



Programová podpora pre sieťový simulátor

Tímový projekt I

Členovia tímu: Bc. Hana Severínová, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka, Bc. Martin Pirháč, Bc. Martin Svetlík
Študijný program: Počítačové a komunikačné systémy a siete
Študijný odbor: 9.2.4 Počítačové inžinierstvo
Miesto vypracovania: Ústav počítačových systémov a sietí, FIIT STU Bratislava
Vedúci projektu: Ing. Peter Magula
Akademický rok: 2010/2011

Obsah

Zoznam obrázkov	III
Zoznam tabuliek	III
Zadanie	4
1 Úvod	5
2 Simulácia v počítačových sieťach	6
3 Simulátory počítačových sietí	8
3.1 Network simulator 2	8
3.1.1 Architektúra ns-2	8
3.1.2 Simulácie pomocou ns-2	10
3.1.3 Podporné nástroje pre ns-2	17
3.1.4 Zhodnotenie ns-2	24
3.2 Network simulator 3	26
3.2.1 Architektúra ns-3	26
3.2.2 Simulácia pomocou ns-3	27
3.2.3 Podporné nástroje pre ns-3	33
3.2.4 Zhodnotenie ns-3	36
3.3 Zhodnotenie analýzy simulátorov	38
4 Špecifikácia a hrubý návrh riešenia	39
4.1 Používatelia	40
4.2 Popis komponentov	41
5 Záver	44
6 Bibliografia	45
Príloha A	1

Zoznam obrázkov

Obrázok 1: Zjednodušená architektúra simulátora ns-2.....	9
Obrázok 2: Organizácia modulov ns-2.34 v rámci inštalačného balíka allinone.....	9
Obrázok 3: Organizácie sieťového zásobníka mobilného uzla implementovaného v ns-2.34	11
Obrázok 4: Architektúra programu NS2 TCP Evaluation Tool.....	18
Obrázok 5: Ukážka výstupného grafu simulácie programu NS2 TCP Evaluation Tool.....	19
Obrázok 6: Ukážky grafov z programu Gnuplot	20
Obrázok 7: Ukážka programu XGRAPH	21
Obrázok 8: Ukážky rozhrania programu Ns-nam	22
Obrázok 9: Ukážka rozhrania programu iNSpect v OS Ubuntu	23
Obrázok 10: Ukážka grafu z programu DrawNetwork	24
Obrázok 11: Schéma simulátora ns-3	26
Obrázok 12: Príklad bezdrôtovej topológie	27
Obrázok 13: Grafické rozhranie programu NetAnim	33
Obrázok 14: Grafické rozhranie vizualizéra PyViz	34
Obrázok 15: Grafické rozhranie programu	35
Obrázok 16: Schéma štatistického systému	36
Obrázok 17: Porovnanie funkcionalít implementovaných v jadrách simulátorov ns-2 a ns-3	38
Obrázok 18: Architektúra riešenia	40
Obrázok 19: Podrobná schéma riešenia	41

Zoznam tabuliek

Tabuľka 1: Zoznam protokolov podporovaných v jadre ns-2.....	25
--	----

Zadanie

Analyzujte možnosti súčasných sieťových simulátorov pričom sa zamerajte na popredné simulátory ns2 a ns3. Analýzu upriamte na simulačné možnosti vybraných typov počítačových a komunikačných sietí so zameraním sa na prípravu a vyhodnotenie prebehnutých simulácií. Sústreďte sa na typické požiadavky vyhodnocovania simulácií vybraných typov sietí. Na základe vykonanej analýzy navrhните a následne implementujte systém, ktorý používateľovi umožní ľahšiu prípravu a vyhodnotenie prebehnutých simulácií. Systém musí poskytovať čo najuniverzálnejšie prostredie pre vyhodnocovanie a zobrazovanie priebehov a štatistík typických parametrov sledovaných pri simuláciách vybraných typov sietí.

1 Úvod

Tento dokument vznikol ako projektová dokumentácia k predmetu Tímový projekt 1na Fakulte informatiky a informačných technológií Slovenskej technickej univerzity v Bratislave. Je súčasťou projektu s názvom Programová podpora pre sieťový simulátor, vedeného Ing. Petrom Magulom. Vypracovaný bol na Ústave počítačových systémov a sietí.

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

Dokument obsahuje v prvej časti analýzu simulátorov ns-2 a ns-3, špecifikáciu a hrubý návrh riešenia. Kapitoly 0 a 2 predstavujú úvod do problematiky simulácií. Nasledujúce kapitoly sú analýzou simulátorov ns-2 a ns-3. Záver je venovaný zhodnoteniu analýzy. V kapitole 4 sú špecifikované požiadavky na systém. V kapitole 4 (Špecifikácia a hrubý návrh riešenia) je uvedený hrubý návrh riešenia. K práci je pripojená dokumentácia k riadeniu projektu.

2 Simulácia v počítačových sieťach

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

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

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

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

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

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

Simulácia počítačových sietí v sebe zahŕňa tri prvky: topológiu siete, teda usporiadanie a vzájomné prepojenie jednotlivých uzlov siete, použité protokoly a tok dát medzi jednotlivými uzlami. Tieto tri časti tvoria vstupné dáta simulácie, na základe ktorých potom simulačný program vypočíta, teda nasimuluje, časový priebeh udalostí, ktoré sa v simulovanom modeli udejú. Výstupom simulácie môže byť napríklad textový súbor obsahujúci postupnosť udalostí, ako sú

napríklad príjem datagramu uzlom, zahodenie paketu, či zaradenie rámca do rady. Simulácia počítačovej siete nám teda povie, čo sa v danej sieti udeje v určitom časovom úseku.

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

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

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

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

3 Simulátory počítačových sietí

V nasledujúcej kapitole si priblížime vzhľadom na zadanie projektu simulátory Network simulator 2 (ns-2) a Network simulator 3 (ns-3).

3.1 Network simulator 2

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

3.1.1 Architektúra ns-2

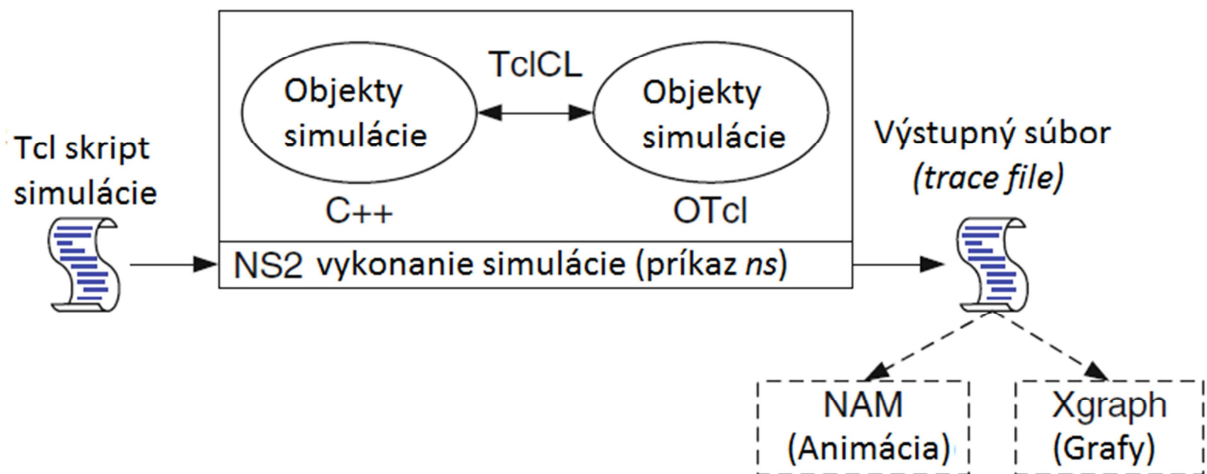
Ns-2 je simulátorom diskretných udalostí a bol vyvinutý pre testovanie a vyhodnocovanie existujúcich prístupov používaných v počítačových sieťach ako aj pre vývoj nových prístupov. Simulátor umožňuje simuláciu TCP/IP sietí, smerovania ako aj rôznych iných protokolov tak v pevných, ako aj v bezdrôtových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

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

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

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

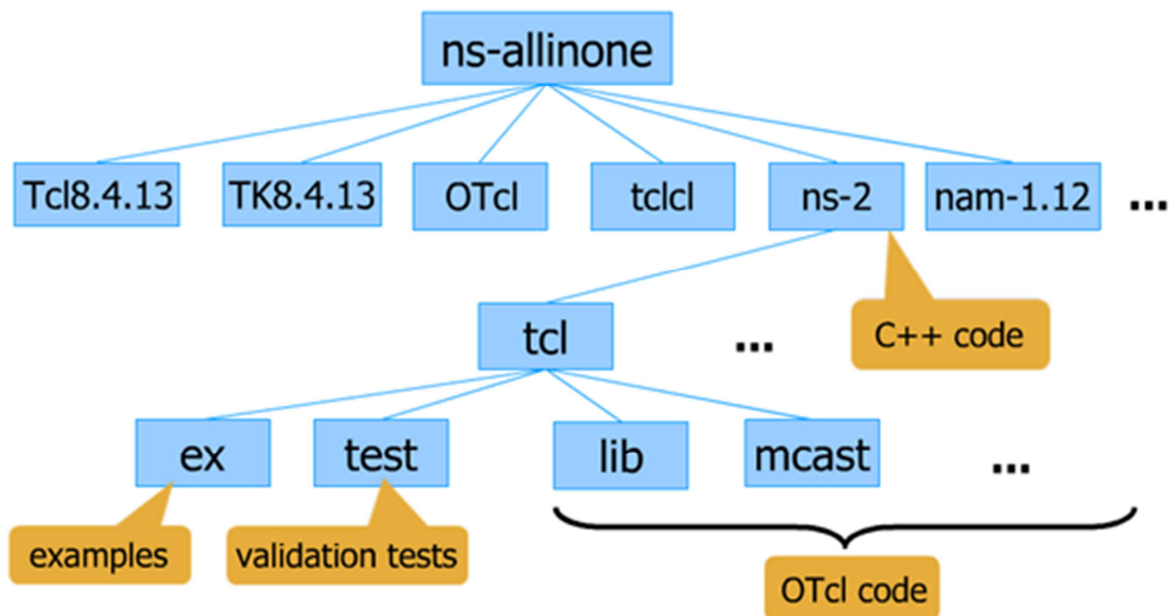
Obrázok 1.



