VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

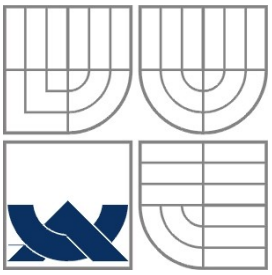
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WIFI SÍŤ 802.11B A JEJÍ SIMULACE

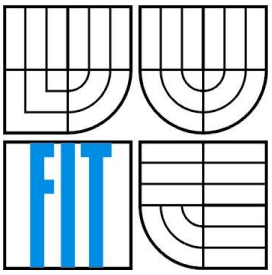
BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE VIKTOR MARTIŠ
AUTHOR

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

WIFI SÍŤ 802.11B A JEJÍ SIMULACE

WIFI NETWORK 802.11B AND ITS SIMULATION

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE
AUTHOR

VIKTOR MARTIŠ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. VÁCLAV ŠIMEK

BRNO 2007

Zadání bakalářské práce/5641/2006/xmarti32

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2006/2007

Zadání bakalářské práce

Řešitel: **Martiš Viktor**
Obor: Informační technologie
Téma: **WiFi síť 802.11b a její simulace**
Kategorie: Počítačové sítě

Pokyny:

1. Seznamte se s oblastí síťových komunikací využívajících bezdrátové technologie. Zaměřte se zejména na standard 802.11b.
2. Zabývejte se problematikou jejich simulace a prozkoumejte možnosti simulátoru NS-2.
3. Sestavte bezdrátovu síť podle standardu IEEE 802.11b se zvolenou topologií.
4. Proveďte zevrubnou analýzu navržené sítě.
5. Odsimulujte danou síť pomocí simulátoru NS-2.
6. Analyzujte výsledky získané simulací a proveďte jejich porovnání se skutečný chováním navržené sítě.

Literatura:

- dle pokynů vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Požadováno splnění prvních dvou bodů zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Šimek Václav, Ing.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2006
Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2
L.S.

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Licenčná zmluva

Licenčná zmluva je uložená v archíve Fakulty informačních technologií Vysokého učení technického v Brně.

Abstrakt:

Táto bakalárska práca sa zaoberá problematikou bezdrôtových sietí a ich simuláciou. Na začiatku je zahrnutý krátky úvod do bezdrôtových sietí a procesu simulovania. Práca je zameraná na jeden typ bezdrôtových sietí a to konkrétne na štandard IEEE 802.11b, tiež nazývaný ako „WiFi sieť“. Ďalej je rozobraný sieťový simulátor „The network simulator - ns2“ a možnosti jeho využitia. Následne je navrhnutá WiFi sieť, z ktorej je urobená hrubá analýza. Topológia je zvolená podľa reálnej siete, ktorá podlieha stálemu pozorovaniu a zaznamenávaniu prevádzky. Táto topológia siete je od simulovaná v spomenutom simulátore ns2, pomocou skriptu napísaného v jazyku TCL. Z dosiahnutých výsledkoch sú urobené analýzy. V závere je uvedené zhrnutie jednotlivých analýz a ich porovnanie s reálnym výkonom siete.

Kľúčové slová:

bezdrôtová sieť, WiFi sieť, abstraktný model, simulačný model, simulátor ns2, simulačný skript, TCL, simulácia WiFi siete, výsledky simulácie, maximálna kapacita linky, porovnanie výsledkov

Abstract:

This work deals with the technological background of wireless networks and the conceptual issues of their simulation. The introductory part provides a brief review of essential facts related to wireless networks and simulation flow. Furthermore, our work will focus on one particular type of such networks, which is governed by the standard IEEE 802.11b, also recognized as WiFi network. The actual simulation will be carried out by means of "The network simulator - NS2". Its general description is included together with some practical comments on possible deployment scheme. Then, we design a simple WiFi network which will be subjected to our analysis. The underlying topology was inspired by a real network scheme where the data traffic can be constatnly monitored and evaluated. Simulation of the proposed network was successsfully performed with help of the previously mentioned NS2 simulator. Script written in TCL language, which containted network definition and necessary control commands, served as the simulation input. The final part of this work contains a general summary of individual analyses and their comparison with real-life network performance.

Key Words:

wireless network, WiFi, abstract model, simulation model, NS2 simulator, simulation script, TCL, simulation of WiFi network, results of simulation, maximum link capacity, evaluation of results

Citácia:

Viktor Martiš: WiFi sieť 802.11b a jej simulácia, Bakalárska práca, Brno, FIT VUT v Brne, 2007

WiFi sieť 802.11b a jej simulácia

Prehlásenie:

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Václava Šimka. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

……
Viktor Martiš
15. máj 2007

Poďakovanie

Ďakujem Ing. Václavovi Šimkovi za odborný dohľad nad riešením mojej práce, za jeho ochotu pri konzultáciách a nespočetné množstvo rád a nápadov, ktorými ma zásobil. Jeho profesionálny pohľad na prezentáciu práce a výsledkov ma neustále viedol ku kvalitnému spracovaniu mojej bakalárskej práce. Ďalej by som sa chcel poďakovať Dr. Ing. Petrovi Peringerovi, ktorý mi pomáhal pri preklade zdrojových kódov vizualizačného nástroja Huginn.

Obsah

1 Úvod

Jednou z prirodzených vecí v živote každého človeka je komunikácia. Tak ako potrebujú komunikovať medzi sebou ľudia, tak si potrebujú medzi sebou vymieňať informácie aj jednotlivé systémy po celom svete.

Počítačové siete a vzájomné spájanie počítačov alebo obecne systémov je už tak štandardný princíp, že by bez neho nemohol svet správne fungovať. Každý deň sa s nimi stretávame a aktívne ich využívame. Veď vďaka nim funguje internet, telefóny, bankomaty a takmer všetky firmy po celom svete. Myšlienka, že by naraz prestali fungovať všetky dátové siete je absolútne nereálna. V počítačových sietiach si svoje vysoké postavenie právom vybudovali aj bezdrôtové siete. Ich najväčšia výhoda pramení už zo samotného slovíčka „bezdrôtové“. Implementácia takejto siete nevyžaduje často veľmi obtiažne manipulácie s káblami ako pri klasických sietiach. Pod pojmom bezdrôtové siete rozumieme mnoho typov a štandardov, ktoré majú svoje špecifické použite. Vybudovanie takejto siete je finančne a časovo náročná činnosť a je potrebné tento proces dobre naplánovať. To nám vo veľkej miere môže uľahčiť jej simulácia. Ďalej bude rozobraný hlavne štandard IEEE 802.11b a proces simulácie siete tohto štandardu v simulátore NS2.

Celá práca sa skladá z týchto kapitol:

Úvod

Krátky predslov, ktorý má za úlohu priblížiť rozoberanú problematiku a ukázať štruktúru celej bakalárskej práce.

Bezdrôtové siete

Objasnenie pojmu „bezdrôtové siete“ so zameraním na štandard IEEE 802.11b. Možnosti zostavenie siete podľa tohto štandardu. Letmý prehľad potrebného hardwaru pre takúto sieť. Spôsoby zabezpečenia bezdrôtových sietí.

Modelovanie a simulácia

Vysvetlenie základných pojmov, princípu a výhod modelovania a simulácie. Vytýčenie cieľa simulácie, ktorého realizácia je hlavnou náplňou celej bakalárskej práce.

Sieťový simulátor ns2

Oboznámenie sa s týmto simulátorom, s možnosťami jeho použitia,so spracovaním výstupov a výhodami či nevýhodami, ktoré pokrýva. Predstavenie TCL, NAM a XGRAPH.

Návrh a simulácia WiFi siete

Vytvorenie konkrétnej bezdrôtovej siete. Znázornenie a vysvetlenie reálneho a abstraktného modelu siete. Vytvorenie a vysvetlenie simulačného modelu (simulačného skriptu).

Analýza výsledkov simulácie

Rozobratie výsledkov simulácií, porovnanie medzi sebou a ich vyhodnotenie.

Záver

Zhrnutie dosiahnutých výsledkov, načrtnutie korektnosti a možnosti rozšírenia celej práce.

2 Bezdrôtové siete

Pod pojmom „bezdrôtové siete“ sa skrýva široká paleta možností a definícií. Keď o nich hovoríme, mohli by sme mať na mysli mobilné telefónne siete a štandard GSM, UMTS, CDMA, či profesionálne siete FWA alebo dokonca televízne či rádiové vysielanie [2].

Jednu vec majú ale všetky tieto siete spoločnú. Vysielajú na určených frekvenciách s určeným výkonom a ich prevádzka musí byť schválená patričným regulačným úradom, ktorý im zaisťuje, že do tohto pásma nesmie žiadna iná sieť zasahovať. Na základe vlastností rádiovej frekvencie, ktorá sa využíva nielen pri bezdrôtových sietiach vzniklo tzv. pásmo IMS (Industrial Scientific and Medical),čo v preklade znamená pásmo vyhradené pre priemyselné, vedecké a lekárske potreby. Pásmo frekvencie 2,4 GHz vymedzil pre tieto účely ako americký regulátor FCC, tak aj európsky ETSI .

2.1 Úvod do bezdrôtových sietí a pojem „WiFi“

Z počiatku každý výrobca zariadení na prenos v tomto pásme vyrábal vlastné technológie, takže akákoľvek kompatibilita zariadení medzi jednotlivými výrobcami bola nemysliteľná. Práve preto boli v r. 1997 zavedené štandardy, ktoré publikoval medzinárodný štandardizačný inštitút IEEE. Špecifikácie štandardu bezdrôtových sietí pracujúcich v pásme ISM boli označené číslom 802.11. Táto bezdrôtová sieť ponúkala rýchlosť prenosu dát až 2Mbps prenesených infračerveným signálom. Originál definoval ako prístup k médiu metódu Carrier Sense Multiple Access with Collision Avoidance (CSMA|CA). Značné percento dostupnej kapacity prenosového kanála je obetované (mechanizmom CSMA|CA) za účelom zvýšenia spoľahlivosti pri prenose údajov cez rôzne heterogénne a znečistené prostredia. V r. 1999 bola táto špecifikácia rozšírená o dve kvalitatívne vyššie špecifikácie pod revíznymi písmenami ako 802.11b, kde išlo o definíciu bezdrôtových sietí pracujúcich v pásme 2,4 GHz rýchlosťou až 11 Mbps a 802.11a, kde frekvenčné pásmo bolo stanovené na 5GHz a rýchlosť prenosu dosahovala až 54 Mbps. Ďalšia významná revízia pribudla v r. 2003 pod označením 802.11g, ktorá na pásme 2,4 GHz priniesla zvýšenie prenosovej rýchlosti na 54 Mbps a teda sa rýchlosťou vyrovnala štandardu 802.11a, ktorý je definovaný v pásme 5 GHz.

Aj keď tieto štandardy jednoznačne určili smer vývoju a výroby zariadení, ostalo v tomto smere veľa možností spresňovania špecifikácií. Na základe tejto skutočnosti vznikla certifikačná autorita WECA testujúca schopnosti zariadení, vzájomne si poskytovať služby a efektívne spolupracovať, hlásiace sa k štandardu 802.11. Výrobkom, ktoré vyhovovali a úspešne splnili všetky testy udelili logo WiFi. Toto logo slúži kupujúcemu zákazníkovi uistiť sa, že dané zariadenie je prepojiteľné s ostatnými zariadeniami s týmto logom. Popularita WiFi sieti stúpla natoľko, že sa v r. 2003 WECA premenovala na WiFi alianciu.

Štandard 802.11 má ale aj veľa nedostatkov a to hlavne v oblasti zabezpečenia a roamingu, kde sa ukázal ako nevyhovujúci. Štandardizátor IEEE ale naďalej pracuje a vydáva revízie, ktoré upravujú tieto nedostatky.

Následne bude okruh bezdrôtových sietí zúžený na problematiku WiFi sietí, teda sietí založených na štandarde 802.11 so zameraním na konkrétnu revíziu 802.11b. Pod pojmom WiFi teda bude ďalej myslený hlavne tento štandard. Pri chápaní celého objemu sietí 802.11 bude tento pojem označovaný ako 802.11, bez upresňujúcej revízie. Pokiaľ bude reč o bezdrôtových sietiach obecne, bude použitá skratka WLAN (Wireless Local Area Network).

2.2 Štandard IEEE 802.11b

Medzinárodný štandardizačný inštitút IEEE je spomenutý v kapitole 2.1. Takisto je v nej naznačený dôvod vzniku jednotlivých štandardov a ich povrchné predstavenie [4]. Nás ale bude zaujímať revízia s označením 802.11b.

Doplnok 802.11b bol k pôvodnému štandardu ratifikovaný v roku 1999. 802.11b definuje maximálnu priepustnosť 11 Mbps a využíva ten istý CSMA|CA prístup ku médiu ako bol definovaný v originálnom štandarde. Kvôli hlavičke CSMA|CA protokolu však v skutočnosti umožňuje maximálnu priepustnosť 5.9 Mbps použitím TCP a 7.1 Mbps použitím UDP rámca.

Produkty rešpektujúce štandard 802.11b sa na trhu objavili veľmi rýchlo vzhľadom na fakt, že 802.11b je priamym rozšírením DSSS (Direct-sequence spread spectrum) modulačnej technológie definovanej v pôvodnom štandarde. Technicky 802.11b štandard využíva komplementárny kódovací kľúč (CCK) ako svoju modulačnú techniku, ktorá je variantom CDMA (Code division multiple access). Preto boli čipsety a produkty ľahko prerobené na rozšírenie 802.11b. Dramatický nárast výkonu 802.11b oproti 802.11 spolu so znížením ceny viedol ku veľmi rýchlej akceptácii 802.11b ako definitívnej WLAN technológie.

802.11b je zvyčajne využívaný v po

-to-multipoint topológiách, kde prístupový bod komunikuje prostredníctvom viacsmerovej antény s jedným alebo viacerými klientmi umiestnenými v oblasti pokrytej prístupovým bodom. Typický rozsah vo vnútri budovy je 30 metrov pri rýchlosti 11 Mbps a 90 metrov pri 1 Mbps. S vysoko výkonnými externými anténami protokol umožňuje aj upravené prepojenie point-to-point. Typické sú vzdialenosti pokrytia do 8 km, avšak niektoré správy dokumentujú aj úspešný prenos do – 120 km v miestach bez prekážok.

Karty 802.11b operujú s rýchlosťami 11 Mbps, ale škálujú sa na 5,5, potom 2 a 1 Mbps (adaptívna selekcia rýchlosti) v prípade zhoršenej kvality signálu. Keďže prenos nižšími rýchlosťami využíva menej komplexné a nadbytočné metódy šifrovania dát, sú menej náchylné na poruchy v dôsledku interferencií alebo poruche signálu. Existujú rozšírenia 802.11b na rýchlosti 22, 33 a 44 Mbps, ale neboli zakotvené v IEEE štandarde. Nazývajú sa aj 802.11b+. Tieto rozšírenia boli odstránené vývojom 802.11g, ktorý definuje prenosové rýchlosti až do 54 Mbps a je spätne kompatibilný s 802.11b.

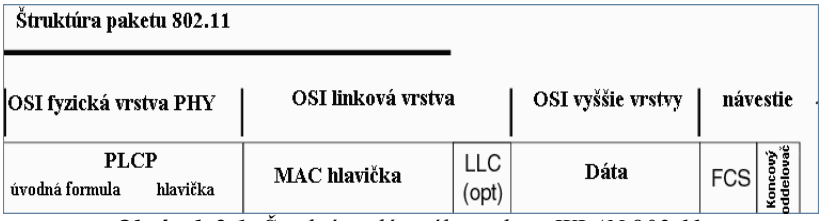
2.3 Štruktúra a typy paketov WLAN

Tak ako zbytok rodiny 802 LAN protokolov aj 802.11 WLAN posiela všetky dáta po sieti v podobe paketov. Tieto môžu byť troch typov: dátové, riadiace a kontrolné.

V kapitole 2.3.1 je rozobraný dátový paket 802.11 a v 2.3.2 riadiace a kontrolné pakety tohto protokolu.

2.3.1 Dátové pakety

Celá funkčnosť protokolu závisí od hlavičiek paketov. Technológia rádiovej frekvencie a mobilita zariadení zaviedli niekoľko spletitých požiadaviek pre 802.11 WLAN siete. Pridanie týchto zložitostí sa odrazilo na hlavičkách protokolu PLCP (physical layer convergence protocol) a takisto aj na objeme MAC hlavičiek. Túto zmenu dobre znázorňuje obrázok 2.1.

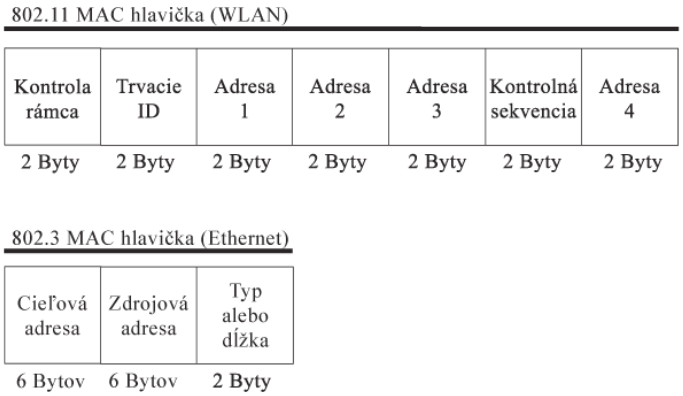


Obrázok 2.1: *Štruktúra dátového paketu WLAN 802.11*

LLC (logical link control) má za úlohu nadviazať a zrušiť spojenie na úrovni linkovej vrstvy. Popritom kontroluje tok dát, detekuje a preposiela zahodené pakety ak sú znova vyžiadané.

FCS (frame check sequence) je kontrolná sekvencia, ktorá slúži pre overenie neporušenosti prijatého rámca.

Pretože 802.11 WLAN musí byť schopná formovať a neustále upravovať členstvo v sieti a pretože podmienky rádiového vysielania sa môžu meniť, stáva sa koordinácia siete náročnejším faktorom. Pre jej funkciu sú vyhradené riadiace a kontrolné pakety. Popritom bežné dátové pakety obsahujú veľké množstvo pridaných informácií o podmienkach v sieti a jej topológii, narozdiel od hlavičky dátového paketu Ethernetu. Tento rozdiel je znázornený na obrázku 2.2.



Obrázok 2.2: *Porovnanie MAC hlavičky 802.11 s hlavičkou Ethernetu*

2.3.2 Riadiace a kontrolné pakety

Kontrolné pakety sú krátke prenosy, ktoré priamo sprostredkujú alebo kontrolujú komunikáciu. Patria medzi ne RTS, CTS a ACK pakety používané v štvorfázvom cykle prenosu, ďalej pakety PSP (power save polling) a pakety pre oznámenie o konci bezkolíznej periódy vo vnútri konkrétnej BSS.

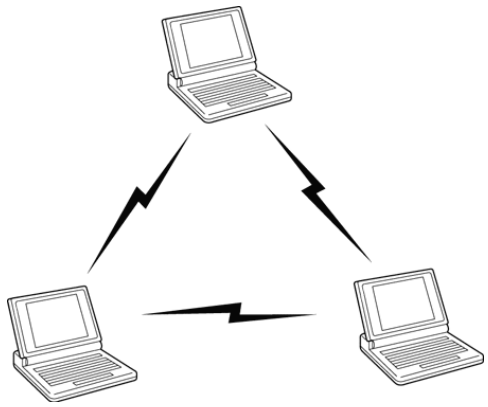
Riadiace pakety sú využívané na podporu autentifikácie, spojovania a synchronizácie. Ich formát je podobný formátu už spomínaných dátových paketov až na menší počet polí v MAC hlavičke. Naviac riadiace pakety môžu mať dátové polia fixnej alebo premenlivej dĺžky, v závislosti na definícii ich presného podtypu.

2.4 Typy WiFi sietí

Základný stavebný blok WiFi sietí označujeme ako **Basic Service Set (BSS)**, teda základný súbor služieb. Jedná sa o skupinu staníc, ktoré spolu komunikujú. Táto spoločná komunikácia prebieha v území vymedzenom prienikmi dosahov týchto staníc. Takéto územie sa nazýva **Basic Service Area (BSA)**. Ak sa stanica nachádza v rámci BSA môže komunikovať s ďalšími členmi BSS. Rozpoznávame dva hlavné typy sietí podľa toho, ako komunikácia medzi členmi BSS prebieha [2].

2.4.1 Ad-Hoc siete

Tento typ sa tiež niekedy nazýva nezávislá sieť. Je to z toho dôvodu, že jednotlivé stanice môžu medzi sebou komunikovať priamo, teda nezávisle na nejakom prostredníkovi. Z toho vyplýva, že pokiaľ chcú stanice spolu komunikovať, musia byť vo vzájomnom rádiovom dosahu. Najčastejším použitím sietí ad-hoc je prepojenie počítačov z nejakého špecifického dôvodu a to na obmedzený čas – napr. LAN party, nárazová výmena dát a pod. Príklad takejto siete je na obrázku 2.3.



