Modelování, simulace a analýza propustnosti protokolů rodiny 802.11

Modelling, simulation and throughput analysis of protocols 802.11 family

Bc. Josef Sviták

Diplomová práce
2007

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2006/2007

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Josef SVITÁK**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Modelování, simulace a analýza propustnosti protokolů rodiny 802.11**

Zásady pro vypracování:

1. **Proveďte literární rešerši dané problematiky.**
2. **Prostudujte dostupné simulační modely sítí Ethernet.**
3. **Vytvořte simulační model sítě založené na protokolech 802.11a/b/g v nejčastěji používaném módu přístupový bod a N klientů.**
4. **Na základě simulačního modelu ověřte následující parametry sítě.**
5. **Ověřte simulované výsledky experimenty s reálnými zařízeními.**
6. **Vytvořte sadu doporučení pro stavbu bezdrátových sítí.**

Rozsah práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

Simulation Investigation of the Ethernet Network Performance ...
Efficient and Accurate Ethernet Simulation
http://standards.ieee.org/getieee802/download/802.11-1999.pdf
Modeling multipath in 802.11 systems
a http://netstumbler.com/2002/10/23/modeling_multpath_in_802.11_systems/

Vedoucí diplomové práce:	**Ing. Tomáš Dulík** Ústav aplikované informatiky
Datum zadání diplomové práce:	**13. února 2007**
Termín odevzdání diplomové práce:	**28. května 2007**

Ve Zlíně dne 13. února 2007

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Tato práce se zabývá analýzou protokolů 802.11a, b a g, zejména z hlediska propustnosti. Budu zkoumat bezdrátové sítě v režimu infrastructure a posuzovat vliv výskytu skrytých stanic na propustnost, odezvu a jitter. K provedení těchto analýz budu vytvářet simulační model. Zaměřím se na simulaci MAC vrstvy bezdrátových sítí.

Klíčová slova: 802.11 MAC protokol, simulace, modelování, simulační knihovna

ABSTRACT

This book deals with protocols 802.11a, b and g analysis, especially in term of troughput. I will study wireless networks in infrastructure mode and consider effect of hidden nodes occurrence to troughput, response and jitter. I will make simulation model. Model will be focused to MAC layer of wireless networks.

Keywords: 802.11 MAC protocol, simulation, modelling, simulation library

Prohlašuji, že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků, je-li to uvolněno na základě licenční smlouvy, budu uveden jako spoluautor.

Ve Zlíně ……

…….

Podpis diplomanta

OBSAH

ÚVOD

Lokální bezdrátové počítačové sítě se u nás těší velké oblibě. Rozvoji těchto sítí velmi napomohla existence několika bezlicenčních rádiových pásem. Počítačová síť bez nutnosti pokládat kabely přináší výhody, ale i nevýhody. Výhodou je rychlá instalace sítě a její rychlá demontáž, možnost rychlého připojení nových uživatelů a také možnost mobility stanic. Nevýhodou bývá nižší propustnost sítě daná velkou režií přístupového protokolu a náchylnost k problémům se zarušením bezlicenčního pásma. Sítě 802.11 také na rozdíl od novějších klasických lokálních sítí pracují v poloduplexním režimu.

Velká část bezdrátových sítí se používá ve venkovním (outdoor) prostředí. Jak vyplývá z názvu *lokální,* byly tyto sítě navrženy především pro použití uvnitř budov, na krátké vzdálenosti a hlavně v prostředí, kde jsou všechny stanice vzájemně v dohledu. Pro optimální provoz bezdrátové sítě je důležitá zejména poslední podmínka, její splnění je však ve venkovním prostředí často neřešitelné. Nesplnění této podmínky vede ke vzniku tzv. skrytých uzlů.

Tato práce se zabývá vytvořením modelu bezdrátové sítě, pracující s protokolem 802.11 a modelováním jejího přístupového protokolu ve venkovním prostředí, to znamená s výskytem skrytých uzlů. Cílem práce je vytvořit a otestovat model sítě v režimu označovaném jako infrastructure, to znamená s přístupovým bodem.

I. TEORETICKÁ ČÁST

1 LITERÁRNÍ REŠERŠE PROBLEMATIKY

- **http://www.opnet.com/:** Komerční prostředí pro simulace sítí, klasické i bezdrátové, mobilní atd.
- **http://nsnam.isi.edu/nsnam/index.php/Main_Page**: opensource simulační prostředí pro síťové aplikace. Začátek vývoje modulu pro 802.11, který však nepokračuje.
- **On the Impact of IEEE 802.11 MAC on Traffic Characteristics**, Omesh Tickoo, Student Member, IEEE and Biplab Sikdar, Member, IEEE: analytický model 802.11 MAC protokolu

2 MODELOVÁNÍ A SIMULACE POČÍTAČOVÝCH SÍTÍ

Žádný ucelený simulační model mi není znám. Uvedené zdroje se zabývají problematikou částečně nebo z jiného úhlu pohledu.

Obecně rozlišujeme dva druhy simulačních modelů. Je to analytický model a počítačová simulace. Simulaci dále rozdělujeme na spojitou a diskrétní.

Analytickým modelem rozumíme nahrazení simulovaného systému (počítačové sítě) sadou rovnic. Řešení těchto rovnic je potom výsledkem modelu. Pro přesné modelování počítačových sítí se analytické modely příliš nehodí. Zejména sítě založené na přístupové metodě CSMA pracují s náhodnými časovými intervaly, se kterými se analyticky pracuje obtížně.

Pro simulaci počítačové sítě je nejvhodnější použití diskrétní simulace. Její výhody jsou následující:

- můžeme ovlivnit úroveň detailů
- podrobné procházení prostoru možných řešení
- kombinace matematických a empirických modelů
- použití hodnot naměřených na reálném systému

Síťový simulátor je samostatný software, může to být komplexní prostředí pro návrh, simulaci a analýzu výkonnosti sítě. Může existovat také jako modul pro univerzální simulační prostředí.

2.1 Modelování a simulace ethernetu

Simulace klasických („drátových“) ethernetových sítí je poměrně propracovaná pro všechny jeho typy. Je to dáno tím, že tyto sítě ve srovnání s bezdrátovými existují dlouhou dobu.

Ethernet můžeme modelovat pomocí analytických metod, statistických metod, nebo sestavíme model jako počítačovou simulaci. V této práci se zabývám modelováním MAC protokolu, proto se zaměřím na počítačové simulace.

Pro simulaci ethernetu je vhodný model řízený událostmi [1]. Jako události se bere průchod paketů přenosovým médiem, požadavky stanic na přenos dat atd. Pakety samotné potom považujeme za zprávy, které si vyměňují jednotlivé objekty modelu. Rozlišujeme dva typy modelu řízeného událostmi, centralizovaný a decentralizovaný model.

Pro simulaci bezdrátových sítí budu vycházet ze starších verzí ethernetu. U novějších verzí

neexistuje sdílené médium a tím jsou vyloučeny kolize. Bezdrátové sítě jsou charakteristické právě sdíleným médiem. Proto zde budu vycházet z modelů a simulací ethernetu 10Base-2 a 10Base-T s použitím hubů jako aktivních prvků.

2.1.1 Centralizovaný model

Zaměřuje se zejména na modelování přenosového média. Na stanice připojené k síti se pohlíží pouze jako na zdroje a příjemce dat. Tento typ modelu je vhodný pro modelování nejnižší síťové vrstvy, fyzikálních vlastností média.

2.1.2 Distribuovaný model

U tohoto typu modelu bereme přenosové médium jako pasivní, slouží pouze k přesunu dat mezi stanicemi [2]. Důraz je kladen na modelování chování stanic. Distribuovaný model se hodí pro modelování protokolů pro přístup k médiu (MAC). U tohoto modelu se také nejlépe uplatní řízení simulace událostmi.

Událostmi se u tohoto modelu rozumí například požadavek na odeslání paketu, přijetí paketu (jeho doručení médiem), ale také vypršení náhodně nastavovaných časovačů pro přístup k médiu.

2.1.3 Výkonnost sítě

Pro sledování výsledků modelu je dobré zavést si kritéria, která můžeme souhrnně pojmenovat jako výkonnost sítě. U počítačové sítě nejčastěji sledujeme její propustnost, odezvu a kolísání odezvy (jitter). U některých typů ethernetu je důležitým kritériem také počet kolizí při přístupu k médiu. Rovněž sledujeme ztrátovost paketů. Výkonnost sítě definujeme jako vztah mezi provozem odeslaným dané stanici a provozem, který tato stanice skutečně přijme [1].

Výkonnost sítě závisí na počtu kolizí. Ten je závislý na počtu stanic v síti, intenzitě provozu a velikosti paketů. Nejlepšího poměru mezi odeslanými a skutečně přijatými daty dosahuje ethernet u menších paketů [1], [2]. K přenosu malého paketu je potřeba kratší čas než k přenosu velkého, s časem se zmenšuje pravděpodobnost kolize.