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

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

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



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

3.1.2 Simulácie pomocou ns-2

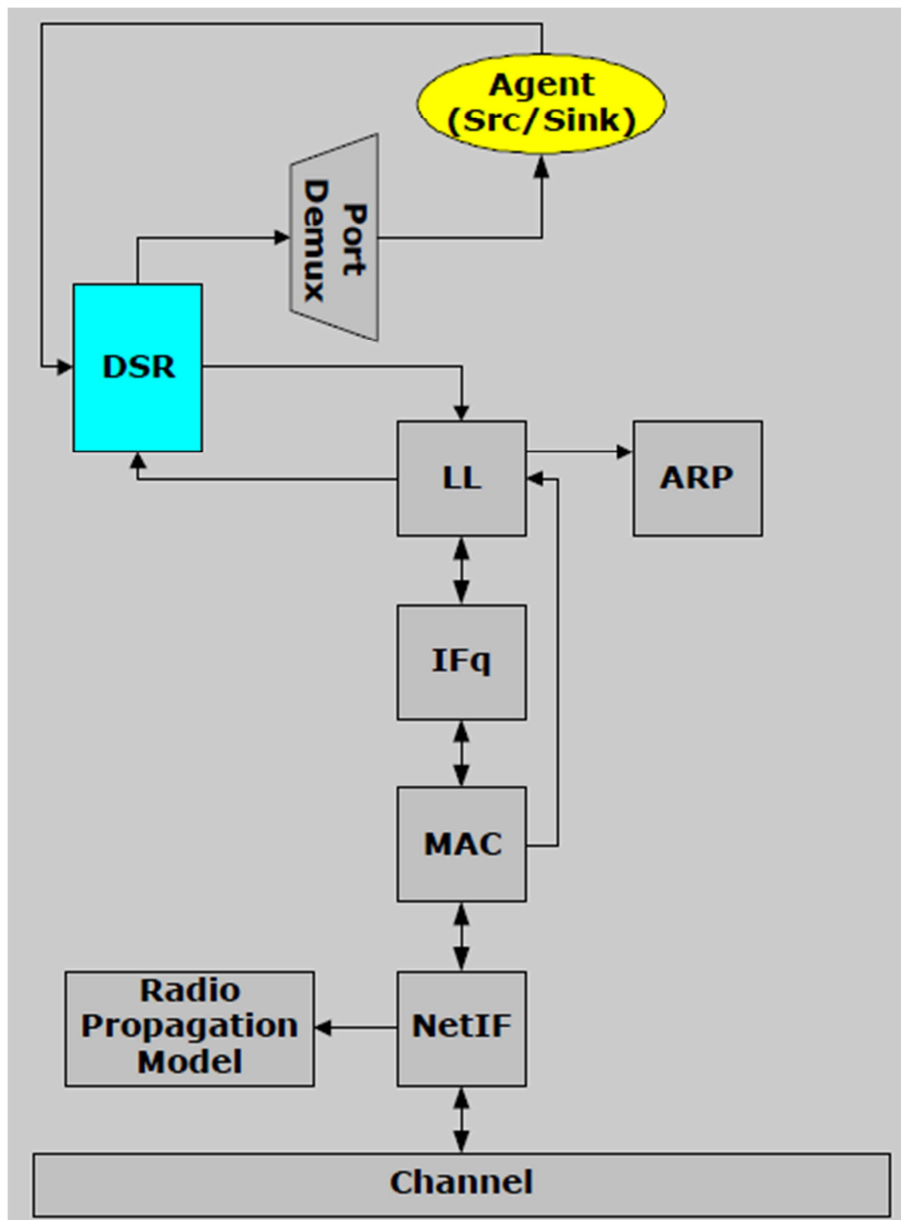
Nasledujúca kapitola je venovaná možnostiam simulátora ns-2 v oblasti simulácie bezdrôtových sietí, na ktoré je projekt prevažne zameraný. Podrobnejšie sú tu rozobrané jednotlivé entity prítomné v tomto druhu simulácií, podpora protokolov druhej a tretej vrstvy RM OSI a tiež špeciálny formát výstupného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

Mobilný uzol

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

Obrázok 3 a tvoria ho nasledovné moduly :

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



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

Kanál

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

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

Smerovacie protokoly

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

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

MAC protokoly

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

1. IEEE 802.11

- a. 802.11 DCF (*Distributed Coordination Function*)
 - Model vysielania typu unicast RTS/CTS/DATA/ACK
 - Model vysielania typu broadcast DATA
 - Možnosť prepojenia bezdrôtových a klasických sietí
- b. 802.11 infraštruktúrne rozšírenia
 - Aktívne a pasívne skenovanie
 - Autentifikácia
 - Komunikácia medzi prístupovými bodmi
 - Podpora mobility (*handoff*)
- c. 802.11Ext (*extended*)
 - Štruktúrovaný návrh modulov funkcionality na MAC vrstve: vysielanie, prijímanie, koordinácia vysielania.
 - Monitorovanie stavu kanála, riadenie stiahnutia sa z vysielania
 - Súhrnné výpočty SINR (*Signal to Interference + Noise Ratio*)
 - Schopnosť zachytávania rámcov
 - Podpora viacerých schém modulácie
 - Sledovanie strát dátových jednotiek na fyzickej vrstve
 - Nakagamiho model slabnutia signálu
- d. 802.11mr (*multirate*)
 - Podpora viacerých prístupov na fyzickej vrstve
 - Simulovanie vysielania na rôznych frekvenciách
 - Modulácia a kódovanie signálu podľa štandardu IEEE802.11b/g.
 - Model chybovosti paketov založený na SINR, zohľadňujúci silu signálu, šum a rušenie.

2. TDMA (*Time Division Multiple Access*)

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

3.1.2.1 Formát výstupného súboru

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

1. Typ udalosti v danom uzle :

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

2. Všeobecná značka :

-t čas
-t *(globálne nastavenie)

3. Značka nastavenia uzla :

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

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

4. Informácia o pakete na sieťovej vrstve :

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

5. Informácia o nasledujúcom skoku :

-Hs: identifikátor nasledujúceho uzla

-Hd: identifikátor pre nasledujúci uzol vzhľadom na cieľ

6. Informácia o pakete na linkovej vrstve :

- Ma:** doba trvania
- Md:** cieľová MAC adresa
- Ms:** zdrojová MAC adresa
- Mt:** typ rámca

7. Informácia o pakete na aplikačnej vrstve :

-P arp ARP (*Address Resolution Protocol*)

- Po:** ARP dotaz/odpoveď
- Pm:** zdrojová MAC adresa
- Ps:** zdrojová IP adresa
- Pa:** cieľová MAC adresa
- Pd:** cieľová IP adresa

-P dsr DSR (*Dynamic Source Routing*)

- Pn:** počet prejdených uzlov
- Pq:** príznak pre smerovací dotaz
- Pi:** sekvenčné číslo smerovacieho dotazu
- Pp:** príznak pre odpoveď na smerovací dotaz
- Pl:** dĺžka odpovede
- Pe:** zdroj zo zdroja smerovanie/cieľ z cieľa smerovanie
- Pw:** chybový príznak
- Pm:** počet chýb
- Pc:** komu oznámiť správu
- Pb:** chyba pri prenose z A do B

-P cbr CBR (*Constant Bit Rate*)

- Pi:** sekvenčné číslo
- Pf:** počet preposlaní paketu
- Po:** optimálny počet preposlaní

-P tcp Informácie o TCP toku

- Ps:** sekvenčné číslo
- Pa:** číslo potvrdenia ACK
- Pf:** počet preposlaní paketu
- Po:** optimálny počet preposlaní

3.1.2.2 Konfigurácia simulácie bezdrôtovej siete

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

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

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

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

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

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

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

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

```
$ns_ node-config
-addressingType flat/hierarchical/expanded
-adhocRouting DSDV or DSR or TORA
-llType LL
-macType Mac/802_11
-propType Propagation/TwoRayGround"
-ifqType "Queue/DropTail/PriQueue"
```

```

-ifqLen                50
-phyType               "Phy/WirelessPhy"
-antType               "Antenna/OmniAntenna"
-channelType           "Channel/WirelessChannel"
-topoInstance          $topo
-energyModel            "EnergyModel"
-initialEnergy         (in Joules)
-rxPower               (in W)
-txPower               (in W)
-agentTrace            ON or OFF
-routerTrace           ON or OFF
-macTrace              ON or OFF
-movementTrace         ON or OFF

```

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

```

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

```

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

```

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

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

```

A následne definovať pohyb zariadenia:

```

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

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

```

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

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

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

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

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

Najprv si uvedieme aké možnosti máme a potom ukážkový príkaz.

