava najmä na oblasť počítačových sietí a svoje vedomosti v oblasti návrhu, konfigurácie a fungovania sietí rozširoval na CISCO akadémii na FIIT STU. Po ukončení bakalárskeho štúdia získal certifikát CCNA. Počas štúdia nadobudol skúsenosti v programovaní v jazykoch C, Java a C#, v ktorom programoval väčšinu projektov, ako napríklad analyzátor sieťovej premávky, softvérový prepínač alebo smerovač. V mnohých projektoch pracoval s databázami najmä s relačnou databázou MySQL. Často pracuje so sieťovým emulátorom GNS3. V



inžinierskom štúdiu chce pokračovať v zdokonaľovaní svojich vedomostí v oblasti sietí a v štúdiu na CCNP. Jeho prínosom k tomuto projektu môžu byť nadobudnuté vedomosti v oblasti počítačových sietí, skúsenosti s inými sieťovými simulátormi a emulátorom GNS3, schopnosť racionálneho rozhodovania a plánovania práce tak, aby bola vykonaná v dohodnutom termíne, skúsenosti získané pri návrhu používateľského prostredia, ale aj skúsenosti z oblasti testovania používateľského rozhrania a testovania funkčnosti programov.

- **Bc. Hucková Ivana** - absolventka bakalárskeho štúdia na FIIT STU v študijnom programe Počítačové a komunikačné systémy a siete. Počas štúdia získala skúsenosti s programovaním v jazykoch C, C#, Java, PHP, VBA a s prácou s databázou MySQL. V rámci štúdia rozširovala svoje vedomosti a praktické skúsenosti z oblasti počítačových sietí v regionálnej CISCO akadémii na fakulte. Absolvovala štyri semestre CCNA a momentálne sa pripravuje na absolvovanie certifikačnej skúšky. Jej prínosom k projektu budú vedomosti a praktické skúsenosti s návrhom a konfiguráciou počítačových sietí, ako aj s prácou s rôznymi sieťovými simulátormi. Popri štúdiu pracuje na vývoji aplikácií pre bankový systém v UniCredit Bank Slovakia, vďaka čomu získala cenné skúsenosti s prácou v tíme, schopnosť pracovať pod tlakom a dodržiavať termíny, čo môže byť výrazným prínosom k celému projektu.
- **Bc. Viktor Šulák** - absolvent študijného programu Počítačové a komunikačné systémy a siete FIIT STU. Počítačové siete študoval aj na Cisco sieťovej akadémii už v čase stredoškolského štúdia a pokračoval aj na univerzite. Počas štúdia tiež získal skúsenosti s programovacími jazykmi C, C++, C#, Java, PHP a niekoľkými ďalšími a taktiež s databázovým systémom MySQL. Prínosom k projektu by mohli byť jeho schopnosť zodpovednej práce v tíme, tímový duch, skúsenosti s bezdrôtovými sieťami a skúsenosti s viacerými sieťovými simulátormi. Sieťové simulátory využíval vo svojej maturitnej aj bakalárskej práci. Taktiež má skúsenosti s grafikou a webdizajnom, ktoré by mohol využiť pri návrhu užívateľského rozhrania.
- **Bc. Juraj Šubín** - absolvent bakalárskeho štúdia na FIIT STU v študijnom programe Počítačové a komunikačné systémy a siete. Počas štúdia sa sústredil na zdokonalenie svojich znalostí a schopností v oblasti počítačových sietí. Okrem toho vypracoval projekty s rôznym zameraním v programovacích jazykoch C, Java, C#, PHP, x86 assembler, čím nadobudol schopnosti programovania v týchto jazykoch. Súčasťou projektov bola neraz aj relačná databáza MySQL. Pri práci často využíva emulátor GNS3. V tomto čase pracuje na získaní certifikátu CCNA, čiže návrhom a konfiguráciou sietí sa aktívne zaoberá. V inžinierskom štúdiu chce pokračovať v doterajšom smerovaní a získať certifikát CCNP. Jeho prínosom

k projektu môže byť schopnosť pracovať v tíme, chuť učiť sa nové veci, vedomosti z oblasti počítačových sietí, skúsenosti s návrhom a testovaním používateľského rozhrania.

## 2 Motivácia

Všetci členovia nášho tímu sú úspešnými absolventami bakalárskeho stupňa štúdia na FIIT STU, odbor PKSS. Každý z členov tímu má vysoký záujem rozvíjať svoje schopnosti najmä v oblasti návrhu a správy počítačových sietí, čo dokazuje aj záujem o nadštandardné vzdelanie vrámci RCNA na FIIT. Všetci členovia tímu majú praktické skúsenosti s návrhom a konfiguráciou počítačových sietí, ako aj znalosti o ich fungovaní a prevádzke. Nakoľko simulátory NS2 alebo NS3 sú jednými zo špičkových simulátorov sietí používaných v súčasnosti, chceli by sme svoje vedomosti rozšíriť aj týmto smerom, naučiť sa s nimi pracovať, ale najmä vytvoriť systém, ktorý poskytne iným používateľom možnosť pracovať s týmito nástrojmi jednoduchým a intuitívnym spôsobom. Cieľom bude pokračovať vo vytváraní takéhoto systému tak, aby spĺňal kritéria požadované v zadaní. Taktiež máme záujem prispieť ku skvalitneniu výučby počítačových sietí na predmetoch ako sú PSIP a WAN, nakoľko systém bude prístupný pre množstvo študentov PKSS, ale aj iných používateľov.

### 3 Kolaboračné nástroje

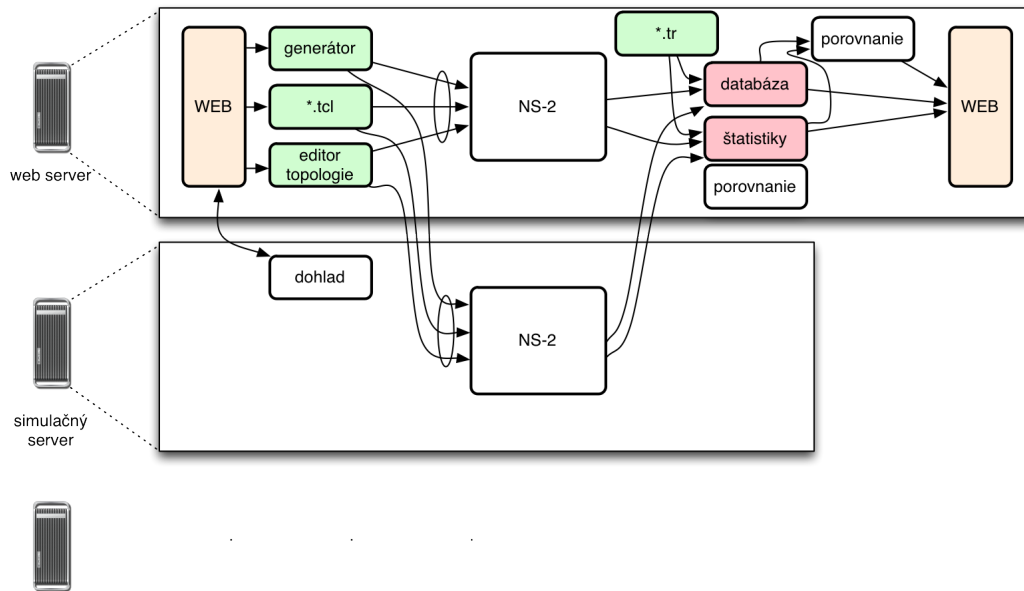
Niekoľkoročné skúsenosti s prácou v tíme nás naučili využívať rôzne kolaboračné technológie.

- SVN - Subversion, systém, vďaka ktorému už uchováваме zdrojové kódy a kompletnú dokumentáciu (tj. všetky dáta projektu) v centrálnom repositári. Každý člen tímu má nezávislý prístup do repositára. SVN nám poskytuje možnosť úplne nezávislého vývoja.
- Trac - kolaboračný nástroj pre tímy, je odvodený od MediaWiki softvéru. V tomto systéme uchovávané všetky informácie o projekte, odkazy na súvisiace zdroje, všetko v prehľadne organizovanej štruktúre, ktorú samostatné stránky poskytujú. Navyše je v Tracu možné vytvárať a priradovať „tickety“ obsahujúce zadanie práce, vykonávateľa, dátum a termín. Takto môžeme presne sledovať ako sa projekt vyvíja. Trac sme integrovali s SVN a je možné cez Trac prehliadať SVN repositár. Trac tiež podporuje zaznamenávanie „roadmap“, čo je plán funkcií pre celý projekt a „milestones“, dôležité celky.
- Mailing list - vytvorili sme si interný mailing list. Mailing list funguje tak, že keď jeden člen pošle do neho e-mail, tak ho ostatní dostanú. Zároveň si mailing list udržiava vetvenú históriu všetkých správ. Tento nástroj slúži na komunikáciu rozsiahlejších rozmerov a diskusie.
- LaTeX - ako správni akademici samozrejme tvoríme dokumentáciu a aj túto ponuku v systéme LaTeX. Vďaka spôsobu jeho písania a možnosti rozdeľovať ho na viacero súborov vieme naraz pracovať na jednom dokumente aj všetci. LaTeXový dokument máme samozrejme uložený v SVN.
- 6. virtuálnych serverov - pri implementácii projektu vytvoríme 6 identických virtuálnych serverov, na ktorých bude funkčná posledná verzia tohoto projektu. Každý server dostanete jeden člen tímu a bude vyvíjať iba na ňom. Vďaka použitiu SVN budeme synchronizovať zdrojové kódy medzi všetkými virtuálnymi servermi.