2.1.4 Další faktory, zahrnuté v modelu

Důležitým faktorem při modelování je čas, který je potřebný k přenosu paketu od zdroje k cíli sdíleným médiem (propagation delay). Příjemce má tedy paket k dispozici s časovým zpožděním oproti okamžiku, kdy odesílatel začal s vysíláním. V modelu dochází k přenosu

zpráv vzhledem k modelovanému systému „okamžitě“. Pro dosažení chování modelu blížícího se realitě musíme tento čas brát v úvahu. V praxi se tento problém řeší pomocí tzv. zpožděného odesílání, kdy se s přenosem zprávy počká až na okamžik, kdy má být doručena. Po tuto dobu je třeba zajistit blokování sdíleného média jiným způsobem.

Pro lepší přiblížení modelu reálné síti s reálnými zařízeními je vhodné zavést omezení pro fronty zpráv. Při zaplnění těchto front potom dochází k zahazování zpráv, stejně jako u reálného systému. Model potom musí být schopen se se ztrátami zpráv vyrovnat.

3 MODELOVÁNÍ A SIMULACE BEZDRÁTOVÝCH POČÍTAČOVÝCH SÍTÍ

V této práci budu vytvářet model MAC protokolu, proto se stejně jako v odstavci 2.1 bude jednat o událostmi řízený diskrétní distribuovaný model.

Přístupová metoda, používaná v bezdrátových sítích 802.11, vyžaduje odlišný přístup k simulaci než klasické ethernetové sítě. Hlavní rozdíl je v nemožnosti detekovat kolizi při vysílání stanicí, která už vysílá. Proto je přístupová metoda založena na vyhýbání se kolizím (CSMA/CA).

Kolizím nebo ztrátám paketů z jiných důvodů se ovšem není možné vyhnout úplně, proto se používá pozitivní potvrzování. Ihned po přenosu každého datového paketu následuje potvrzení příjemcem. Pokud nedorazí odesílateli do určité, doby přenos se opakuje.

Na rozdíl od drátových sítí existuje v bezdrátové síti více možností ztrát paketů. Kromě zaplnění bufferů aktivních síťových prvků může být paket znehodnocen kvalitou bezdrátového prostředí (rušení jinými sítěmi, vzdálenost, odrazy signálu, počasí,...) natolik, že se ho nepodaří správně přijmout. Zejména ve venkovním prostředí je častý další jev, který vede ke ztrátám paketů. Jde o takzvané *skryté uzly.* Z důvodu existence překážek pro šíření rádiového signálu se stává, že ne všechny stanice v síti na sebe navzájem „vidí“, to znamená, že ne každá stanice je schopná zachytit vysílání všech ostatních stanic. Viditelnost je přitom důležitá právě pro detekci volného média. Skrytý uzel tedy není schopen detekovat vysílání jiných stanic, s určitou pravděpodobností jejich vysílání naruší. Naopak ostatní stanice nejsou schopny detekovat vysílání skrytého uzlu, mohou tedy narušovat jeho vysílání.

3.1 Bezdrátová síť s protokolem 802.11

První verze protokolu byla vydána v roce 1999 jako standard pro rádiové a infračervené bezdrátové sítě. Definuje parametry fyzické vrstvy, řízení přístupu k médiu, postup připojování do sítě. Volitelně také zabezpečení bezdrátové komunikace šifrováním a power management.

Zavádí dva možné způsoby uspořádání bezdrátové sítě, režimy *ad-hoc* a *infrastructure*, v oficiální dokumentaci [3] pojmenované jako IBSS (independent basic service set) a BSS (basic service set). Režim ad-hoc je velmi podobný klasickému ethernetu.

3.1.1 Přístupový bod (access point)

V režimu infrastructure jsou bezdrátová zařízení v síti rozdělena na přístupový bod a stanice. V každé lokální bezdrátové síti (podle [3] nazvané BSS) je právě jeden přístupový bod. V tomto režimu komunikuje každá stanice pouze s přístupovým bodem. Ten potom předává pakety konečnému příjemci. Přístupový bod má v síti několik důležitých funkcí.

- určuje, na kterém kanále se bude komunikovat
- zajišťuje zabezpečení provozu
- spojuje bezdrátovou síť s jinou sítí, nejčastěji s klasickým ethernetem

Pro zajištění těchto funkcí řeší přístupový bod autentizaci a autorizaci, asociaci a deasociaci stanic, směrování provozu do bezdrátové sítě nebo ethernetu. Za účelem tohoto směrování si přístupový bod udržuje tabulku připojených (asociovaných) stanic a tabulku stanic v připojené ethernetové síti. Z důvodu usnadnění předávání dat z jedné sítě do druhé byl u bezdrátových sítí převzat formát ethernetových MAC adres. Přístupový bod funguje jako ethernetový bridge mezi bezdrátovou a drátovou sítí. Za tímto účelem bývá přístupový bod osazen minimálně jedním ethernetovým portem.

Dále většinou obsahuje diagnostické funkce, které slouží pro hledání problémů v bezdrátové síti, účtování přenesených dat a/nebo doby připojení. Novější přístupové body podporují rozšířené autentizační a autorizační funkce například protokolem RADIUS.

Kromě toho musí přístupový bod ovládat také funkce bezdrátové stanice pro korektní příjem a odesílání dat.

Přístupové body je možné sdružovat do větších sítí. Každá stanice, asociovaná k některému z nich, může komunikovat s kteroukoliv jinou stanicí připojenou ke stejnému nebo jinému bodu. Přístupové body jsou mezi sebou propojeny *distribučním systémem* a podle [3] dohromady s připojenými stanicemi tvoří *ESS* (extended service set). Distribuční systém tvoří nejčastěji ethernetová síť, novější přístupové body podporují bezdrátový distribuční systém (WDS, wireless distribution systém). V rámci jedné ESS se mohou bezdrátové stanice pohybovat, a to i mezi různými přístupovými body. Stanice je potom deasociována od jednoho bodu a asociována k jinému. Pohyb stanic v rámci ESS se nazývá *roaming*. Problematikou roamingu, zejména možností výpadku přenosu dat v okamžiku přepojení mezi přístupovými body, se zabývá [4].

Je zřejmé, že přístupový bod je nejvíce zatíženým prvkem bezdrátové sítě, ať už při komunikaci mezi jednotlivými stanicemi nebo při komunikaci bezdrátové stanice se stanicí v

připojeném ethernetu (viz např. Obrázek 9 na straně 41). Proto bývá výběr dostatečně výkonného přístupového bodu rozhodující pro výkonnost sítě.

Obrázek 1.: Produktová řada firmy Orinoco

Obrázek 1 ukazuje ucelenou produktovou řadu zařízení pro bezdrátové sítě. Jedná se o přístupové body i stanice.

3.1.2 Bezdrátové stanice

Podle původního protokolu 802.11 slouží bezdrátová stanice pro připojení jediného počítače (nebo jiného zařízení) do bezdrátové sítě. Jediného proto, že přístupový bod má ve své tabulce asociovaných stanic uloženu právě jednu MAC adresu pro každou stanici.

Stanice se podle provedení dělí na externí a interní (viz Obrázek 1). Ani externí stanice není podle protokolu 802.11 možné použít pro připojení více zařízení do bezdrátové sítě. Typickým zástupcem této skupiny externích zařízení je Orinoco ethernet adapter, na obrázku 1 v levém dolním rohu. I když je adaptér vybaven ethernetovým portem, není možné za něho přímo připojit switch a další zařízení.

Modernější zařízení umožňují toto omezení obejít. Taková stanice má ještě svoji vlastní interní tabulku MAC adres dalších připojených zařízení. Navenek (směrem k bezdrátové síti) vystupuje pod jedinou, svojí vlastní MAC adresou.

Novější externí stanice jsou schopny fungovat v několika režimech, jeden z těchto režimů

bývá Access point. Tím se v podstatě stírá rozdíl mezi stanicemi a přístupovými body, jedno zařízení je schopné fungovat v obou režimech. Nic se ale nemění na tom, že v jedné bezdrátové síti je jediné zařízení v režimu AP (access point, přístupový bod) a ostatní zařízení musí být v klientském režimu. Některé stanice jsou dále vybaveny dalšími režimy, které už ale nejsou obsaženy v protokolu 802.11. Těchto funkcí je dosaženo rozšířením software jednotek a v převážné většině pracují pouze mezi zařízeními jednoho výrobce nebo dokonce jenom mezi určitými verzemi jednotek. Nejčastěji jde o režimy pro transparentní propojení dvou ethernetových sítí nebo různé turbo režimy pro dosažení vyšších přenosových rychlostí, než předepisuje protokol 802.11.

3.2 MAC vrstva protokolu 802.11

Funkce MAC vrstvy bezdrátové sítě můžeme rozdělit na dvě skupiny. Základní metodou přístupu ke sdílenému médiu je *DCF* (distributed coordination function). Tato metoda je známější jako *CSMA/CA* (carrier sense multiple access with collision avoidance). Podle [3] je DCF povinná pro každou bezdrátovou stanici v síti v obou režimech, ad-hoc i infrastructure. Druhou metodou přístupu k médiu je *PCF* (point coordination function). Tato metoda je použitelná pouze v síti v režimu infrastructure. U této metody řídí celou komunikaci přístupový bod, ten určuje, která stanice má právo vysílat. Řízení komunikace je realizováno využitím mechanismu virtuální nosné a nastavování NAV (viz dále) pomocí Beacon servisních rámců.