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

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

Príklady:

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

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

```
./setdest [-n num_of_nodes] [-p pausetime] [-s maxspeed]
[-t simtime] \
[-x maxx] [-y maxy] > [outdir/movement-file]
```

Príklad:

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

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

3.1.3 Podporné nástroje pre ns-2

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

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

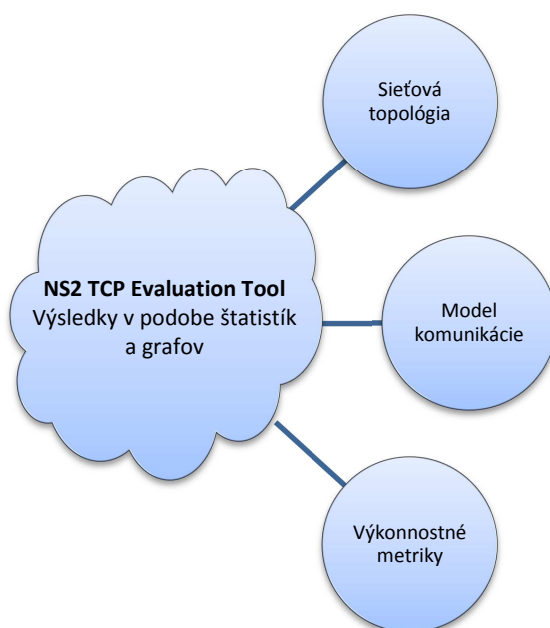
- Nástroje pre štatistické vyhodnotenie simulácie
- Kreslenie grafov výstupných štatistík
- Kreslenie topológií sietí s podporou bezdrôtových sietí
- Nástroje pre štatistické vyhodnotenie

3.1.3.1 NS2 TCP Evaluation Tool

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

Medzi hlavné vlastnosti a výhody nástroja patrí:

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



Obrázok 4: Architektúra programu NS2 TCP Evaluation Tool

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

1. Sieťová topológia

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

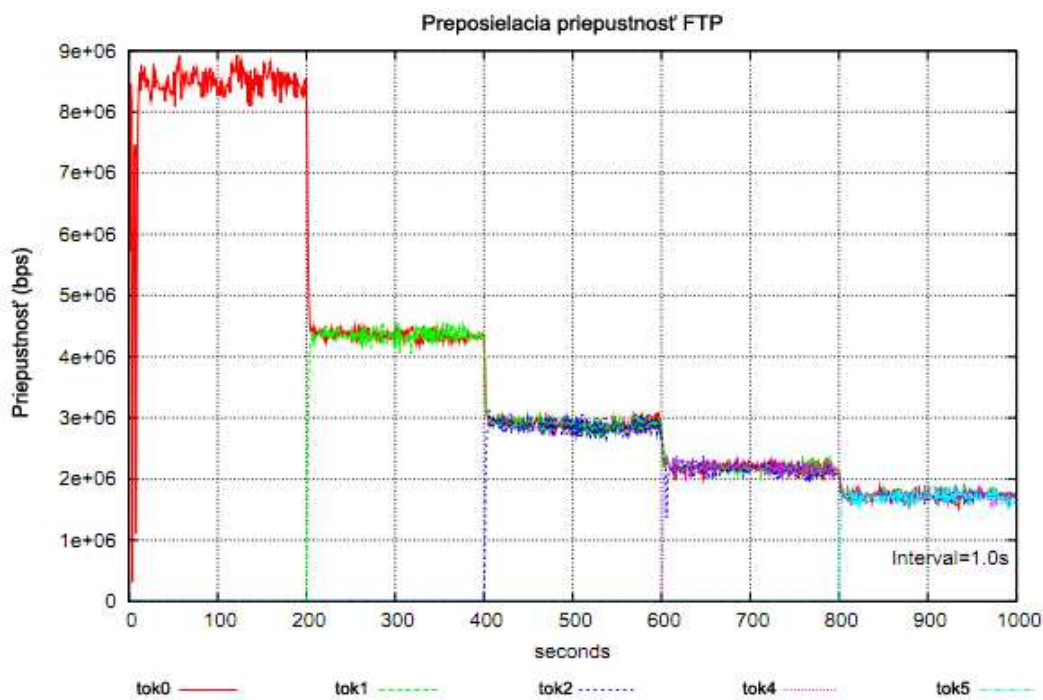
1. Model komunikácie

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

2. Výkonnostné metriky

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

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



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

3.1.3.2 GnuPlot

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

Medzi výhody nástroja patrí:

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

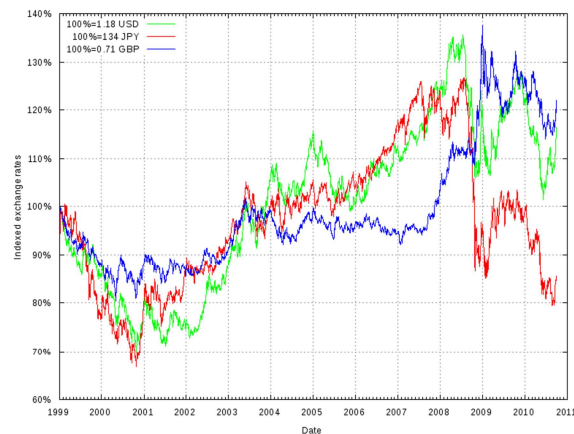
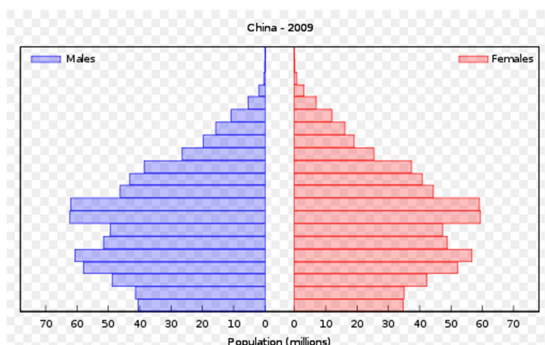
Ukážka skriptu pre generovanie grafu:

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

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

data = throughputOverTime.throughputOverTime( flows )

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



Obrázok 6: Ukážky grafov z programu GnuPlot (9)

3.1.3.3 Xgraph

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

Medzi jeho nesporné výhody patrí:

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

Xgraph môže byť použitý dvoma spôsobmi:

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



Obrázok 7: Ukážka programu XGRAPH (10)

3.1.3.4 ThroughputOverTime

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

Ukážka výstupu nástroja:

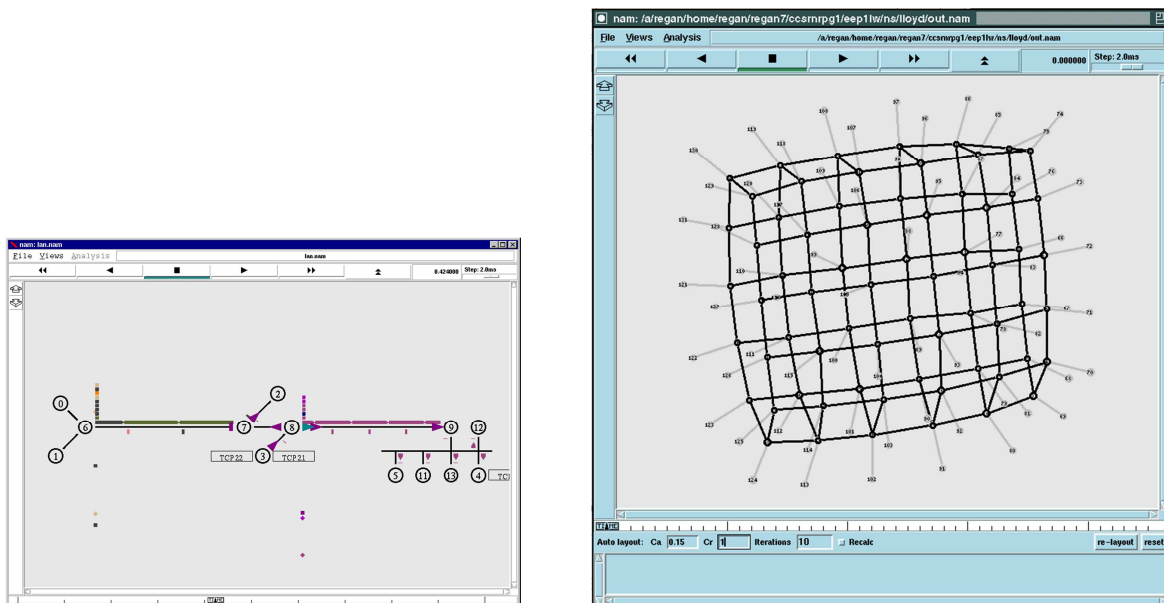
```
Time    0:0 -> 4:0    4:0 -> 0:0
1.0      0.0          0.0
2.0      0.0          0.0
3.0      0.0          0.0
```

3.1.3.5 Ns-nam

Sieťový animátor pre NS (Ns-nam, network animator for ns) bol vytvorený začiatkom deväťdesiatych rokov ako jednoduchý nástroj pre animovanie sietí a toku rámcov zo simulačných súborov (trace files). Tie sa dajú získať ako výstupy sieťového simulátora *ns2* alebo odchytením reálnej prevádzky programom *TcpDump*. V súčasnosti je stále vo vývoji na vedeckom inštitúte *ISI* (*Information Sciences Institute*) (11). Obrázok 8 znázorňuje rozhranie programu Ns-nam.