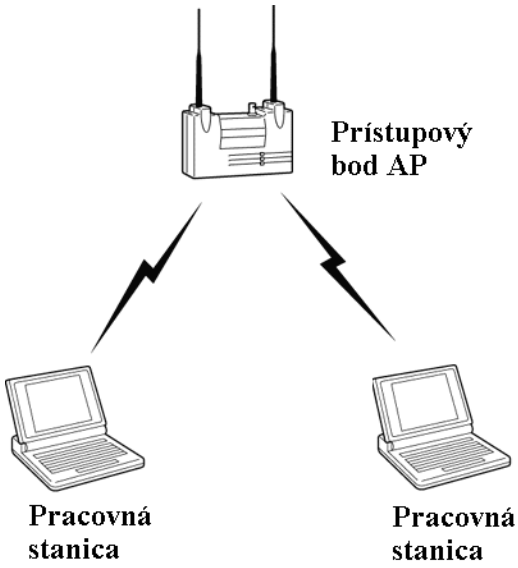
Obrázok 2.3: *Jednoduchá ad-hoc sieť*

2.4.2 Infraštruktúrne siete

Už názov prezrádza, že tieto siete majú presne vymedzenú infraštruktúru. Ako spojovací element medzi zariadeniami slúži prístupový bod (AP – acces point). Je to rozhranie medzi bezdrôtovou a drôtovou sieťou plniace funkciu dátového mostu (bridge). Prístupový bod je schopný komunikovať s viac ako jednou stanicou. Preto môže prepojovať navzájom medzi sebou aj stanice, ktoré sú v jeho dosahu nezávisle na tom, či tieto stanice chcú používať most do káblového Ethernetu.

Pokiaľ chceme aby jedna bezdrôtová stanica komunikovala s inou stanicou v infraštruktúrnej sieti, musia dáta putovať cez prístupový bod.

Infraštruktúrne siete poskytujú komplexnejšie využitie než siete ad-hoc. Je to hlavne z dôvodu pokrytia. Tieto siete sa pomocou smerových antén dajú rozšíriť aj na väčšie vzdialenosti, kde pri ad-hoc sietiach je to prakticky nemožné. Ďalej umožňujú centrálnu správu, čo pri ad-hoc sietiach takisto nie je možné. Príklad takejto siete je na obrázku 2.4.



Obrázok 2.4: *Sieť s infraštruktúrou*

2.5 Hardware a antény pre WiFi siete

V súčasnej dobe je na trhu naozaj pestrý výber bezdrôtových zariadení od rôznych výrobcov. Medzi takéto zariadenia patria napr. prístupové body (AP), bezdrôtové smerovače, dátové mosty, bezdrôtové opakovače (repeater), karty PCI, adaptéry USB a mnohé ďalšie. Poskytujú väčšinou rôzne možnosti spravovania a to pomocou SNMP, webového rozhrania či telnetu. Dôležitou súčasťou hardwaru sú antény, bez ktorých sa WiFi siete vonkoncom nezaobídu.

Srdcom každého WiFi zariadenia je jeho čipová sada (chipset), podobne ako je tomu aj v počítačoch. Ona rozhoduje o tom, aké funkcie WiFi zariadenie ponúka, výkon procesoru rozhoduje o tom, ako rýchlo ich toto zariadenie spracováva a ovládací software rozhoduje o tom, čo všetko môže užívateľ z týchto funkcií zmeniť.

Následne podrobnejšie rozoberiem prístupové body a antény. Najmä tieto zariadenia sú použité v simulovanej sieti a ďalej nieje potrebné rozoberať všetky ostatné.

Prístupový bod

Jeho výkon je individuálny a odvíja sa od neho aj zvládané množstvo súčasne pripojených zariadení. Tento počet sa môže pohybovať v rozmedzí desiatok až stoviek pripojených jednotiek. Napájanie prístupových bodov môže byť pomocou sieťového adaptéru alebo u kvalitnejších zariadení môže byť využité napájanie cez nevyužité vodiče ethernetového káblu (PoE – power over Ethernet). Takýto spôsob napájania sa využíva hlavne v priestoroch s komplikovaným prístupom k sieťovému zdroju.

Antény

Patria k neodmysliteľnej časti bezdrôtovej siete. Integrované antény alebo antény dodávané k zariadeniam postačujú k prepojeniu zariadení na malé vzdialenosti a to väčšinou v priestoroch jednej budovy.

Antény sa rozdeľujú podľa smeru, do ktorého je signál distribuovaný, na:

- Všesmerové - šíria signál do všetkých strán, teda vykrývajú uhol o šírke 360 stupňov.

- Sektorové - vyžarujú signál pod určitým uhlom a vykrývajú uhol od 0 do 180 stupňov.
- Smerové – sú nasmerované na presný bod a len tam vyžarujú signál.

Ďalším dôležitým parametrom antény je jej zisk (gaint). Od neho sa odráža schopnosť antény zachytávať vzdialené signály.

Na šírenie signálu má dopad veľké množstvo vonkajších vplyvov, ako napr. počasie, terén prenosu a pod. Často krát signál rušia a v niektorých prípadoch dokonca úplne hltia.

2.6 Zabezpečenie sietí 802.11

Jeden z hlavných nedostatkov bezdrôtových sietí je ich bezpečnosť. Dosahy WiFi sietí sa nedajú presne určiť. Neplatí tu pravidlo ako u káblových sietí, že pripojiť sa dá iba v určitých miestach, kde je vyvedená prípojka do siete. Tým, že sú jednotlivé dáta prenášané vzduchom, sú priam poskytnuté na odchytávanie potenciálnym útočníkom.

Bezpečnosť bezdrôtových sietí môžeme teda rozdeliť do dvoch hlavných skupín:

- Šifrovanie - teda zabezpečenie prenášaných dát pred odpočúvaním.
- Autentifikácia - teda riadenie prístupu oprávnených užívateľov.

2.6.1 Šifrovanie prenášaných dát

WiFi obsahuje možnosť šifrovania dát už vo svojom štandarde 802.11. Je to samozrejmý predpoklad k tomu, aby bol dôveryhodný pre nasadenie v spoločnostiach.

WEP – Wired Equivalent Privacy

Vo WiFi sietiach sa o zabezpečenie stará WEP. Je to štandard pre zabezpečenie rádiovej časti siete. Štandard WEP používa symetrickú prúdovú šifru RC4, teda šifru s tajným kľúčom. Podstatou tejto šifry je, že sa odosielaná správa kóduje podľa nejakého kľúča (obvykle slova alebo sekvencie znakov) a na cieľovom bode sa zase podľa tohto kľúča rozkóduje.

Kľúč sa expanduje v pseudonáhodný prúd rovnakej dĺžky, ako je šifrovaná správa. O pseudonáhodnosť sa stará pseudonáhodný generátor čísel PRGN. Šifrovanie prebieha jednoducho tak, že na šifrovanej hodnote sa urobí logická operácia XOR s kľúčovým prúdom. Rozšifrovanie prebieha rovnako.

Problémom tajného kľúča WEP zabezpečenia je v tom, že štandard žiadno nerieši jeho automatickú distribúciu a je na jednotlivých výrobcoch, ako distribúciu kľúča realizujú. Keďže WiFi zariadenia väčšinou dokážu kľúč prijať iba vo forme zápisu do ich konfigurácie a teda nie bez zásahu ľudského faktora, je obtiažne hovoriť o skutočnom tajnom kľúči. Z toho plynie nedokonalosť zabezpečenia pomocou WEP.

2.6.1 Autentifikácia – riadenie prístupu do siete

Druhou dôležitou bezpečnostnou stratégiou je riadenie prístupu do siete, teda autentifikácia užívateľa.

802.11 špecifikuje dve metódy pre autentifikáciu:

- **Open-system autentifikácia** – táto metóda spočíva v tom, že prístupový bod príjme klientské zariadenie na základe údajov, ktoré mu toto poskytne bez toho, aby ich overoval.
- **Shared-key autentifikácia** – táto metóda je už sofistikovanejšia a v prípade jej nasadenia je nutné v sieti používať WEP. Jej podstata spočíva v kľúči, ktorý pozná každé zariadenie,

ktoré chce pristupovať do siete.

2.6.2 Ďalšie bezpečnostné štandardy

Vlastné bezpečnostné mechanizmy 802.11 nepostačujú a tak museli byť vytvorené ďalšie štandardy.

Extensible authentication protocol (EAP) a 802.1x

IEEE 802.1x je obecný bezpečnostný rámec pre všetky typy sietí zahrňujúci autentifikáciu užívateľov, integritu správ (šifrovanie) a distribúciu kľúčov. Overovanie pre bezdrôtové siete sa realizuje na úrovni portov prístupového bodu WLAN s tým, že protokol nie je špecifikovaný pre bezdrôtové siete.

802.1x je založený na protokole EAP. Overovanie v bezdrôtovej sieti uskutočňuje prístupový bod pre klientov na základe ich výzvy, pomocou zoznamu alebo externého autentifikačného systému založeného na servere Kerberos alebo RADIUS. 802.1x používa k šifrovaniu dátovej komunikácie pre každé autentizované zariadenie dynamické kľúče.

WPA (WiFi protected access)

Hlavnou výhodou WPA je prijatie mechanizmov z vtedy ešte len vznikajúceho štandardu 802.11i a to ako pre šifrovanie komunikácie, tak aj pre riadenie prístupu do bezdrôtovej siete. Pre šifrovanie komunikácie je používaný TKIP (Temporal Key Integrity Protocol). TKIP využíva rovnaký šifrovací algoritmus ako WEP, používa štandardne 128bitový kľúč (WEP len 64) a na rozdiel od WEP obsahuje dynamické dočasné kľúče.

Pre všetky výhody sa o WPA a TKIP uvažovalo ako o dočasnom riešení, ktoré bolo užitočné do príchodu štandardu 802.11i.

802.11i

Tento štandard známy tiež ako WPA2 je platný pre všetky bezdrôtové siete. Bol schválený 24. júla 2004. Používa blokovú šifru Advanced Encryption Standard (AES), zatiaľ čo skorší WEP a WPA používajú prúdovú šifru RC4. 802.11i architektúra obsahuje následujúce komponenty: IEEE 802.1X pre autentifikáciu (používa teda Extensible Authentication Protocol (EAP) a autentifikačný server), Robust Security Network (RSN) pre udržovanie záznamu asociácií a na AES založený Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP), ktorý poskytuje utajenie, integritu a autentifikáciu. Ďalším dôležitým prvkom autentifikačného procesu je štvorcestné uvítanie.

3 Modelovanie a simulácia

Pre pochopenie celej práce je nevyhnutné rozumieť základným pojmom, princípom a výhodám modelovania a simulácie. Tento krátky prehľad poskytuje nutný základ z tejto oblasti. Ten postačuje k pochopeniu postupu, ktorým je riadený celý proces simulácie WiFi siete.

3.1 Základné pojmy

Pochopenie princípov modelovania a simulácie vyžaduje aspoň neformálnu definíciu niekoľko základných pojmov, s ktorými budem v ďalšom texte pracovať [1].

- **Systém** môžeme obecne definovať ako súbor elementárnych častí (prvkov systému), ktoré majú medzi sebou určité väzby.
  Systémy môžeme deliť podľa existencie na:
  - Reálne (existujúce) systémy.
  - Nereálne (fiktívne, ešte neexistujúce) systémy.

  Ďalej podľa zmeny stavu na:
  - Statické systémy – nemenia svoj stav v čase.
  - Dynamické systémy – menia stav v čase.

  Pre simuláciu sú zaujímavé hlavne dynamické systémy
- **Model** je napodobenina systému iným systémom (napr. počítačovým programom). Model systému musí napodobňovať všetky pre naše účely podstatné vlastnosti systému.
- **Modelovanie** je proces vytvárania modelov systému na základe našich znalostí. Kvalita vytvoreného modelu zásadným spôsobom ovplyvní výsledky získané experimentovaním s modelom.
- **Simulácia** je metóda získavania nových znalostí o systéme experimentovaním s jeho modelom. Pre tieto účely simulácie musí byť model popísaný odpovedajúcim spôsobom – nie každý model je pre simuláciu vhodný.

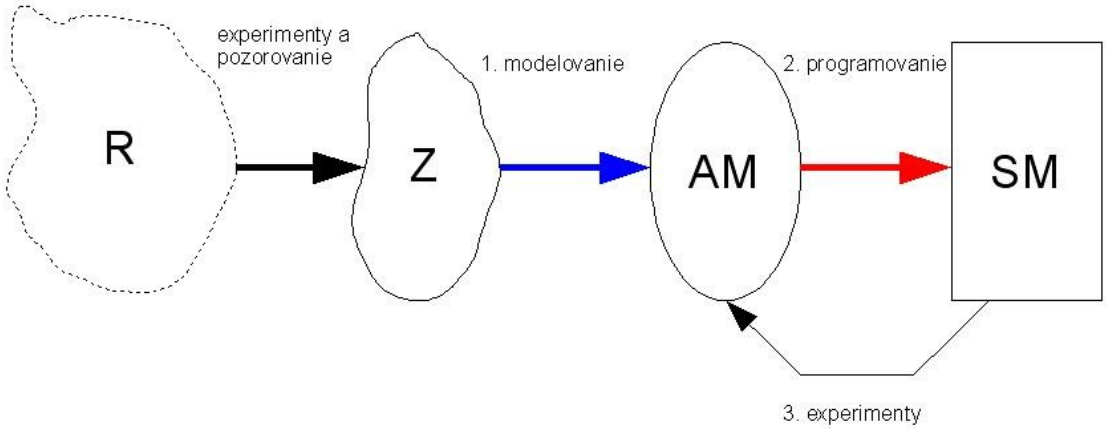
Podrobnejšie špecifikácie a súvislosti týchto pojmov viď [1].

3.2 Princíp modelovania a simulácie

Cieľom simulácie je získavať nové znalosti o skúmanom systéme. Aby sme mohli uskutočňovať simuláciu, je nutné vytvoriť vhodný model tohto systému. Postupujeme tak, že:

1. Najskôr vytvoríme tzv. *abstraktný model*, ktorý nezahrňuje všetky naše znalosti o modelovanom systéme, ale sú v ňom vybrané len tie, ktoré sú pre naše účely podstatné. Tým dosiahneme zjednodušenie modelu na zvládnuteľnú úroveň. Abstraktný model môže mať napríklad formu matematických rovníc.
2. Na základe abstraktného modelu vytvoríme *simulačný model*, ktorý už ďalej nič nezjednodušuje a musí zahrňovať všetky vlastnosti abstraktného modelu. Rozdiel medzi abstraktným a simulačným modelom je iba v možnosti uskutočňovať experimenty (simulačný model je napr. spustiteľný program, ktorý počíta výsledky podľa zadaného počiatočného stavu, vstupov a parametrov modelu).
3. So simulačným modelom robíme simulačné experimenty a ich výsledky analyzujeme. Výsledkom sú informácie o správaní systému z ktorých zobecnením získame nové znalosti.

Obrázok 3.1 znázorňuje celý proces získavania znalostí s využitím simulácie.



Obrázok 3.1: *Realita→Znalosti→Abstraktný model→Simulačný model [1]*

3.3 Výhody simulačných metód

Obecne môžeme za hlavné výhody simulácie v porovnaní s reálnymi experimentami považovať nižšiu cenu. Väčšinou i menšiu časovú náročnosť experimentovania a naprostú bezpečnosť pri uskutočňovaní experimentov.

- Cena je dôležitým kritériom pre nasadenie simulácie. Experimenty s reálnym systémom môžu byť veľmi drahé – napríklad havarijné testy automobilov sa dnes robia iba pre validáciu výsledkov simulačných experimentov a preto je spotreba automobilov pri týchto testoch výrazne menšia.
- Rýchlosť experimentovania je ďalším dôležitým faktorom pre simuláciu. Simuláciu môžeme v rámci možností zrýchliť alebo spomaliť podľa potreby (zrýchľovanie je samozrejme obmedzené výkonom nášho počítača). Napríklad rast rastlín je veľmi pomalý proces, ktorý možno výrazne urýchliť a dosiahnuť výsledok podstatne skôr. Naopak pre sledovanie veľmi rýchlych dejov v reálnych systémoch je potrebné veľmi drahé experimentálne vybavenie (alebo dokonca nemožno tieto javy sledovať vôbec), kde pri simulačných modeloch môžeme tieto deje ľubovoľne spomaliť. Príkladom môže byť sledovanie atómov v nanotechnologických systémoch alebo putovanie prenášaných paketov v dátovej sieti, ktorú budeme neskôr simulovať.
- Bezpečnosť je pri niektorých reálnych experimentoch veľký problém, kedy často nemožno tieto experimenty z bezpečnostných dôvodov vôbec uskutočniť. Príklady môžu byť niektoré jadrové reakcie, šírenie epidémií a podobne. Simulačné experimenty na počítačoch sú naprosto bezpečné a dovoľujú nám uskutočňovať čokoľvek.
- Simulácia dovoľuje experimentovať s modelmi veľmi zložitých systémov, sme obmedzení iba možnosťami simulačného prostredia, ktoré používame. Reálne experimenty by pri takto veľkých systémoch boli neuskutočniteľné.
- Niekedy je simulácia jediný spôsob ako experimentovať (napr. simulácia zrážky dvoch

galaxií vo vesmíre a pod.).

Pretože výkon počítačov neustále rastie, je ekonomicky výhodnejšie, rýchlejšie a často jediné možné experimentovať s modelom, než s originálom.

3.4 Cieľ simulácie

Úlohou simulovania je získanie nových znalostí o simulačnom modele a podľa miery validity tohto modelu s reálnym systémom tieto znalosti uplatniť. Preto je nutné zvoliť si cieľ výsledných znalostí, ktoré chceme získať.

V predvádzanej simulácií sa dá zamerať na množstvo vlastností, ktoré sú v sieti zaujímavé. Ja som si vytýčil cieľ, ktorý je podľa mňa najviac diskutabilný. Budem zisťovať, akú maximálnu rýchlosť pripojenia do internetu je daná sieť schopná zvládnuť. Konkrétne mám na mysli, aká maximálna rýchlosť bude zvládaná zariadeniami v sieti. Zistenie hranice rýchlosti je dôležité kvôli tomu, aby nebolo pripojenie do internetu k sieti zbytočne preplácané a zároveň, aby sieť pracovala s plným vyťažením a ponúkala tak jednotlivým klientom maximálny komfort v podobe rýchleho pripojenia do internetu.

Vďaka metóde CSMA|CA, ktorá je v štandarde 802.11b použitá, je maximálna rýchlosť prenosu znížená z 11 Mbps na 5,9 Mbps. Táto rýchlosť je ale teoretická a v reálnych podmienkach prakticky nedosiahnuteľná. Pri bezdrôtových spojoch veľmi záleží od okolitého prostredia a to obzvlášť pri implementácii siete vo vonkajšom prostredí (mimo priestorov budovy). Výrazný vplyv má tiež kvalita použitých zariadení, od ktorej priamo závisí ich výkon. Nemožno zaprieť fakt, že kvalita zariadení je veľmi závislá od značky výrobcu.

V NS2 nieje možnosť definovať zariadenia od konkrétnych výrobcov. Sú v ňom uvažované obecné zariadenia. Pre presnosť našej simulácie to ale úplne postačuje. Nastaveniami, ktoré je v NS2 možné vykonať na bezdrôtových uzloch som sa snažil čo najviac priblížiť k reálnej sieti.

3.5 Sieťové simulátory

Práca sa zaoberá simuláciou počítačovej siete. Nástrojov na jej realizáciu by sme našli veľké množstvo. Existuje mnoho komerčných simulátorov, ktoré sa zameriavajú najmä na vyššie vrstvy ISO/OSI modelu. Pre nás sú ale zaujímavejšie voľne dostupné nástroje, pre ktorých použitie nebudeme potrebovať žiadne finančné prostriedky. Tých je takisto veľké množstvo. Väčšinou sú výsledkom nejakého projektu a preto často nepokrývajú celú oblasť problematiky simulovania počítačových sietí, ale iba určité štandardy a pod.

Medzi takéto nástroje patria:

- **Cnet Network Simulator –** Simulátor protokolov linkovej, sieťovej a transportnej vrstvy založených na rôznych kombináciách liniek bod – bod a segmentov Ethernetu (štandard IEEE 802.3). Je napísaný a vyvíjaný v jazyku C (s grafickým rozhraním) od roku 1991. Určený je predovšetkým na výukové účely. Cnet network simulator je vytvorený len pre operačné systémy platformy UNIX. Pre ostatné platformy nie je dostupný. Program povoľuje nastavovať veľkosti správ, počty chýb v kanáli (stratu i poškodenie rámcov) či réžiu prenosu. Obsahuje debugger pre ladenie napísaných protokolov. Program je voľne dostupný, šíriteľný a modifikovateľný pod GNU General Public License. Domovská stránka tohto projektu viď [9].
- **Data-link NP Simulator –** Jednoduchý simulátor protokolov linkovej vrstvy. Často sa v ňom objavuje nedeterministické spávanie a nie je veľmi prehľadný. Je implementovaný

ako Java applet. Umožňuje simulovať selektívne opakovanie. Má jednoduché nastavovanie (veľkosť okienka, strata či poškodenie rámca a pod.). Stránka kde je tento nástroj vystavený viď [10].