Metoda PCF je navržena tak, aby byla schopná pracovat v síti, kde se už používá DCF.

3.2.1 Přístupová metoda DCF

Tato metoda umožňuje sdílení přenosového média použitím CSMA/CA a náhodného časového intervalu po detekci volného média. Každý datový rámec je okamžitě potvrzován servisním rámcem ACK. Pokud odesílatel neobdrží potvrzení odeslaného rámce, přenos se opakuje. Protokol CSMA/CA je navržen pro snížení pravděpodobnosti kolize tam, kde je její největší pravděpodobnost. Ta je po ukončení předchozího vysílání. Všechny stanice, které mají data k odeslání, na tento okamžik totiž čekají a všechny ho také pomocí funkce *CS* (carrier sense, detekce nosné) prakticky ve stejném okamžiku detekují. Pokud by v tomto okamžiku začalo vysílat několik stanic, došlo by ke kolizi. Proto musí každá stanice před pokusem o vysílání čekat ještě náhodný časový okamžik. Důvod je ten, že jakmile začne stanice vysílat, už není schopna detekovat kolizi.

3.2.2 Metody detekce nosné

Protokol 802.11 zavádí dva způsoby detekce nosné, a tím i detekce obsazeného nebo volného média. Jsou to metody virtuální a fyzická (naslouchání médiu). Fyzickou detekci nosné zajišťuje fyzická vrstva bezdrátové sítě, virtuální detekci nosné zajišťuje MAC vrstva.

Základem metody virtuální detekce nosné je předávání informací o době přenosu a udržování této informace každou stanicí ve vnitřní proměnné, zvané *NAV* (Network Allocation Vector). Informace o době obsazení média se předává rámci *RTS* (Request To Send) a *CTS* (Clear To Send). Tyto rámce v poli *Duration* obsahují informaci o délce časového intervalu, potřebné pro přenos následujícího datového rámce a ihned následujícího potvrzení ACK o přijetí.

Všechny stanice v dosahu, které zachytí rámec RTS nebo CTS, jsou povinné z nich převzít informaci o následující době obsazení média a v této době nesmí vysílat (berou médium jako obsazené). Stanice musí reagovat i na rámce RTS/CTS, které nejsou adresovány jim, a podle [3] je každá bezdrátová stanice povinna reagovat na tyto rámce, i když sama nevyužívá mechanismus RTS/CTS pro rezervaci média.

Pokud stanice nedostane na požadavek RTS odpověď CTS do protokolem definované doby, odesílá se znova rámec RTS. Ten je výrazně kratší, než bývají datové rámce, proto jeho odesílání zabere méně času.

Další přínos virtuální detekce nosné je v možnosti koordinace provozu mezi různými bezdrátovými sítěmi, pracujícími ve stejné oblasti na stejném kanále. Stanice jedné sítě mohou zachycovat RTS/CTS rámce od stanic jiné sítě, nebo alespoň jeden z dvojice těchto rámců. To v případě, že z jiné sítě jsou v dohledu pouze některé stanice nebo jenom přístupový bod.

Používání mechanismu RTS/CTS přidává do sítě další režii, proto se nevyplatí pro malé datové rámce. Hranice, od které se bude pro větší rámce používat, se nastavuje pro každou stanici zvlášť. RTS/CTS také není možné použít u broadcastových a multicastových rámců.

3.2.3 Intervaly mezi jednotlivými rámci

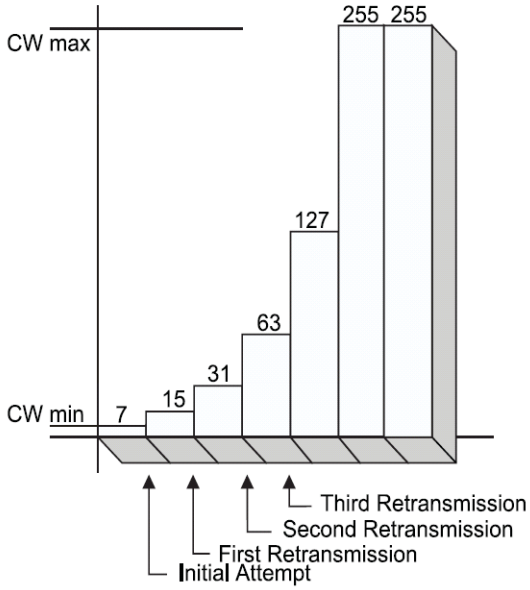
Jednotlivé rámce jsou od sebe odděleny krátkým časovým intervalem. Těchto intervalů je několik druhů. Pro tuto práci je důležitý pouze interval *DIFS* (DCF interframe space). Je ze všech intervalů nejdelší. Ostatní tři intervaly, včetně vztahů mezi nimi, jsou popsány v [3] na straně 74 a dalších. Délku intervalu DIFS získáme výpočtem podle vztahu (1) z parametrů, daných protokolem.

$$DIFS = aSIFSTime + 2 \cdot aSlotTime \tag{1}$$

3.2.4 Náhodný čekací časový interval

Při ošetření některých situací předepisuje protokol 802.11 náhodně dlouhé čekání. Tento náhodný časový interval je vždy násobkem základního časového intervalu, zvaného *aSlotTime* a definovaného protokolem fyzické vrstvy.

$$BackoffTime = Random() \cdot aSlotTime \tag{2}$$



Obrázek 2.: Vývoj hodnot parametru CW

Délka náhodného časového intervalu před pokusem o vysílání se určí podle vztahu (2). Funkce `Random()` vrací náhodné číslo z intervalu (0,CW), kde *CW* (contention window) leží v intervalu (aCWmin, aCWmax) daném fyzickou vrstvou protokolu. Proměnný CW roste od aCWmin do aCWmax po mocninách 2 zmenšených o 1. Tento mechanismus se nazývá také *exponential random backoff time*. Omezením horní hranice intervalu pomocí aCWmax je dosaženo stability algoritmu při vysokém zatížení sítě. Vývoj hodnot parametru CW ukazuje Obrázek 2. Tento obrázek jsem převzal z [3].

3.2.5 Mechanismus RTS/CTS

V bezdrátové síti není zaručeno, že každá stanice je v dohledu všech ostatních stanic. Ty stanice, které nejsou v dohledu ostatních, se nazývají *skryté uzly*. Taková stanice není schopna detekovat vysílání jiné stanice, a může tedy považovat sdílené přenosové médium

za volné, i když volné není. Následným vysíláním znehodnotí přenos dat od jiné stanice a také svůj vlastní, dojde ke kolizi, kterou navíc ani jedna ze stanic není schopna detekovat. Všechny stanice, účastnící se kolize, musí čekat na potvrzení ACK svých odeslaných dat. Toto potvrzení po kolizi nepřijde, všechny stanice tedy musí vysílání opakovat. Přitom není vyloučen vznik další kolize.

Za účelem minimalizace tohoto problému je v protokolu 802.11 zaveden mechanismus RTS/CTS. Odesláním rámce RTS žádá stanice o rezervaci média na dobu, po kterou bude trvat přenos následujícího datového rámce včetně jeho potvrzení rámcem ACK. Pokud cílová stanice obdrží požadavek RTS a je schopná přijímat data, potvrdí příjem rámcem CTS. Zdrojová stanice (odesílatel) čeká po odeslání RTS po dobu aCtsTimeout na odpověď CTS. Pokud do této doby odpověď nedostane, odesílá se znova RTS.

Parametr aCtsTimeout není předepsaný protokolem. Některá zařízení ho mají nastaven napevno, u některých je možné ho měnit.

Každá stanice je podle protokolu 802.11 povinna reagovat na všechny RTS/CTS rámce, které zachytí. Stanice zachytí všechny tyto rámce, které odešle kterákoliv jiná stanice v dohledu. V rámci RTS je v položce Duration uložena doba, po kterou má trvat následující přenos, včetně následného CTS, datového rámce a jeho potvrzení ACK. Do odpovědi CTS uloží cílová stanice dobu z přijatého RTS zmenšenou o dobu trvání přenosu RTS. Rámce RTS i CTS mají konstantní velikost, určení doby trvání jejich přenosu tedy není problém.

Každé bezdrátové zařízení v síti si udržuje v proměnné NAV dobu, po kterou ještě bude obsazeno médium. Pokud kterákoliv stanice zachytí některý z dvojice RTS/CTS a položka Duration má větší hodnotu než aktuální NAV, je zařízení povinno upravit si hodnotu NAV na zachycenou hodnotu Duration. Hodnota Duration v rámci CTS je vždy nižší než v předchozím RTS, který k němu patří. Tím je zaručeno, že stanice, která před zachycením CTS zachytila jemu odpovídající RTS, si už nebude upravovat NAV. Tím také odpadá nutnost rozlišovat, ke kterému RTS patří CTS. Stanice, která (například z důvodu viditelnosti) předchozí RTS nezachytila, si nastaví NAV podle rámce CTS.