Hlavné vlastnosti:

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



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

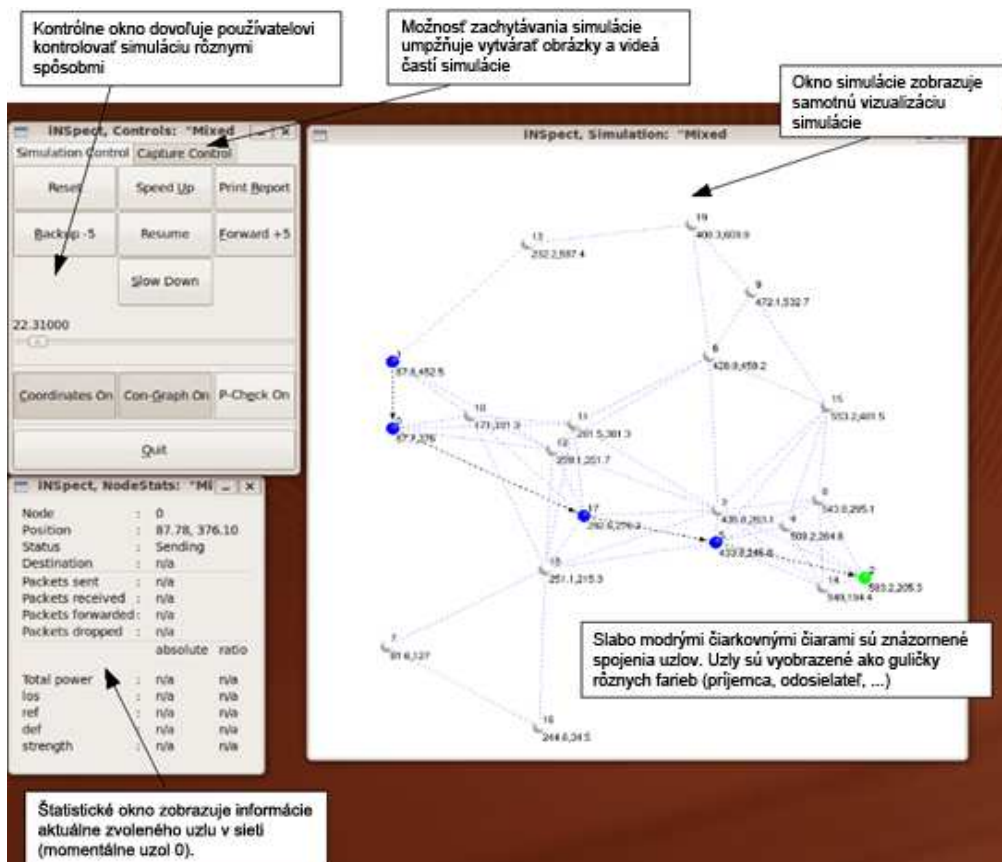
3.1.3.6 iNSpect

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

Medzi jeho hlavné vlastnosti a prednosti patrí:

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

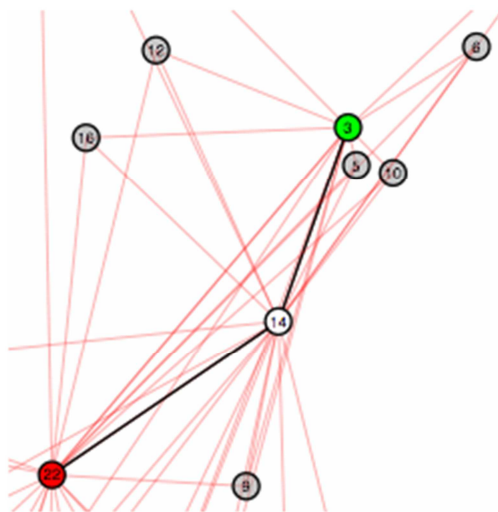
Popis rozhrania programu iNSpect znázorňuje Obrázok 9.



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

3.1.3.7 DrawNetwork

DrawNetwork (13) je jednoduchý program vytvorený v grafickom prostredí *GnuStep* na vykresľovanie topológie pre beh simulácie (Obrázok 10). Program je zameraný hlavne na kreslenie bezdrôtových sietí.



Obrázok 10: Ukážka grafu z programu DrawNetwork [22]

3.1.4 Zhodnotenie ns-2

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

Výhody ns-2

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

Nevýhody ns-2

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

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

Tabuľka 1: Zoznam protokolov podporovaných v jadre ns-2

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

3.2 Network simulator 3

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

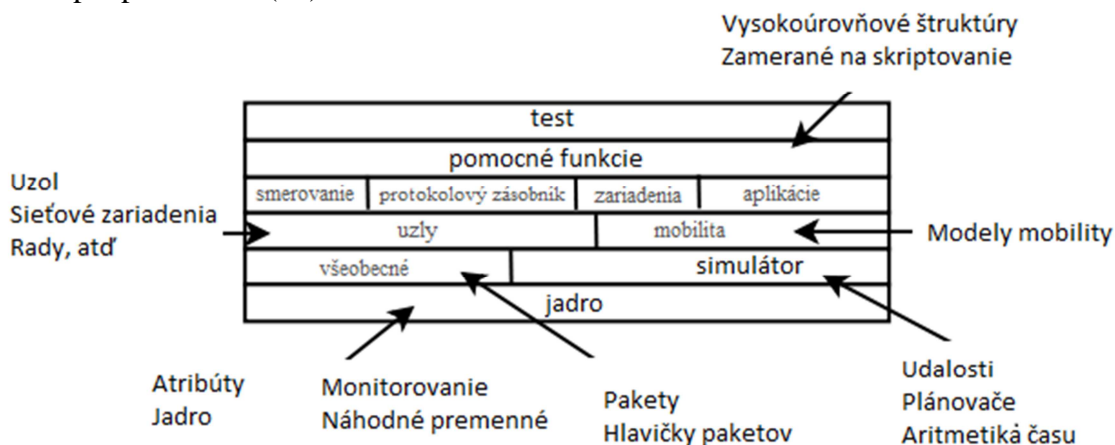
3.2.1 Architektúra ns-3

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

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

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

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



Obrázok 11: Schéma simulátora ns-3 (18)

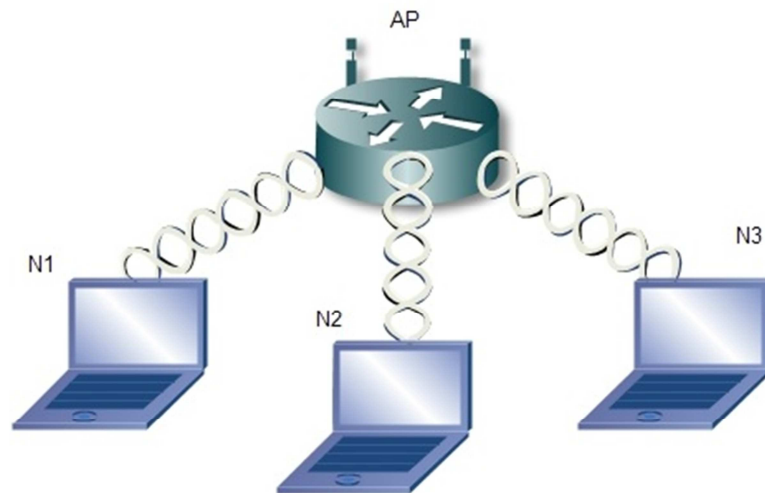
3.2.2 Simulácia pomocou ns-3

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

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

3.2.2.1 Vstupný súbor simulátora ns-3

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



Obrázok 12: Príklad bezdrôtovej topológie

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

```
#include "ns3/core-module.h"
#include "ns3/simulator-module.h"
#include "ns3/node-module.h"
#include "ns3/helper-module.h"
#include "ns3/wifi-module.h"
#include "ns3/mobility-module.h"
using namespace ns3;
Vo funkcii main() si ako prvé zdefinujeme počet počítačov do
premennej pocetPocitacov, ktoré budú v sieti.
uint32_t pocetPocitacov = 3;
```

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

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

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

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

Vytvoríme si ďalší uzol v sieti pre prístupový bod.

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

Vytvoríme si bezdrôtový kanál.

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

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

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

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

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

Pomenujeme sieť.

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

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

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

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

```
mac.SetType("ns3::NqapWifiMac", "Ssid", SsidValue  
(ssid), "BeaconGeneration", BooleanValue(true), "BeaconInterval", Ti  
meValue(Seconds(2.5)));  
NetDeviceContainer apDevices;
```

```
apDevices = wifi.Install (phy, mac, wifiAP);
```

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

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

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

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

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

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

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

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

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

```
UdpEchoServerHelper echoServer (9);  
ApplicationContainer serverApps =  
echoServer.Install(wifiPocitace.Get(0));  
serverApps.Start (Seconds (6.0));  
serverApps.Stop (Seconds (10.0));
```

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

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

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

Naplníme smerovacie tabuľky.

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

Po desiatich sekundách simuláciu zastavíme.

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

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

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

Spustíme simuláciu a po skončení dealokujeme pamäť použitú pre simulátor.

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

3.2.2.2 Výstupný súbor simulátora ns-3

Sledovanie simulácie sa v simulátore ns-3 môže realizovať niekoľkými spôsobmi. Buď sa použijú už implementované prostriedky na sledovanie simulácie, ako je napríklad ASCII sledovanie, alebo výstup simulácie vo forme súboru typu .pcap, ktoré si popíšeme neskôr.

Ďalšou možnosťou sledovania simulácie sú výpisy implementované priamo vo vstupnom súbore simulácie, v ktorých si môžeme vypisovať ľubovoľne zadefinované hlásenia, napríklad pri vytvorení nového uzla, alebo podobne.

Poslednou možnosťou je upraviť priamo jadro simulátora, kde si môžeme nechať vypisovať hlásenia priamo vo vbudovaných funkciách simulátora. (20)

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

V nasledujúcom texte sa oboznámime s už implementovanými spôsobmi sledovania behu simulácie, ako je sledovanie pomocou ASCII výstupu či PCAP výstupu.

ASCII aj PCAP sledovanie sa definuje priamo vo vstupnom súbore simulátora, pričom sa musí určiť, na ktorom uzle v simulovanej sieti sa toto sledovanie bude vykonávať. Sledovanie každej entity v simulácii sa ukladá do samostatného súboru, ktorý sa neskôr spracováva.

To, že vo vstupnom súbore simulátora povieme, ktoré komunikácie na ktorých uzloch chceme zaznamenávať do akého súboru, nám značne uľahčuje orientáciu vo výstupných súboroch. Vždy vieme, v ktorom nájdeme požadované dáta a preto nemusíme prácne prehľadávať záznamy z rôznych uzlov, aby sme našli komunikáciu, ktorú potrebujeme.