Všetky tieto nástroje (okrem virtuálnych serverov) už máme funkčné a pracujeme s nimi v rámci tohoto projektu.

## 4 Návrh

Vychádzajúc z vyčerpávajúcej dokumentácie predchádzajúceho tímu na Obr. ?? navrhujeme modifikovanú architektúru:



Obr. 1: Navrhovaná architektúra

Súčasne prekladáme predbežný návrh vylepšení.

### 4.1 Vektorový editor pohybu

Vytvoríme editor pohybu zariadení tak, aby používateľ mohol jednoduchým vložení a potiahnutím šípky definovať smer a vzdialenosť, ktorou sa bude objekt pohybovať a následne do vlastností vektoru definuje čas/rýchlosť, akou sa bude objekt daným smerom pohybovať. Takto definované vektory sa budú dať kombinovať, čím dosiahneme plne grafické vytváranie pohybových akcií. Výstup z editora pohybov bude vstupom do editora topológie (prípadne integrujeme editor pohybu priamo do editora topológie).

### 4.2 Integrácia NS-3

Preskúmame možnosť integrácie NS-3 do systému tak, aby bola zachovaná funkčnosť NS-2. Používateľ si pri definícii simulácie bude môcť zvoliť, ktorý simulátor použije. Systém následne upraví ponuku tak, aby zodpovedala schopnostiam konkrétneho zvoleného simulátora.

Podľa aktuálnej dokumentácie je systém naprogramovaný modulárne, avšak autori upozorňovali na možné problémy pri integrácií. Tieto problémy sme pripravení zdolať.

### **4.3 Upravovanie skriptov vygenerovaných autogenerátorom**

Ako bolo spomenuté v predchádzajúcej dokumentácii, nepodarilo sa implementovať upravovanie skriptov z autogenerátora. Túto funkciu implementujeme.

### **4.4 Granularita grafov**

Predpokladáme, že niektoré grafy sú časovo závislé a používajú program RRDTool. Implementujeme spôsob akým bude môcť používateľ dynamicky vyseknúť z grafu časový interval, ktorý sa mu v reálnom čase približí.

### **4.5 TORA a DSDV**

Prirodzene zaktualizujeme (alebo budeme používať najnovšiu verziu) simulátora ns-2. Ak sa problém so smerovacími protokolmi TORA a DSDV doteraz neodstránil, aplikujeme naše skúsenosti s programovaním veľkých projektov v C/C++ a opravíme ho my.

### **4.6 Distribuované prostredie**

Analyzujeme súčasný stav simulovania v distribuovanom prostredí. V prípade NS-3 je táto funkcia podporovaná. V každom prípade navrhujeme implementovať distribúciu jednotlivých simulácií na viacero počítačov (Obr).

### **4.7 LiveCD**

Na záver projektu vytvoríme LiveCD, ktoré bude použiteľné aj úplným začiatočníkom bez potreby inštalovania akéhokoľvek softvéru.

### **4.8 Pridelovanie výkonu**

Implementujeme pridelovanie výkonu a prioritizovanie jednotlivých používateľov, aj simulácií jedného používateľa. Tiež implementujeme štatistiky výkonu (ako dlho prebiehala simulácia, koľko systémových prostriedkov využila, ...).

### **4.9 Export výsledkov simulácií**

Pridáme možnosť exportovať výsledky simulácií do TeX/PDF. Myslíme si, že táto vlastnosť systému bude prospešná hlavne pre vedeckých pracovníkov.

#### **4.10 Vytváranie topológií zo šablón**

Pri vytváraní topológie bude používateľ môcť zvoliť existujúcu topológiu ako základ novej. Toto umožní rýchly vývoj topológií s rovnakým základom.

#### **4.11 Ďalšie vylepšenia a funkcie**

Samozrejme toto ani z ďaleka nie sú všetky vylepšenia, ktoré plánujeme zaviesť. Po hĺbkovej analýze celého riešenia vytvoríme novú špecifikáciu, do ktorej zahrnieme ďalšie funkcie a požiadavky vedúceho tímu.

## 5 Priorita tém

1. Simulácia bezdrôtových sietí
2. Aplikácia pre platformu Funtoro
3. Inovatívne multimedialne služby
4. Zdieľaná identita vo WebRTC doménach
5. Testovanie znalostí s grafickým vstupom
6. Interaktívny prezentačný systém



## 6 Rozvrh

	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	
Pondelok	Martin Čechvala		BKS		OP			DPS							
	Ivana Hucková		BKS					DPS							
	Jakub Obetko		BKS												
	Richard Roštecký		BKS												
	Juraj Šubín		BKS												
Viktor Šulák			BKS												
Utorok	Martin Čechvala			DPS		KSS				BPS					
	Ivana Hucková			DPS						BPS					
	Jakub Obetko									BPS					
	Richard Roštecký	KOD								BPS					
	Juraj Šubín									BPS					
Viktor Šulák	KOD								BPS						
Streda	Martin Čechvala			BKS		KSS		BPS				Cisco			
	Ivana Hucková			BKS				BPS							
	Jakub Obetko					BKS		BPS							
	Richard Roštecký					BKS		BPS							
	Juraj Šubín					BKS		BPS							
Viktor Šulák			BKS				BPS								
Štvrtok	Martin Čechvala				ZK		Preferovaný čas stretnutia			APS			VSP1		
	Ivana Hucková										APS			VSP1	
	Jakub Obetko			TDS							APS			VSP1	
	Richard Roštecký	KOD									APS			VSP1	
	Juraj Šubín			TDS							APS			VSP1	
Viktor Šulák									APS			VSP1			
Piatok	Martin Čechvala		ZK		OP			Cisco							
	Ivana Hucková														
	Jakub Obetko							TDS							
	Richard Roštecký														
	Juraj Šubín							TDS							
Viktor Šulák															

## Plán projektu

Nižšie je rozpisovaný podľa aktuálneho progresu aktualizovaný plán projektu s uvedením dôležitých termínov a plánovaných cieľov.

**1.10.2012 :**

- vypracovanie ponuky pre tému Simulácia bezdrôtových sietí
- programová podpora pre podporu tímovej práce (mailing list, SVN, trac)

**15.10.2012 :**

- časový harmonogram prác na zimný semester
- rozdelenie úloh v tíme

**22.10.2012 :**

- funkčná webstránka
- zriadenie projektovej dokumentácie
- dostupnosť riešenia predchádzajúceho tímu

**29.10.2012 :**

- analýza sieťového simulátora ns-2
- analýza sieťového simulátora ns-3
- analýza riešenia predchádzajúceho tímu

**5.11.2012 :**

- zhodnotenie riešenia predchádzajúceho tímu
- zhodnotenie analýzy
- výber sieťového simulátora ns-2 alebo ns-3 pre implementáciu
- podrobnejšia analýza podporných programov pre zvolený simulátor
- špecifikácia požiadaviek, vstupov a výstupov

**12.11.2012 :**

- hrubý návrh

**16.11.2012 :**

- dokument opisujúci doterajšie riešenie projektu pre odovzdanie

**19.11.2012 :**

- posudok dokumentácie konkurenčného tímu

**26.11.2012 :**

- dopracovanie zistených nedostatkov a návrh prototypu vybraných častí projektu

**3.12.2012 :**

- implementácia prototypu vybraných častí

**10.12.2012 :**

- odovzdanie prototypu a kompletnej dokumentácie, prezentácia riešenia

## Úlohy členov tímu

V tejto časti sú uvedené krátkodobé a dlhodobé úlohy členov tímu. Krátkodobé úlohy sú jednorazové a automaticky znamenajú autorstvo príbuznej kapitoly v dokumentácii k inžinierskemu dielu alebo autorstvo časti vytvoreného diela. Dlhodobé úlohy majú členovia tímu počas celého trvania riešenia projektu.