Bezdrátová síť musí být uspořádána tak, aby minimálně přístupový bod byl v dohledu všech stanic. Tím je dáno, že každá stanice má možnost zachytit minimálně jeden z dvojice RTS/CTS rámců. Tato možnost však není zaručena, vždy existuje možnost ztráty některého rámce.

Jak už bylo napsáno v odstavci 3.2.2, mechanismus RTS/CTS přidává do sítě další režii. Rámec RTS je velký 20 oktetů [3], rámec CTS je ještě o 6 oktetů menší. Z důvodu možnosti

výskytu různých přenosových rychlostí v jedné síti musí být RTS/CTS vysíláno tak, aby je byly schopny zachytit všechny stanice. To znamená, že jsou vysílány nejnižší přenosovou rychlostí, která se v síti může vyskytnout, a tím pádem může být doba jejich přenosu srovnatelná s dobou přenosu delších datových rámců, vysílaných vyšší rychlostí.

Není proto vhodné používat mechanismus RTS/CTS pro všechny datové rámce. Zvláště u malých datových rámců se efekt RTS/CTS ztrácí. Každá stanice má nastavený vlastní práh, od kterého se pro větší rámce používá tento mechanismus. Obecně doporučovaná hodnota pro sítě 802.11b je podle dokumentace k access pointům Orinoco 500 oktetů.

Podle protokolu 802.11 je povinna na RTS/CTS rámce reagovat každá stanice v síti nastavením svého NAV, a to i taková, která sama nemá použití RTS/CTS mechanismu zapnuto.

4 SIMULAČNÍ KNIHOVNA

Simulační model jsem se rozhodl vytvořit v jazyku C++. Samotný jazyk není pro simulace nijak přizpůsoben. Pro vývoj simulačního modelu se v tomto jazyku nabízí dvě možnosti. Buď naprogramovat vše od základu, nebo využít už existující prostředky.

Zvolil jsem druhou variantu. Pro řešení této práce jsem zvolil událostmi řízený model, v kterém bude přenosové médium pasivní. U většiny simulačních knihoven se ukázalo, že by byl problém se souběhem událostí a s detekcí kolizí na sdíleném médiu. Potřebné požadavky splňuje knihovna OMNeT++ [5], kterou jsem se rozhodl při své práci využít. Knihovna podporuje pouze diskrétní simulace, což je pro tuto práci dostačující.

OMNeT++ je objektově orientovaná, modulární, na grafických komponentách založená simulační knihovna s veřejnými zdrojovými kódy. Jednotlivé komponenty modelu spolu komunikují zasíláním zpráv buď prostřednictvím definovaných vstupně/výstupních bran, na které jsou napojeny „spojky“, nebo přímým posíláním zpráv mimo výstupní bránu přímo na vstupní bránu příjemce. V terminologii OMNeT++ se grafické komponenty nazývají také moduly.

Knihovna obsahuje vlastní grafické prostředí pro zobrazení běhu simulace, kompletní simulační jádro pro běh simulace a její řízení, sadu předdefinovaných tříd pro základní prvky modelu, pro načítání parametrů modelu a ukládání výsledků simulace. Je vyvíjena pro potřeby simulací v telekomunikačním průmyslu a datových sítí. Vedle nekomerční verze OMNeT++ existuje i komerční, s názvem OMNEST.

Knihovna OMNeT++ používá pro popis modelu vlastní jazyk *NED* (Network Description), jednotlivým komponentám modelu odpovídají C++ třídy. Jazyk NED se používá pro popis propojení a komunikace jednotlivých komponent, třídy C++ pro definici funkce nejzákladnějších modulů simulačního modelu. Popis jazyka NED viz dokumentace na stránkách [5], ukázka je na obrázku Obrázek 4.

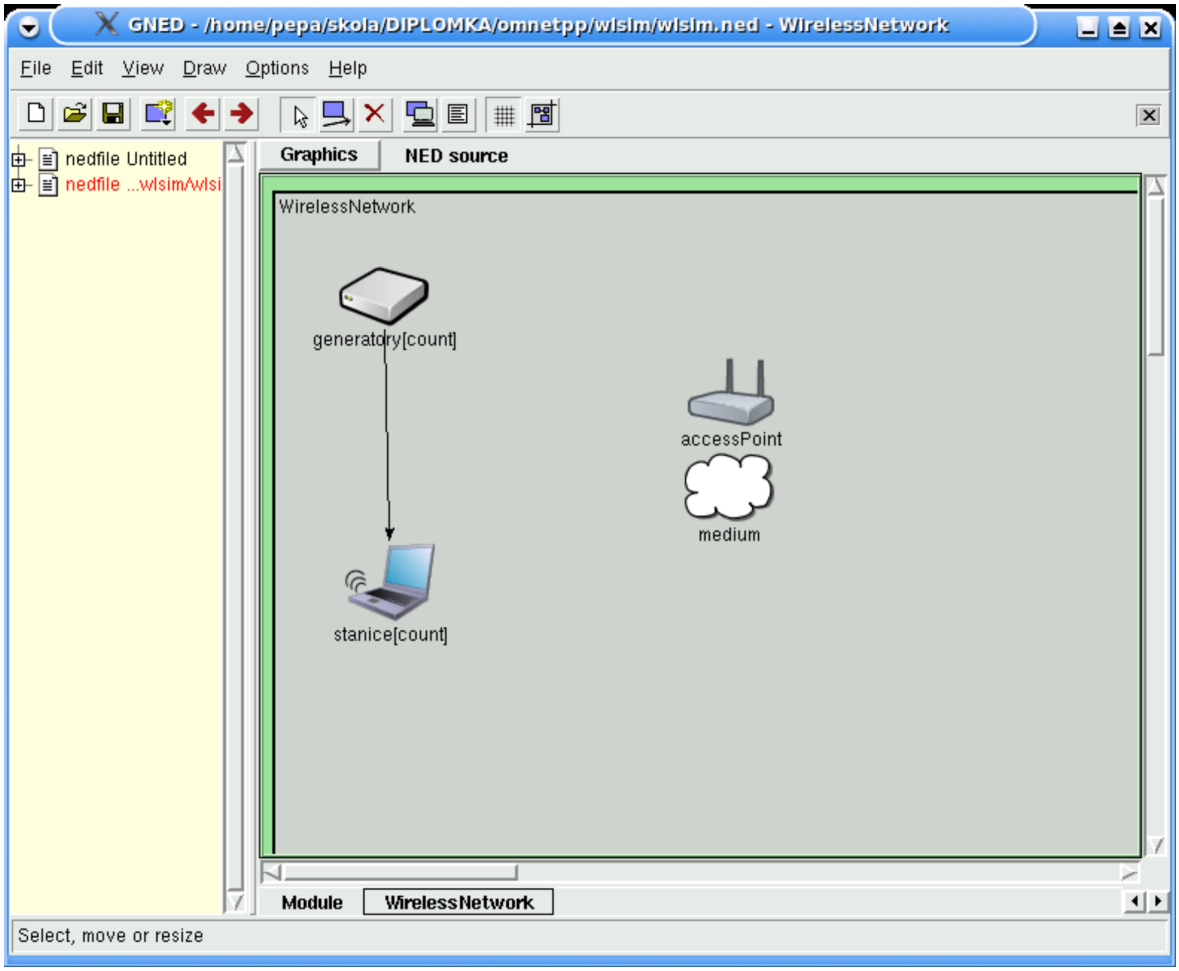
OMNeT++ je přeložitelná na více platformách. Grafické rozhraní přiložených nástrojů využívá jazyka Tcl a jeho grafické nadstavby Tk. Stejně tak grafické prostředí pro běh simulací využívá Tcl/Tk. Model je tedy možné vytvořit a odladit např. v Linuxu a potom ho přenést do Windows. Takový postup jsem použil ve své práci.

4.1 Prostředí OMNeT++

Prostředí OMNeT++ se skládá z několika aplikací. Kromě prostředí pro běh simulace obsahuje grafický editor modelů, aplikaci pro grafické zobrazování výstupních dat a

podpůrné scripty pro sestavení modelů.