ASCII sledovanie simulácie

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

1. Typ udalosti na uzle, udalosti sa týkajú paketu: pridanie do radu (+), odobranie z radu (-), prijatie (r), zahodenie (d).
2. Časová pečiatka udalosti.
3. Ďalší reťazec znakov je ns-3 špecifická cesta k udalosti, ktorá dáva informáciu o tom, na ktorom sieťovom rozhraní ktorého uzlu sa udalosť stala, zároveň aj poskytuje informáciu o type udalosti.

4. Ďalší reťazec je zoznam typov hlavičiek, zakódovaných ako názvy tried ns-3 simulátora, ktoré sa v pakete/rámci nachádzajú. Každá hlavička je nasledovaná informáciou indikujúcou informácie špecifické pre daný protokol.

Príklad ASCII .tc výstupného súboru simulátora:

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

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

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

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

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

PCAP sledovanie simulácie

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

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

```
0.000025 Beacon ( ) [6.0* 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
IBSS
0.000263 Assoc Request ( ) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0
Mbit]
0.000279 Acknowledgment RA:00:00:00:00:00:01
0.000357 Assoc Response AID(0) :: Successful
0.000501 Acknowledgment RA:00:00:00:00:00:04
0.000748 Assoc Request ( ) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0
Mbit]
0.000764 Acknowledgment RA:00:00:00:00:00:02
0.000842 Assoc Response AID(0) :: Successful
0.000986 Acknowledgment RA:00:00:00:00:00:04
0.001242 Assoc Request ( ) [6.0 9.0 12.0 18.0 24.0 36.0 48.0 54.0
Mbit]
0.001258 Acknowledgment RA:00:00:00:00:00:03
0.001336 Assoc Response AID(0) :: Successful
0.001480 Acknowledgment RA:00:00:00:00:00:04
```



```
2.500000 Beacon () [6.0* 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
IBSS
5.000000 Beacon () [6.0* 9.0 12.0 18.0 24.0 36.0 48.0 54.0 Mbit]
IBSS
6.000112 ARP, Request who-has 192.168.10.2 (ff:ff:ff:ff:ff:ff)
tell 192.168.10.4, length 32
6.000128 Acknowledgment RA:00:00:00:00:00:03
6.000206 ARP, Request who-has 192.168.10.2 (ff:ff:ff:ff:ff:ff)
tell 192.168.10.4, length 32
6.000464 ARP, Reply 192.168.10.2 is-at 00:00:00:00:00:01, length
32
```

3.2.2.3 Podporované bezdrôtové technológie

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

ns-3.5 (Júl 2009)

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

ns-3.6 (Október 2009)

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

ns-3.7 (Január 2010)

- 802.11p PHY
- AODV
- Waypoint mobility
- NetAnim
- IPv6 Extension and Option headers

ns-3.8 (Máj 2010)

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

ns-3.9 (August 2010)

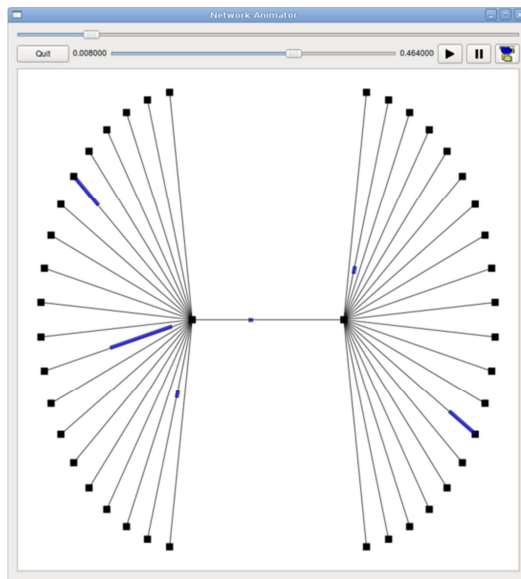
- OFDM
- UAN (Underwater Acoustic Network)
- Energy model (16)

3.2.3 Podporné nástroje pre ns-3

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

3.2.3.1 NetAnim

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



Obrázok 13: Grafické rozhranie programu NetAnim

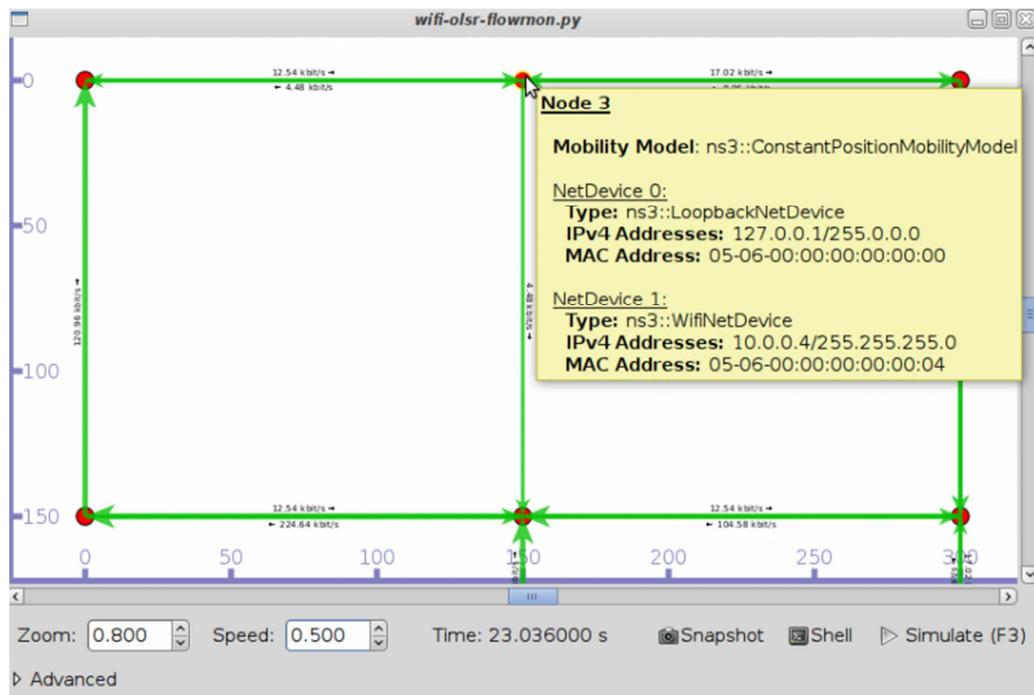
Zaujímavou vlastnosťou tohto vizualizéra je špecifikácia umiestnenia uzlov na schéme (Obrázok 13). Potreba špecifikovať umiestnenie jednotlivých uzlov naznačuje, že program neobsahuje modul na samostatné generovanie schémy (t.j. nemá implementované algoritmy optimalizujúce zobrazenie grafu ako planárneho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

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

3.2.3.2 PyViz

PyViz je vizualizér simulácií v reálnom čase, t.j. nepoužíva žiadne výstupne súbory a do simulácie je možné vstupovať prostredníctvom konzoly, cez ktorú je možné meniť stav simulovaných objektov za behu. PyViz podporuje simulácie napísané v- jazyku Python aj v C++

(23). Tento vizualizér nie je súčasťou hlavnej vetvy simulátora, ale existujú balíky, kde sú PyViz a ns-3 zlúčené. Rozhranie vizualizéra znázorňuje Obrázok 14.