- **NS2** – simulátor TCP, UDP, smerovacích, multicastových protokolov a to cez drôtové i bezdrôtové siete (miestne i satelitné). Je vyvíjaný od roku 1989 a bol založený na REAL network simulátore. Jeho implementácia v jazyku C++ je podporovaná operačnými systémami založenými na platforme UNIX, Solaris, Windows (Cygwin). Je dobre zdokumentovaný. Podporuje spúšťanie cez príkazový riadok, ale existujú nástroje aj pre grafický výstup. Jeho zdrojový kód je voľne dostupný na internete. Tento nástroj bude v práci použitý ako primárny simulátor a v kapitole 4 je podrobne rozobraný.
- **SSFNet** – moduly IP, TCP, UDP, BGP4, OSPF, a ďalšie s voľne šíriteľným zdrojovým kódom založená na Jave. Pre výukové účely sú k dispozícii zdarma, inak sú platené. Domovská stránka viď [11].
- **Jasper** – simulátor umožňujúci interaktívne simulácie celej rady komunikačných protokolov. Je implementovaný v Jave. Medzi preddefinované protokoly patria: ABP, ABRA, BOOTP, HTTP, IP, SMTP, SWP, TCP, TFTP a UDP. Obsahuje framework pre vytváranie nových protokolov. Simulátor je možno spúšťať z internetu. Simuláciu v ňom možno krokovať, ale nie je možné určovať rýchlosť animácie. Program je voľne dostupný, šíriteľný a modifikovateľný pod GNU General Public License. Jeho domovská stránka viď [12].
- **NCTUns – s**imulátor rady protokolov v prostredí drôtových i bezdrôtových sietí. Je vyvinutý pre OS Linux. Pre jeho beh je potrebný výkonnejší hardware (min 1,5 GHz CPU, 256 MB pamäte). Podpora protokolov IP, RIP, OSPF, UDP, TCP, IEEE 802.11(b), Diffserv (QoS), RTP/RCTP/SDP a rada ďalších. Demonštračné videá sú k dispozícií na domovskej internetovej stránke projektu [13].
- **Network simulator – s**imulátor napísaný ako Java applet. Vytvára simuláciu ideálneho média (bez chýb). Ponúka množstvo rôznych nastavení ale je neprehľadný. Domovská stránka viď [14].
- **P2psim – s**imulátor p2p protokolov pre platformu UNIX. Jeho rozhranie je čisto konzolové. Má neprehľadný štýl zápisu (3 textové súbory na vstupe). Výstupom programu je iba štatistika výsledku. Neexistuje k nemu prakticky žiadna dokumentácia. Domovská stránka viď [15].
- **Ďalšie simulátory –** UMTSProSIM, Peersim, Opnet, Flow Control Protocol Simulator, JavaSim a iné.

4 Sieťový simulátor NS2

NS2 (netwok simulator) je nástroj na vytváranie diskrétnych simulácií so zameraním na pozorovanie a testovanie sietí. NS2 poskytuje podporu pre simuláciu TCP, smerovania a multicastových protokolov káblových aj bezdrôtových sietí (lokálnych i satelitných).

NS2 vznikol ako variant *REAL network simulator*-u v r. 1989 a za pár rokov sa podstatne vyvinul. V r. 1995 vývoj NS2 podporila agentúra ministerstva obrany USA – DARPA (Defense Advanced Research Projects Agency), ktorá jeho vývoj podporuje až doteraz.

NS2 je postavený na dvoch programovacích jazykoch:

- **C++ -** Objektovo orientovaný programovací jazyk, v ktorom je NS2 implementovaný.
- **OTcl - Objektové** rozšírenie skriptovacieho jazyka Tcl, ktorý sa používa na preklad simulačných skriptov popisujúcich navrhovanú sieťovú simuláciu.

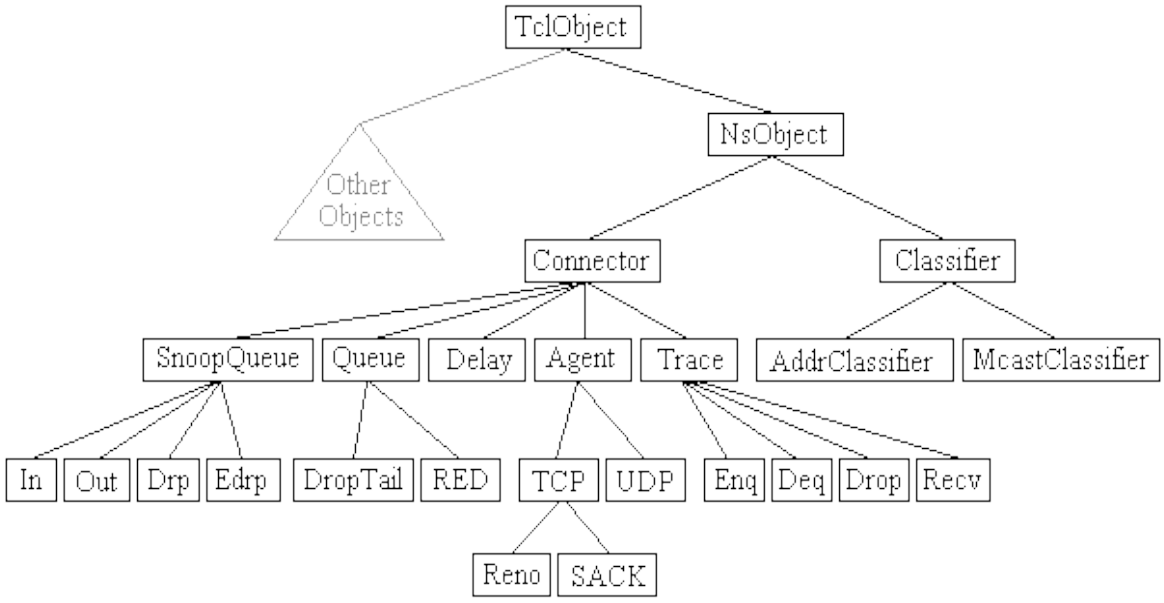
NS2 obsahuje bohaté knižnice na popis sietí a protokolov. Existujú teda dve hierarchie tried, ktoré sa dajú uplatniť: preložená C++ a OTcl hierarchia, ktoré na seba nadväzujú.

C++ hierarchia nám dovoľuje docieliť efektívne simulácie a rýchlejšie spúšťacie časy. Je obzvlášť dôležitá pre detailnejšie definície a operácie protokolov. Umožňuje znížiť počet paketov a udalostí v jednotlivých procesoch simulácie.

OTcl skript slúži na podrobné zadefinovanie topológie siete so špecifickými protokolmi a aplikáciami (ktorých správanie sa je už zadefinované v preloženej hierarchii tried), ktoré chceme simulovať a formu výstupu, akú požadujeme aby nám simulátor vytvoril [6].

4.1 Sieťové komponenty NS2

Pre chápanie logického zloženia NS2 je potrebné poznať aspoň čiastočnú hierarchiu tried, ktorou je NS2 implementovaný. Na obrázku 4.1 je znázornená táto čiastočná hierarchia. Pre úplnú hierarchiu viď domovskú stránku NS2 [5].

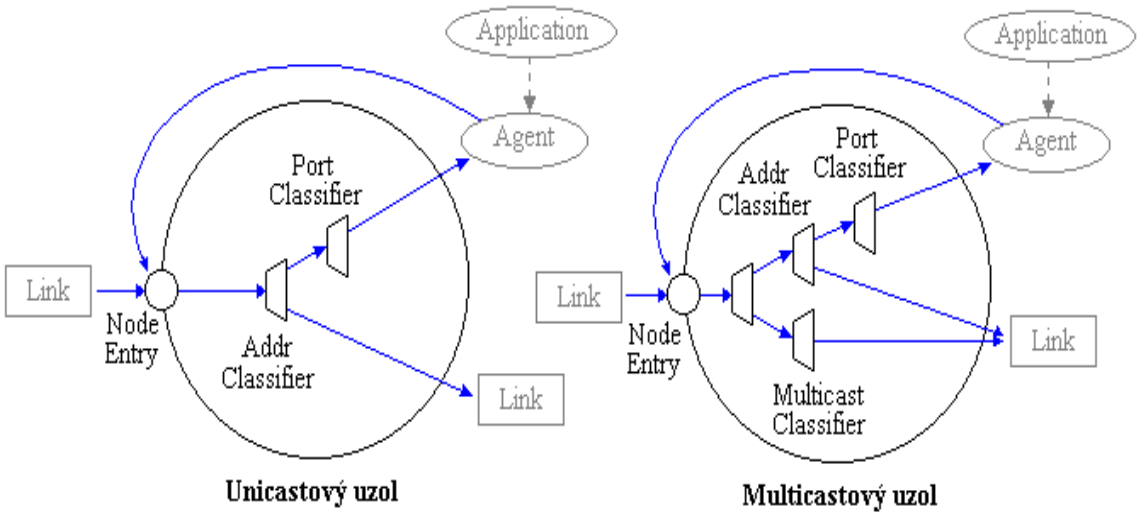


Obrázok 4.1: *Čiastočná hierarchia tried NS2*

Koreň hierarchie tvorí trieda TCLObject, ktorá je supertriedou všetkých OTcl knižných objektov. Z TclObject triedy dedí trieda NsObject, ktorá je supertriedou všetkých základných sieťových komponentných objektov pracujúcich s paketmi, ktoré môžu upravovať zloženie skriptu tvorené sieťovými objektami ako sú uzly a linky. Základné sieťové komponenty sa ďalej delia na dve podtriedy. Connector a Classifier založené na počte možných ciest dátových výstupov. Základné sieťové komponenty, ktoré poskytujú jednu cestu dátového výstupu patria do triedy Connector a prepínajúce objekty, ktoré poskytujú viacero ciest dátového výstupu patria do triedy Classifier.

4.1.1 Uzly a smerovanie

Uzol je tvorený objektom, ktorý predstavuje vstup uzlu a objektom classifier. V NS2 existujú dva typy uzlov. Unicastový vlastní address classifier, ktorý poskytuje smerovanie a triedenie portov unicastového prenosu. Multicastový uzol obsahuje classifier, ktorý oddeľuje multicastové pakety od unicastových a ďalej vykonáva smerovanie multicastu. Oba typy sú znázornené na obrázku 4.2.

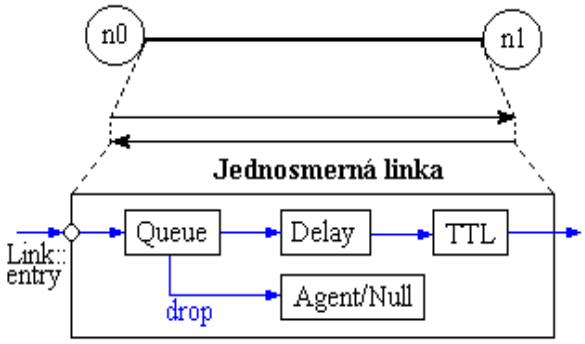


***Obrázok 4.2**: Uzly v NS2*

4.1.2 Linky

Druhým hlavným objektom v NS2 je linka. Keď užívateľ vytvorí linku použitím funkcie *duplex-link* objektu simulátora, vytvorí vlastne dve jednosmerné linky v oboch smeroch, ako je naznačené na obrázku 4.3.

Výstupná fronta uzlu je v NS2 implementovaná ako časť jednoduchej linky. Pakety vybrané z fronty sú poslané do Delay objektu, ktorý simuluje oneskorenie linky. Zahodené pakety sú poslané Null Agentovi, pomocou ktorého sú uvoľnené. Nakoniec TTL objekty počítajú Time To Live (čas platnosti) pre každý prijatý paket a obnovujú TTL pole jednotlivých paketov.



Obrázok 4.3: *Linka v NS2*

4.2 TCP v NS2

Implementácia TCP v NS2 má niekoľko zaujímavých vlastností. Za prvé, TCP v NS2 neimplementuje riadenie toku (flow control) ako prevenciu proti zahlteniu prijímača. Je implementované iba riadenie zahltenia (congestion control) ako prevencia proti zahlteniu siete. Okienko zahltenia (cwnd) môže rásť až do limitu špecifikovaného premennou *window* objektu TCP vysielača.

Za druhé, TCP v NS2 neimplementuje blokujúce volanie pre predávanie dát z vysielacej aplikácie do TCP vysielača. Musíme preto u linky odchádzajúcej z vysielacej stanice nastaviť dostatočne veľkú frontu, aby nikdy nedošlo k jej naplneniu v dôsledku veľkého objemu dát od vysielacej aplikácie.

Za tretie, TCP v NS2 neposkytuje priamo žiadnu informáciu o aktuálnej priepustnosti. Je na užívateľovi, aby si sám vytvoril vhodnú podtriedu štandardných tried, ktorá bude poskytovať informáciu o aktuálnej priepustnosti.

4.3 Výstup NS2

Výsledok simulácie v NS2 má podobu textového súboru s presnou syntaxou. Tá pozostáva z jednotlivých riadkov, kde každý riadok predstavuje prenos jedného paketu medzi dvoma uzlami v topológii. Takéto riadky majú obecný formát. Paket prenášaný bezdrôtovo má trochu odlišný formát od paketu prenášaného medzi dvoma pevnými uzlami.

4.3.1 Obecný formát výstupného súboru

Pri tomto type prenosu sa každý riadok výstupu skladá z dvanástich políčok, ktoré sú znázornené na obrázku 4.4.

Udalosť	Čas	Zdroj. uzol	Cieľ. uzol	Typ pkt	Veľkosť pkt	Príznaky	FID	Zdroj. adresa	Cieľ. adresa	Porad. číslo	Pkt ID

Obrázok 4.4: *Polia obecného výstupu NS2*

Následne sú rozobrané jednotlivé polia v zhodnom poradí ako sú znázornené na obrázku 4.4:

1. Udalosť nadobúda hodnoty jednej zo štyroch možností r, +, -, d, ktoré odpovedajú prijatiu paketu (na konci linky), zaradeniu či odobraniu paketu z fronty a zahodeniu

paketu.

2. Druhé pole obsahuje čas kedy k danej udalosti došlo.
3. Vstupný uzol linky na ktorej došlo k udalosti.
4. Výstupný uzol linky na ktorej došlo k udalosti.
5. Typ paketu (napr. CBR alebo TCP; tento typ súhlasí s typom používaným pri zadefinovanom spojení v simulácii).
6. Veľkosť paketu.
7. Príznaky prenosu.
8. FID (flow ID) je identifikácia toku IPv6, ktorú si môže užívateľ nastaviť pre každý dátový tok na začiatku OTcl skriptu. Toto pole sa ďalej môže využiť pre analyzačné ciele alebo na špecifikáciu farby prenosov v aplikácii NAM.
9. Zdrojová adresa určená formou „node.port“
10. Cieľová adresa daná tou istou formou.
11. Poradové číslo paketu sieťovej vrstvy. V skutočných sietiach pri prenose UDP toto číslo neexistuje. V NS2 sa toto číslo pri UDP používa pre analyzačné ciele.
12. Posledné pole je unikátna identifikácia paketu.

4.3.2 Syntax výstupu pri bezdrôtovom prenose

Ako je uvedené v 4.3, pri bezdrôtovom prenose sa formát výstupu odlišuje od obecnej syntaxe. Na obrázku 4.5 je znázornená štruktúra riadku takéhoto formátu.

Udalosť	Čas	Číslo uzlu	Typ pkt	---	Globálne porad. číslo	Detail. typ pkt	Veľkosť pkt	MAC inf.	---	Sieťové inf.	TCP inf.

Obrázok 4.5: *Polia výstupu pri bezdrôtovom prenose*

Následne sú rozobrané jednotlivé polia v zhodnom poradí ako sú uvedené na obrázku 4.5:

1. Udalosť môže nadobúdať hodnotu r, s, f, D odpovedajúce prijatiu, poslaniu, preposlaniu a zahodeniu paketu. Tiež sa ako hodnota môže vyskytnúť písmeno M, ktoré vyjadruje lokalizáciu uzlu alebo jej zmenu.
2. Čas v ktorom sa udalosť vyskytla.
3. Číslo uzlu.
4. Typ paketu nám určuje, či sa daný paket týka MAC vrstvy alebo vyššej - transportnej vrstvy (AGT) alebo či sa jedná o smerovací paket (RTR). Takisto môže toto pole nadobudnúť hodnotu IFQ, ktorá indikuje udalosť vo fronte rozhrania (napr. zníženie pozície vo fronte).
5. --- (nevyužité pole).
6. Globálne poradové číslo paketu (toto nie je poradové číslo transportnej vrstvy v TCP).
7. Detailnejšie informácie o type paketu (napr TCP,ACK alebo UDP).
8. Veľkosť paketu.
9. Toto pole je vyznačené hranatými zátvorkami, v ktorých sú 4 čísla. Obsahujú informácie o MAC vrstve. Prvé číslo v hexadecimálnom tvare udáva čas očakávaný na poslanie cez bezdrôtový kanál. Druhé číslo je MAC-id posielajúceho uzlu a tretie naopak uzlu

prijímajúceho. Posledné číslo je tzv. MAC typ.

10. --- (nevyužité pole).

11. Ďalšie pole je takisto uzavreté v hranatých zátvorkách. Čísla v nich odpovedajú IP adresám zdrojového i cieľového uzlu a hodnote TTL (Time To Live), čo je čas platnosti paketu.

12. Posledné pole je tiež zložené z hranatých zátvoriek a obsahuje informácie týkajúce sa TCP. Obsahuje poradové číslo a číslo potvrdzovacieho paketu (ACK).

4.4 Použitie NS2, jeho výhody a nevýhody

Tento simulátor je určený na simulovanie počítačových sietí a ich správania sa v prevádzke. Možnosti jeho využitia sú naozaj veľké. Dokáže simulovať ako siete káblové tak aj siete založené na bezdrôtovom prenose (satelitné i lokálne).

Ďalej sa zameriam hlavne na popis využitia simulátora pre lokálne bezdrôtové siete.

V NS2 je možné simulovať bezdrôtové siete iba so štruktúrou typu ad-hoc. Pre popis ad-hoc sietí viď 2.3.2. Pre túto štruktúru ponúka NS2 štyri typy smerovacích protokolov (DSDV, AODV, DSR, TORA). Podľa týchto protokolov získavajú mobilné uzly v simulácií informácie o smerovaní paketov v sieti. Ďalej máme možnosť podrobne zadefinovať MAC vrstvu týchto uzlov (MAC vrstva definuje napr. prenosovú rýchlosť zariadenia, rôzne vlastnosti zariadenia, podľa ktorých potom pracuje s prenášanými rámcami), typy antén, frônt paketov v uzloch, maximálne počty paketov vo frontách, veľkosť simulovanej topológie a mnoho ďalších vlastností.

Prenosové spojenia, ktoré NS2 podporuje sa definujú pomocou tzv. agentov, ktorí môžu byť rôzneho typu (TCP, UDP). Môžeme definovať veľkosti paketov, intervaly medzi posielaním paketov a ďalšie zaujímavé vlastnosti.

Ďalej uzlom môžeme zadať presné súradnice umiestnenia v topológii. Medzi deje, ktoré sa v ad-hoc sietiach často vyskytujú patria pohyby bezdrôtových klientov v sieti. Aj toto nám NS2 umožňuje definovať a presne určiť čas, miesto a rýchlosť, kde, kedy a akou rýchlosťou sa má daný uzol premiestniť.

Za veľkú výhodu NS2 považujem podrobnosť nastavenia simulovaných sietí a jeho univerzalitu použitia pre drôtové ako aj bezdrôtové siete. Pre simuláciu počítačovej siete, považujem NS2 za komplexný produkt. Ďalšou výhodou je voľná dostupnosť celého programu na internete, spolu s jeho zdrojovým kódom a s tým spojená možnosť doprogramovania či prispôsobenia tohto nástroja na vlastnosti, ktoré potrebujeme pre simuláciu.

Naopak medzi nevýhodu môžem zaradiť len to, že neexistuje alebo sa mi aspoň nepodarila vypátrať akákoľvek podpora pre NS2 v tlačenej podobe. Tento projekt je síce kvalitne zdokumentovaný, ale k jeho počiatočnému pochopeniu je potrebné preštudovanie kvalitného tutoriálu, ktorý som nikde nenašiel. Dopátral som sa len k menším tutoriálom [6], ktoré mi boli nápomocné pri začínaní s prácou v NS2. Ani jeden z nich ale nebol tak podrobný, aby som podľa neho vypracoval aspoň väčšiu časť práce. Najviac názorné mi boli diskusné fóra, kde sa preberali konkrétne problematiky tvorby simulácie. K ďalším nevýhodám možno zaradiť netriviálnu inštaláciu a nutnosť doplnenia ďalších programov pre grafické zobrazenie výsledkov, čo je ale malá daň za výhodu voľnej licencie a voľne šíriteľného zdrojového kódu.