### **Bc. Martin Čechvala**

Krátkodobé úlohy:

- mailing list, SVN a trac
- analýza riešenia predchádzajúceho tímu
- návrh a špecifikácia riešenia
- server a "back-endový skript

### **Bc. Ivana Hucková**

Dlhodobé úlohy:

- vedúca tímu
- tvorba zápisnice

Krátkodobé úlohy:

- analýza riešenia predchádzajúceho tímu
- všeobecný úvod do simulácií

### **Bc. Jakub Obetko**

Dlhodobé úlohy:

- sledovanie projektových noviniek a termínov

Krátkodobé úlohy:

- analýza ns-2
- analýza podporných programov pre ns-3
- simulačný skript pre prototyp

### **Bc. Richard Roštecký**

Dlhodobé úlohy:

- správca webového sídla

Krátkodobé úlohy:

- analýza ns-3

- zhodnotenie analýzy
- GUI prototypu

**Bc. Juraj Šubín**

Dlhodobé úlohy:

- zastrešenie finálnej formy dokumentu
- dokument k riadeniu projektu

Krátkodobé úlohy:

- analýza ns-2
- analýza podporných programov pre ns-3

**Bc. Viktor Šulák**

Krátkodobé úlohy:

- logo tímu
- analýza ns-3
- analýza vstupov a výstupov ns-3
- GUI prototypu

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 1

**Dátum:** 27.9.2012  
**Čas:** 13:00  
**Miesto:** FIIT STU, blok D, 3. poschodie  
**Prítomní:** Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Stanovenie termínu stretnutí tímu na štvrtok 13.00
- Určenie organizačných funkcií v tíme:
  - Riadenie tímu: Bc. Martin Čechvala
  - Tvorba zápisnice: Bc. Ivana Hucková
- Vytvorenie mailing-listu pre zjednodušenie komunikácie členov tímu
- Vytvorenie SVN
- Vytvorenie systému Trac pre zjednodušenie riadenia práce v tíme
- Rozdelenie úloh na prípravu ponuky a prezentácie

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 2

**Dátum:** 4.10.2012  
**Čas:** 13:00  
**Miesto:** FIIT STU, blok D, 3. poschodie  
**Prítomní:** Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Oslava získania vytúženej témy :)
- Vytvorenie loga tímu

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 3

**Dátum:** 8.10.2012  
**Čas:** 16:00  
**Miesto:** FIIT STU, blok E, 7. poschodie  
**Prítomní:** Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Oficiálne pridelenie témy
- Zmena termínu stretávania sa na pondelok 17:00
- Návrh plánu práce
- Návrh časového harmonogramu



## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 4

**Dátum:** 15.10.2012

**Čas:** 17:00

**Miesto:** FIIT STU, blok E, Softvérové štúdio

**Prítomní:** Ing. Peter Magula  
Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Definovanie organizačných úloh členov tímu:
  - Tvorba zápisnice - Bc. Ivana Hucková
  - Vedúca tímu (komunikácia s iným tímom) - Bc. Ivana Hucková
  - Správca webového sídla - Bc. Richard Roštecký
  - Zastrešenie finálnej formy dokumentu - Bc. Juraj Šubín
  - Sledovanie noviniek a termínov - Bc. Jakub Obetko
  - Tvorba projektového denníka

### Povinnosti do ďalšieho stretnutia:

- Zistiť formu a obsah projektového denníka
- Zistiť podmienky umiestnenia webstránky
- Zabezpečiť CD riešenia predchádzajúceho tímu
- Analýza riešenia predchádzajúceho tímu
- Analýza možností na implementáciu funkcií
- Skompletizovať zápisnice

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 5

**Dátum:** 22.10.2012  
**Čas:** 17:00  
**Miesto:** FIIT STU, blok E, Softvérové štúdio  
**Prítomní:** Ing. Peter Magula  
Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Rozdelenie tímu na dvojice pre analýzu simulátorov a riešenia predchádzajúceho tímu:
  - Martin a Ivana - analýza riešenia predchádzajúceho tímu
  - Jakub a Juraj - analýza ns-2
  - Richard a Viktor - analýza ns-3
- Dokončenie plánu práce:
  - 5. týždeň - analýza
  - 6. týždeň - špecifikácia požiadaviek
  - 7. týždeň - špecifikácia požiadaviek a hrubý návrh
  - 8. týždeň - odovzdanie analýzy, špecifikácie a návrhu, analýza a vytvorenie posudku pre iný tím
  - 9. týždeň - odovzdanie posudku analýzy iného tímu
  - 10. týždeň - dopracovanie zistených nedostatkov a návrh prototypu vybraných častí projektu
  - 11. týždeň - implementácia prototypu vybraných častí
  - 12. týždeň - odovzdanie prototypu a kompletnej dokumentácie, prezentácia riešenia
- Vytvorenie priestoru pre projektový denník v systéme Trac
- Doriešenie problému s databázou na webstránke
- Vytvorenie konta v mailing liste aj v systéme Trac pre vedúceho

**Úlohy do ďalšieho stretnutia:**

- Analýza riešenia predchádzajúceho tímu
- Analýza ns-2 a ns-3
- Kontaktovať P. Lacka kvôli databáze na webstránke
- Úprava obsahu webu

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 6

**Dátum:** 29.10.2012

**Čas:** 17:00

**Miesto:** FIIT STU, blok E, Softvérové štúdio

**Prítomní:** Ing. Peter Magula  
Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Diskusia ohľadom analýzy riešenia predchádzajúceho tímu
- Diskusia ohľadom analýzy ns-2 a ns-3
- Dohodnutie vytvorenia nového projektu, ktorý nebude nadväzovať na riešenie predchádzajúceho tímu
- Určenie simulátora ns-3 pre použitie v projekte

### Úlohy do ďalšieho stretnutia:

- Úprava obsahu web stránky
- Vyriešenie problému s kontaktným formulárom na web stránke
- Napísať všeobecný úvod ohľadom simulácií
- Napísať zhodnotenie analýzy
- Analýza podporných programov pre ns-3
- Špecifikácia požiadaviek, vstupov, výstupov

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 7

**Dátum:** 5.11.2012  
**Čas:** 17:00  
**Miesto:** FIIT STU, blok E, Softvérové štúdio  
**Prítomní:** Ing. Peter Magula  
Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Hrubý návrh:
- Vstupy:
  - Generátor topológie
  - Editor topológie
  - Knižnica - predpripravené skripty a úložisko používateľových topológií a kódov
  - Parametrizovateľnosť skriptu
- Výstupy:
  - .pcap použiť pre TShark
  - .xml, .tr použiť pre FlowMonitor/ iný nástroj na vytvorenie štatistík
- Chybové hlášky kompilátora pre používateľa
- Knižnica samostatne pre každého používateľa
- Zdieľaná knižnica pre všetkých

### Úlohy do ďalšieho stretnutia:

- Prekresliť hrubý návrh (Martin)
- Popísať návrh, ktoré časti budeme implementovať my a ktoré preberieme (Martin)
- Špecifikácia riešenia (Martin)

- Analýza konvertora ns-2 do ns-3 (Jakub)
- Úvod do simulácií (Ivana)
- Zhodnotenie analýzy (Rišo)
- Analýza riešenia predchádzajúceho tímu (Ivana)
- Riadenie projektu (Juraj)
- Organizovať stretnutie a prezentovanie s iným tímom
- Skontrolovať dokument (gramaticky)
- Analýza vstupov a výstupov ns-3 (Viktor)

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 8

**Dátum:** 12.11.2012

**Čas:** 17:00

**Miesto:** FIIT STU, blok E, Softvérové štúdio

**Prítomní:** Ing. Peter Magula  
Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Vytvorenie špecifikácie
- Diskusia ohľadom posúdenia iného tímu: formálna stránka, písmo, označenie obrázkov, ...