Obrázok 14: Grafické rozhranie vizualizéra PyViz

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

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

nasledujúcim kódom

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

a v C++ simulácii vložiť riadok

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

a nahradiť

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

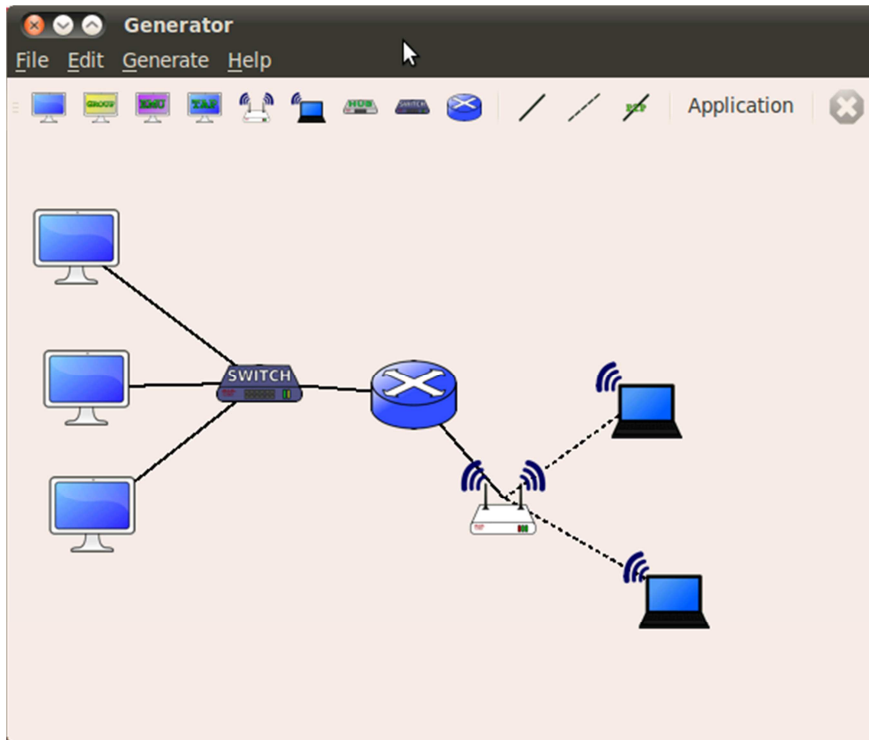
kódom

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

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

3.2.3.3 Ns3Generator

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



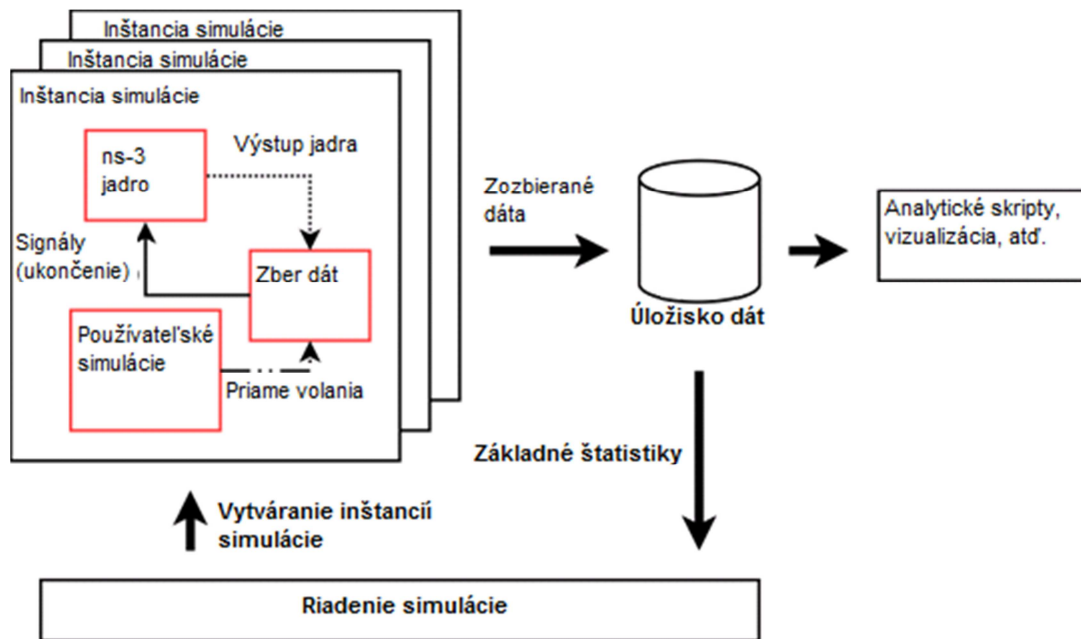
Obrázok 15: Grafické rozhranie programu

3.2.3.4 Statistical Framework for Network Simulation

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

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

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



Obrázok 16: Schéma štatistického systému (24)

Statistical Framework for Network Simulation,

http://www.nsnam.org/wiki/index.php/Statistical_Framework_for_Network_Simulation, máj 2008

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

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

Vyvíjané sú prioritne tieto ďalšie moduly:

- On-line štatistický kód
- Funkcie v dátových zberačoch na ukončenie simulácií (pri rôznych podmienkach)
- Dátové zberače na zber vzoriek a výstupy v rôznych formátoch

3.2.4 Zhodnotenie ns-3

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

Výhody

- Vývoj s dôrazom na nové bezdrôtové technológie (802.11, 802.16, GSM, GPRS,...)
- Škálovateľnosť
- Podpora spolupráce simulátora s reálnou sieťou
- Podpora bežných formátov výstupov (.pcap)
- Jednoduchšie používateľské rozhranie
- Natívna podpora 64 bitových a multiprocessorových strojov

- Beh reálneho kódu v rámci ns-3 (reálne aplikácie, protokolové zásobníky)
- Natívna podpora štatistického vyhodnotenia simulácií

Nevýhody

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

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

3.3 Zhodnotenie analýzy simulátorov

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

Porovnanie podpory protokolov v oboch simulátoroch ilustruje Obrázok 17.

	Existujúce možnosti jadra ns-2	Existujúce možnosti jadra ns-3
Aplikačná vrstva	ping, vat, telnet, FTP, multicast FTP, HTTP, probabilistické a súborové generátory prevádzky, vyrovnávací pamäť webu	Vypnutá/zapnutá aplikácia, asynchrónne sokety, paketové sokety
Transportná vrstva	Mnoho variánt TCP, UDP, SCTP, XCP, TFRC, RAP, RTP Multicas: PGM, SRM, RLM, PLM	UDP, TCP
Sieťová vrstva	Unicast: IP, mobilné IP, generické dist. vektor, generické link state, IPvIP, smerovanie na základe zdroju, Nixvector Multicast: SRM, generické centralizované MANET: AODV, DSR, DSDV, TORA, IMEP	Unicast: IPv4, globálne statické smerovanie Multicast: statické smerovanie MANET: OLSR
Linková vrstva	ARP, HDLC, GAF, MPLS, LDP, Diffserv Radenie paketov: DropTail, RED, RIO, WFQ, SRR, sémantické radenie, REM, prioritné radenie, Prístupové metódy: CSMA, 802.11b, 802.15.4 (WPAN), satelitná Aloha	PointToPoint, CSMA, 802.11 MAC s kontrolou prenosových rýchlostí
Fyzická vrstva	TwoWay, Shadowing, všesmerové antény, energetický model, satelitný opakováč	802.11a, Friis model straty signálu, model straty signálu na veľkú diaľku, základný model straty signálu (strata, oneskorenie)
Podporné prostriedky	Generátory náhodných čísel, sledovanie, monitorovanie, podpora matematického spracovania, animácie (nam), modely chýb, testovacie scenáre	Generátory náhodných čísel, monitorovanie, testy jednotiek, sledovanie, vizualizátor mobility modely chýb

Obrázok 17: Porovnanie funkcionalít implementovaných v jadrách simulátorov ns-2 a ns-3 (28)

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

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

4 Špecifikácia a hrubý návrh riešenia

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

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

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

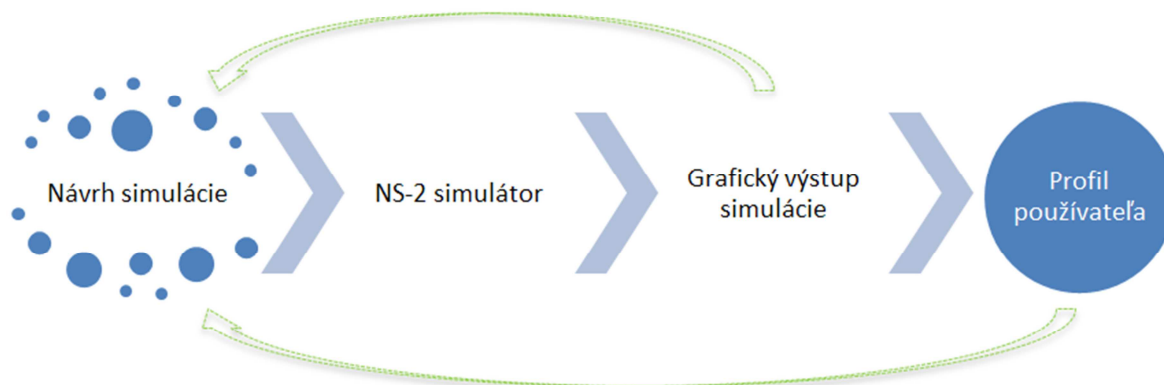
Aplikáciu sme sa rozhodli navrhnuť ako klient-server aplikáciu. Dôvodom takéhoto riešenia je, že niektorí používatelia môžu mať výkonnejší hardvér než akým je vybavený server. Simulácia bude prebiehať lokálne na počítači používateľa, čo spôsobí jej urýchlenie. Výstup simulácie potom používateľ iba odošle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

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

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

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

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



Obrázok 18: Architektúra riešenia

4.1 Používatelia

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

Typy používateľov:

- neprihlásený používateľ
- prihlásený používateľ
- administrátor

Neprihlásený používateľ má nasledujúce práva:

- zaregistrovať sa
- prezerat' výsledky skončených simulácií
- porovnávať vybrané simulácie

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

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

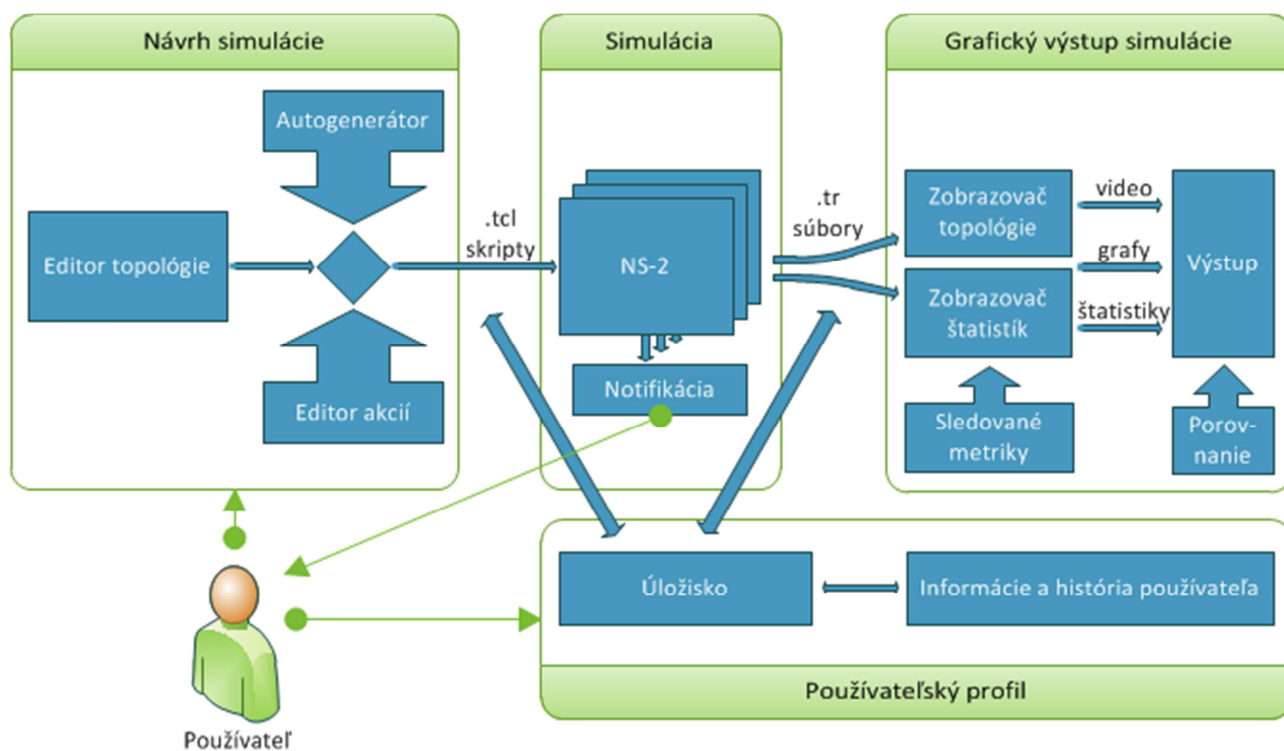
- uložiť si vstupný súbor typu *.tcl
- uložiť si výstupný súbor simulácie
- uložiť si výsledky simulácie v podobe obrázkov, grafov a/alebo videa
- nahrať na server svoj vlastný vstupný súbor typu *.tcl
- nahrať na server svoj vlastný výstupný súbor simulácie pre zobrazenie výsledkov pomocou obrázkov, grafov a/alebo videa
- vyžiadať si zaslanie e-mailu zo systému po skončení simulácie
- spustiť počas jedného prihlásenia naraz 3 simulácie
- pridať poznámky k výsledkom simulácie