4.5 Jazyk TCL

TCL (Tool Command Language) [3] je skriptovací jazyk, ktorého vznik sa datuje do roku 1988.

Od začiatku bol jazyk vyvíjaný tak, aby bol zároveň jednoduchý, rozšíriteľný, multiplatformový a jednoducho zaimplementovateľný do aplikácií, čo takisto rada aplikácií využíva .

TCL je interpretovaným jazykom. Preto nie je prekladaný do samostatne spustiteľného strojového kódu, ale potrebuje interpretér. Vďaka nemu je TCL na platforme nezávislý, čo z neho robí ešte mocnejší nástroj.

Distribúcia TCL prevažne pozostáva z knižnice *libtcl*, ktorá implementuje celý intrerpretér a môže byť ľahko použitá v ďalších aplikáciach. Obsahuje tiež jednoduchý interpretér *tclsh* a samozrejme sadu manuálových stránok.

Práve spomínaná možnosť zabudovania TCL do ďalších aplikácií z TCL je vyhľadávaná vlastnosť - pokiaľ potrebujeme mať vo svojej aplikácii napísaný napr. v C/C++ skriptovací jazyk a nechceme sa zaoberať tvorbou vlastného interpreta, je TCL ideálnou voľbou. Tento princíp sa využíva aj v NS2 simulátore. Rovnako tak pomocou TCL môžeme písať jednoduché utility a pomocou knižnice TK môžeme veľmi jednoducho vytvoriť aplikácie s grafickým užívateľským rozhraním. Možnosti využitia TCL sú teda pomerne veľké.

TCL je bežnou súčasťou linuxových distribúcií, alebo si ho môžeme zdarma získať na internete. Celé je naviac distribuované pod benevolentnou BSD licenciou - môžeme teda TCL ľubovoľne redistribuovať, modifikovať, atď. Na internete sú k dispozícii ako zdrojové súbory TCL, tak i binárne balíčky pre Linux, Mac OS a Windows.

Syntax jazyka je veľmi jednoduchá a názorná. Ďalej už preto nebude rozobraná. Časti kódu v tomto jazyku, ktoré sa budú neskôr v texte vyskytovať, budú dôkladne vysvetlené a mal by ich pochopiť i programový laik. Pre detailnejšie spoznanie jazyka TCL viď [3].

4.6 Aplikácie pre grafický výstup

Na zobrazenie výsledkov simulácie sú použité dve grafické X-window aplikácie. Tie sú implementované pre všetky platformy Unix s X-window. Obe sú voľne dostupné na internete.

4.6.1 NAM – Network Animator

NAM je nástroj na zobrazenie funkčnosti simulovanej počítačovej siete a putovania paketov po takejto sieti. Je založený na animáciách jazyka TCL/TK. Obsahuje vrstvu pre zobrazenie topológie siete, animáciu posielaných a prijímaných paketov a nástroje na kontrolu prenášaných dát.

Na výslednej animácií pomocou NAM-u teda bude priamo vidno navrhnutú topológiu. Na nej budú simulované spojenia, ktoré vyžadujú prenos paketov. Tieto pakety sa v animácií zobrazujú presne tak, ako budú posielané. Ďalej sa vo výslednej animácií zobrazia fronty paketov, ktoré môžu vznikať na jednotlivých uzloch siete. Vďaka tomu všetkému je možné lepšie pochopiť fungovanie simulovanej siete a nájsť prípadné komplikácie, ktoré sa v nej vyskytujú.

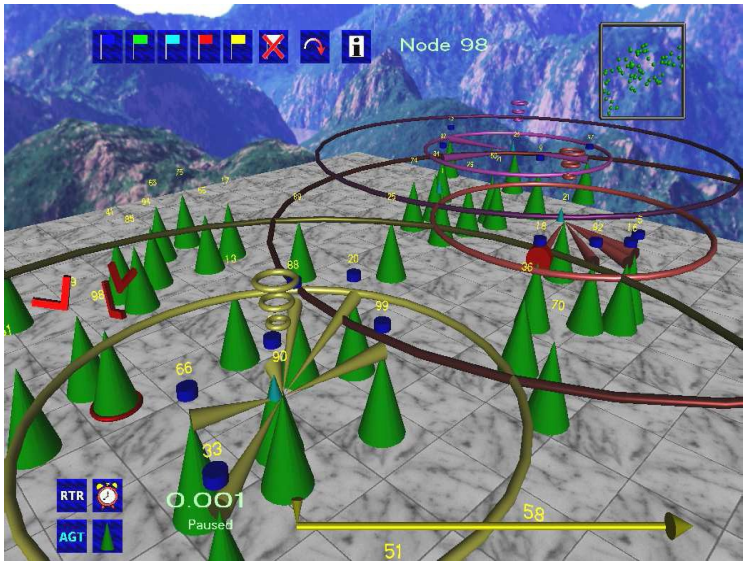
4.6.2 Xgraph

Táto aplikácia je použitá na vykreslenie výsledných grafov simulácie. NS2 ako výstup simulácie vytvára textové súbory, ktoré nám ale bez kvalitného prednesenia nič nepovedia. Takýto súbor treba prefiltrovať a vhodne zobraziť. V tejto simulácii budú grafy použité na zobrazenie pomeru prenášaných paketov a paketov, ktoré boli kvôli určitým príčinám zahodené.

4.6.3 Ďalšie nástroje pre grafické zobrazenie simulácie

Medzi ďalšie animátori výstupu nástroja NS2 patrí výsledok projektu s názvom Huginn. Je to 3D vizualizér, ktorý pracuje takisto ako NAM s výstupným súborom NS2 bez nutností jeho úpravy. Výsledky tohto vizualizéru je atraktívne 3D video, ktoré znázorňuje priebeh simulácie.

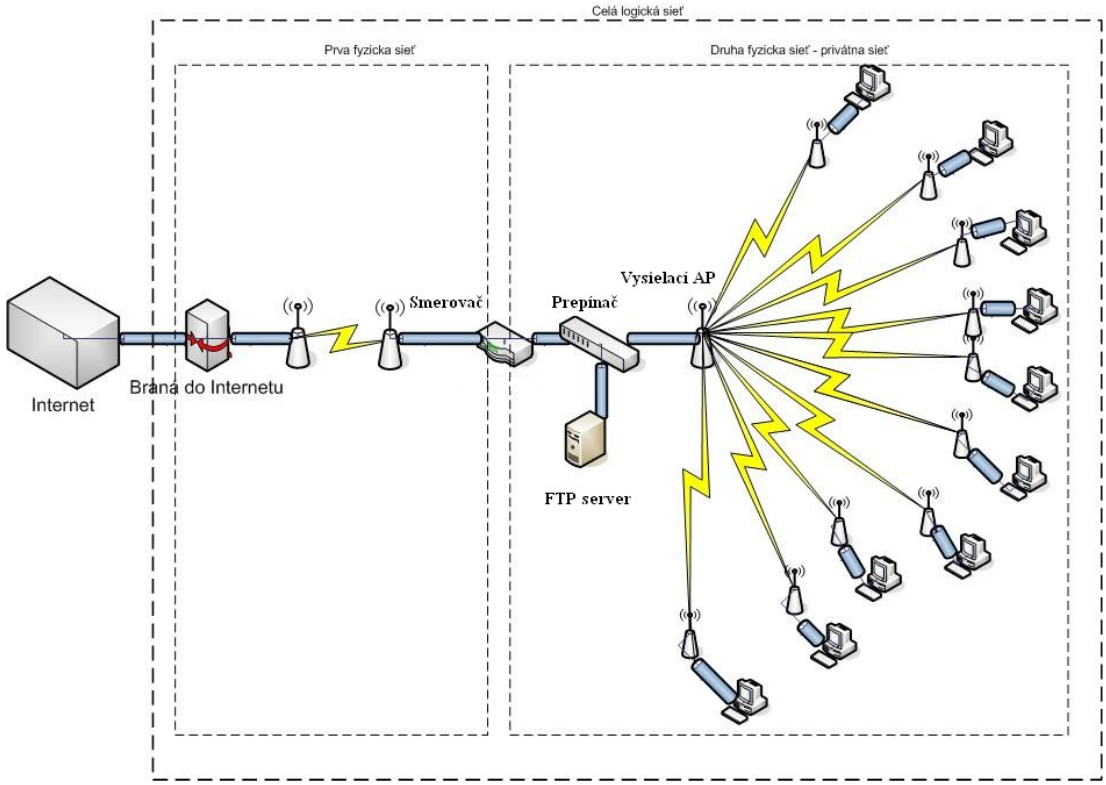
Tento produkt má voľne šíriteľný zdrojový kód. Internetová stránka o ňom je ale v štádiu rekonštrukcie a tak nie je možné získať viac informácii [8]. Na stránke je naznačený postup inštalácie a potrebné nástroje na jej realizáciu. Tento postup je netriviálny a vyžaduje vyššiu znalosť programovania a hlavne operačného systému UNIX [7]. Mne sa tento nástroj podarilo preložiť a spustiť. Pre správnu funkčnosť je ale potreba vizualizačné prostredie, ktoré podporuje vykresľovanie 3D priestoru. Zdrojové kódy prostredia sa mi ale kvôli dočasnému výpadku domovskej stránky tohto projektu nepodarilo získať. Preto sa mi vizualizácia simulácie v Huginne nepodarila spustiť. Na obrázku 4.6 je ukážka výstupu tohto vizualizéru. Tento obrázok je prevzatý z domovskej stánky projektu a znázorňuje smerovanie paketov prebiehajúce v bezdrôtovej sieti.



Obrázok 4.6: *Ukážka výstupu vizualizačného nástroja Huginn [8]*

5 Návrh a simulácia WiFi siete

Topológia simulovanej siete je vytvorená na základe skutočnej siete. Skladá sa z dvoch častí. Prvá z nich predstavuje bránu do internetu a druhá samotnú privátnu sieť. Obrázok 5.1 zobrazuje toto fyzické rozloženie.



Obrázok 5.1: *Topológia reálnej siete*

5.1 Rozbor reálnej siete

Ako už bolo spomenuté skôr, reálna sieť sa skladá z dvoch častí. Tie nepodliehajú spoločnej správe, a preto je ich nutné rozobrať jednotlivo.

Spojovacím bodom oboch častí je smerovač (router). Ten prepája obe časti pomocou NAT (network address translation). Tým dosiahneme ukrytie druhej časti siete pred vstupom z Internetu a vznik privátnej siete. Na pripojenie do internetu nám bude vďaka NAT stačiť jediná verejná IP adresa. Na smerovači sú spustené ešte ďalšie služby pre privátnu (napr. DHCP, NTP, QoS atď.). Nie sú však pre našu simuláciu podstatné a tak sa nimi nebudem ďalej zaoberať.

V nasledujúcich podkapitolách hlbšie rozoberiem obe časti celej siete.

5.1.1 Prvá fyzická sieť

Spravovanie tejto časti má za úlohu poskytovateľ pripojenia do internetu. Spočiatku bola táto linka tvorená štandardom IEEE 802.11b. Toto spojenie sa však časom stalo nedostatočné a to nie z dôvodu šírky pásma, ktorú podporuje, ale kvôli počtu klientov, ktorých pripája. V tomto prípade myslím klientov poskytovateľa internetu. Ten totiž touto linkou pomocou všesmerovej antény pripájal

ďalších svojich zákazníkov. Pre tieto dôvody bola linka skvalitnená. To znamená, že zo štandardu 802.11b sa prešlo na 802.11a (IEEE 802.11a, čiže frekvencia prenosového pásma je 5 GHz so šírkou prenosu 54 Mbps). Tento spoj už nie je zdielaný inými klientmi, ale je časťou chrbticovej siete poskytovateľa. Touto inováciou linky sa úplne stratili nepriaznivé dopady predchádzajúceho pripojenia.

5.1.2 Druhá fyzická sieť

Druhá sieť je už spravovaná užívateľom pripojenia a preto ju je možné modifikovať podľa potreby. Je postavená na štandarde IEEE 802.11b. Skladá sa z vysielacieho AP pripojeného na smerovač a z desiatich klientov.

Signál vysielacieho AP je zvyšovaný pomocou všesmerovej antény. Tá je umiestnená na najvyššej budove v oblasti, ktorú daná sieť pokrýva. Výkon prijímacích bodov jednotlivých klientov zvyšujú smerové antény. Tie sú vždy umiestnené tak aby mali na anténu vysielacieho AP priamy výhľad.

Všetci klienti používajú pre akúkoľvek komunikáciu vysielacie AP. Nech prebieha výmena informácií navzájom medzi klientmi alebo nech ide o ich prístup do internetu, vždy celý prenos dát sprostredkuje vysielacie AP. Toto je jednoznačný znak siete s infraštruktúrou. Všetky bezdrôtové stanice, či už sa jedná o vysielacie AP alebo prijímacie jednotky klientov siete pracujú v režime mostov. Práve spoje medzi týmito mostami tvoria úzke miesto celej siete.

Z obrázku 5.1 je jasné, že spoj medzi smerovačom a vysielacím AP je realizovaný pomocou sieťového prepínača. Do neho je napojený jeden server, na ktorom beží služba anonymného FTP (file transfer protocol). V budúcnosti sa plánuje zvýšiť počet služieb a serverov v tomto mieste. Pre našu simuláciu sú ale nepodstatné a preto, ako uvidíme neskôr, budú úplne vypustené.

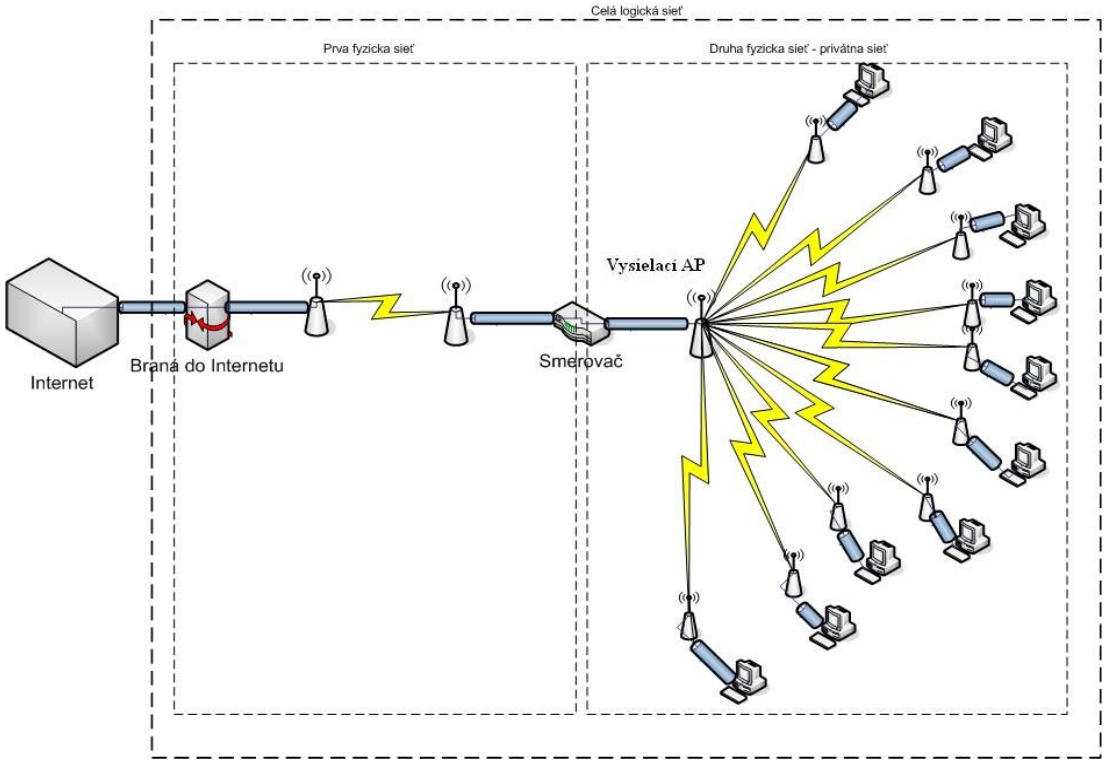
5.2 Abstraktný model siete a jeho analýza

Z reálnej topológie je pre účely simulácie potreba vytvoriť abstraktný model. Ten bude obsahovať všetky potrebné vlastnosti na správne odsimulovanie vytýčeného cieľa a práve naopak nebudú v ňom zahrnuté vlastnosti, ktoré nemajú vplyv na výsledok a je ich možné zanedbať. Výsledný abstraktný model je znázornený na obrázku 5.2.

Jediným podstatným faktom, ktorý som do abstrakcie zahrnul, bolo nahradenie bezdrôtového spoju medzi bránou do internetu a smerovačom. Jedná sa teda o prvú fyzickú sieť. Vzhľadom na fakt, že je tento spoj spravovaný poskytovateľom internetu, je práve on zodpovedný za kvalitu linky. Z tejto zodpovednosti vyplývajú povinnosti prispôsobovať šírku pásma potrebnú na pripojenie do internetu a podľa toho prípadne meniť typ spojenia. Preto je možné nahradiť tento spoj v simulácii drôtovým pripojením s dostatočnou šírkou prenosu. Tým si vytvoríme možnosť delegovať tento spoj, čo budeme využívať na simulovanie rýchlosti pripojenia do internetu. To znamená, že v simulačnom skripte budeme nastavovať kapacitu tejto linky podľa toho, aké pripojenie k internetu budeme chcieť simulovať.

V privátnej sieti som nahradil u každého klienta prepojenie bezdrôtového zariadenia a PC abstraktným klientom. V praxi to znamená to, že uvažujeme pripojenie medzi dvoma bezdrôtovými zariadeniami a už je na jednotlivých užívateľoch ako sú schopní prepojiť svoje PC so svojím bezdrôtovým zariadením. Väčšinou sa na to využíva drôtové pripojenie s veľmi krátkou vzdialenosťou, čo môžeme v porovnaní so štandardom IEEE 802.11b považovať za dokonalé spojenie a môžeme ho teda zanedbať.

Nastavením zariadení som sa snažil zosúladiť s nastaveniami v reálnej sieti. Tým som chcel výsledok simulácie čo najviac priblížiť k správaniu sa siete v skutočných podmienkach.



Obrázok 5.2: *Abstraktný model simulovanej siete*

5.3 Vytvorenie simulačného modelu

Ďalší krok, ktorý treba uskutočniť, je vytvorenie simulačného modelu. Ako simulátor bude použitý nástroj NS2 a preto musí mať tento model podobu skriptu napísaného v jazyku TCL. V nasledujúcich podkapitolách budú uvedené kľúčové časti simulačného skriptu s následným detailným vysvetlením.

5.3.1 Rozdelenie siete do domén

Ako prvé je nutné rozdelenie siete na dve časti. V NS2 sa to urobí pomocou definície domén, ktorým sa priradí počet clustrov a počet uzlov, ktoré každý cluster obsahuje. Pri našej topológii vytvoríme dve domény, ktoré odpovedajú jednotlivým fyzickým sietiam. Prvá fyzická sieť má dva clustre, ktoré značia vstupnú bránu do internetu a smerovač. Druhá sieť má len jeden cluster - smerovač, pod ktorý spadá aj vysielací AP. Smerovač je teda zahrnutý v obidvoch doménach siete.

```
AddrParams set domain_num_ 2                ;# pocet domen
lappend cluster_num 2 1                     ;# pocet clusterov v kazdej domene
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 1 2                   ;# pocet uzlov v kazdom clustri kazdej
                                               domeny
```

5.3.2 Vytvorenie jednotlivých uzlov siete

Bezdrôtoví klienti sú špeciálny prípad a preto sa nezahŕňajú do definície clusterov. Ich účasť v siete prebieha na základe registrácie (kde premenná *val(nn)* udáva počet bezdrôtových zariadení – v našom prípade sú to všetci klienti a zároveň aj vysielací AP).

```
create-god $val(nn)                          ;#registracia bezdrotovych klientov
```

Pridelenie tzv. „home agenta“ čo je vysielací AP.

```
set HAaddress [AddrParams addr2id [$node_(0) node-addr]]   #urcenie vys. AP
[$node_($i) set regagent_] set home_agent_ $HAaddress      #priradenie home agenta
```

Ďalej treba vytvoriť uzly reprezentujúce východziu bránu poskytovateľa internetu a hlavný smerovač vnútornej siete. Tie už musia byť priradené do správnej domény.

```
set WP [$ns node 0.0.0]                          ;#provider - pevny bod
set WR [$ns node 0.1.0]                          ;#hlavny router - pevny bod
```

Bezdrôtových klientov sme síce registrovali, ale ešte ich je treba vytvoriť. Tomu musí predchádzať veľmi dôležité nastavenie, ktoré spôsobí, že uzly reprezentujúce týchto klientov budú odpovedať štandardu IEEE 802.11b. Ide hlavne o definovanie fyzickej vrstvy a definovanie MAC hlavičky jednotlivých rámcov. To je vykonané postupnosťou príkazov:

```
Phy/WirelessPhy set freq_ 2.4e+9                 ;# Frekvencia pasma 2.4 GHz
Mac/802_11 set SlotTime_          0.000020       ;# 20us
Mac/802_11 set SIFS_              0.000010       ;# 10us
Mac/802_11 set PreambleLength_    144            ;# 144 bit
Mac/802_11 set PLCPHeaderLength_  48             ;# 48 bits
Mac/802_11 set PLCPDataRate_      1.0e6          ;# 1Mbps
Mac/802_11 set dataRate_          11.0e6         ;# 11Mbps
Mac/802_11 set basicRate_         1.0e6          ;# 1Mbps
Mac/802_11 set RTSThreshold_ 2347                ;# RTS treshold
Mac/802_11 set FragmentationThreshold_ 2346      ;# Fragmentation Treshold
```

Tým sme dostali správne nastavenia spomínaných vlastností modelu pre bezdrôtových klientoch, ktoré odpovedajú použitému hardwaru a jeho konfigurácii. V ďalšom kroku zadefinujeme premenné, ktoré budú vystupovať ako vlastnosti vytváraných uzlov. Týmto uzlom musí byť udelené, že sú bezdrôtoví klienti. Rovnako dôležité je zadanie použitej MAC hlavičky. Až tu využijeme parametre zadefinované skôr. Ďalej sú tu definované parametre ako typ pozičnej fronty, ktorý nám určuje kde bude fronta vykreslená vo výslednej simulácii, maximálny počet paketov vo fronte, smerovací protokol, ktorý ale priamo nevyužijeme, pretože simulujeme sieť s infraštruktúrou a typ linkovej vrstvy. Takisto sú v tejto časti skriptu zadefinované aj iné premenné, ktoré nám určujú počet bezdrôtových klientov, veľkosť topológie a čas konca simulácie.