### Úlohy do ďalšieho stretnutia:

- Prekresliť hrubý návrh (Martin)
- Popísať návrh, ktoré časti budeme implementovať my a ktoré preberieme (Martin)
- Riadenie projektu (Juraj)
- Organizovať stretnutie a prezentovanie s iným tímom
- Skontrolovať dokument (gramaticky)
- Zjednotiť terminológiu
- Vytvoriť preberací protokol

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 9

**Dátum:** 19.11.2012

**Čas:** 17:00

**Miesto:** FIIT STU, blok E, Softvérové štúdio

**Prítomní:** Ing. Peter Magula  
Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Rozdelenie úloh na vytvorenie posudku:
  - Návrh - Martin
  - Zdroje - Jakub
  - Formálna stránka dokumentu - Juraj (gramatika, postupnosť sekcií, kompletnosť...)
  - Analýza
  - Špecifikácia
- Rozdeliť posudok z vecnej stránky a formálnej stránky
- Vytvoriť preberací protokol

### Úlohy do ďalšieho stretnutia:

- Vytvoriť posudok druhému tímu
- Začať s prototypom - server a web



## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 10

**Dátum:** 26.11.2012

**Čas:** 17:00

**Miesto:** FIIT STU, blok E, Softvérové štúdio

**Prítomní:** Ing. Peter Magula  
Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- Definovanie štábnej kultúry - jazyk použitý v projekte
- Definovanie štandardov pri programovaní
- Analýza bezpečnosti webservera a webstránky
- Vytvorenie preberacieho protokolu

### Úlohy do ďalšieho stretnutia:

- Aktualizovať webstránku (Rišo)
- Pokračovať v práci na prototype
- Spísanie nedostatkov webservera a upovedomenie administrátora (kto bude mať nadmerné množstvo voľného času)
- Napísať vyjavenie k posudku
- Doplniť zoznam obrázkov
- Upraviť referencie k obrázkom

## Zápisnica zo stretnutia tímu UFANS

### Stretnutie č. 11

**Dátum:** 3.12.2012

**Čas:** 17:00

**Miesto:** FIIT STU, blok E, Softvérové štúdio

**Prítomní:** Ing. Peter Magula  
Bc. Martin Čechvala  
Bc. Ivana Hucková  
Bc. Jakub Obetko  
Bc. Richard Roštecký  
Bc. Juraj Šubín  
Bc. Viktor Šulák

### Náplň stretnutia:

- analýza detailov vstupného rozhrania - uzly a ich vlastnosti, pohyb uzlov a vlastnosti - rýchlosť, čas
- mierka, zoom vstupného rozhrania
- výstupy vstupného rozhrania: súradnice každého objektu, spojenia medzi objektami, dosah uzla - diameter, komunikačné toky - zdroj a cieľ (nie graficky - vypísať)

### Úlohy do ďalšieho stretnutia:

- generovanie skriptu do templaty
- spracovanie skriptu ns-3
- vytvorenie templaty
- naplánovať prezentáciu prototypu

## Štandardy kódovania

Dokument k projektu má nasledovné formátovanie:

- okraj zhora 2,5 cm
- okraj zdola 2,5 cm
- okraj sprava 2 cm
- okraj zľava 3,5 cm
- kódovanie UTF8
- typ písma Times New Roman
- veľkosť písma 12
- farba písma čierna
- riadkovanie 1,5

Použité jazyky a nástroje pre implementáciu servera a prototypu:

- Ubuntu 12.04
- Linux 3.2.0-29
- Apache 2.2.22
- PHP 5.3.10-1-3.4
- JavaScript
- balík nástrojov jQuery
- balík nástrojov jQuery UI

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

## **Tím č. 2**

Bc. Martin Čechvala

Bc. Ivana Hucková

Bc. Jakub Obetko

Bc. Richard Roštecký

Bc. Juraj Šubín

Bc. Viktor Šulák

## **Posudok tímu č. 7**

**Analýza, Špecifikácia, Návrh**

Vedúci tímu: Ing. Peter Magula

november 2012

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Posudok z vecného hľadiska</b>	<b>2</b>
2.1	Posudok kapitoly Úvod . . . . .	2
2.2	Posudok kapitoly Analýza . . . . .	2
2.2.1	Posudok kapitoly Analýza problematiky . . . . .	2
2.2.2	Posudok kapitoly Existujúce riešenia v danej oblasti . . . . .	2
2.2.3	Posudok kapitoly Analýza technológií . . . . .	3
2.3	Posudok kapitoly Návrh architektúry . . . . .	3
2.3.1	Posudok kapitoly Webový portál . . . . .	4
2.3.2	Posudok kapitoly Databáza . . . . .	4
2.3.3	Posudok kapitoly Multimedialný portál . . . . .	4
<b>3</b>	<b>Posudok z formálneho hľadiska</b>	<b>5</b>
3.1	Gramatika . . . . .	5
3.2	Použitie slovenských slov . . . . .	7
3.3	Konzistenia formátovania textu . . . . .	7
3.4	Číslovanie strán, obrázkov, kapitol . . . . .	8
3.5	Zdroje . . . . .	9
<b>4</b>	<b>Posudok dokumentácie k riadeniu projektu</b>	<b>10</b>
<b>5</b>	<b>Záver</b>	<b>11</b>

# 1 Úvod

Tento dokument predstavuje posudok k projektovej dokumentácii projektu *Zdielaná identita vo WebRTC doménach* vytvorenej tímom č. 7 v rámci predmetu *Tímový projekt I*. Posudok je členený na tri časti. Prvá časť sa zaoberá hodnotením projektovej dokumentácie z vecného hľadiska, druhá časť hodnotí dokumentáciu z formálneho hľadiska. Posledná časť posudku hodnotí dokumentáciu k riadeniu projektu.

## 2 Posudok z vecného hľadiska

Predložená dokumentácia k projektu pozostáva zo štyroch častí - *Úvod*, *Analýza*, *Návrh architektúry* a *Zdroje*. Chýba časť *Špecifikácia*, ktorá by mala obsahovať špecifikáciu vstupov a výstupov projektu, použité technológie a zdroje, prípadne v skratke zhrnuté možnosti použitia navrhovaného systému.

### 2.1 Posudok kapitoly Úvod

V prvej kapitole (*Úvod*) je predstavené zloženie tímu, zadanie projektu, ciele projektu a prehľad dokumentu, ktorý však nie je v zhode s obsahom tohto dokumentu. V Úvode je tiež umiestnený zoznam použitých skratiek, ktorý by bolo vhodnejšie umiestniť k zoznamu obrázkov a tabuliek.

### 2.2 Posudok kapitoly Analýza

Druhá časť práce je *Analýza*, ktorá je rozdelená na tri celky - *Analýza problematiky*, *Existujúce riešenia v danej oblasti*, *Analýza technológií*. *Analýza problematiky* predstavuje úvod do problematiky a jednoducho oboznámi čitateľa s aktuálnou situáciou v analyzovanej oblasti. Popisuje existujúce problémy aj ich riešenia spolu s konkrétnymi príkladmi použitia v praxi.

Na záver kapitoly *Analýza* chýba zhodnotenie, alebo aspoň stručné zhrnutie obsahujúce informácie o tom, ktoré analyzované prostriedky budú použité pri implementácii riešenia, prípadne zdôvodnenie daného výberu.

Napriek spomenutým nedostatkom obsahuje analýza z vecného hľadiska všetky požadované náležitosti a je vypracovaná na dobrej úrovni.

#### 2.2.1 Posudok kapitoly Analýza problematiky

Ku analýze problematiky nemáme žiadne vecné pripomienky. Obsah bol vyvážený teoretickými aj praktickými informáciami. Bol uvedený správny pomer vysvetlení a príkladov. Všetky dôležité prvky, ktoré súvisia s projektom boli riadne popísané.

#### 2.2.2 Posudok kapitoly Existujúce riešenia v danej oblasti

Kapitola *Existujúce riešenia v danej oblasti* všeobecne popisuje výhody a nevýhody poskytovaných multimediálnych služieb na Internete. Nespomína však žiadne konkrétne existujúce projekty súvisiace s témou. Tiež nie je pravda, že „Webové prehliadače pri poskytovaní takejto služby (pozn. poskytovanie multimediálnych video služieb) musia mať nainštalovanú aktuálnu verziu softvéru tretej strany na zobrazenie videa. Zvyčajne ide o flash player“. Svetlým príkladom je HTML5.

Nadpis *Existujúce riešenia v danej oblasti* nie je úplne korektný, pretože táto kapitola neopisuje riešenia v danej oblasti (komunikácie), ale len v oblasti sprístupňovania obsahu. Napr. TatraBanka minimálne rok implementuje rozhovor s operátorom cez web.

### 2.2.3 Posudok kapitoly Analýza technológií

V kapitole *Analýza technológií* sú opísané hlavné technológie umožňujúce nasadenie WebRTC do reálneho prostredia. Detailne je opísané OpenID spolu s grafickými príkladmi použitia. Taktiež je podrobne analyzované WebRTC, jeho architektúra, komponenty, výhody a nevýhody. Ďalšie časti analýzy tvorí technológia WebSocket, Client-Side JavaScript a Server-Side JavaScript. Všetky časti tejto kapitoly sú dobre spracované a preto k nim nemáme z vecného hľadiska žiadne závažné pripomienky.

Za zmienku stojí asi iba to, že WebSocket nie je TCP protokol, WebSockets je samostatný protokol 7. vrstvy RM ISO OSI. Odporúčame do budúcnosti nečerpať z Wikipédie. Tiež veta „Ako je to s každou novou technológiou, aj Websockety majú svoje nedostatky. Problémom sú niekedy proxy servery, ktoré sú realizované na báze HTTP, pretože po dohodnutí spojenia sa spojenie razom stáva už len TCP spojením, čo sa niektorým proxy serverom nepáči a tak zrušia spojenie.“ nedáva zmysel. Skutočným dôvodom, prečo Websocket nefunguje cez HTTP proxy je ten, že Websockets je úplne iný protokol ako HTTP, a keďže štandardne sa Websocket spojenie otvára na port 80, tak toto spojenie prechádza cez HTTP proxy. Riešením je použiť SOCKS5 proxy alebo HTTP proxy s podporou WebSocket.