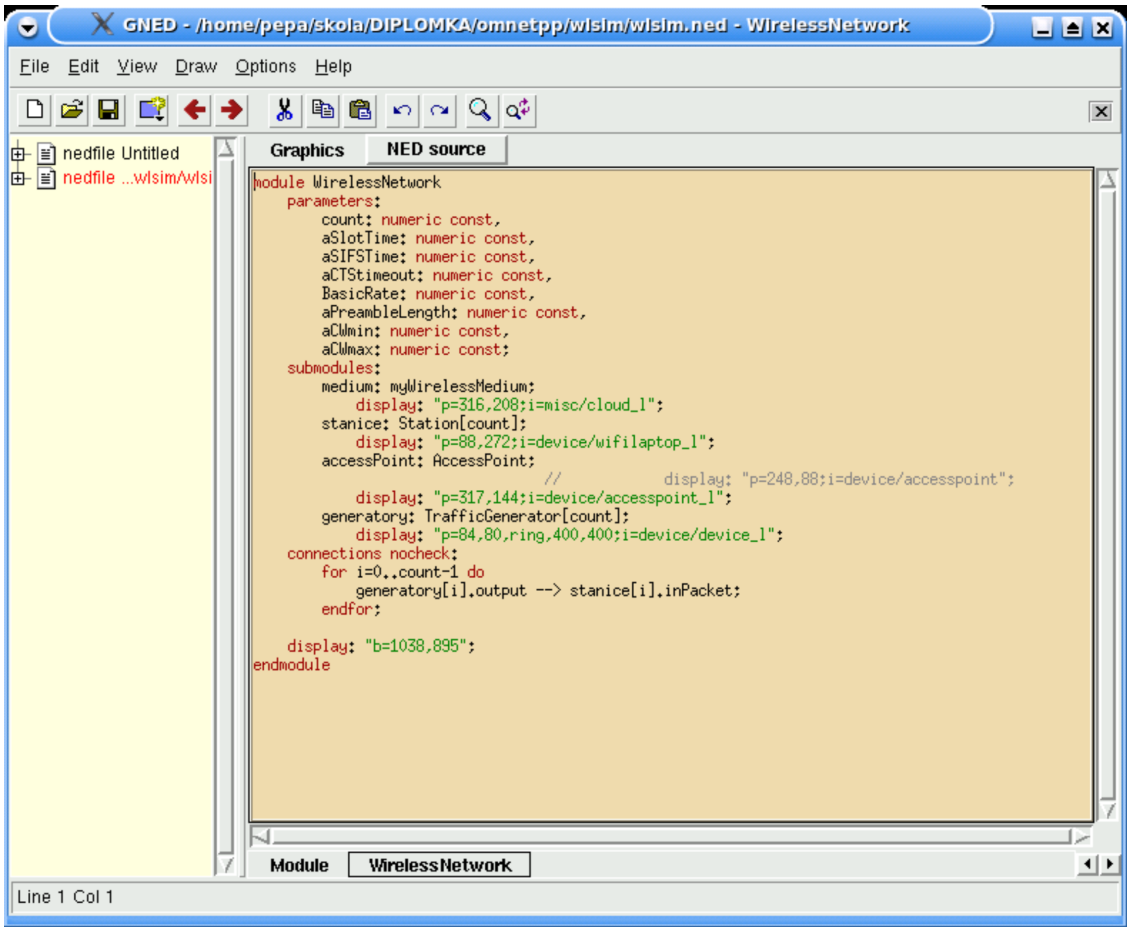
4.1.1 GNED, grafický editor modelu



Obrázek 3.: Prostředí grafického editoru modelu GNED

V prostředí OMNeT++ se simulace sestavuje z jednotlivých komponent. Celý simulační model je popsán jazykem NED. Popis je uložen v textovém souboru, je možné ho editovat běžným ASCII editorem nebo můžeme použít grafický editor GNED. Pomocí grafického editoru sestavíme model, vytvoříme propojení mezi jednotlivými komponentami, upravíme vlastnosti spojení jako směr nebo vlastnosti přenosového kanálu, nastavíme vzhled modelu: umístění komponent, ikony, barvy atd. Editor GNED ukazuje Obrázek 3.

V editoru GNED můžeme také editovat přímo textový tvar popisu modelu. Tuto možnost ukazuje Obrázek 4, který složí zároveň jako ukázka jazyka NED.



Obrázek 4.: Textová editace simulačního modelu

Na obrázku dobře vidíme rozdělení na deklaraci parametrů modulu nebo sítě (síť je hlavní objekt simulačního modelu), případných podřízených modulů s jejich vlastnostmi a popis spojení mezi moduly (komponentami).

4.1.2 PLOVE, zobrazení výstupních dat modelu

Při běhu modelu máme možnost ukládat tzv. výstupní vektory. Jde o dvojici hodnot, kdy jedna hodnota je vždy simulační čas (může se lišit od aktuálního simulačního času) a druhá je libovolná hodnota, kterou chceme uložit. Aplikace PLOVE potom graficky zobrazuje uložené hodnoty v závislosti na čase. Na obrázku je zobrazeno hlavní okno aplikace s výběrem vektorů k zobrazení.

Další obrázek (Obrázek 6) ukazuje zobrazení konkrétních grafů.

4.1.3 Další podpůrné scripty a aplikace

Pro překlad zdrojových kódů modelu a jeho sestavení je nejdůležitější aplikace *nedtool,* která slouží k vygenerování C++ kódu ze souborů s popisem modelu v jazyce NED. Script

opp_makemake slouží k vytvoření souboru Makefile, potřebného pro překlad a sestavení modelu jako samostatné aplikace. Knihovna OMNeT++ umožňuje definovat vlastní strukturu zpráv, které se posílají mezi jednotlivými komponentami. K překladu této struktury ze souborů s popisem zprávy do tvaru C++ třídy slouží script *opp_msgc*.

Pro lepší přehlednost NED definice je možné přímo do souboru s popisem modelu vkládat dokumentaci. K vytvoření konečné dokumentace z této vložené slouží aplikace *opp_neddoc.*

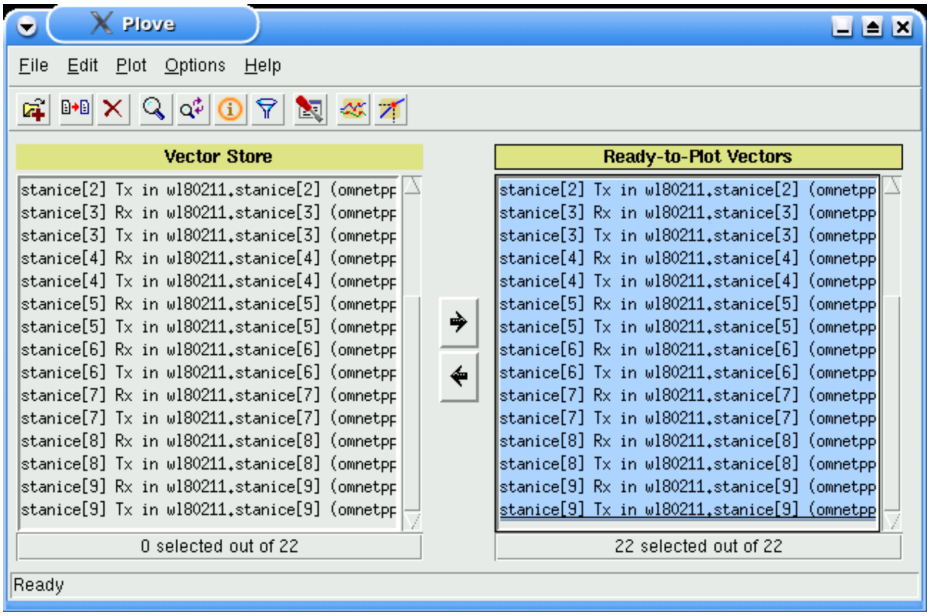
4.1.4 Prostředí pro běh simulace

Ke knihovně OMNeT++ existují dvě prostředí pro běh simulací. Textové prostředí *Cmdenv* pro běh simulací v konzoli a *Tkenv* pro grafické zobrazení simulace. Ve své práci jsem použil grafické prostředí, proto se dále budu zabývat pouze touto verzí.