Takéto priradenie vlastností do premenných nám veľmi uľahčí prácu v budúcnosti. Keď budeme chcieť zmeniť určité parametre, stačí zmeniť hodnotu v konkrétnej premennej a zmena, ktorú požadujeme, sa prejaví v celom skripte. Zároveň sa celý skript stáva čitateľnejším a ľahšie pochopiteľným.

```
# Definicie parametrov
set val(chan)           Channel/WirelessChannel    ;# typ kanalu
```

```
set val(prop)           Propagation/TwoRayGround   ;# radio-propagacny model
set val(netif)          Phy/WirelessPhy            ;# sietove rozhranie
set val(mac)            Mac/802_11                 ;# MAC typ
set val(ifq)            Queue/DropTail/PriQueue    ;# Typ pozitej fonty
set val(ll)             LL                         ;# typ linkovej vrstvy
set val(ant)            Antenna/OmniAntenna        ;# model anteny
set val(ifqlen)         50                         ;# max pocet packetov vo fronte
set val(nn)             11                         ;# pocet bezdrotovych klietnov
set val(rp)             DSDV                       ;# rutovaci protokol
set val(x)              300                         ;# X suradnica topologie
set val(y)              300                         ;# Y suradnica topologie
set val(stop)           250                         ;# cas konca simulacie
```

Tesne pred tým ako ideme vytvárať bezdrôtových klientov, musíme NS2 zadať vlastnosti vytváraného uzlu. To urobíme pomocou premenných zadefinovaných vyššie v práci. Je ale pár výnimiek, ktoré majú iba pravdivostné hodnoty v podobe ON alebo OFF. Tie zadefinujeme až tu.

Všetky tieto vlastnosti sa priamo týkajú len jedného bezdrôtového uzlu a tým je vysielací AP. Ten sa líši do ostatných bezdrôtových uzlov iba v schopnosti smerovať pakety cez drôtové spoje. Táto vlastnosť vyplýva z toho, že len tento bod je pripojený drôtovými linkami.

```
#  Konfiguracia pre vytvaranie mobilnych uzlov
        $ns node-config  -mobileIP ON \
                        -adhocRouting $val(rp) \
                        -llType $val(ll) \
                        -macType $val(mac) \
                        -ifqType $val(ifq) \
                        -ifqLen $val(ifqlen) \
                        -antType $val(ant) \
                        -propType $val(prop) \
                        -phyType $val(netif) \
                        -channelType $val(chan) \
                        -topoInstance $topo \
                        -wiredRouting ON \
                        -agentTrace ON \
                        -routerTrace ON \
                        -macTrace OFF \
                        -movementTrace ON
```

Až teraz je možné správne vytvorenie vysielacieho AP. K tomu je ešte nutné zadať správne umiestnenie v doméne. Pre úplnosť vypneme vlastnosť náhodného pohybu, čo sa využíva pri sietiach typu Ad-Hoc. Pre viac zrozumiteľný grafický výstup je zadaná farba uzlu a popis, ktorý bude pri ňom zobrazený.

```
#vytvorenie prijimacieho bodu od mobilnych uzlov
set node_(0) [$ns node 1.0.0]
$node_(0) random-motion 0
$node_(0) color forestgreen
$node_(0) label Vysielace_AP
```

V tomto momente už môžeme doladiť nastavenia uzlov, ktoré sú potrebné pre klientov. Toho sa priamo týka vypnutie smerovania medzi pevnými uzlami. Aby sme mohli bezdrôtových klientov vytvoriť jedným cyklom, musíme definovať pomocnú premennú typu pole. Tá nám poslúži na uloženie hodnôt pre zaradenie do domény, ktoré budeme v cykle postupne uzlom priraďovať. Taktiež tu určíme funkciu vysielacieho AP. Takto nastavený bezdrôtový uzol môžeme v nasledujúcom cykle priraďovať každému klientovi ako agenta, vďaka ktorému bude komunikovať s ostatnými komponentami v sieti. Pre našu simuláciu je to nevyhnutný krok, od ktorého sa odvíja

správne fungovanie celej simulácie.

```
#vypntutie smerovania
$ns node-config -wiredRouting OFF

#pomocna premenna sluziaca na definovanie domen pre jednotlive mobilne uzly
set temp {1.0.1 1.0.2 1.0.3 1.0.4 1.0.5 1.0.6 1.0.7 1.0.8 1.0.9 1.0.10}

#urcenie primacieho AP
set HAaddress [AddrParams addr2id [$node_(0) node-addr]]
```

Teraz, keď máme všetky nastavenia uzlov v správnom stave, môžeme jedným cyklom vytvoriť všetkých bezdrôtových klientov. Ako prvé v cykle vytvoríme uzol. Potom vypneme možnosť náhodného pohybu a určíme agenta pre daný uzol. V našom prípade je pre všetkých klientov spoločný agent, ktorým je vysielací AP. Ďalej v cykle priradíme každému klientovi malý popis. Vďaka tomuto cyklu a použitej premennej, ktorá predstavuje počet klientov, bude v budúcnosti veľmi jednoduché zvýšiť počet klientov a tak aktualizovať simulačný skript.

```
#Vytvorenie volnych bezdrotovych klientov
for {set i 1} {$i < $val(nn)} { incr i } {

        #vytvaranie jednotlivych uzlov a priradovanie im primacie AP
        set node_($i) [$ns node [lindex $temp [expr $i - 1]]]
        $node_($i) random-motion 0
        [$node_($i) set regagent_] set home_agent_ $HAaddress

        #nastavenie navestia
        set AP_label "AP_"
        append AP_label $i

        #priradenie navestia
        $node_($i) label $AP_label
}
```

5.3.3 Vytvorenie liniek medzi uzlami

Po zadefinovaní a vytvorení všetkých uzlov nám ku kompletnej definícií celej topológie treba ešte vytvoriť prepojenia medzi uzlami. Tieto prepojenia, tzv. Linky, je treba vytvoriť medzi východzou bránou poskytovateľa internetu a smerovačom, a takisto medzi smerovačom a vysielacím AP.

Kľúčovú úlohu bude v simulácii hrať práve linka medzi poskytovateľom internetu a smerovačom. Vďaka určovaniu šírky pásma budeme simulovať rýchlosť pripojenia do internetu. Pre výraznejší grafický výstup použijeme rôzne farby. Ďalej určíme orientáciu linky. Užitočný nám bude samozrejme popis a nastavenie pozície fronty.

V nasledujúcom úseku kódu je simulovaná rýchlosť pripojenia do internetu 700 Kbps.

```
#pevne spojenie medzi poskytovatelom internetu a hlavnym smerovacom
$ns duplex-link $WP $WR 700Kb 15ms DropTail

#nastavenie vlastnosti pre tuto linku
$ns duplex-link-op $WP $WR color chocolate
$ns duplex-link-op $WP $WR orient down
$ns duplex-link-op $WP $WR label Wired_512Kb
$ns duplex-link-op $WP $WR label-color red
```

```
$ns duplex-link-op $WP $WR queuePos 0.5

#pevne spojenie medzi hlavnym smerovacom a vysielacim AP
$ns duplex-link $WR $node_(0) 100Mb 3ms DropTail
#nastavenie vlastnosti pre tuto linku
$ns duplex-link-op $WR $node_(0) color  blue
$ns duplex-link-op $WR $node_(0) orient down
$ns duplex-link-op $WR $node_(0) label Wired_100Mb
$ns duplex-link-op $WR $node_(0) label-color red
$ns duplex-link-op $WR $node_(0) queuePos 0.5
```

5.3.4 Vytvorenie simulovaných spojení

V tomto čase už máme vytvorený presný model siete a je čas robiť s ním experimenty.

V našom prípade ide o simulovanie určitých spojení vnútornej siete s internetom. Zvolil som prípad plného zaťaženia. Je to prípad, kde všetci bezdrôtoví užívatelia v tom istom čase sťahujú dáta z internetu. Sťahovaním dát sa myslí nadviazanie spojenia protokolom TCP/IP a následný prenos dát smerom z Internetu na bezdrôtovú stanicu. Takéto prenosy sú veľmi časté napr. pomocou FTP (File Transfer Protocol), HTTP (Hypertext Tranfer Protokol), IMAP (Internet Message Access Protocol) a mnohých ďalších protokolov. Budeme sledovať chovanie vnútornej siete a predovšetkým linky, ktorá predstavuje pripojenie do internetu.

Nižšie v texte je uvedené jedno takéto spojenie. Všetky ostatné sa líšia len v indexe „x“ premennej `$node_(x)`. Nastavenie triedy je kvôli priradeniu farby na prenášané pakety. Ďalej je nastavená maximálna veľkosť paketu. Spojenia sú zadefinované pomocou tzv. agentov. Je tu nastavený čas, za aký bude poslaný potvrdzujúci paket ACK, vzťahujúci sa na prenesený paket. Agentov treba vzájomne spojiť a nastaviť na tomto spojení druh prenosu. Je asi zrejmé, že som zvolil prenos pomocou FTP, ktorý slúži na prenášanie dát. Je pre našu simuláciu plne vyhovujúci. Nakoniec je nastavený čas nadviazania spojenia na 0.0, teda akonáhle sa simulácia spustí.

```
set tcp1WP [new Agent/TCP]
$tcp1WP set class_ 2
$tcp1WP set window_ 2000
$tcp1WP set packetSize_ 984
Agent/TCPSink/DelAck set interval_ 100ms
set sink1WP [new Agent/TCPSink/DelAck]
$ns attach-agent $WP $tcp1WP
$ns attach-agent $node_(1) $sink1WP
$ns connect $tcp1WP $sink1WP
set ftp1WP [new Application/FTP]
$ftp1WP attach-agent $tcp1WP
$ns at 0.0 "$ftp1WP start"
```

5.3.5 Ukončenie simulačného skriptu

Na záver treba v skripte ešte zadať reštartovacie príkazy pre každý uzol. Tie ukončia simulované spojenia a zastavia celú simuláciu.

```
#restartovanie jednotlivych uzlov
$ns at $val(stop).0 "$WP reset";
$ns at $val(stop).0 "$WR reset";
$ns at $val(stop).0 "$node_(0) reset";
$ns at $val(stop).0 "$node_(1) reset";
```

```
...
$ns at $val(stop).0 "$node_(10) reset";
#koniec simulacie
$ns at $val(stop).01 "stop"
$ns at $val(stop).01 "puts \"end simulation\" ; $ns halt"
```

Týmto spôsobom korektne ukončíme beh simulácie.

5.4 Spustenie simulácie

Pre uľahčenie spúšťania grafického zobrazenia simulácie je na konci skriptu zahrnutých pár príkazov. Tieto nám z výsledného súboru simulácie vyberú len potrebné hodnoty a vytvoria tak súbory vhodné na vykreslenie grafov pomocou aplikácie XGRAPH. Ďalej je spustená aplikácia NAM, ktorá je vyvinutá priamo na podporu NS2, a preto je výsledný súbor predaný tomuto nástroju priamo, bez akýchkoľvek korektúr.

Všetky tieto príkazy zapríčinia, že pri správnom nastavení prostredia operačného systému UNIX, je pre spustenie celej simulácie aj s grafickým výstupom potrebný jeden príkaz a to `ns nazov_skriptu`, kde nazov_skriptu je názov súboru simulačného skriptu.

Beh simulácie trvá pár minút. Po jeho skončení nám na obrazovku samé povyskakujú grafické zobrazenia výsledkov.

5.5 Výsledok simulácie

Výsledok simulácie je jeden súbor so špecifickou syntaxou popísaný v kapitole 4.2. Výsek z nášho výsledného súboru je vložený v prílohe A.

Nás ale viac zaujímajú výstupy spracované grafickými aplikáciami. Ako je uvedené v 5.4, aplikácii NAM predáme výsledný súbor bez akýchkoľvek zásahov. Avšak pre vykreslenie grafov pomocou XGRAPH-u musíme vybrať len potrebné hodnoty. To sa nám podarí pomocou aplikácie AWK [7]. Preto nám vznikajú po dokončení behu simulácie ďalšie 4 súbory. Až tie sú použité pre vykreslenie kriviek v XGRAPH-e. Vysvetlenie významu týchto grafických výsledkov je predmetom nasledujúcej kapitoly.

6 Analýza výsledkov simulácie

Ako výsledok simulácie budeme považovať grafické výstupy. Bolo by nelogické a nemožné rozoberať výstupný súbor NS2. Pre dosiahnutie nášho cieľa v kapitole 3.4, musíme odsimulovať viacero variánt siete. Jednotlivé varianty sa navzájom líšia len v zadefinovanej šírke pásma linky, ktorá predstavuje pripojenie do internetu.

Začneme simuláciou, kde je šírka pásma stanovená na 700 Kbps. Túto hodnotu budeme postupne zvyšovať na 1, 2, 3, 4, 5 a 10 Mbps. Pre jednotlivé simulácie sú vytvorené analýzy, ktorých účelom je rozobrať a zhodnotiť spávanie siete s príslušným pripojením do internetu.

Výsledok simulácie tvoria 2 grafy vytvorené programom XGRAPH a simulačná grafika, ktorá zobrazuje putovanie paketov po sieti.

Graf naľavo v každej dvojici na obrázkoch B.1, B.2, B.3, B.4, B.5, B.6 a B.7 znázorňuje súčet úspešne doručených paketov všetkých klientov (zelená krivka) a počet zahodených paketov vysielacím AP (červená krivka). Pomocou týchto kriviek sa dá jednoducho zistiť, či vnútorná sieť stíha prijímať pakety prichádzajúce z internetu k jednotlivým bezdrôtovým klientom. Stačí sa pozrieť na červenú krivku. Ak je táto krivka na nulovej hodnote, znamená to, že vysielacie AP a teda aj celá vnútorná sieť sú schopné spracovávať dáta prichádzajúce do siete rýchlosťou prenosu, ktorá je v konkrétnej simulácii použitá.

Na druhom z dvojice grafov je znázornený počet požadovaných paketov z internetu (zelená krivka) a počet zahodených paketov ešte pred vstupom do vnútornej siete (červená krivka). Tu zas na prvý pohľad vidno, či vnútorná sieť produkuje viac požiadaviek na prenos, ako je linka do internetu schopná zvládnuť. Rovnako ako v prvom grafe pozeráme na červenú krivku a vidíme či kvôli nízkej linke pripojenia do internetu dochádza k strate paketov. V tomto prípade ale ešte pred vstupom do vnútornej siete.

Zelené krivky v grafoch nám poslúžia na znázornenie pomeru koľko paketov bolo z internetu poslaných a koľko z nich bolo skutočne k jednotlivým bezdrôtovým klientom vnútornej siete doručených.

Grafika siete vyobrazená v animátore NAM na obrázkoch A.1, A.2, A.3, A.4, A.5, A.6 a A.7 vykresluje priebeh simulácie. Uzly v sieti, ktoré sú drôtového charakteru, sú znázornené ako štvorce. Naopak vysielací AP a bezdrôtoví klienti sú znázornení ako kruhy. Každý takýto uzol a aj linky medzi nimi sú opatrené popisom pre jednoduchšiu orientáciu. Jednotlivé preposielané pakety sú znázornené v drôtovej časti siete červenými, v bezdrôtovej časti čiernymi šípkami. Keďže je simulácia založená na dátovom prenose pomocou protokolu TCP štvrtej vrstvy ISO/OSI modelu, je na každý úspešne doručený paket poslaný potvrdzujúci paket ACK. Ten má tvar malej červenej alebo čiernej čiarky v závislosti na prostredí, v ktorom je prenášaný. Výsledkom nepostačujúcej linky do internetu alebo naopak nedostatočnou kapacitou prenosu vnútornej siete sú vznikajúce fronty pri určitých uzloch a zahadzovanie paketov. Tieto fronty majú tvar prerušovanej čiary vpravo od uzlu, pri ktorom vzniká. U takejto fronty môže pri preplnení zásobníka dochádzať k strate paketov. Táto situácia je v NAM naznačená padajúcimi štvorčekmi smerom nadol z konca fronty.

Ďalej sú rozobrané grafy a výstupy v NAM pre jednotlivé šírky pásma. Pri 700 Kbps, 1, 2, 3 Mbps majú výstupy podobný charakter a tak sú zhrnuté do jednej analýzy. Inak je to už pri simuláciách so šírkou pásma 4 Mbps a viac. Tieto sa síce odlišujú od predchádzajúcich simulácii, ale v porovnaní medzi sebou majú takisto podobné výsledky. Preto sú tiež zhrnuté do spoločnej analýzy.

6.1 Analýza výsledku - pripojenie od 700 Kbps do 3 Mbps

V čase vzniku tejto práce je aktuálna šírka pásma linky pripojenia do internetu 700 Kbps. Práve preto je simulácia s takouto kapacitou prenosu vykonaná ako prvá.

Rovnaké charakteristiky, ktoré vznikajú pripojením s linkou 700 Kbps, vykazujú aj pripojenia s linkami, ktoré sú kapacity 1, 2 alebo 3 Mbps. Pri porovnaní grafov na obrázku B.1, B.2, B.3 a B.4, zbadáme tieto podobnosti.

V ľavom grafe každej dvojice červená krivka splýva s nulovou hodnotou. Naopak v každom pravom grafe červená krivka vystupuje nad nulovú hodnotu. Z toho je jasné, že pri takýchto pripojeniach k internetu vnútorná sieť zvláda príjem dát, ktorý je v simulácii nastavený. Inak je to ale s linkou pripojenia na internet. Tá tvorí úzke miesto prenosu a jej kapacita je v týchto prípadoch nepostačujúca.

Zelené krivky nás v týchto prípadoch veľmi nezaujímajú. Vykazujú hodnoty, ktoré odpovedajú simulovanej rýchlosti pripojenia, ale vzhľadom na fakt, že vnútorná sieť zvláda spracovávať príjem dát, nie sú pre nás tieto krivky extra zaujímavé.

Na ukážkach z programu NAM (obrázok C.1, C.2, C.3 a C.4) vidno frontu paketov, ktorá vznikla na strane poskytovateľa internetu a stratu paketov z nej. Ostatné pohybujúce sa elementy ako dátové pakety a k nim odpovedajúce pakety ACK nám v simulácii nič významné nenaznačia.

6.2 Analýza výsledku - pripojenie od 4 do 10 Mbps

Pri pripojení vnútornej siete do internetu s linkou kapacity 4 Mbps je už situácia úplne iná. Všetky simulácie s takouto alebo vyššou kapacitou majú podobné charakteristiky, a preto sú rozobrané spolu v tejto podkapitole.

Od predchádzajúcich prípadov v 6.1 sa výsledok rapídne líši. V každom ľavom grafe z dvojice na obrázkoch B.5, B.6 a B.7 už vystupuje červená krivka nad nulovú hodnotu. To nám naznačuje vznikajúce problémy na vysielacom AP. Tento uzol teda začal zahadzovať pakety.

Naopak v grafe na obrázkoch B.5, B.6, B.7 vpravo v každej dvojici nám červená krivka splynula s nulovou hodnotou. Pri simulácii s linkou 4 Mbps ešte síce zbadáme na obrázku B.5 nejakú stratu paketov na strane poskytovateľa internetu, je ale jasné, že ide o hranicu, kde sa láme celý výsledok simulácie.