Kedže bol analyzovaný protokol WebSocket, do časti AJAX by sme doplnili porovnanie s WebSocket, výhody prenosu cez WebSocket oproti prenosu cez AJAX. Tiež by stálo za uváženie vôbec nepoužiť AJAX a na všetkú asynchrónnu komunikáciu klient <-> server použiť WebSocket.

## 2.3 Posudok kapitoly Návrh architektúry

Pred návrh by bolo vhodné umiestniť špecifikáciu požiadaviek. To znamená, čo má systém vykonávať, aké majú byť vstupy a výstupy systému a pod.

Na úvod návrhu by bolo vhodné umiestniť najprv koncepciu celého riešenia. Priam nevyhnutný je architekturný koncepčný obrázok, ktorý by popisoval celé riešenie (jeden obrázok je niekedy hodný tisíc slov). Tiež by bolo ideálne popísať jednotlivé komponenty takéhoto obrázku.

Úvod návrhu naopak vyzerá ako prípad použitia, respektíve Flow Chart diagram prepísaný do textu. Toto nie je nevhodné, ale nemyslíme si, že by mal byť tento opis akcií používateľa a reakcií systému v úvode návrhu.



### 2.3.1 Posudok kapitoly Webový portál

Prvá časť podkapitoly (Autentikácia) je časť, ktorá by mala byť v analýze. V návrhu bolo potrebné popísať ako bude fungovať autentifikácia v kontexte systému (kde a ako budú uložené autentifikačné údaje, konkrétny model, ktorý sa bude implementovať, akým spôsobom budú šifrované používateľské údaje). Zvyšok kapitoly je v poriadku.

### 2.3.2 Posudok kapitoly Databáza

V kapitole Databáza by bolo vhodné do budúcnosti umiestniť logický a fyzický databázový model.

### 2.3.3 Posudok kapitoly Multimediálny portál

Použitie WebSocket je zdôvodnené nedostatočne. Akým spôsobom pomôže vlastnosť multiplexovania tomu, aby systém dokázal obslúžiť viac klientov? Multiplexovanie vo WebSocket funguje iba medzi klientom a serverom. Ak má jeden klient viac spojení v HTTP (tj. viac TCP spojení), toto možno implementovať vo WebSocket ale jedno TCP spojenie - tj. multiplexovať. Ak ale je na server pripojených 100 klientov a každý má jedno spojenie, tak sa stráca význam použiť WebSocket iba z dôvodu multiplexovania. Tento odstavec hodnotíme ako nepodložený a bez jasných dôvodov.

Pozitívne ale hodnotíme obrázok *Obr. č.8 Návrh architektúry*, aj keď je veľmi vysokoúrovňový.

### 3 Posudok z formálneho hľadiska

Posudzovaný dokument je prehľadný a vhodne štrukturovaný. Obsahuje titulnú stranu, obsah, zoznam obrázkov, zoznam tabuliek a samotný text dokumentu ukončený odkazmi na použité zdroje. Text je vhodne členený do kapitol a podkapitol maximálne tretej úrovne, čo je pre dokument tohto rozsahu (25 strán) vhodné.

Dokument vyhovuje požiadavkám na základnú štruktúru (Úvod, Analýza, Návrh architektúry, Zdroje), z obsahovej stránky však chýba časť Špecifikácia. Z formálneho hľadiska by mal dokument spĺňať aj ostatné náležitosti ako gramatika, použitie slovenských slov, konzistencia formátovania textu, číslovanie strán, číslovanie kapitol, číslovanie obrázkov, formát odkazov na zdroje. Preto tieto náležitosti postupne popíšeme.

#### 3.1 Gramatika

Nakoľko dokument je oficiálnym dokumentom k tímovému projektu, mal by obsahovať gramaticky správny text. V texte dokumentácie sa nachádza pomerne veľké množstvo gramatických chýb. Prvým typom chýb sú preklepy:

„Akademický. rok“ (úvodná strana), „skrátkach“ (kap. 1, s. 1), „používateľa“ (kap.1.2, s. 1), „havne“ (kap. 1.2.1, s. 1), „umožňujúca“ (kap. 1.2.1, s. 1), „učtovanie“ (kap. 1.2.1, s. 1), „používateľa“ (kap.1.3, s. 2), „Comunication“ (kap. 1.3.2, s. 3), „technológii“ (kap. 2, s. 4), „vytvaraní“ (kap. 2, s. 4), „tímového“ (kap. 2, s. 4), „identitu pod ktorou“ (kap. 2.1, s. 4), „organizácií“ (kap. 2.1, s. 4), „ku overeniu“ (kap. 2.1, s. 4), „heslo“ (kap. 2.1, s. 4), „o delegovaní“ (kap. 2.1, s. 5), „Príklad je ilustrovaná“ (kap. 2.1, s. 5), „komunikácií“ (kap. 2.1, s. 5), „užívateľských“ (kap. 2.1, s. 5), „odvisievanej“ (kap. 2.2, s. 7), „odvisievaná“ (kap. 2.2, s. 7), „Analýza technológii“ (kap. 2.3, s. 8), „technológii“ (kap. 2.3, s. 8), „implementáciach“ (kap. 2.3, s. 8), „používateľom“ (kap. 2.3.1, s. 8), „vytvaraní“ (kap. 2.3.2, s. 11), „vývojarske“ (kap. 2.3.2, s. 11), „ktróá“ (kap. 2.3.2, s. 11), „vidieťarchitektúru“ (kap. 2.3.3, s. 13), „dva vrstvy“ (kap. 2.3.3, s. 13), „nezavislý“ (kap. 2.3.4, s. 14), „Všetka“ (kap. 2.3.4, s. 14), „pokial“ (kap. 2.3.4, s. 14), „a ja“ (kap. 2.3.5, s. 16), „farmeworky“ (kap. 2.3.5, s. 17), „zamestatný“ (kap. 2.3.6, s. 19), „spoločnosťou“ (kap. 2.3.6, s. 19), „realtimových“ (kap. 2.3.6, s. 19), „aplikácií“ (kap. 2.3.6, s. 19) (4x), „najmá“ (kap. 2.3.6, s. 19), „optimalizácií“ (kap. 2.3.6, s. 19), „rozrahníe“ (kap. 2.3.6, s. 19), „robit“ (kap. 2.3.6, s. 19), „problemóv“ (kap. 2.3.6, s. 19), „chvilu“ (kap. 2.3.6, s. 19), „prehravať“ (kap. 2.3.6, s. 19), „mame“ (kap. 2.3.6, s. 19), „pômožu“ (kap. 2.3.6, s. 19), „umožňujú“ (kap. 2.3.6, s. 20), „všetka“ (kap. 2.3.7, s. 20), „pride“ (kap. 3, s. 21), „svojim“ (kap. 3, s. 21), „užívateľ“ (kap. 3, s. 21), „identifikuje“ (kap. 3, s. 21), „príjmať“ (kap. 3.1.1, s. 22), „účtovníctvo“ (kap. 3.1.1, s. 22), „Databáze“ (kap. 3.2, s. 23), „účtovníctvo“ (kap. 3.2, s. 23)

(2x), „implementácií“ (kap. 3.3, s. 23), „näm“ (kap. 3.3, s. 23), „byt“ (kap. 3.3, s. 23).

Druhým typom chýb je vynechanie čiarky:

- „stránky respektíve organizácie“ (kap. 2.1, s. 4),
- „meno a heslo uložené v centrálnej databázovej štruktúre voči ktorému“ (kap. 2.1, s. 4),
- „na jednom mieste pričom dôjde“ (kap. 2.1, s. 5),
- „nie sú vlastníkmí a tam ak nemajú“ (kap. 2.2, s. 7),
- „kvalitu čo znamená“ (kap. 2.2, s. 7),
- „najmä vtedy ak má“ (kap. 2.2, s. 7),
- „jeden profil a ten“ (kap. 2.3.1, s. 8),
- „prostredníka a tým“ (kap. 2.3.1, s. 8),
- „využíva OpenID sa vymení“ (kap. 2.3.1, s. 8),
- „profilu teda“ (kap. 2.3.1, s. 10),
- „vyhladzovanie snímánie obrazu a to“ (kap. 2.3.3, s. 13),
- „nepáči a tak“ (kap. 2.3.4, s. 15),
- „vybral pretože“ (kap. 2.3.6, s. 19),
- „Node.js používa JavaScript je práca“ (kap. 2.3.6, s. 19),
- „ktorý hovorí o Node.js je“ (kap. 2.3.6, s. 19),
- „kontom. ktoré“ (kap. 3, s. 21),
- „obsah zobrazíť sa“ (kap. 3, s. 21).