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

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

4.2 Popis komponentov

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



Obrázok 19: Podrobná schéma riešenia

Editor topológie

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

Auto-generátor

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

Editor akcií

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

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

Simulátor NS2

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

Notifikácia

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

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

Merané metriky

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

Zobrazovač topológie

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

Zobrazovač štatistík

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

Porovnanie

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

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

Úložisko

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

Informácie a história používateľa

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

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

5 Záver

V tejto práci sme sa zaoberali programovou podporou pre sieťový simulátor. Analyzovali sme popredné simulátory ns-2 a ns-3, pričom sme sa zamerali na niektoré ich vlastnosti, ako funkčnosť, vstupné a výstupné súbory simulátorov, typy bezdrôtových sietí, ktoré podporujú, či pre nich dostupné podporné prostriedky. Na základe týchto vlastností sme si pre ďalšiu prácu na tomto projekte vybrali sieťový simulátor ns-2. Špecifikovali sme riešenie a navrhli základnú funkcionálnosť.

Pri ďalšej práci na tomto projekte plánujeme vytvoriť podrobný návrh riešenia, a tento návrh implementovať s dôrazom na vlastnosti obsiahnuté v špecifikácii.

6 Bibliografia

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

Príloha A

Dokumentácia k riadeniu projektu

1 Riadenie projektu

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

Ponuka

Tímový projekt

Team 

Autori:

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

predmet: Tímový projekt 1
akademický rok: 2010/2011

Predstavenie členov tímu

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

Motivácia

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

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

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

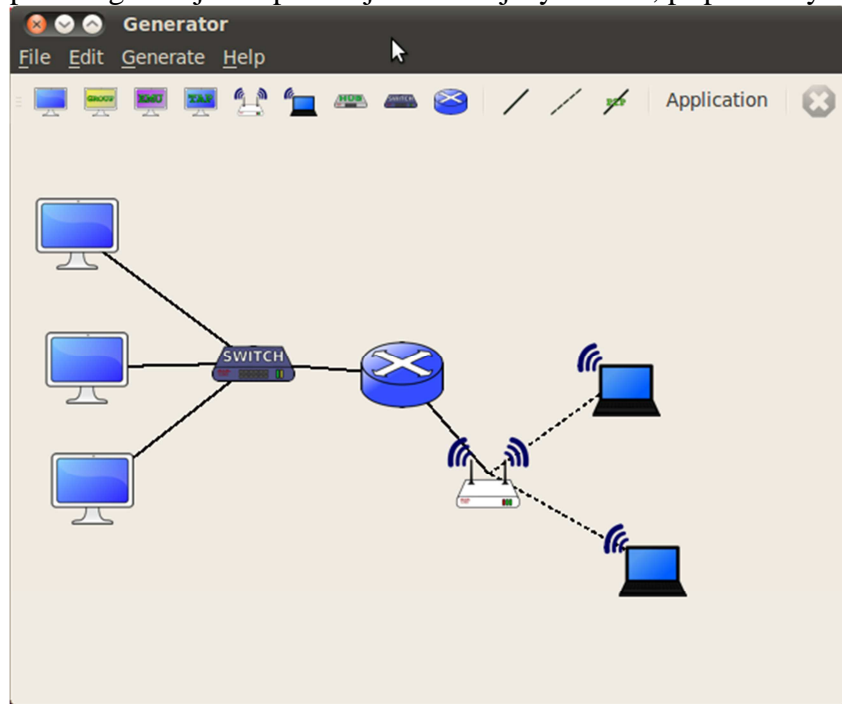
Existujúce riešenia

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

Rozširujúce moduly pre simulátor ns-3

Ns3Generator¹

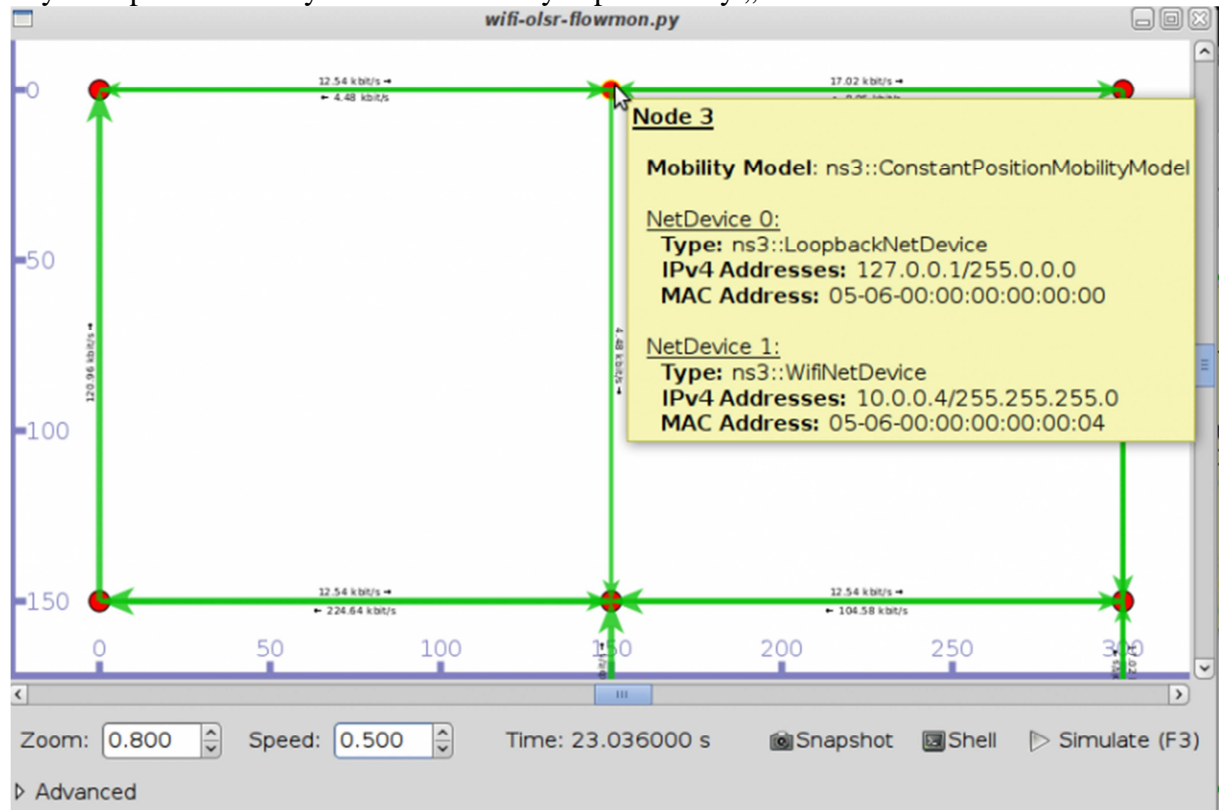
Modul umožňuje prostredníctvom GUI intuitívne navrhnúť topológiu pre simuláciu, pričom generuje zodpovedajúci kód v jazyku C++, prípadne Python.



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

PyViz²

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

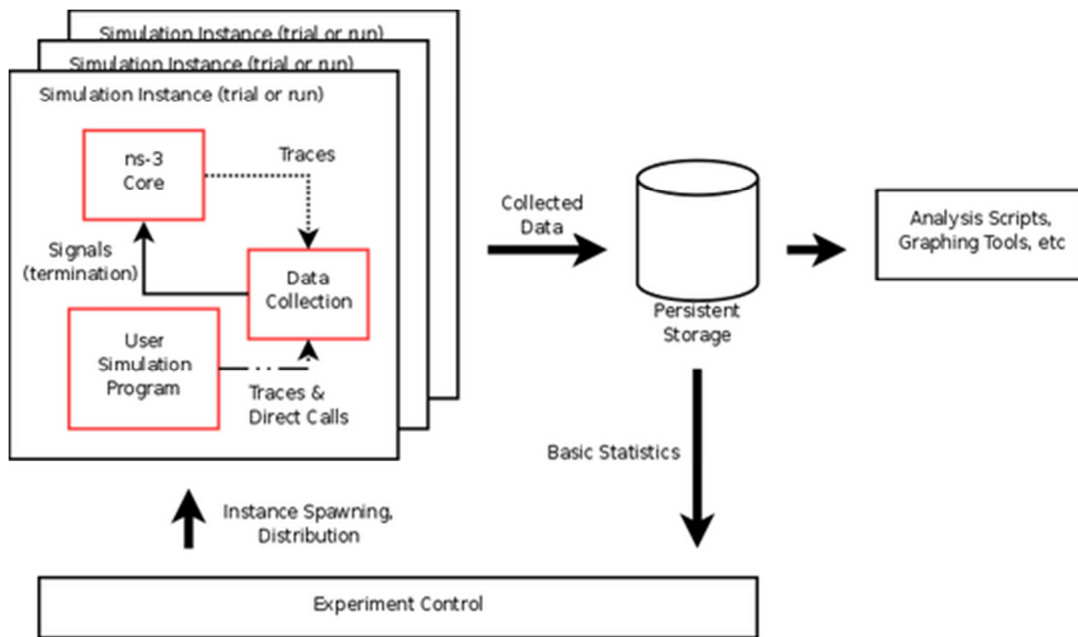


Statistical Framework for Network Simulation³

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

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

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



Rozširujúce moduly pre ns-2

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

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

Návrh riešenia

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

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

Aplikáciu sme sa rozhodli navrhnuť ako klient-server aplikáciu. Dôvodom takéhoto riešenia je, že niektorí používatelia môžu mať výkonnejší hardvér než akým je vybavený server. Simulácia bude prebiehať lokálne na počítači používateľa, čo spôsobí jej urýchlenie. Výstup simulácie potom používateľ iba odošle na server, ktorý ho následne spracuje