Obe tieto zmeny sa navzájom potvrdzujú a naznačujú, že nastal prelom. Zo stavu nedostatočnej šírky pásma linky predstavujúcej pripojenie do internetu, sieť prešla do stavu, pri ktorom naopak nestíha vnútorná sieť.

Teraz sú pre nás dôležité i zelené krivky v grafoch. Všímame si hlavne tých v ľavých grafoch na obrázkoch B.5, B.6, B.7. V každom grafe vidíme dosiahnutie približne rovnakej hodnoty 3800. Táto hodnota predstavuje počet paketov za sekundu, kde každý paket má veľkosť presne 1024 bitov. Celá krivka je teda rýchlosť prenosu, ktorou boli pakety prenášané v jednotkách Kbps. Zelenej krivky na pravej strane si všímať nemusíme, pretože časť týchto paketov bude neskôr zahodená vysielacím AP.

Nakoniec ešte rozoberieme výslednú grafiku v NAMe. Fronta, ktorá bola pri simuláciách s nižšou rýchlosťou pri uzle, ktorý predstavuje bránu do internetu, na obrázkoch C.5, C.6 a C.7 sa zmenšila alebo úplne zmizla. NAM nepodporuje vykresľovanie fronty pri bezdrôtových uzloch a preto nevidíme frontu, ktorá teraz vznikla pri vysielacom AP. Môžeme ale zbadať, že v ňom dochádza k strate paketov. Tento jav má podobu čiernych štvorčekov padajúcich smerom nadol.

Čím je rýchlosť prenosu vyššia, tým je samozrejme vyšší aj počet prenášaných paketov. To je jasné zo zvýšenej hustoty šipiek v animácii. Pre nás ale nie je tento fakt extra podstatný.

6.3 Zhrnutie všetkých analýz

Pomocou výsledných grafov a simulačných grafík sme zistili, že pripojenia bezdrôtovej siete do internetu linkou so šírkou pásma 700 Kbps až 3 Mbps je nedostatočné. Pri týchto simuláciách dochádzalo k vytváraniu fronty a následne k strate paketov z nej ešte pred vstupom do vnútornej siete.

Iný prípad nastal pri linke pripojenia do internetu s kapacitou 4 Mbps a viac. V týchto simuláciách už k strate paketov na strane poskytovateľa internetu nedochádza. Linka zvláda posielanie paketov na požiadavky z vnútornej siete. K strate paketov ale dochádza vo vnútornej sieti. Vysielacie AP, od ktorého závisí celá vnútorná sieť, zahadzuje časť paketov smerujúcich z internetu k jednotlivým bezdrôtovým klientom.

Podľa kapitoly 4.1 môžeme zahadzovanie paketov pri vstupe do vnútornej siete zamedziť nastavením nadmernej dĺžky fronty. Presne tak by sa správala reálna sieť pri prenose protokolom TCP. Ja som ale dĺžku fronty pri linkách úmyselne nezvyšoval. Takto jednoznačne vidieť, kedy kapacita linky pripojenia do internetu postačuje pre plné využitie celej siete.

Výsledná kapacita linky pripojenia do internetu, ktorá nám v simuláciách vyšla ako maximálne využiteľná, je cca 3800 Kbps. Treba ale pripomenúť, že táto rýchlosť prenosu je počítaná iba z paketov TCP, ktoré boli úspešne doručené jednotlivým bezdrôtovým klientom. Nie sú do nej započítané iné pakety ako napr. potvrdzujúce pakety ACK. Preto treba k hodnote 3800 Kbps ešte pripočítať spotrebu pásma týmito paketmi. Podľa mojich odhadov by pripojenie k internetu linkou s prenosovou kapacitou 4000 až 4200 Kbps plne postačovalo pre maximálne využitie s ohľadom vnútornú sieť. Toto pripojenie by malo poskytovať najlepší pomer: cena za pripojenie versus využiteľnosť siete.

Celkové zhrnutie výsledku simulácie a zhodnotenie celej bakalárskej práce je uvedené nižšie v poslednej kapitole.

7 Záver

V práci som si za cieľ vytýčil zistiť maximálnu kapacitu linky pripojenia do internetu pre privátnu bezdrôtovú sieť. Dodržal som postup tvorenia obecnej simulácie. Z reálnej siete som vytvoril abstraktný model. Z neho ďalej simulačný model, ktorý mal podobu skriptu napísaného v jazyku TCL. Spúšťaním simulácií v nástroji NS2 som získal výsledky, ktoré som následne analyzoval.

7.1 Zhrnutie dosiahnutých výsledkov

Simulácia skúma správanie siete pri pripojení 10 bezdrôtových klientov v lokálnej sieti. Každý z nich prenáša dáta pomocou protokolu TCP z verejnej dátovej siete Internet. Pre dosiahnutie cieľu je urobených niekoľko simulácii, ktoré sa medzi sebou líšia len kapacitou linky, ktorá predstavuje pripojenie do internetu.

Prvá simulácia s kapacitou linky 700 Kbps, čo je v čase vzniku práce aktuálne pripojenie v reálnej sieti, rovnako ako simulácie s kapacitou 1 až 3 Mbps, vykazuje nedostatočné pripojenie. Až pripojenie linkou so šírkou pásma 4 Mbps a viac spôsobí iné správanie siete. V týchto simuláciách už nezvláda pripojenie vnútorná sieť a to hlavne vysielací AP, ktorý tvorí jej kľúčový element.

Po zvážení výsledkov simulácií a všetkých vedľajších aspektov som dospel k záveru, že sieť najlepšie využije pre pripojenie do internetu linku s kapacitou prenosu 4200 Kbps. Táto rýchlosť sa blíži k reálne dosiahnuteľným hodnotám. Kvôli tomu, že nie je v mojej kompetencii meniť pripojenie siete k internetu podľa potreby, nemôžem overiť výsledok simulácie. Teoreticky by dosiahnuteľná rýchlosť prenosu v sieti vytvorenej podľa štandardu IEEE 802.11b mala byť až 11 Mbps. Túto hodnotu ale znižuje metóda CSMA|CA, použitá pri prenose jednotlivých paketov na cca 5,9 Mbps. Aj toto je však v praxi nedosiahnuteľná rýchlosť. Na bezdrôtové siete vplýva mnoho vedľajších aspektov, a to tým viac, ak sa jedná o ich vonkajšie nasadenie. Preto považujem dosiahnutý výsledok simulácie 4200 Kbps v dostatočnej miere za reálny.

Na obrázkoch D.1, D.2, D.3 a D.4 v prílohe D sú grafy zo smerovača, ktorý je použitý v reálnej sieti. Dá sa z nich vypozorovať, že aktuálne pripojenie linkou kapacity 700 Kbps, ktoré je v tom čase v sieti zavedené, je naozaj v určitých okamihoch naplno využívané. Je otázkou času, kedy sa bude linka skvalitňovať a či výkon, ktorý bude sieť dosahovať splní očakávania dosiahnuté v simuláciách.

Dosiahnutými výsledkami som dospel k záveru, že pri plnom vyťažení vnútornej bezdrôtovej siete bude pre jej pripojenie do verejnej sieti Internet najviac výhodná linka s kapacitou 4200 Kbps.

7.2 Prínos a nové poznatky z vypracovania bakalárskej práce

Túto prácu som vytváral postupne. Získavanie znalostí o NS2 bolo naozaj náročné. Projekt je síce kvalitne zdokumentovaný, ale chýba k nemu obsiahlejší tutoriál. Všetky texty boli v anglickom jazyku, čo bola ale najmenšia prekážka. Získanie orientácie v NS2 v tomto prípade bolo ešte náročnejšie, pretože sa jednalo o simuláciu bezdrôtovej siete. Tá má oproti klasickej drôtovej sieti mnoho vlastností navyše.

Samozrejmosťou pre vytvorenie skriptu simulácie bola dobre naštudovaná problematika bezdrôtových sietí. K tomu sa ďalej viazalo dobré chápanie ISO/OSI modelu a znalosť všetkých jeho vrstiev so zameraním na prvé 4.

Celý proces vytvárania simulácie sa riadil obecným postupom z kapitoly 3. Vďaka tomu simulácia obsiahla z reálnej siete iba podstatné časti a ostatné nepotrebné časti boli zanedbané.

Pri vypracovaní tejto bakalárskej práce som dôkladnejšie pochopil princípy bezdrôtových sietí a zároveň i ISO/OSI modelu. Otestoval som obecný postup simulácie na diskrétnej simulácii počítačovej siete. Utvrdil som si tým nutnosť vytvorenia abstraktného a z neho vychádzajúceho simulačného modelu, s ktorým som vykonával experimenty. Na základe analýz a vyhodnotení som dospel ku konkrétnym záverom. V práci som spoznal simulátor NS2 a hlavne možnosti, ktoré ponúka v oblasti simulácii bezdrôtových sietí. Tomu však muselo predchádzať naštudovanie syntaxe skriptovacieho jazyka TCL. Pre grafické výstupy simulácie boli použité aplikácie NAM a XGRAPH, ktorých vstupné dáta tvorili upravené súbory výstupu simulátoru NS2. Tieto úpravy boli vykonávané prevažne utilitou AWK. Všetky tieto programy sú podporované iba operačnými systémami platformy UNIX, s ktorými som pri inštalácii programov a tvorení či ladení výsledného skriptu získal skúsenosti.

7.3 Ďalšie možnosti rozšírenia práce

Ďalšie rozšírenie tejto práce pre budúcnosť by som hľadal najmä vo vizualizačných nástrojoch výsledku NS2. Nástroj, ktorý by vykresloval grafy podľa definovaných kritérií s príjemným užívateľským prostredím, by podľa mňa dosiahol veľký úspech. Predišlo by sa tak použitiu rôznych iných aplikácii pre vydolovanie dát z výstupného súboru NS2, čo je potrebné pre zobrazenie grafu v XGRAPH-e. Najmä táto nutnosť robí z grafových výstupov zložitú vec. Podobné vylepšenie platí aj pre aplikáciu na výslednú animáciu simulácie. NAM síce plní svoj funkčný účel, ale pre dnešný svet, v ktorom hrá veľkú úlohu design a estetičnosť nemá šancu na úspech. Projekt Huginn rozobraná v kapitole 4.5.3 je podľa môjho názoru dobrý odrazový mostík takýchto vylepšení. Chýbajú k nemu však akékoľvek verejné informácie a tak je nutné kontaktovať priamo jeho autorov. Nemuselo by sa však jednať o 3D animátor, ako je to v prípade projektu Huginn. Takýto program nie je triviálne naprogramovať a bolo by pre to potreba tím ľudí. Vypracovanie takéhoto projektu jedným autorom je prakticky nemožné. Preto by som sa zameral skôr na prepracovanejší animátor s 2D rozhraním. Ani jeho implementácia by nebola triviálna, ale s porovnaním 3D zobrazenia by bola určite jednoduchšia.

Ďalším pokračovaním, ktoré by nadväzovalo na túto bakalársku prácu, by mohlo byť vytvorenie kvalitného tutoriálu pre NS2. Ako som spomenul v kapitole 4.3, takýto dokument mi pri vytváraní tejto práce veľmi chýbal. Pre NS2 je síce vytvorená kvalitná dokumentácia. Tá je ale bez názorných príkladov a vysvetlenia ich použitia takmer nepoužiteľná. Zameral by som sa na skompletizovanie všetkých doteraz nájdených menších tutoriálov. Z nich načerpané vedomosti a názorné príklady by som doplnil preberanými problémami, ktoré sú vyriešené na internetových fórach o NS2. Takto získané informácie by bolo treba vhodne usporiadať a doplniť funkčnými zdrojovými kódmi, ktoré by jasne znázorňovali vysvetľovanú problematiku. Takisto by sa nemohlo zabudnúť na základnú syntax jazyka TCL. Celý skript NS2 je napísaný v tomto jazyku a aj napriek tomu postačuje poznať iba jeho syntax. Takto spracovaný dokument spolu s dokumentáciou, ktorá je už teraz voľne dostupná na internete, by bol podľa mňa neoceniteľným sprievodcom pre mnoho začínajúcich programátorov simulácií v NS2.

Medzi ďalšie rozšírenie by sa dalo zaradiť doprogramovanie určitých knižníc pre NS2 v programovacom jazyku C++. NS2 nám túto možnosť ponúka vďaka jeho voľne šíriteľným zdrojovým kódom. Existuje však viacero takto doprogramovaných modulov a preto treba dobre

preskúmať zdroje na internete a neriešiť už hotovú vec nanovo. Toto rozšírenie je vhodné iba pre konkrétne simulácie, napr. pre štandardy, ktoré nie sú bežne nasadzované (len veľmi zriedka v špeciálnych prípadoch) a preto nie sú implicitne implementované v NS2.

NS2 obsahuje zaujímavý objekt s názvom Emulation. Ten sprístupňuje nasadiť NS2 priamo do reálnej siete. To môže mať dve podoby závislé od toho, či sa simulátor tvári ako koncová stanica typu smerovač alebo ako iná koncová stanica. V prvom prípade pakety reálnej siete putujú cez simulátor (transparentne ku koncovým bodom siete) a ovplyvňujú tak objekty vo vnútri simulátora alebo ostatnú premávku v sieti. V druhom prípade sa simulátor môže tváriť ako zdroj alebo príjemca paketov a bude tak priamo komunikovať so zariadeniami reálnej siete. Prvá možnosť je v súčasnosti viac používaná a teda aj viac vyvíjaná. V sieti, ktorá bola simulovaná v tejto bakalárskej práci, je dostupný anonymný FTP server pracujúci na voľne dostupnom operačnom systéme FreeBSD [16]. Práve pod týmto operačným systémom bol modul Emulation vyvíjaný a testovaný. Pre žiadne iné operačné systémy testovaný nebol. Prakticky by bolo možné nahradiť hlavný smerovač týmto serverom. Potom by už nič nebránilo nasadiť NS2 na tento server a sledovať pomocou neho správanie siete, zaznamenávať prevádzku a nakoniec porovnať výsledky s hodnotami, ktoré sme dosiahli čistou simuláciou. Takéto preklopenie funkcie smerovača však nie je v súčastnosti možné. Server, na ktorý by sme funkciu smerovača chceli preniesť, je veľmi starý a nestabilný. Bola by ohrozená funkčnosť celej siete, čo si nemôžeme dovoliť. Je ale možné na FTP serveri využiť nasadenie druhej možnosti, ktorú objekt Emulation ponúka. Simulátor sa bude tváriť ako koncová stanica, ktorá môže vytvárať a prijímať spojenia s ostatnými reálnymi zariadeniami v sieti. Pre toto nasadenie ale treba vymyslieť vhodné uplatnenie aby mal celý proces zmysel a vznikli tak užitočné informácie. V opačnom prípade by došlo k zbytočnému zaťaženiu FTP servera a hlavne k zbytočnému využitiu šírky pásma celej siete. Pre ďalšie rozšírenie práce je možnosť nasadenia NS2 do reálnej siete veľmi zaujímavá. Vyžadovala by ale stabilný server s operačným systémom FreeBSD. Tento by musel byť správne nakonfigurovaný. Jednak by musel vykonávať funkciu smerovača, ale takisto aj mnoho ďalších funkcií, ktoré sú spustené a poskytované aktuálnym smerovačom v sieti.

Literatúra

[1] Petr Peringer, Dc. Ing.: *Modelování a simulace IMS*, študijná opora predmetu IMS, Brno, VUT v Brne, 2006

[2] Patrick Zandl: *Bezdrátové sítě WiFi praktický průvodce*, Brno, Computer Press, 2005, ISBN 80-722-6632

[3] Zdzislaw Wrzeszcz: *TCL/TK podrobný průvodce programátora*, Brno, Computer Press, 2003, ISBN 80-251-0781-7

[4] Internetová encyklopédia Wikipedia, dostupná na URL: *http://en.wikipedia.org/* (máj 2007)

[5] Simulátor NS2, dostupný na URL: *http://www.isi.edu/nsnam/ns/* (máj 2007)

[6] Eitman Altman a Tania Jiménez: *NS simulator for beginners*, prednáškové poznámky, 2003, dokument dostupný na URL: *http://www-sop.inria.fr/maestro/personnel/Eitan.Altman/COURS-NS/n3.pdf* (máj 2007)

[7] Internetový server o linuxe, dokumenty dostupné na URL: *https://linuxzone.cz* (máj 2007)

[8] Huginn - Visualizing Network Simulation in 3D, dostupný na URL: *http://www.informatik.uni-mannheim.de/pi4.data/content/projects/huginn/* (máj 2007)

[9] Cnet Network Simulator, dostupný na URL: *http://www.csse.uwa.edu.au/cnet/* (máj 2007)

[10] Data-lik NP Simulation – java applet, dostupný na URL: *http://www.cs.bham.ac.uk/~gkt/Teaching/SEM335/dlsim/Simulation.html* (máj 2007)

[11] SFFNet, dostupný na URL: *http://www.sffnet.org* (máj 2007)

[12] Jasper, dostupný na URL: *http://www.cs.stir.ac.uk/~kit/software/comms/jasper.html* (máj 2007)

[13] NCTUns, dostupný na URL: *http://nsl10.csie.ntctu.edu.tw* (máj 2007)

[14] Network Simulator, dostupný na URL: *http://vip.cs.utsa.edu/nsf/net/* (máj 2007)

[15] P2Psim, dostupný na URL: *http://pdos.csail.mit.edu/p2psim* (máj 2007)

[16] OS FreeBSD, dostupný na URL: *http://freebsd.org* (máj 2007)

Zoznam príloh

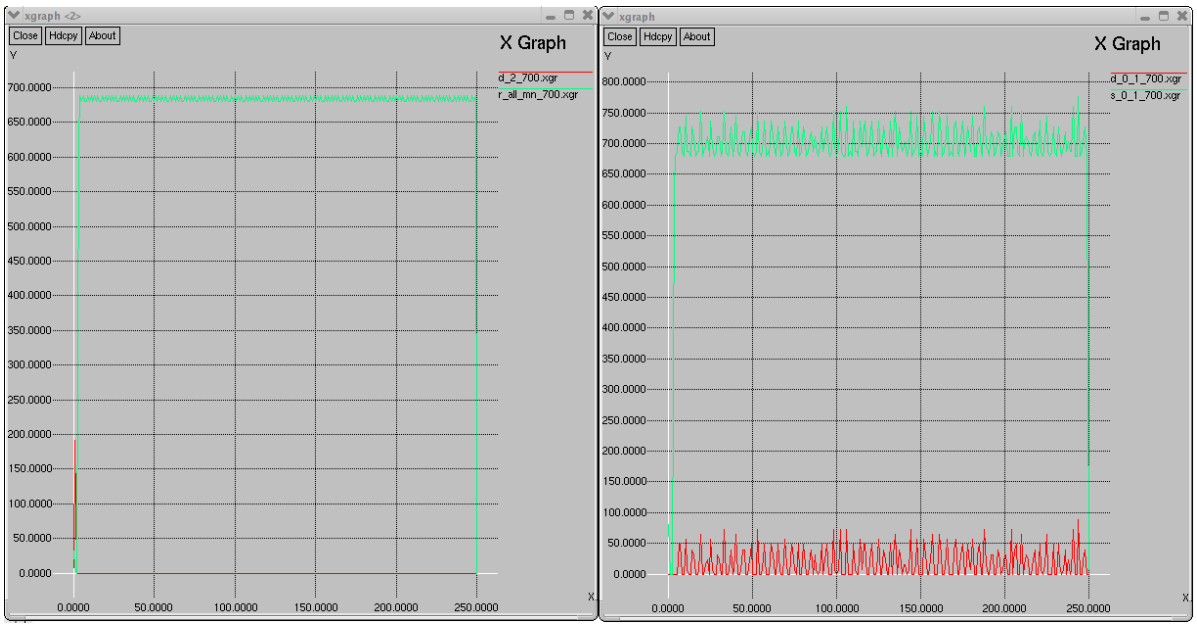
Príloha A: Výstupný súbor simulácie získaný pomocou nástroja NS2.
Príloha B: Grafy znázorňujúce výsledky simulácie.
Príloha C: Pohľady na animácie jednotlivých simulácií.
Príloha D: Záznamy z prevádzky na smerovači v reálnej sieti.