Tretím typom chýb je čiarka navyše v texte:

- „Úvod do dokumentu, a prehľad“ (kap. 1.3.1, s. 2),
- „tým pádom, nie je“ (kap. 2.3.1, s. 8),
- „heslo, alebo napríklad“ (kap. 2.3.1, s. 8).

Nakoniec sa ešte našli dva prípady, kedy chýbalo slovo vo vete:

- „kodek \_\_\_\_ úzkopásmovú komunikáciu“ (kap. 2.3.3, s. 14),

- „platnosť niekoľko \_\_\_\_\_“ (kap. 3, s. 21),

a jeden prípad, kedy bola v zátvorke bodkou ukončená veta, ktorá ďalej pokračovala:

- „(pokiaľ sa nedohodlo ináč.),“ (kap. 2.3.4, s. 14).

## 3.2 Použitie slovenských slov

Text dokumentu by mal obsahovať spisovné slová slovenského jazyka. Dokument ale obsahuje niekoľko slov prevzatých z anglického jazyka, ktoré majú slovenský ekvivalent. Použitie takýchto slov v texte je preto nesprávne. Nájdené prípady:

- „real-time“ (kap. 1.2, s. 1),
- „hashované“ (kap. 2.1, s. 4),
- „username“ (kap. 2.1, s. 4),
- „provider“ (a jeho pády) (kap. 2.1, s. 5; kap. 2.3.1, s. 10),
- „plug-in“ (a jeho pády) (kap. 2.1, s. 5; kap. 2.3.2, s. 11; kap. 2.3.3, s. 12),
- „stand-alone“ (kap. 2.1, s. 5),
- „streamy“ (kap. 2.2, s. 7),
- „framework“ (kap. 2.3.1, s. 8; kap. 2.3.5, s. 17; kap. 2.3.6, s. 20; kap. 3.1.1, s.22),
- „full-duplex“ (kap. 2.3.4, s. 14),
- „zahashuje“ (kap. 2.3.4, s. 15),
- „plain“ (kap. 2.3.5, s. 18),
- „engine“ (kap. 2.3.6, s. 19).

## 3.3 Konzistenia formátovania textu

V celom dokumente by mal byť použitý jednotný štýl formátovania textu, či už sa jedná o odsadenie textu na začiatku odseku, formát písania rovnakých slov, typ písma, veľkosť písma, formátovanie textu pri vymenovávaní a pod.

Typ a veľkosť písma sú v celom dokumente jednotné, to je v poriadku. V obsahu je kapitola 2.2 *Existujúce riešenia v danej oblasti* odsadená viac ako ostatné. Odsadenie textu na začiatku odsekov nie je jednotné, v dokumente sa nachádzajú odseky, ktoré majú odsadený text, ďalej odseky, ktoré nemajú

odsadený text (väčšina), a nakoniec odsek, ktorý ma dvakrát odsadený text (kap. 2.3.4, s. 14).

Našli sme nezrovnalosti pri písaní rovnakého slova (Internet), raz s malým „i“ a raz s veľkým „I“, napríklad:

- „po internete“ (kap. 2.1, s. 4),
- „na Internete“ (kap. 2.2, s. 7),
- „ku internetu“ (kap. 2.2, s. 7),
- „Internetu“ (kap. 2.3, s. 8).

Rovnaká chyba bola nájdená aj pri slove „HyperText“, ktoré je raz napísané s malým „t“ a raz s veľkým „T“ (kap. 1.3.2, s. 3).

Nekonzistencia bola nájdená aj pri vymenovávaní vrstiev štruktúrneho modelu webovej stránky (kap. 2.3.5, s. 16), kde pri dvoch možnostiach sa nachádza priblíženie obsahu danej vrstvy v zátvorke hneď za názvom:

- „Štruktúrna vrstva (obsahová)“,
- „Funkčná vrstva (správania)“,

a pri tretej možnosti sa toto priblíženie nenachádza, resp. je zadané až v textovom bloku bližšie popisujúcim danú vrstvu:

- „Prezentačná vrstva – alebo vrstva štýlu...“.

Treba podotknúť, že táto nezrovnalosť nemá vplyv na výpovednú hodnotu textu.

### 3.4 Číslovanie strán, obrázkov, kapitol

V podobných dokumentoch je zaužívané, že obsah sa čísluje malými rímskymi písmenami a samotný text arabskými číslicami. Úvodná strana sa nečísluje. Tu nastala drobná chyba, keďže úvodná strana má číslo i, ktoré mal mať až obsah. Samotný text má správne číslovanie strán.

Pre číslovanie obrázkov sa najčastejšie používajú dva formáty. Buď sa obrázky číslujú od čísla 1 ďalej, pričom číslovanie nadväzuje v celom dokumente, alebo sa číslujú vo formáte *číslo\_kapitoly.číslo\_obrázku*. V texte dokumentu sú obrázky číslované prvým spomenutým formátom, v tabuľke obrázkov sú číslované druhým spomenutým formátom. Ďalšou podstatnou náležitou požiadavkou použitia obrázku v dokumente je odvolanie sa na daný obrázok v texte. Obrázok, ktorý nie je v texte spomenutý, ako by tam ani nebol. Obrázky číslo 1,3,5,6 a 7 sú spomenuté v texte. Na obrázky č. 2 a 8 nie je v texte odvolanie. Obrázok č. 4 má v texte odvolanie, ktoré ale obsahuje číslo 5. Čísla strán obrázkov uvedené v zozname obrázkov sa až na jednu výnimku (č. 1) nezhodujú s číslami strán,

na ktorých sa dané obrázky nachádzajú v dokumente. Nekonzistencia formátu popisu obrázkov nastáva pri obrázku *Obr. 4 Architektúra WebRTC*. Pri popise ostatných obrázkov je použitý formát *Obr. č. číslo\_obrázku*. Pri obrázku *Obr. 4* je však tento formát porušený.

Kapitoly sa číslujú buď od nuly alebo od jednotky. Kapitola Úvod má v obsahu číslo 1, v prehľade dokumentu (kap. 1.3.1, s. 2) je ale napísané: „Kapitola 0: Úvod do dokumentu, a prehľad jeho súčastí“. Našli sme aj chybu v číslovaní kapitol, kedy kapitola *CouchDB* je označená číslom 2.2.7 namiesto 2.3.7. Až na tieto dve nezrovnalosti je číslovanie v poriadku. Takisto čísla strán kapitol uvedené v obsahu zodpovedajú číslam strán, na ktorých sa dané kapitoly v dokumente nachádzajú.

### 3.5 Zdroje

Bibliografické odkazy sú z formálneho hľadiska správne štrukturované podľa normy STN ISO 690. Nakoľko uvedené zdroje sú väčšinou oficiálne webové stránky prezentujúce použité technológie, osoba autora sa neudáva. V každom bibliografickom odkaze je namiesto dátumu vzniku daného zdroj, dátum posledného prístupu k zdroju. Toto je v poriadku, nakoľko norma pri elektronických zdrojoch presne neudáva, ktorý dátum sa má použiť v štruktúre bibliografického odkazu.

Formátovanie zdrojov nebolo úplne jednotné, odkaz na webovú stránku bol pri troch zdrojoch (1,2,7) nepodčiarknutý a napísaný čiernym písmom, ale pri ostatných zdrojoch (3,4,5,6) bol odkaz na webovú stránku podčiarknutý a napísaný modrým písmom. Nastáva tu nekonzistencia formátovania zdrojov.

Po prezretí dokumentu sa zistili nasledovné nedostatky súvisiace so zdrojmi:

- Zdroj [1] nie je v texte citovaný ani naň nie je v texte uvedený žiaden odkaz. Po prezretí obsahu zdroja sa predpokladá, že z neho bol prebratý obrázok *Obr. 4 Architektúra WebRTC*.
- Zdroj [2] nie je v texte citovaný ani naň nie je v texte uvedený žiaden odkaz. Po prezretí obsahu zdroja sa predpokladá, že z neho bol prebratý obrázok *Obr. č. 5 Signalizácia a WebRTC*.
- Zdroj [3] takisto nie je v texte citovaný ani naň nie je v texte uvedený žiaden odkaz. Po prezretí obsahu zdroja sa predpokladá, že z neho bol prebraný obrázok *Obr. č. 6 Komunikácia cez Websockety*.

Ostatné zdroje (4,5,6,7) sú z tohto hľadiska v poriadku, v texte sa na každý z nich nachádza aspoň jeden odkaz. Všetky zdroje reálne existujú, aj odkazy na stránky sú platné.