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

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

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

Alternatívne riešenia

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

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

Požiadavky

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

Prečo by táto téma mala byť pridelená práve nám

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

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

Priority jednotlivých tém

- 1. Programová podpora pre sieťový simulátor**
- 2. Vývoj vzorových aplikácií pre IPTV v prostredí Microsoft Mediaroom Framework**
- 3. Diagnostika porúch diskretných udalostných systémov založená na modeloch a aplikácie v informatike**
- 4. Prostredie pre návrh digitálnych systémov (Digital System Designer)**

- 5. Simulátor komunikácie v počítačovej sieti**
- 6. Využitie vlastností HP Tablet PC na inováciu vyučovania predmetu Testovateľnosť digitálnych systémov**
- 7. Návrh vnoreného systému (Embedded Development)**
- 8. Využitie vlastností HP Tablet PC na inováciu vyučovania**
- 9. Distribuovaný odkladací priestor virtuálnej pamäte**

Plán projektu

V tabuľke 1 je znázornený plán vypracovávania projektu. Tento plán je priebežne aktualizovaný, preto sa v neskorších verziách tohto dokumentu bude pravdepodobne líšiť od súčasnej verzie.

Dátum	Činnosť
September 2010	Príprava ponuky
4.10.2010	Pridelenie úloh v tíme, zriadenie projektovej dokumentácie
11.10.2010	Úvodné analýzy simulátorov ns-2 a ns-3
18.10.2010	Hĺbková analýza simulátora ns-2, zadefinovanie rozhodovacích kritérií pre výber simulátora
4.11.2010	Hĺbková analýza simulátora ns-3, porovnanie simulátorov ns-2 a ns-3, konečný výber simulátora
8.11.2010	Kompletizácia časti analýza a ucelenie časti návrh a špecifikácia prvej verzie dokumentu.
November 2010	Vytváranie podrobného návrhu riešenia
November 2010	Spojzdenie servera, na ktorom pobeží výsledné riešenie.
December 2010	Vytvorenie skriptov na spracovanie výstupných súborov simulátora ns-2.
December 2010	Vytvorenie prototypu riešenia

Tabuľka 1. Plán projektu

Úlohy členov tímu

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

Bc. Martin Nagy

- dlhodobé úlohy:
 - riadenie tímu
- krátkodobé úlohy:
 - analýza simulátora ns-3 – časť úvod a inštalačné manuály
 - časť dokumentu Návrh
 - formálna úprava dokumentu

Bc. Dávid Oros

- krátkodobé úlohy:
 - analýza ns-2 – podpora bezdrôtových sietí, TCL
 - špecifikácia riešenia

Bc. Roman Panenka

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

Bc. Martin Pirháč

- dlhodobé úlohy:
 - kontrolovanie projektových termínov
- krátkodobé úlohy:
 - prezentácia o simulátore ns-2
 - analýza ns-2 – ns-2 prehľad, výhody oproti ns-3
 - finálne úpravy dokumentu

Bc. Hana Severínová

- dlhodobé úlohy:
 - vytváranie zápisníc zo stretnutí
- krátkodobé úlohy:
 - analýza ns-3 – vstupné a výstupné súbory simulátora ns-3, podpora bezdrôtových sietí
 - úvod a riadiaca dokumentácia projektu

Bc. Martin Svetlák

- krátkodobé úlohy:
 - analýza ns-3 – programová podpora
 - zhodnotenie a záver dokumentu

Zápisnica zo stretnutia Tímu 3

Stretnutie č.1

Dátum: 4.10.2010

Čas: 16:00

Miesto: Softvérové štúdio

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

Náplň stretnutia :

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

Zápisnica zo stretnutia Tímu 3

Stretnutie č.2

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

Kontrola plnenia úloh stanovených na poslednom stretnutí

- e-mailová skupina pre tím vytvorená

Náplň stretnutia

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

Ciele do najbližšieho stretnutia

- predstavenie webstránky – Bc. Roman Panenka
- prednáška Úvod do skriptovacieho jazyka TCL a simulátora ns2 – Bc. Martin Pirháč
- inštalácia ns2 aj ns3
- porovnanie simulátorov ns2 a ns3 z vybraných hľadísk
- porovnanie ns2 a ns3, potom hrubá analýza a návrh.

Zápisnica zo stretnutia Tímu 3

Stretnutie č.3

Dátum: 18.10.2010
Čas: 17:30
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy (meškanie 1/2 hodiny), Bc. Dávid Oros, Bc. Roman Panenka, Bc. Martin Pirháč, Bc. Hana Severínová. Bc. Martin Svetlák

Kontrola plnenia úloh stanovených na poslednom stretnutí

- webstránka vytvorená
- prednáška odprezentovaná

Náplň stretnutia

- prezentácia sieťového simulátoru ns2 a TCL skriptov
- stanoviť náhradný termín stretnutia v siedmom týždni – štvrtok 18:00
- v šiestom týždni: porovnanie simulátorov ns2 a ns3 v dvoch tímoch – ktorý simulátor je lepší?
- vybraný typ sietí: bezdrôtové (802.11, Bluetooth,...)
- rozdelenie do dvoch tímov:
 1. analýza ns2: Dávid Oros, Martin Pirháč, Martin Svetlák
 2. analýza ns3: Martin Nagy, Roman Panenka, Hana Severínová

Ciele do najbližšieho stretnutia

- zistiť veľkosť identifikátoru vo výstupnom súbore ns2 simulátora
- premyslieť vizualizáciu topológie zo zdrojového súboru
- prezentácia o ns3
- popísať ns2, jeho vlastnosti, architektúra, na základné porovnanie
- analýza existujúcich riešení
- podľa zadania do tímu naštudovať simulátor so zameraním na bezdrôtové siete
- do siedmeho týždňa: analýza ns2 a ns3, definitívny výber simulátora, definitívny výber typu sietí, veličiny pre vybraný typ sietí

Zápisnica zo stretnutia Tímu 3

Stretnutie č.3.1

Dátum: 27.10.2010

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

Ciele do najbližšieho stretnutia

- Dávid Oros:
 - podpora bezdrôtových sietí
 - TCL
- Martin Pirháč:
 - ns-2 overview, výhody oproti ns-3
 - zistiť veľkosť identifikátoru vo výstupnom súbore ns2 simulátora
- Roman Panenka:
 - možnosti vizualizácie topológie
 - podporné nástroje tretích strán
- Termín odovzdania: 31.10.2010

Zápisnica zo stretnutia Tímu 3

Stretnutie č.3.2

Dátum: 2.11.2010

Miesto: blok D

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

○ **Ciele do najbližšieho stretnutia**

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

Zápisnica zo stretnutia Tímu 3

Stretnutie č.4

Dátum: 4.11.2010
Čas: 18:00
Miesto: blok C, tretie poschodie
Účast': Ing. Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová

Kontrola plnenia úloh stanovených na poslednom stretnutí

- diskusia ns2 verzus ns3
- výber ns2, pretože ns3 má veľa chýb a ešte nie je veľmi doimplementované, nedá sa spoľahnúť

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

Ciele do najbližšieho stretnutia

- dopísať úvod – prečo je vhodné simulovať (lebo je náročné a drahé robiť si laboratórium keď treba veľa hardvéru, ...) - Hana Severínová
- analýza toho, aké riešenia už pre ns2 existujú
- vlastnosti ns2, hĺbková analýza simulátora – čo vie ponúknuť
- všetky obrázky prepísať do slovenčiny
- preveriť, či na TPcup treba pracovať podľa nejakej metodiky – Martin Nagy
- premyslieť budúcnosť nášho projektu, kam chceme smerovať
- do 8.11. hotová predbežná analýza
- zistiť, či sa na konci semestra dokladá CD, či treba mať funkčný prototyp, alebo stačia ukážky – Martin Pirháč
- hĺbková analýza ns2 – Dávid Oros, Roman Panenka, Martin Pirháč
- zhodnotenie analýzy – Martin Svetlík
- formálne úpravy a manažment okolo dokumentu, spájanie čiastkových dokumentov – Martin Nagy

Zápisnica zo stretnutia Tímu 3

Stretnutie č.5

Dátum: 8.11.2010
Čas: 18:00
Miesto: Softvérové štúdio
Účast': Ing. Peter Magula, Bc. Martin Nagy, Bc. Dávid Oros, Bc. Roman Panenka,
Bc. Martin Pirháč, Bc. Hana Severínová, Bc. Martin Svetlík

Náplň stretnutia

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

Ciele do najbližšieho stretnutia

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

Manažment verzií, konfigurácií a zmien

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

Dropbox

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

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

Google Groups

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

Microsoft Word 2007

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

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