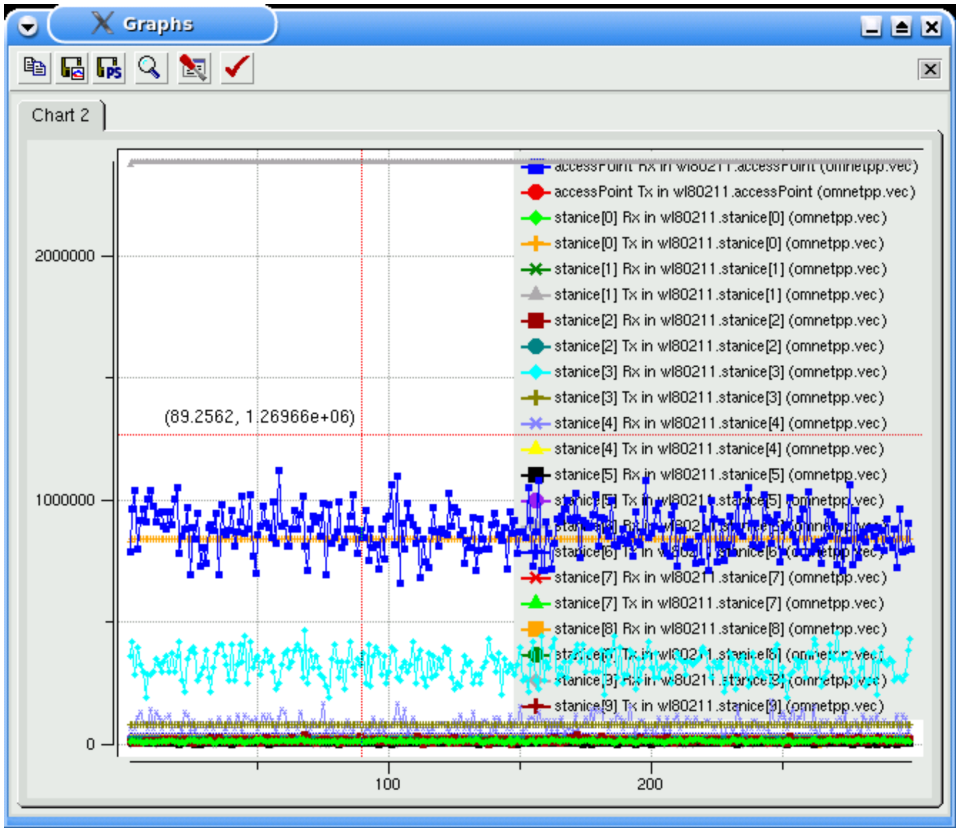
Prostředí pro běh simulace je k aplikaci přilinkováno společně se simulačním jádrem a dalšími částmi OMNeT++. V Linuxu je možnost zvolit statické nebo dynamické linkování, ve Windows je možné pouze statické linkování. Při volbě statických knihoven je ke spustitelnému souboru přilinkována pouze knihovna OMNeT++, další knihovny jako Tcl/Tk je nutné zajistit zvlášť.

Obrázek (Obrázek 7) ukazuje grafické prostředí pro běh simulace. Prostředí Tkenv umožňuje simulaci spouštět různými rychlostmi, zastavovat, krokovat, zjišťovat stav jednotlivých komponent, aktuální hodnoty parametrů atd.

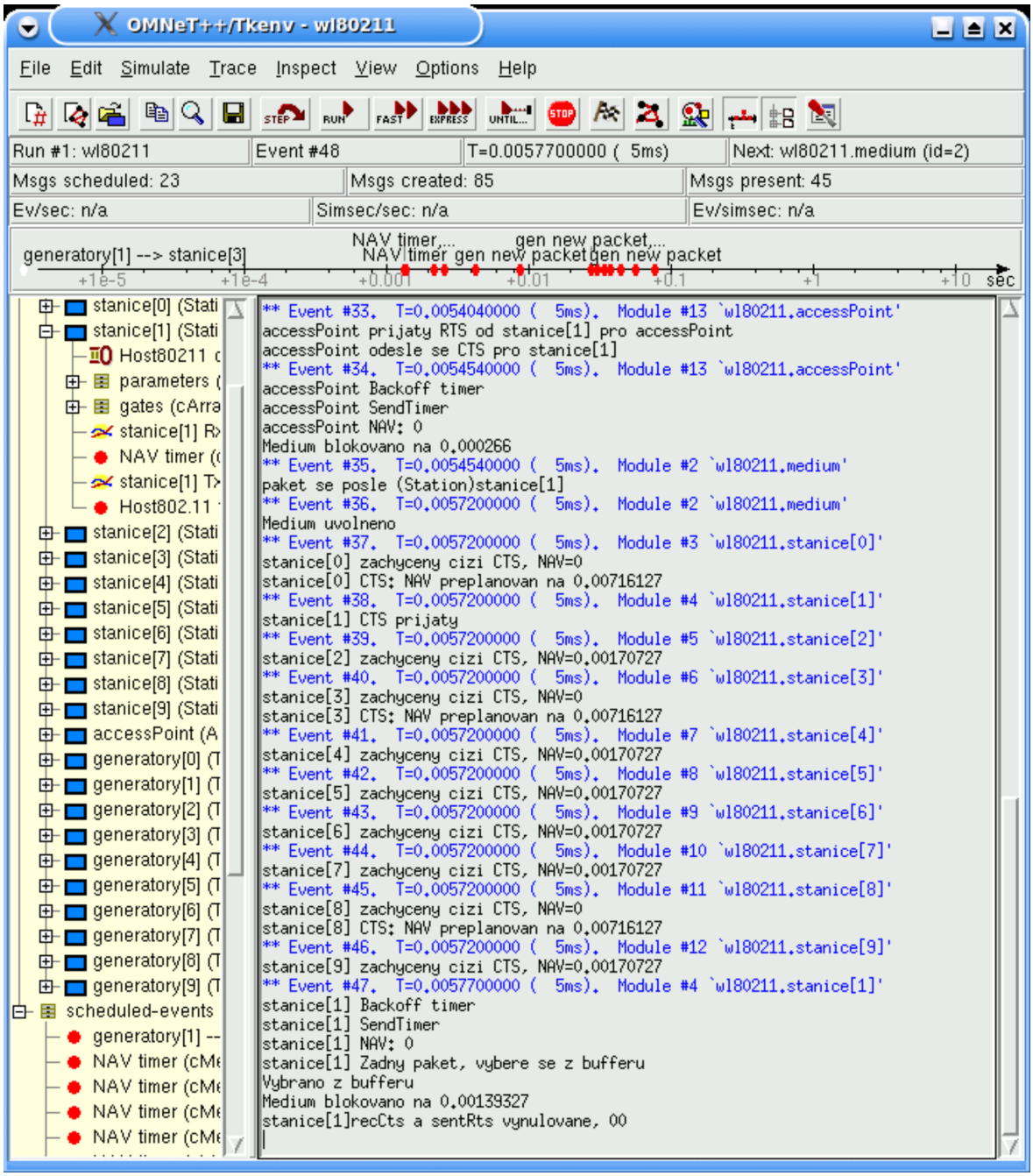
Hlavní okno prostředí se skládá ze tří částí. V horní části je menu a nástrojová lišta. Pomocí tlačítek se šipkami a tlačítka Stop ve střední části nástrojové lišty se ovládá běh simulace.



Obrázek 5.: Hlavní okno aplikace Plove



Obrázek 6.: Ukázka konkrétního grafu



Obrázek 7.: Hlavní okno prostředí Tkenv

Pod lištou jsou zobrazeny stavové informace: kterou konfiguraci sítě máme spuštěnu, kolikátá událost právě probíhá, jaký je aktuální simulační čas, která komponenta bude zpracovávat příští událost, kolik je do budoucnosti naplánovaných událostí, kolik celkově existuje v modelu zpráv v daném okamžiku a parametry, které dávají přehled o rychlosti průběhu simulace.

Pod stavovými informacemi je zobrazena časová osa s vyznačením příštích událostí a s výpisem názvů těchto událostí.

Pod časovou osou je okno rozděleno na dvě části. V levé části je výpis všech komponent

modelu s možností sledovat aktuální obsah parametrů, zprávy zpracovávané komponentou, obsah front zpráv atd. Pod výpisem komponent jsou vypsány plánované události (zprávy), které už zobrazuje časová osa. Zde je možné podrobně prohlížet strukturu a obsah těchto zpráv.

V pravé a největší části je podrobně zapsán průběh simulace. Simulační jádro knihovny OMNeT++ zde vypisuje minimálně číslo události, simulační čas, číslo a název modulu, který danou událost zpracovává. Dále se do tohoto okna vypisují uživatelské výstupy jednotlivých komponent, sloužící pro další upřesnění průběhu simulace nebo pro ladění modelu.

4.1.5 Komunikace mezi jednotlivými komponentami OMNeT++

Podrobný manuál knihovny OMNeT++ je na domovské stránce [5]. Jednou z nejdůležitějších vlastností knihovny, používané při simulaci datových sítí, je předávání zpráv mezi jednotlivými komponentami. Simulační model, který je součástí této práce, je postaven na zasílání zpráv mezi komponentami, proto se v tomto odstavci budu věnovat zasílání zpráv podrobněji.

Každá komponenta, která má komunikovat s jinými komponentami, je vybavena minimálně jednou branou. Každá brána je buď vstupní, nebo výstupní a jsou definovány v popisu modelu v jazyku NED. Brány jednotlivých komponent spolu mohou být propojeny. Potom je zpráva odeslaná výstupní branou doručena na vstupní bránu jiné komponenty, na kterou je výstupní brána připojena. Spojení je možné pouze mezi právě dvěma branami, z nichž jedna musí být výstupní a jedna vstupní. Zpráva může mezi branami putovat přenosovým kanálem. Zpráva může mít nastavenu velikost, přenosový kanál může mít nastavenu přenosovou rychlost. Zpráva potom dorazí k cíli se zpožděním, odpovídajícím její velikosti a přenosové rychlosti kanálu.

Kanál také může mít nastavenu chybovost. Zpráva s sebou nese příznak chybného přenosu. Na základě chybovosti kanálu se s odpovídající pravděpodobností nastavuje chybový příznak přenášené zprávy. Zpráva se příjemci doručí vždy, je potom na příjemci, jak naloží se zprávou s nastaveným chybovým příznakem.