Z vecného sú použité zdroje relevantné, úzko súvisia s témou projektu. Odkazy na zdroje sú vhodne umiestnené v texte a umožňujú čitateľovi textu bližšie sa zoznámiť s danou technológiou spomínanou v texte.

## 4 Posudok dokumentácie k riadeniu projektu

Dokument o riadení projektu obsahuje všetky požadované náležitosti - úvod, ponuka, plán projektu, záznamy zo stretnutí a preberací protokol. Dokument je vhodne štrukturovaný a prehľadný.

Kapitola *Úvod* stručne opisuje účel a obsah dokumentu. Kapitola *Ponuka* obsahuje všetky potrebné časti - zadanie projektu, predstavenie členov tímu, motiváciu, cieľ projektu a úlohy jednotlivých členov tímu. Záznamy zo stretnutí sú detailné, každé obsahuje dátum, čas stretnutia, miesto a zoznam prítomných členov tímu. Každé stretnutie obsahuje zoznam úloh určených na vykonanie, ako aj kontrolu úloh z predchádzajúceho stretnutia. Jednotlivé úlohy sú pridelované členom tímu a prehľadne kontrolované pomocou *statusu* úlohy.

Jediným negatívom tohto dokumentu je po formálnej stránke prázdna očíslovaná strana (strana č. 3) a taktiež prázdny priestor v rámci predstavenia členov tímu (strana č. 6). Gramatické chyby v texte taktiež uberajú z kvality dokumentu.

## 5 Záver

Posudok tímu č. 7 sme rozdelili na štyri kapitoly. V prvej kapitole sme sa zaoberali vecným hľadiskom projektu a to konkrétnejšie predstavením tímu a analýzou riešenej problematiky. V tejto časti sme objavili len malé nedostatky, ktoré sa dajú ľahko opraviť. Všeobecne by sa ale dala táto časť zhodnotiť ako zvládnutá a tím č. 7 svojimi vedomosťami dokázal spracovať na dobrej úrovni celú analýzu. Ďalšou časťou, ktorou sme sa zaoberali bol posudok z formálneho hľadiska. Vzhľadom na to, že toto je oficiálny dokument k tímovému projektu, dôkladne sme skontrolovali gramatickú stránku, štylistickú stránku a v neposlednom rade aj formátovanie dokumentu. Tu sa objavilo množstvo chýb, ktoré by dokument daného typu nemal obsahovať. Po tejto stránke považujeme dokument za nie veľmi dobre zvládnutý a odporúčame sa takýmto chybám v budúcnosti vyhýbať. Dokument má byť svojím spôsobom vizitkou tímu a má reprezentovať celý projekt. V poslednej časti sme sa zamerali na riadenie projektu. Riadenie projektu je až na menšie nezrovnalosti v poriadku, ale tiež sa nezaobišlo bez gramatických chýb.



Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

Tím 7

**Oponentský posudok**

*Tímový projekt*

**Číslo tímu:** 7

**Členovia tímu:** Bc. Šimon Šiplák, Bc. Maroš Medvec, Bc. Marián Sekeráš, Bc. Adam Števko, Bc. Roman Zelenaj

**Vedúci tímového projektu:** Ing. Ján Murányi

**Ročník, typ štúdia:** 1, inžinierske štúdium

**Akademický. rok:** 2012/2013

## **Obsah**

<b>Obsah .....</b>	<b>ii</b>
<b>1 Úvod .....</b>	<b>1</b>
<b>2 Zhodnotenie .....</b>	<b>2</b>
<b>2.1 Dokumentácia k projektu .....</b>	<b>2</b>
2.1.1 Úvod .....	2
2.1.2 Simulácia v bezdrôtových sieťach .....	2
2.1.4 Analýza predchádzajúceho projektu .....	2
2.1.5 Celková analýza .....	3
2.1.6 Špecifikácia .....	3
2.1.7 Návrh riešenia .....	3
2.1.7 Formálna stránka dokumentu .....	3
<b>2.2 Dokumentácia riadenia k projektu .....</b>	<b>4</b>
<b>3 Záver .....</b>	<b>5</b>

# 1 Úvod

Tento dokument je výsledkom práce tímu č.7 v rámci predmetu Tímový projekt na Fakulte informatiky a informačných technológií a zaoberá sa hodnotením dvoch dokumentácií tímu číslo 2: dokumentácia k projektu a dokumentácia k riadeniu projektu.

Dokument riadenia hovorí o aktivitách tímu ako celku a jeho jednotlivých členov na projekte počas 8 týždňov v zimnom semestri akademického roka 2012/2013. Pri tomto dokumente sme sa skôr zamerali na formálnu stránku a kontrolovali sme prítomnosť dôležitých kapitol.

V dokumentácií k projektu sme sa zamerali viac na obsah a súvislosť jednotlivých kapitol a takisto konkrétne pri špecifikácií a návrhu riešenia. V tomto dokumente sme menšou mierou hodnotili formu a jazykový prejav

Spomínané dokumentácie sme si prečítali a pokúsili sme sa o objektívnu kritiku a zhodnotenie.

## **2 Zhodnotenie**

V prvej časti hodnotíme dokumentáciu k projektu, ktorá je štruktúrovaná na časti podľa jednotlivých kapitol daného dokumentu. V závere je hodnotenie formálnej stránky dokumentu, ktorej sme prikladali v tomto dokumente menší dôraz.

V druhej časti zhodnotenia je hodnotená dokumentácia k riadeniu projektu, kde sme sa zamerali na dodržiavanie formálnej stránky dokumentu a tiež či daný dokument obsahuje všetky potrebné náležitosti pre identifikovanie práce jednotlivých členov na projekte.

### **2.1 Dokumentácia k projektu**

#### **2.1.1 Úvod**

Úvod do daného dokumentu vzhľadom na cieľ tejto kapitoly považujeme obsahovo za nedostatočný. V úvode je opísaná štruktúra dokumentu vyhovujúco a zároveň stručne, na druhej strane nám však veľmi chýba zadanie a ciele aké má tento dokument a samotný projekt spĺňať. Ako čitatelia dokumentu by sme pri čítaní úvodu chceli vedieť, aké sú stručné ciele tohto projektu, zadanie a čo konkrétne chce daný tím vytvoriť.

#### **2.1.2 Simulácia v bezdrôtových sieťach**

Táto kapitola je písaná štylisticky a obsahovo na dobrej úrovni a čitateľ tak môže získať dobrý prehľad o danej problematike. Kapitola je aj dobre čitateľná pre človeka, ktorý sa v daných technológiach nevyzná.

#### **2.1.4 Analýza predchádzajúceho projektu**

Oponentský tím nedisponoval všetkými potrebnými informáciami a dokumentmi na analýzu už existujúceho riešenia. Zo získaných údajov sa im však podarilo dobre opísať fungovanie predchádzajúceho projektu a čitateľom dáva ucelený prehľad o danej aplikácii.

### 2.1.5 Celková analýza

Analýzu sieťových simulátorov ns-2 a ns-3 považujeme za jednu zo slabších strán tohto dokumentu. V týchto analýzách považujeme viacero častí za irelevantné k riešeniu daného projektu. Daná analýza sa nám pri čítaní v niektorých častiach spájala s opisom funkcionality a návodom na použitie daných programov. Za hrubšiu chybu tiež považujeme písanie fragmentov zdrojových kódov do analýzy, nakoľko analýza má ponúknuť čitateľovi náhľad ako dané technológie fungujú a nie samotný návod na konfiguráciu daných technológií, používanie aplikácii alebo prehľad manuálových stránok k programom.

- Obrázok 4.1 je po anglicky. Vyzerá, že je stiahnutý z internetu - nie je uvedený zdroj a ani to, čo znamenajú skratky NAM Xgraph.
- Obrázky 4.2, 5.2, 5.3 sú po anglicky a mali by byť preložené, keďže dokumentácia je v slovenčine. Nemajú uvedený zdroj, ak sú stiahnuté. Zdroje obrázkov by mali byť uvedené - preložené z.
- Obrázok 4.3 je po anglicky.
- Obrázok 5.11 - Raz sa zdroj označuje ako [číslo] a raz [link]. Toto navrhujeme zjednotiť pri všetkých obrázkoch.

### 2.1.6 Špecifikácia

Špecifikácia ponúka čitateľovi dobrú možnosť získania predstavy o výslednom produkte daného tímu. V kapitole je jasne a výstižne popísané funkcie danej výslednej aplikácie. Aj keď je táto kapitola na dobrej úrovni čitateľ získal až pri jej čítaní šancu vedieť o tom, čo je cieľom tohto projektu.

### 2.1.7 Návrh riešenia

K tejto kapitole nemáme žiadne pripomienky. Je vhodne rozdelená a primerane popisuje hrubý návrh riešenia.