Príloha A

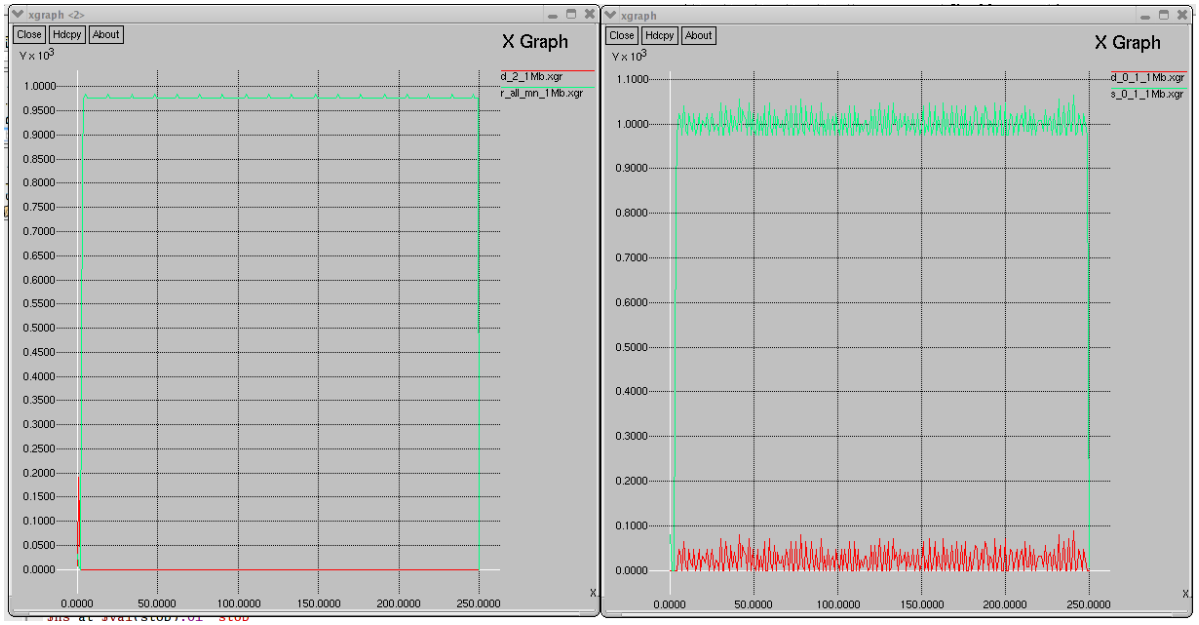
Výsek z výsledného súboru NS2 pri simulácií pripojenia bezdrôtovej siete do internetu linkou kapacity 700 Kbps:

```
- 72.653013 2 1 ack 120 ------- 2 1.0.4.2 0.0.0.3 650 10987
r 72.656023 2 1 ack 120 ------- 2 1.0.4.2 0.0.0.3 650 10987
+ 72.656023 1 0 ack 120 ------- 2 1.0.4.2 0.0.0.3 650 10987
- 72.656023 1 0 ack 120 ------- 2 1.0.4.2 0.0.0.3 650 10987
- 72.656036 0 1 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 765 10926
r 72.659333 0 1 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 600 10922
+ 72.659333 1 2 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 600 10922
- 72.659333 1 2 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 600 10922
r 72.662415 1 2 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 600 10922
r 72.662414721 _2_ RTR  --- 10922 tcp 1024 [0 0 0 0] ------- [0:7 4194312:2 30 0] [600 0] 0 0
f 72.662414721 _2_ RTR  --- 10922 tcp 1024 [0 0 0 0] ------- [0:7 4194312:2 29 4194312] [600 0] 0 0
r 72.663832066 _10_ AGT  --- 10922 tcp 1024 [13a 8 0 800] ------- [0:7 4194312:2 29 4194312] [600 0] 1 0
- 72.667739 0 1 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 766 10927
r 72.671036 0 1 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 601 10923
+ 72.671036 1 2 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 601 10923
- 72.671036 1 2 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 601 10923
r 72.672395 1 0 ack 120 ------- 2 1.0.4.2 0.0.0.3 650 10987
+ 72.672395 0 1 tcp 1024 ------- 2 0.0.0.3 1.0.4.2 656 10988
+ 72.672395 0 1 tcp 1024 ------- 2 0.0.0.3 1.0.4.2 657 10989
r 72.674118 1 2 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 601 10923
r 72.674117579 _2_ RTR  --- 10923 tcp 1024 [0 0 0 0] ------- [0:7 4194312:2 30 0] [601 0] 0 0
f 72.674117579 _2_ RTR  --- 10923 tcp 1024 [0 0 0 0] ------- [0:7 4194312:2 29 4194312] [601 0] 0 0
r 72.675234923 _10_ AGT  --- 10923 tcp 1024 [13a 8 0 800] ------- [0:7 4194312:2 29 4194312] [601 0] 1 0
s 72.675234923 _10_ AGT  --- 10990 ack 100 [0 0 0 0] ------- [4194312:2 0:7 32 0] [601 0] 0 0
r 72.675234923 _10_ RTR  --- 10990 ack 100 [0 0 0 0] ------- [4194312:2 0:7 32 0] [601 0] 0 0
f 72.675234923 _10_ RTR  --- 10990 ack 120 [0 0 0 0] ------- [4194312:2 0:7 32 4194304] [601 0] 0 0
+ 72.676239 2 1 ack 120 ------- 2 1.0.8.2 0.0.0.7 601 10990
- 72.676239 2 1 ack 120 ------- 2 1.0.8.2 0.0.0.7 601 10990
r 72.679248 2 1 ack 120 ------- 2 1.0.8.2 0.0.0.7 601 10990
+ 72.679248 1 0 ack 120 ------- 2 1.0.8.2 0.0.0.7 601 10990
- 72.679248 1 0 ack 120 ------- 2 1.0.8.2 0.0.0.7 601 10990
- 72.679441 0 1 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 767 10928
r 72.682739 0 1 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 765 10926
+ 72.682739 1 2 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 765 10926
- 72.682739 1 2 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 765 10926
r 72.68582 1 2 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 765 10926
r 72.685820436 _2_ RTR  --- 10926 tcp 1024 [0 0 0 0] ------- [0:8 4194313:2 30 0] [765 0] 0 0
f 72.685820436 _2_ RTR  --- 10926 tcp 1024 [0 0 0 0] ------- [0:8 4194313:2 29 4194313] [765 0] 0 0
r 72.687238027 _11_ AGT  --- 10926 tcp 1024 [13a 9 0 800] ------- [0:8 4194313:2 29 4194313] [765 0] 1 0
- 72.691144 0 1 tcp 1024 ------- 2 0.0.0.0 1.0.6.2 484 10930
s 72.693675123 _8_ AGT  --- 10991 ack 40 [0 0 0 0] ------- [4194310:2 0:0 32 0] [483 0] 0 0
r 72.693675123 _8_ RTR  --- 10991 ack 40 [0 0 0 0] ------- [4194310:2 0:0 32 0] [483 0] 0 0
f 72.693675123 _8_ RTR  --- 10991 ack 60 [0 0 0 0] ------- [4194310:2 0:0 32 4194304] [483 0] 0 0
+ 72.694212 2 1 ack 60 ------- 2 1.0.6.2 0.0.0.0 483 10991
- 72.694212 2 1 ack 60 ------- 2 1.0.6.2 0.0.0.0 483 10991
r 72.694441 0 1 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 766 10927
+ 72.694441 1 2 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 766 10927
- 72.694441 1 2 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 766 10927
r 72.69562 1 0 ack 120 ------- 2 1.0.8.2 0.0.0.7 601 10990
+ 72.69562 0 1 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 604 10992
+ 72.69562 0 1 tcp 1024 ------- 2 0.0.0.7 1.0.8.2 605 10993
r 72.697216 2 1 ack 60 ------- 2 1.0.6.2 0.0.0.0 483 10991
+ 72.697216 1 0 ack 60 ------- 2 1.0.6.2 0.0.0.0 483 10991
- 72.697216 1 0 ack 60 ------- 2 1.0.6.2 0.0.0.0 483 10991
r 72.697523 1 2 tcp 1024 ------- 2 0.0.0.8 1.0.9.2 766 10927
r 72.697523293 _2_ RTR  --- 10927 tcp 1024 [0 0 0 0] ------- [0:8 4194313:2 30 0] [766 0] 0 0
f 72.697523293 _2_ RTR  --- 10927 tcp 1024 [0 0 0 0] ------- [0:8 4194313:2 29 4194313] [766 0] 0 0
r 72.699040884 _11_ AGT  --- 10927 tcp 1024 [13a 9 0 800] ------- [0:8 4194313:2 29 4194313] [766 0] 1 0
s 72.699040884 _11_ AGT  --- 10994 ack 100 [0 0 0 0] ------- [4194313:2 0:8 32 0] [766 0] 0 0
r 72.699040884 _11_ RTR  --- 10994 ack 100 [0 0 0 0] ------- [4194313:2 0:8 32 0] [766 0] 0 0
f 72.699040884 _11_ RTR  --- 10994 ack 120 [0 0 0 0] ------- [4194313:2 0:8 32 4194304] [766 0] 0 0
+ 72.699865 2 1 ack 120 ------- 2 1.0.9.2 0.0.0.8 766 10994
- 72.699865 2 1 ack 120 ------- 2 1.0.9.2 0.0.0.8 766 10994
- 72.702847 0 1 tcp 1024 ------- 2 0.0.0.1 1.0.2.2 577 10932
r 72.702875 2 1 ack 120 ------- 2 1.0.9.2 0.0.0.8 766 10994
```

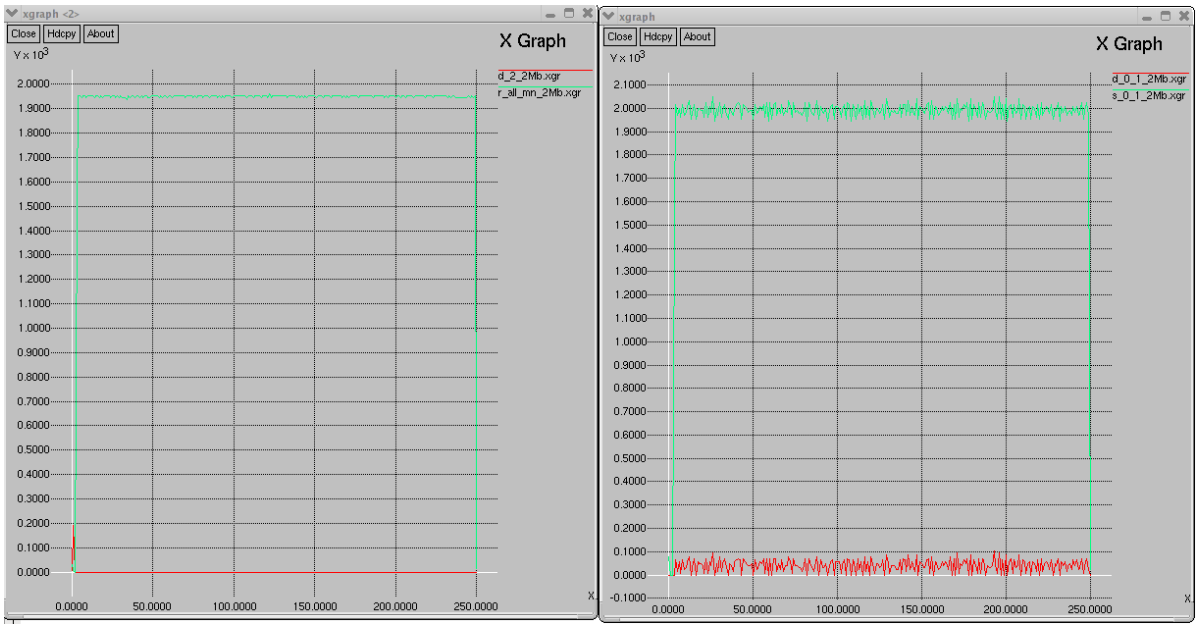
Príloha B



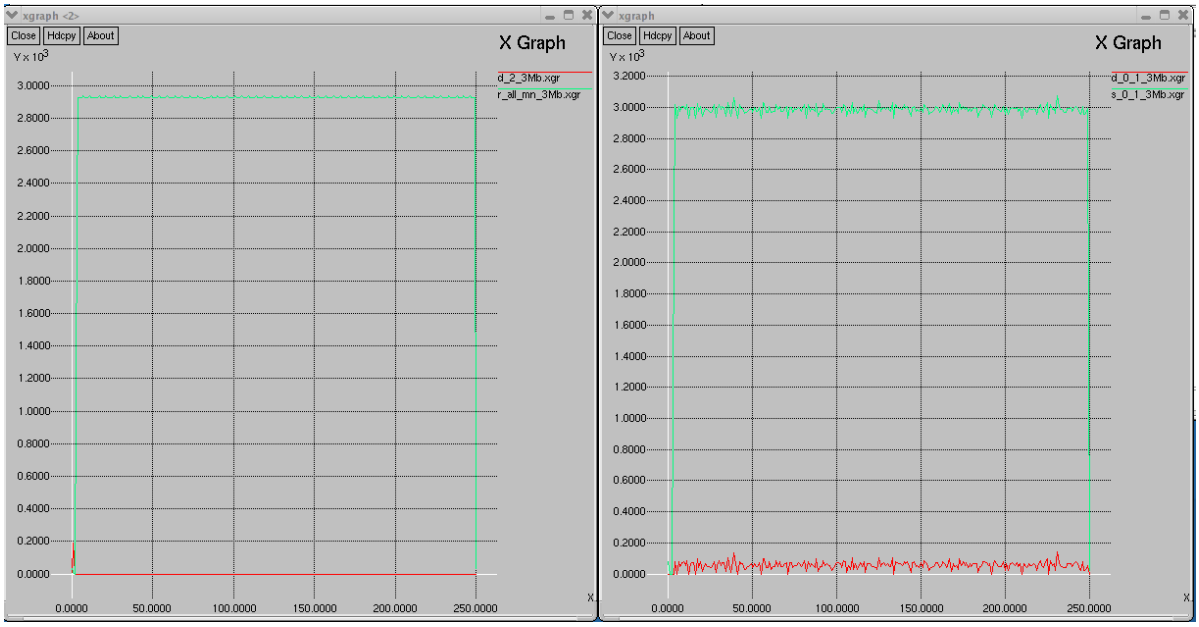
Obrázok B.1: *Výsledne grafy simulácie s linkou 700 Kbps*



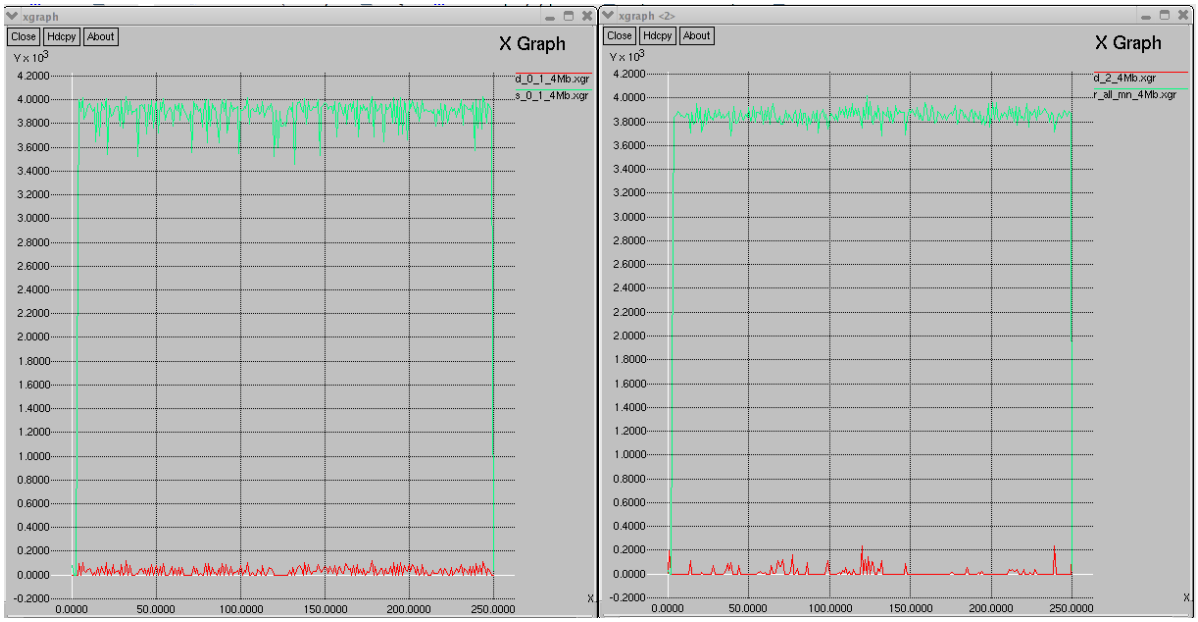
Obrázok B.2: *Výsledne grafy simulácie s linkou 1 Mbps*



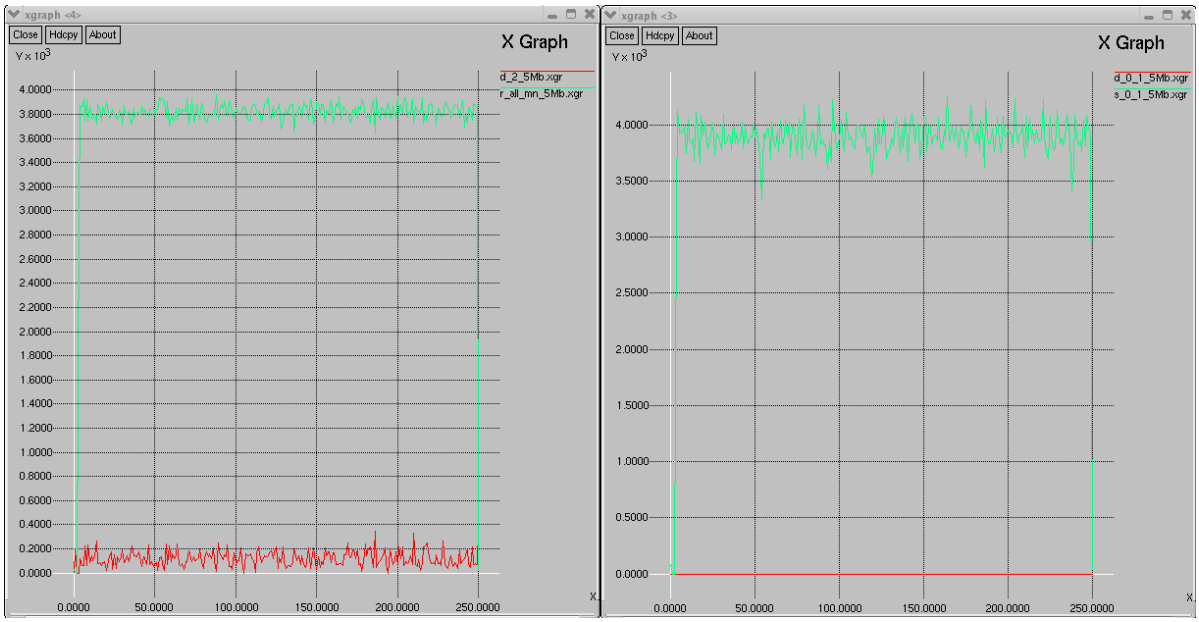
Obrázok B.3: *Výsledne grafy simulácie s linkou 2 Mbps*



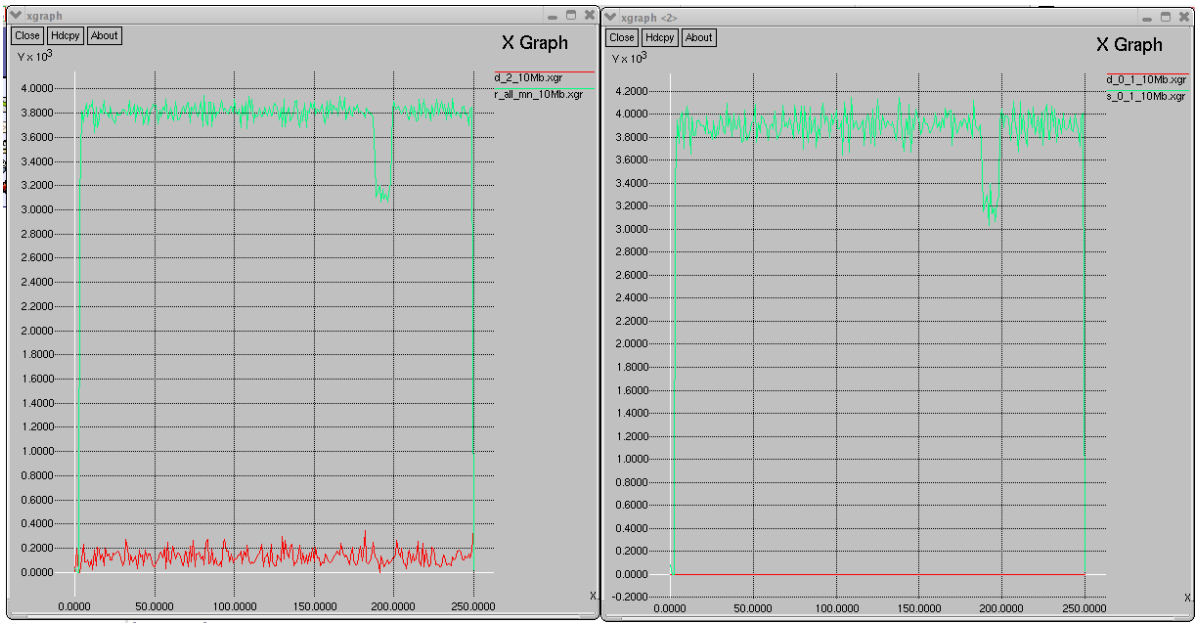
Obrázok B.4: *Výsledne grafy simulácie s linkou 3 Mbps*