Způsob propojení bran komponent je ideální pro simulace point-to-point spojů. Pro simulaci bezdrátové sítě se však nehodí. V modelu, který je součástí této práce, je zapotřebí z jediné výstupní brány odesílat zprávy několika komponentám. Proto podporuje knihovna OMNeT++ ještě druhý způsob předávání zpráv mezi komponentami. Zpráva se nepředává pro odeslání výstupní bráně, ale předá se ke zpracování přímo vstupní bráně cílové

komponenty. Tímto způsobem lze předávat vstupní bráně zprávy od několika komponent bez jakéhokoliv pevného propojení, což lépe odpovídá charakteru simulované bezdrátové sítě.

Při použití přímého posílání zpráv není možné využít přenosu kanálem s definovanými vlastnostmi, který byl popsán výše. Ve vytvářeném modelu budeme po přenosovém kanálu vyžadovat další vlastnosti, kterými standardní kanál nedisponuje. Přenos „reálným“ bezdrátovým prostředím tedy budu řešit vlastními prostředky.

4.1.6 Zpracování událostí

Každá komponenta, která je schopna přijmout zprávu, obsahuje virtuální funkci `handleMessage (cMessage* msg)`, které je simulačním jádrem předána doručená zpráva. Tato funkce je prakticky nejdůležitější pro vytváření komponent s námi požadovanými funkcemi.

Pro simulaci komunikačních protokolů je důležité přesně měřit časové intervaly, čekat přesně definovanou dobu a podobně. Knihovna OMNeT++ nemá třídu pro časovač, který by se hodil k těmto účelům. Je však možné, aby libovolná komponenta vytvořila zprávu a naplánovala na potřebný okamžik její odeslání sobě samé. Zpráva je předána ke zpracování funkci handleMessage, stejně jako jakákoliv jiná zachycená zpráva. Ve funkce handleEvent můžeme použít funkce OMNeT++ pro zjištění, jestli jde o zprávu poslanou sobě samým, nebo o zprávu, která byla doručena od jiné komponenty. Je také možné zjistit, přes kterou bránu byla zpráva doručena, od kterého odesílatele a podobně.

II. PRAKTICKÁ ČÁST

5 MODEL BEZDRÁTOVÉ POČÍTAČOVÉ SÍTĚ S PROTOKOLEM 802.11

Model je vytvořen v jazyce C++ s využitím standardních knihoven a knihovny pro diskrétní simulace OMNeT++. Je tvořen těmito základními komponentami (moduly):

- TrafficGenerator
- Station
- AccessPoint
- WirelessMedium
- datový paket

5.1 Stručný popis principu činnosti modelu

Modelový provoz je generován modulem TrafficGenerator. Tento generátor odesílá modelová data připojené stanici na zvláštní vstupní bránu, která slouží pouze pro příjem takto generovaných paketů. Pakety jsou generovány podle parametrů, nastavitelných pro každý generátor zvlášť.

Data jsou ze stanic odesílána access pointu, modulu AccessPoint, přes sdílené bezdrátové médium. Toto médium je v modelu zastoupeno modulem WirelessMedium, který obstarává nejzákladnější funkce přenosového média. Zajišťuje zpoždění paketů, předání paketů příjemci, blokování média během přenosu, rozesílání „broadcastových“ paketů a také se stará o vznik kolizí.

Moduly Station, stejně jako AccessPoint, obstarávají největší část komunikace. Provádí odesílání a příjem dat, starají se o funkce MAC vrstvy protokolu 802.11 a ukládají data pro pozdější vyhodnocení simulace.

Každá stanice odesílá data modulu AccessPoint, bez ohledu na příjemce. AccessPoint si udržuje tabulku připojených stanic (obdoba tabulky asociovaných stanic u reálného access pointu) a na jejím základě předává data konečným příjemcům.

Datový paket v sobě nese adresovací informace, má definován typ dat, velikost, rychlost, kterou se přenáší, délku v čase a další informace, potřebné pro některé typy dat.

5.2 Hierarchie tříd a jejich činnost

Při tvorbě hierarchie tříd modelu jsem vycházel převážně z existujících tříd knihovny

OMNeT++. V této kapitole uvedu nejdůležitější třídy, ze kterých se skládá model.

5.2.1 TrafficGenerator (soubor TrafficGenerator.cc)

Tato třída je potomkem třídy cSimpleModule, která je součástí OMNeT++ a v modelu nemá žádné další potomky. Třída TrafficGenerator slouží ke generování paketů konstantní velikosti, konstantní rychlostí a stále stejnému příjemci.

K dodržení intervalu generování paketů používá třída časovač *event*. Interval plánování tohoto časovače se určí z parametru rate jako jeho převrácená hodnota. Parametr rate tedy udává počet vygenerovaných paketů za jednu sekundu. Při doručení této události obslužné funkci handleMessage se vytvoří nový paket, nastaví se jeho velikost, typ a cílová stanice. Pro přehlednost se nastaví také jméno objektu zprávy složené z názvu zdrojového modulu a cílové stanice. Takto vytvořený paket se odešle připojené stanici a naplánuje se časovač pro vytvoření příštího paketu.

5.2.2 WlPacket (soubor packet.msg)

Knihovna OMNeT++ obsahuje mechanismus pro definování struktury zpráv. Tohoto mechanismu jsem využil při definování struktury paketu, který se přenáší bezdrátovou sítí. Tato třída nezpracovává žádné události, slouží jenom k uchování a přenosu dat.

Každý objekt (instance třídy) má v knihovně OMNeT++ své jedinečné identifikační číslo a svoje jméno. Tyto údaje mohou dostatečně identifikovat každý objekt v modelu. Nezaváděl jsem tedy pro stanice další údaj, který by sloužil pro adresování. Z důvodu přehlednosti konfigurace modelu jsem použil jména stanic jako adresy. Z tohoto důvodu jsou v třídě WlPacket proměnné, sloužící pro adresaci, typu string. Dále budu jména stanic ve vztahu k adresování nazývat adresy.

Bezdrátový paket WlPacket obsahuje celkem čtyři adresy, dvě zdrojové a dvě cílové. Jde o adresování na dvou různých úrovních, důvod bude popsán v odstavcích o třídách AccessPoint a Station.

Třída WlPacket dále obsahuje údaje o časové délce paketu, čase odeslání, čase, který se vyžaduje pro rezervaci média, rychlosti, kterou se má paket přenášet, a sekvenční číslo paketu. Další potřebné vlastnosti získala třída WlPacket děděním z třídy OMNeT++ cMessage.

Pro zjednodušení modelu jsem zavedl použití pouze jednoho typu paketu, který obsahuje všechny položky potřebné ve všech simulovaných typech reálných paketů. Na rozdíl od

reálného zařízení nezávisí délka paketu na přenášených datech, ale lze ji nastavovat libovolně. Po všechny typy paketů tedy model používá tuto jedinou třídu, pro každý typ paketu nastaví potřebné položky a ostatní zůstanou nevyužity. Velikost každého paketu se nastaví taková, jakou má reálný paket potřebného typu. Například RTS paket má nastavenu velikost 20 oktetů a s takovou velikostí se v modelu počítá, i když nese prostor pro data o sekvenčním čísle, které je určeno pro měření odezvy.

5.2.3 myWirelessMedium (WirelessMedium.cc)

Tato třída je v modelu zavedena jako náhrada standardního mechanismu přenosových kanálů v OMNeT++. Pro náhradu tohoto mechanismu jsem se rozhodl z důvodu nemožnosti vzniku kolizí a hlavně detekce stanic, které mohou kolizi způsobit, na standardním kanále. Další důvod je ten, že standardní přenosový kanál je přizpůsoben pro jednosměrné pevné spojení dvou bran různých modulů. Pro model bezdrátové sítě, kde je stanic více, je tento způsob nevyhovující.

Tato třída je potomkem třídy cSimpleModule, z hlediska OMNeT++ tedy jde o běžný modul. Pro zajištění funkčnosti jako přenosové médium poskytuje funkce Block, Release a Stav. Tyto funkce slouží k blokování média, jeho uvolnění a zjištění stavu. Pokud je médium zablokované, znamená to, že probíhá přenos dat a dalším požadavkem na přenos by došlo ke kolizi.

Třída myWirelessMedium také pomáhá se simulací problému skrytých stanic, kdy vrací jiný stav pro skryté stanice a jiný pro ty, které skryté nejsou. Pokud je médium blokováno vysíláním některé stanice, je pro dotazy ostatních stanic vrácen takový stav, který by v reálném prostředí skutečně detekovaly. Pokud je médium blokováno vysíláním skryté stanice, je na dotazy ostatních stanic vrácen stav volno, potom při vysílání dojde ke kolizi tak jako v reálném prostředí. Naopak skrytá stanice má v dohledu pouze přístupový bod a ostatní stanice jsou mimo její dohled.