### 2.1.8 Formálna stránka dokumentu

Aj keď sme na formálnu stránku tohto dokumentu neprikladali najväčší dôraz, podarilo sa nám nájsť viacero nezrovnalostí.

- Nejednotná forma referencií pre obrázky v dokumente. Niekde sú zobrazené ako referencia inokedy ako link k obrázku. V iných prípadoch je referencovaný text namiesto obrázka.

- Pri niektorých obrázkoch chýba referencia (4.2 a 4.3).
- V dokumente sa tiež nachádzajú občasné gramatické chyby.
- Dĺžka dokumentu sa nám nezdá byť primeraná a daný dokument je príliš dlhý a obsahuje nerelevantné časti. Ide hlavne o kapitolu Analýza.
- Chýba zoznam obrázkov.

## **2.2 Dokumentácia riadenia k projektu**

Dokumentácia k riadeniu projektu je vypracovaná na dobrej úrovni. Avšak, z nášho pohľadu by si dokumentácia ešte zaslúžila lepšie spracovanie zápisníc, nakoľko v niektorých nie je uvedené komu je daná úloha pridelená a tiež stav jednotlivých úloh z minulých stretnutí.

### **3 Záver**

Cieľom tohto dokumentu bolo zhodnotiť dokumentáciu projektu a riadenia tímu č. 2 na predmete Tímový Projekt I. Dokumentácia projektu je vypracovaná dobre a ponúka čitateľovi dobrý, aj keď príliš rozsiahly úvod do danej problematiky. Medzi tieto časti patria najmä duplikovanie informácií, ktoré nájdeme v dokumentácií k jednotlivým programom a úkazy vzorových programov. Špecifikácia a návrh riešenia je viac abstraktný a všeobecný, čo dáva slabšiu predstavu o tom, ako bude prototyp implementovaný.

Dokument projektu aj s dokumentom riadenia sú v súlade s požiadavkami na dokumentáciu, ktoré sú uvedené na webovej stránke predmetu TP1. Dokumenty sú dobre štruktúrované aj keď za nie príliš vhodné riešenie považujeme spojenie do jedného dokumentu. Celkovo hodnotíme dokumenty tímu č.2 za veľmi dobré.

## Vyjadrenie k posudku

Po preštudovaní prijatého posudku a analyzovaní jednotlivých pripomienok, ktoré nám boli vytknuté tímom č. 7, sme sa zhodli na podniknutí nasledovných krokov:

### Úvod

Obsah tejto kapitoly je podľa nášho názoru úplne v poriadku a nemyslíme si, že práve v úvode sa ma nachádzať zadanie a ciele projektu. Práve naopak, ich prítomnosť v kapitole "Úvod" sa nám zdá neprimeraná, keďže patrí do časti dokumentácie k riadeniu projektu. Z tohto dôvodu túto pripomienku nezapracujeme.

### Celková analýza

Analýzu sieťových simulátorov ns-2 a ns-3 sme vypracovali veľmi podrobne hlavne kvôli tomu, aby nám uľahčila prácu v budúcnosti pri implementácii riešenia. Keďže sme pri písaní analýzy ešte nevedeli, ktorý zo sieťových simulátorov použijeme, boli oba detailne analyzované. Ukážky zdrojových kódov v analýze sme tiež použili zámerne, aby sme vedeli porovnať zložitosť podporných nástrojov oboch simulátorov. Je potrebné podotknúť, že analýzu vytvárali členovia tímu v dvojiciach a preto mala každá dvojica podrobný prehľad iba o tej časti analýzy, ktorú spracovala. Po prečítaní analýzy ostatných dvojíc však ľahko získala podrobný prehľad o celej analýze k projektu. Preto nesúhlasíme s výrokom, že analýza obsahuje irelevantné časti.

Za relevantné pripomienky považujeme chýbajúce zdroje k obrázkom a fakt, že niektoré obrázky sú po anglicky. Tieto nedostatky odstránime. Taktiež upravíme formát referencií na jednotný.

### Formálna stránka dokumentu

Vyššie spomínané referencie pri obrázkoch upravíme a zjednotíme, k obrázkom, pri ktorých chýba referencia, bude pridaná. Taktiež doplníme zoznam obrázkov.

Nesúhlasíme s výrokom, že v dokumente sa nachádzajú občasné gramatické chyby. Celý dokument bol niekoľkokrát kontrolovaný členmi tímu a na túto kontrolu bola vytvorená špeciálna funkcia pre jedného člena tímu.

Dĺžka dokumentu je podľa nášho názoru primeraná, vzhľadom na vyššie spomínané dôvody podrobnej analýzy. Opäť nesúhlasíme s výrokom, že dokument obsahuje irelevantné časti.

### Dokumentácia k riadeniu projektu

Jedinou pripomienkou k tejto časti projektu bolo lepšie spracovanie zápisníc - konkrétne chýbajúce pridelenie úloh konkrétnemu členovi a kontrola stavu úloh



z predchádzajúcich stretnutí. Pri úlohách, ktoré nemajú prideleného riešiteľa nebolo v čase stretnutia jasné, kto konkrétne ich bude riešiť. Táto otázka sa často zodpovedala až neskôr, vzhľadom na iné úlohy jednotlivých členov. Meniť znenie zápisnice dodatočne je podľa nášho názoru nesprávne a neprofesionálne, preto sme ich nechali presne v tom stave, v akom boli po skončení stretnutia. Stav jednotlivých úloh sa považuje za dokončený, ak sa úloha neopakuje v dvoch zápisniciach za sebou.

### **Zhodnotenie**

Ako bolo spomenuté vyššie, niektoré z prijatých pripomienok sme zapracovali do dokumentu. Naopak, iné považujeme za celkom irelevantné a preto sme sa ich rozhodli do dokumentu nezapracovať. Celkovo hodnotíme posudok tímu č. 7 za primeraný, obsahoval niekoľko cenných rád, vďaka ktorým sme dokázali vylepšiť kvalitu nášho projektu.

## Manažment verzií, konfigurácií a zmien

Prostriedky pre podporu manažmentu verzií, konfigurácií a zmien používané naším tímom sú:

- **SVN** (Subversion) - systém, vďaka ktorému uchovávame zdrojové kódy a kompletnú dokumentáciu (tj. všetky dáta projektu) v centrálnom re-  
pozitári. Každý člen tímu má nezávislý prístup do repozitára. SVN nám poskytuje možnosť úplne nezávislého vývoja.
- **Trac** - Kolaboračný nástroj pre tímy, je odvodený od MediaWiki softvéru. V tomto systéme uchovávane všetky informácie o projekte, odkazy na súvisiace zdroje, všetko v prehľadne organizovanej štruktúre, ktorú samostatné stránky poskytujú. Navyše je v Tracu možné vytvárať a priradovať „tickety“ obsahujúce zadanie práce, vykonávateľa, dátum a termín. Takto môžeme presne sledovať ako sa projekt vyvíja. Trac sme integrovali s SVN a je možné cez Trac prehliadať SVN repozitár. Trac tiež podporuje zaznamenávanie „roadmap“, čo je plán funkcií pre celý projekt a „milestones“, dôležité celky. V systéme si každý člen tímu vedie aj svoj projektový denník.
- **Mailing list** - Na účely komunikácie v rámci tímu sme vytvorili interný mailing list. Mailing list funguje tak, že keď jeden člen do neho pošle e-mail, ostatní ho dostanú. Zároveň si mailing list udržiava vetvenú históriu všetkých správ. Tento nástroj slúži na komunikáciu rozsiahlejších rozmerov a diskusie.
- **LaTeX** - Tvorba dokumentácie je realizovaná v jazyku LaTeX. Vďaka spôsobu jeho písania a možnosti rozdeľovať ho na viacero súborov vieme naraz pracovať na jednom dokumente aj všetci. LaTeXový dokument máme samozrejme uložený v SVN. Výsledný dokument je potom vo formáte pdf, ktorý je vyhovujúci aj pre vedúceho tímu na pridávanie poznámok a komentárov k dokumentu.

# Preberací protokol

Tím č. 2 - UFANS

Vedúci tímu: Ing. Peter Magula

Členovia tímu:

- Bc. Martin Čechvala
- Bc. Ivana Hucková
- Bc. Jakub Obetko
- Bc. Richard Roštecký
- Bc. Juraj Šubín
- Bc. Viktor Šulák

Tím č. 7,

v zastúpení \_\_\_\_\_, svojím podpisom potvrdzuje prebratie dokumentácie k tímovému projektu od tímu č. 2, ktorá pozostáva z dokumentu "Dokumentácia k projektu" v počte strán \_\_\_\_\_.

V Bratislave, dňa \_\_\_\_\_

\_\_\_\_\_  
Podpis