Obrázok B.5: *Výsledne grafy simulácie s linkou 4 Mbps*

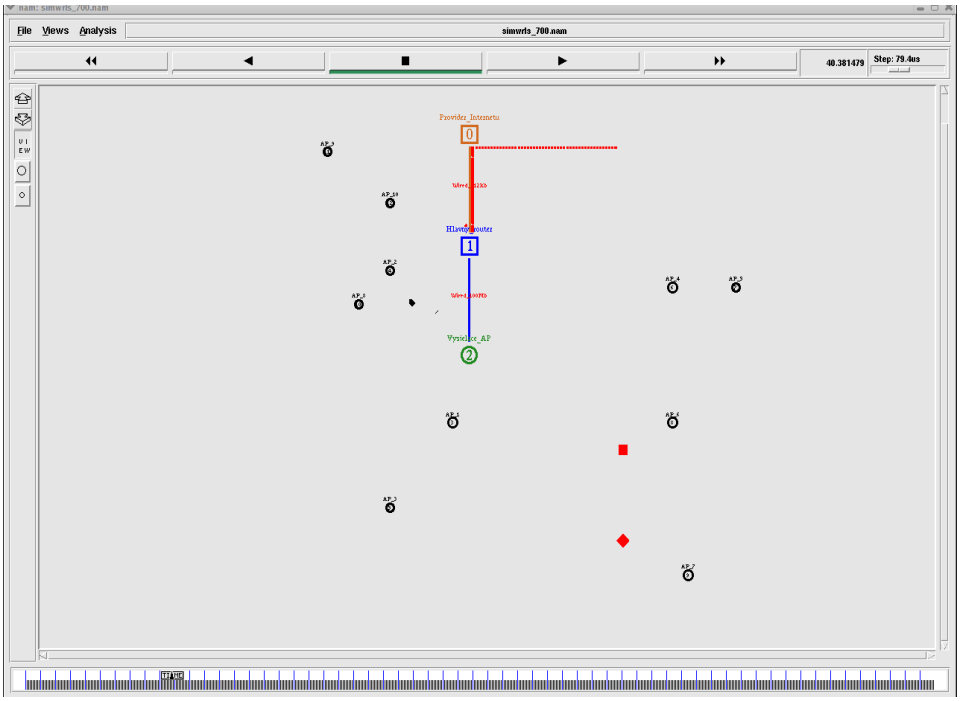


Obrázok B.6: *Výsledne grafy simulácie s linkou 5 Mbps*

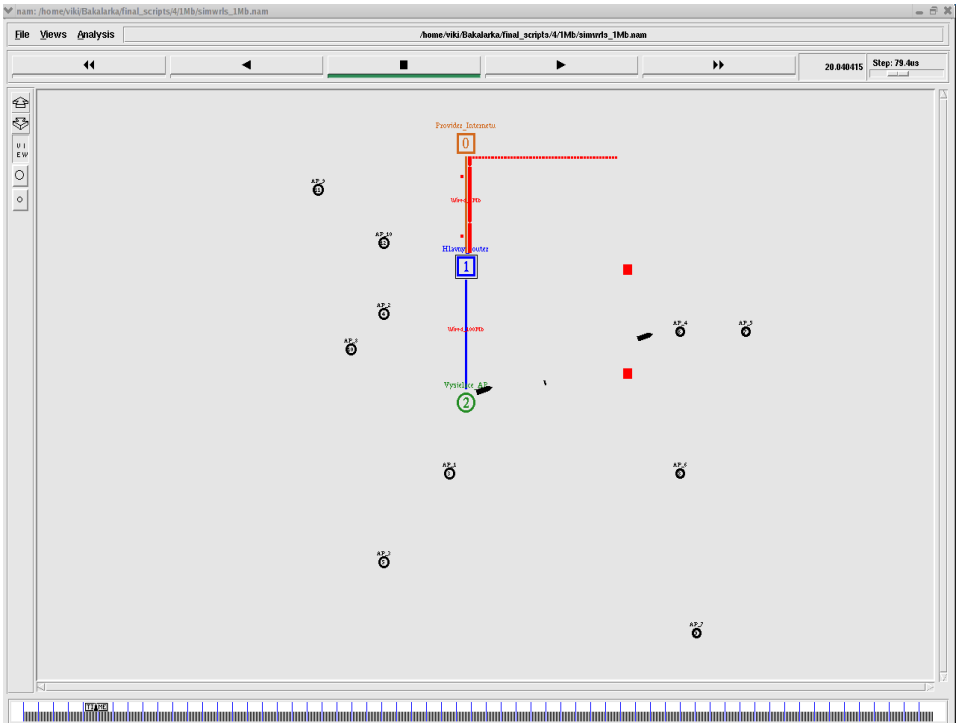


Obrázok B.7: *Výsledne grafy simulácie s linkou 10 Mbps*

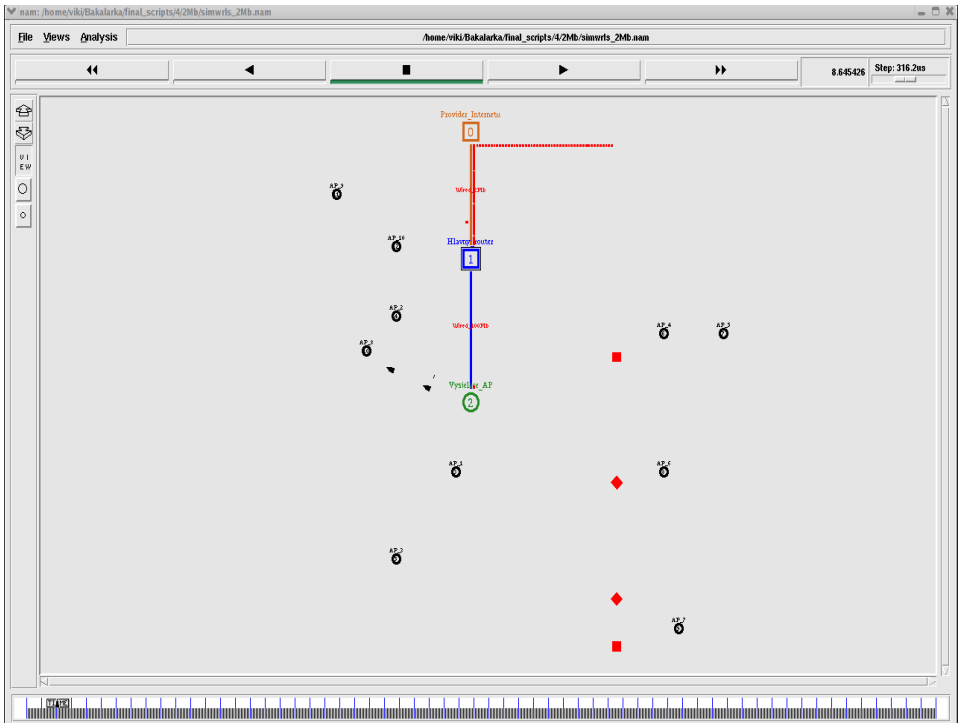
Príloha C



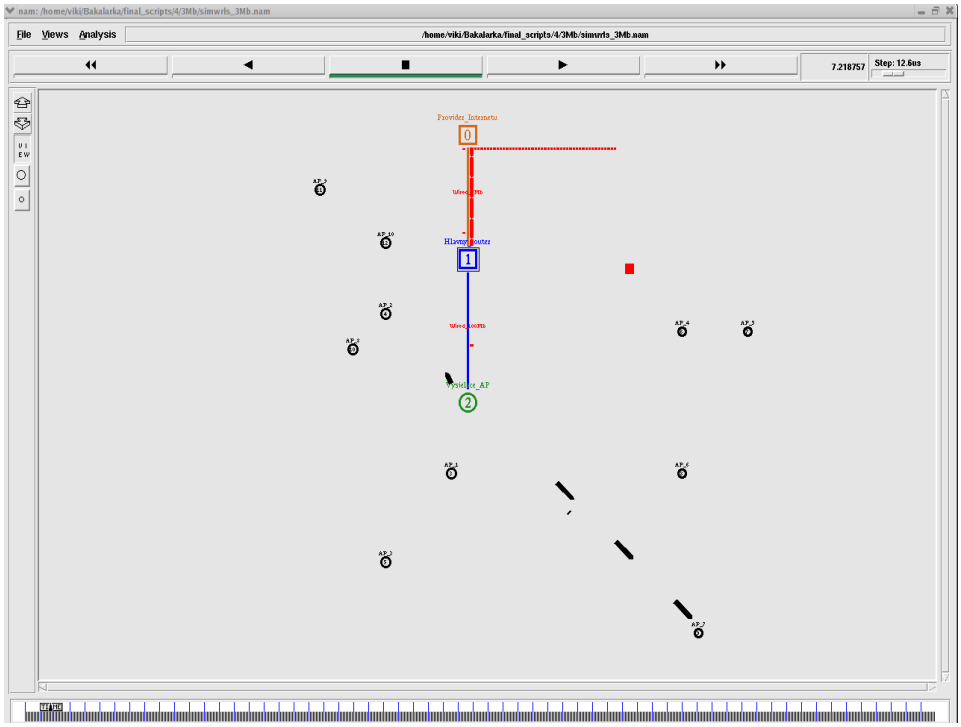
Obrázok C.1: *Pohľad na výslednú simuláciu s linkou 700 Kbps*



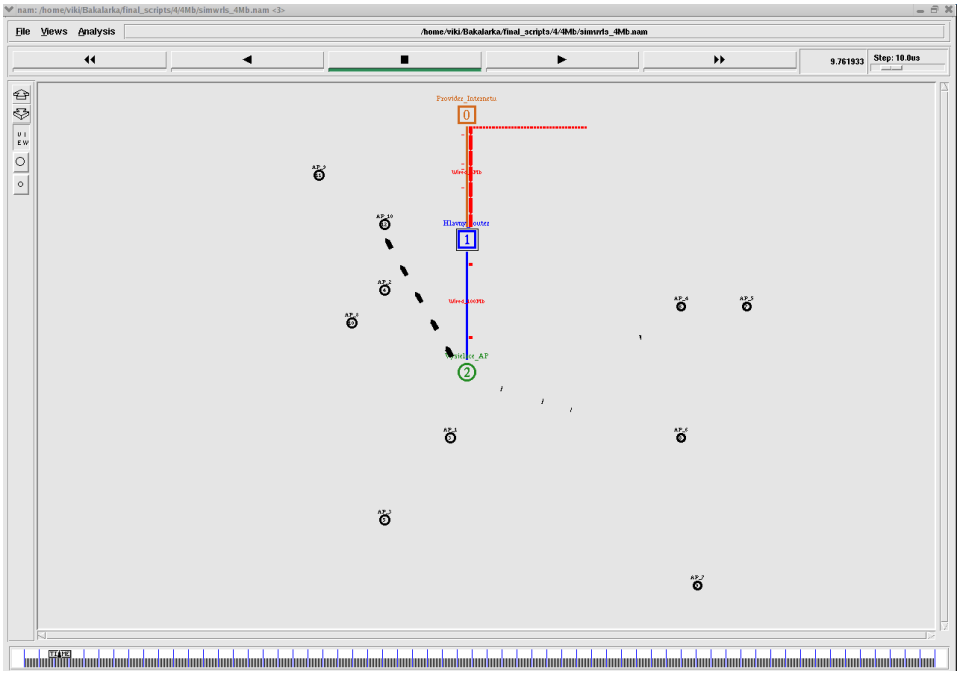
Obrázok C.2: *Pohľad na výslednú simuláciu s linkou 1 Mbps*



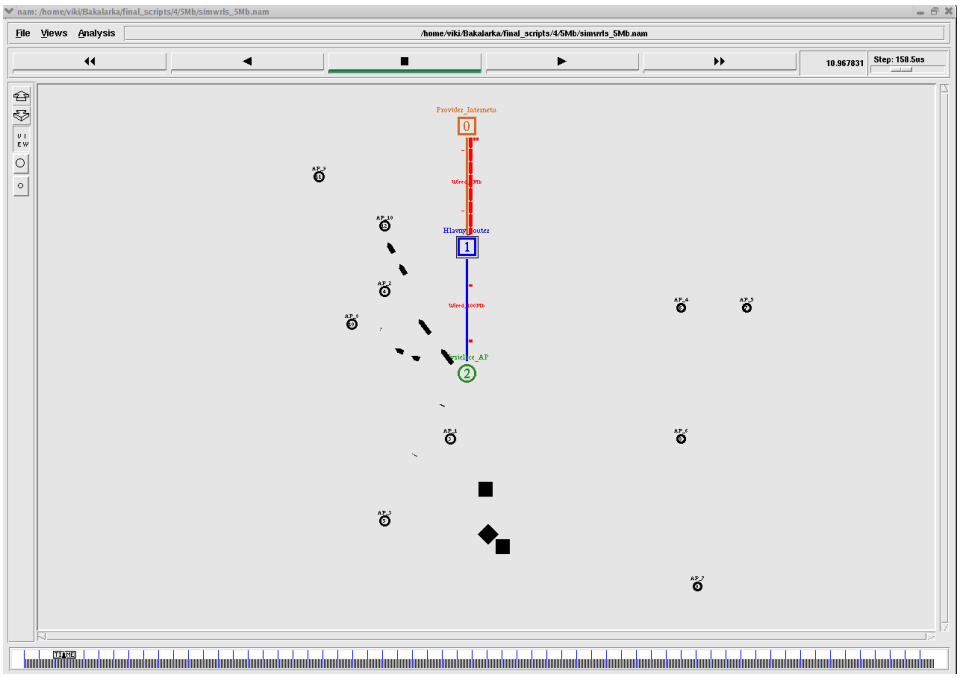
Obrázok C.3: *Pohľad na výslednú simuláciu s linkou 2 Mbps*



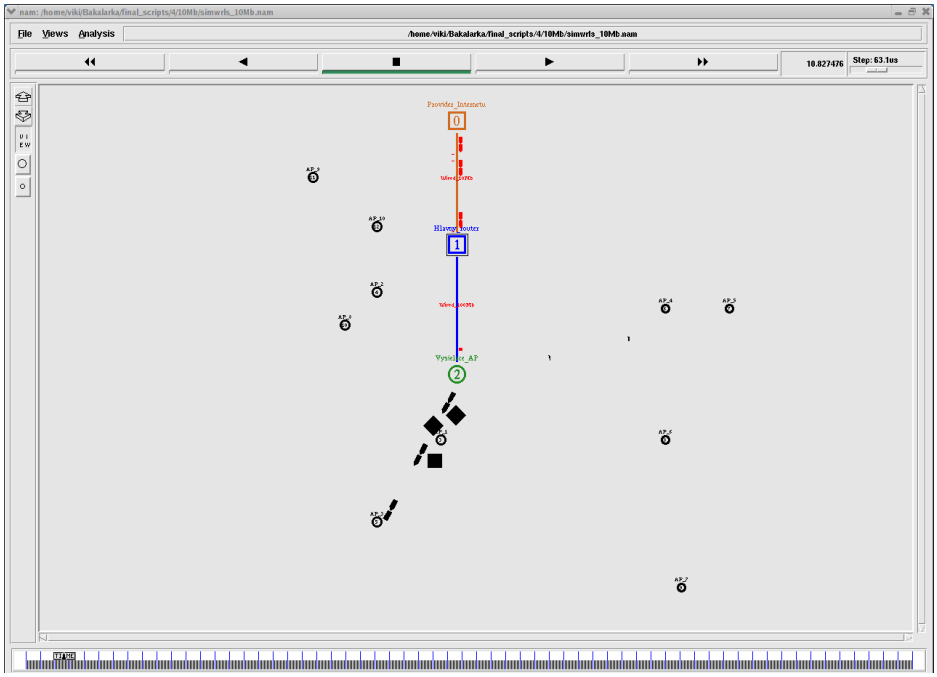
Obrázok C.4: *Pohľad na výslednú simuláciu s linkou 3 Mbps*



Obrázok C.5: *Pohľad na výslednú simuláciu s linkou 4 Mbps*

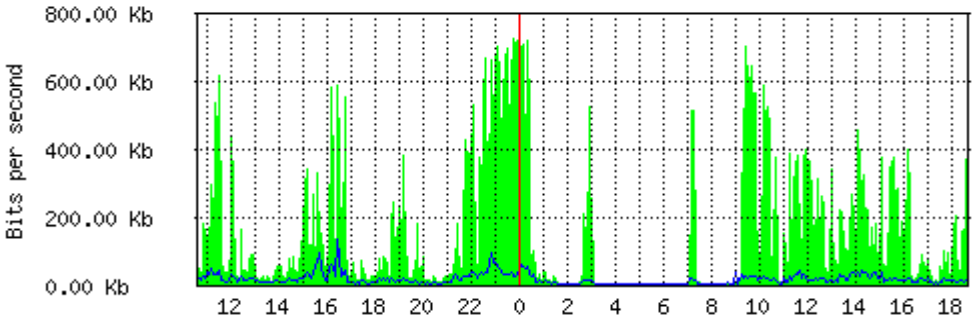


Obrázok C.6: *Pohľad na výslednú simuláciu s linkou 5 Mbps*

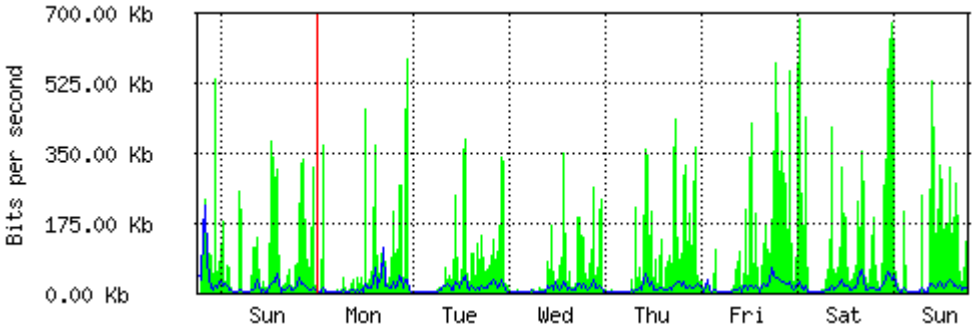


Obrázok C.7: *Pohľad na výslednú simuláciu s linkou 10 Mbps*

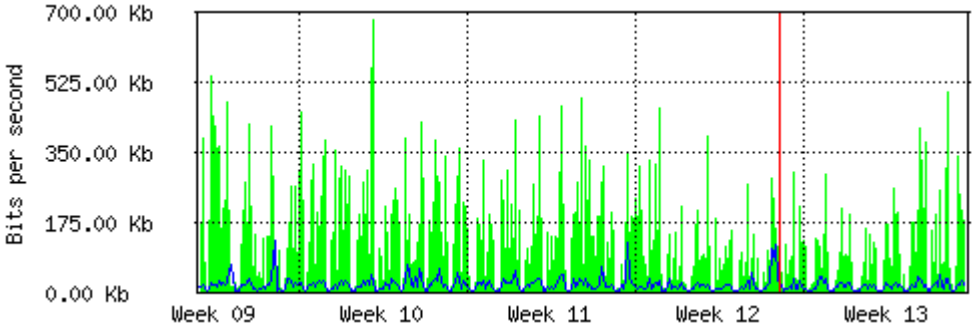
Príloha D



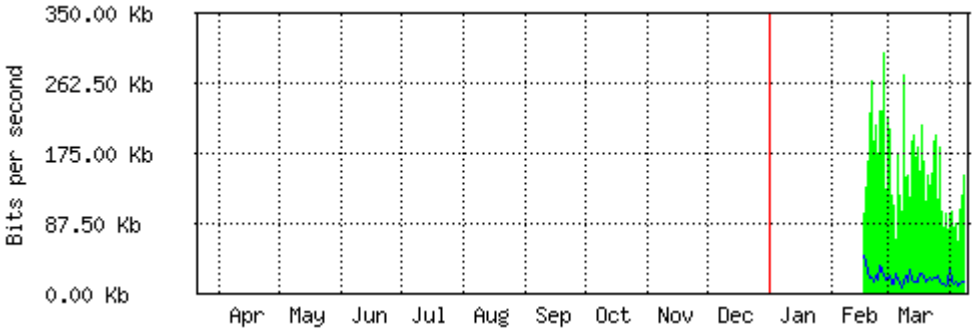
Obrázok D.1: *Denný graf prevádzky siete zo dňa 8.4.2007*



Obrázok D.2: *Týždňový graf prevádzky siete zo dňa 8.4.2007*



Obrázok D.3: *Mesačný graf prevádzky siete zo dňa 8.4.2007*



Obrázok D.4: *Ročný graf prevádzky siete zo dňa 8.4.2007*