Pokud dojde ke kolizi, je nutné určit, jak dlouho má být médium ještě blokováno. Jak jsem uvedl v kapitole 3, jakmile začne stanice vysílat, není schopná detekovat kolizi. To znamená, že vysílání proběhne až do konce i v případě kolize. Je tedy možné, že v případě kolize může dojít v době mezi vznikem kolize a ukončením vysílání všech kolidujících stanic k další kolizi. Pro ošetření této situace je médiem v případě detekce prodloužena doba blokování média až do ukončení vysílání poslední z kolidujících stanic.

V reálném prostředí je každý paket zachytitelný každou stanicí v dohledu odesílatele. V našem případě však každá stanice, která zachytí paket, který není určen pro ni, zahodí. Pro

lepší přehlednost grafického znázornění simulace je paket médiem doručen jen těm stanicím, které by ho nezahodily. To znamená, že datové pakety jsou doručovány jen příjemci, „broadcastové“ pakety RTS a CTS jsou doručovány všem stanicím v dohledu.

Vlastnosti, popsané v této kapitole, vyplývají z fyzikální podstaty reálného bezdrátového přenosového prostředí. Pro simulaci bylo třeba přidat třídě myWirelessMedium funkce, které reálné médium nemá, aby bylo dosaženo stejných vlastností. Na rozdíl od reálného média je simulované médium vybaveno určitou rozhodovací schopností v tom, které stanici bude daný paket doručen a jak se bude které stanici odpovídat na dotaz o blokování média.

Pro dosažení těchto vlastností si třída myWirelessMedium vede, podobně jako přístupový bod, tabulku obsluhovaných bezdrátových stanic. Na rozdíl od přístupového bodu jsou však v této tabulce zanesena všechna zařízení v síti, tedy se všemi stanicemi i přístupový bod. Při startu simulace se všechna zařízení „asociují“ k médiu, které si na základě tohoto procesu vytvoří tabulku všech stanic. Tato asociace k médiu v reálných podmínkách neexistuje.

5.2.4 Host80211 (Host80211.cc)

Tato třída je pro simulaci MAC protokolu sítí 802.11 spolu s předchozí třídou myWirelessMedium velmi důležitá, protože implementuje mechanismus přístupu k médiu a ošetřuje servisní pakety.

Je potomkem třídy cSimpleModule, slouží jako rodičovská třída pro třídy Station a AccessPoint.

Při inicializaci objektu této třídy nebo odvozené třídy zajišťuje asociaci k médiu, zmíněnou v předchozí kapitole, načtení konkrétních hodnot parametrů zvoleného protokolu a vytvoření potřebných objektů pro vnitřní časovače.

Model počítá s tím, že požadavky na odesílání dat generovaných objektem třídy TrafficGenerator mohou přicházet rychleji, než mohou být odesílány. Pro tento případ obsahuje třída Host80211 vnitřní frontu, do které se řadí data k odeslání v případě, že je není možné ihned předat k odeslání. Tato fronta je typu LIFO.

V modelu se nevyskytují objekty třídy Host80211. Tato třída zpracovává pro své potomky pouze vnitřní časovače a servisní pakety, data zpracovávají odvozené třídy.

Požadavek na odeslání paketu je objektu této třídy předán prostřednictvím funkce `SendPacket (WlPacket *packet)`. Tato funkce vloží paket do fronty k odeslání a nastaví vnitřní časovače tak, aby byl dodržen protokol přístupu k médiu.

Prakticky všechny další činnosti, spojené s protokolem 802.11, vykonává funkce `handleMessage (cMessage *msg)`. Ta zachycuje jak události, generované třídou samotnou (vnitřní časovače), tak události související s doručením paketu. Jako první ošetřuje funkce handleMessage vnitřní časovače, které souvisí s vypršením některého z timeoutů.

Potom už jsou zpracovávány přijaté pakety. Nejdříve se testuje, jestli je přijatý paket RTS, a pokud ano, jestli je adresován dané stanici, nebo jiné. Pokud je adresován stanici, která provádí test, je po splnění dalších podmínek vygenerována odpověď CTS a připravena k odeslání. Pokud je RTS adresován jiné stanici, je ošetřena změna NAV, viz kapitola 3.2.5. Dále je testován přijatý paket, jestli je CTS, a když ano, pro kterou stanici. Pokud je pro danou stanici, připraví se k odeslání paket, na základě kterého byl vytvořen RTS a na který je v tomto kroku přijaté CTS odpovědí. Pokud je CTS určen jiné stanici, zkoumá se, jestli se má upravit nastavení NAV.

V další části funkce handleMessage se opět zpracovávají zbývající vnitřní časovače. V této části funkce už z vnitřních časovačů zbývají pouze časovače, související s odesíláním dat. Ošetří se detekce volného média (podle zásad uvedených na konci kapitoly 5.2.3) a pokud není připraven žádný paket k odeslání, vybere se z fronty. Otestuje se, jestli je velikost paketu větší než nastavený práh pro použití mechanismu RTS/CTS. V modelu se zvlášť nenastavuje, jestli má daná stanice používat RTS/CTS. Pokud stanice tento mechanismus používat nemá, nastaví se práh takový, aby nikdy nedošlo k jeho překročení.

Jestliže se zjistí, že velikost paketu je větší než práh RTS/CTS, paket vyzvednutý z fronty se vrátí na její *začátek* a jako paket připravený k odeslání se vygeneruje RTS požadavek. Po přijetí odpovídajícího CTS je paket znovu vyzvednutý z fronty a odeslán příjemci.

5.2.5 Station (Station.cc)

Třída Station je odvozena od třídy Host80211. Navíc z konfiguračního souboru modelu načítá parametry *rate*, *hidden*, *errTx* a *errRx*.

- **rate:** rychlost, na které stanice komunikuje. V síti se mohou vyskytovat různé rychlosti, proto je tento parametr zadán pro každou stanici zvlášť.
- **hidden:** určuje, jestli je stanice v dohledu ostatních stanic nebo ne
- **errTx** a **errRx:** chybovost příjmu a odesílání, v modelu parametry pro každou stanici zvlášť

Po načtení parametrů stanice se vytvoří paket s požadavkem na asociaci a odešle se přístupovému bodu. Protože je vytvářený model navržen pro analýzu provozu, proces

asociace je zjednodušen o proces hledání bodu. Podrobně tuto fázi zkoumá např. [4]. Inicializace stanice končí nastavením objektů pro výstup údajů o přijatých a odeslaných datech.

Další činnost je prováděna funkcí handleMessage. Ta předává události, související s časováním, funkci handleMessage rodičovské třídy. Tím je zajištěna obsluha událostí, souvisejících s MAC vrstvou sítí 802.11.

Vlastní zpracování událostí, souvisejících s třídou Station, začíná uložením objemů prošlých dat v obou směrech. Data se ukládají jednou za sekundu simulačního času, výstupní grafy po zobrazení aplikací Plove (kap. 4.1.2) tedy udávají propustnost přímo v bps (bitech za sekundu).

Stanice rozlišuje přijaté pakety od paketů k odeslání podle toho, kterou branou modulu paket (zpráva) přišel. V režimu sítě AP to klient probíhá komunikace tak, že stanice se asociuje k přístupovému bodu a komunikuje pouze s ním. Všechny odesílané pakety jsou tedy na nižší úrovni adresovány přístupovému bodu a až na vyšší úrovni cílové stanici. Z přijatých paketů se RTS a CTS opět předají ke zpracování rodičovské třídě.

Po úspěšném přijetí paketu se vynulují příznaky čekání na RTS a CTS. Běžné datové pakety se dále nezpracovávají, proto se zahodí. Dále se zpracovávají pouze požadavky a odpovědi na odezvu (ping).

Měření odezvy je v modelu vyřešeno podobně jako v protokolu ICMP. Při odeslání požadavku na odezvu poznamená odesílatel do paketu čas odeslání. Cílová stanice požadavek přijme a vytvoří odpovídající odpověď. Do této odpovědi zkopíruje čas odeslání z přijatého požadavku a paket odešle zpět původnímu odesílateli. Příjemce přečte ze zachycené odpovědi na svůj požadavek čas odeslání původního požadavku, odečte ho od aktuálního (simulačního) času a tím dostane dobu odezvy. Díky přenosu času odeslání požadavku není nutné vést tabulku odeslaných paketů vyhledávat ke kterému patří přijatá odpověď atd. Paket s požadavkem na odezvu nese také sekvenční číslo. To slouží k detekci případných ztrát paketů a vícenásobných odpovědí na jeden požadavek. V modelu není vznik vícenásobných odpovědí možný, není tedy nijak ošetřen.

5.2.6 AccessPoint (AccessPoint.cc)

Každá bezdrátová síť v režimu infrastructure obsahuje právě jeden přístupový bod. Ten musí být v dohledu *všech* ostatních stanic v síti. Vzhledem k přístupovému bodu tedy není žádná stanice skrytá. Z toho vychází i způsob modelování přenosového média (kap. 2.1).

Modelový přístupový bod udržuje tabulku asociovaných stanic, na rozdíl od reálného přístupového bodu, za účelem převodu modelové adresy (jména stanice) na ukazatel na ni. Ukazatele jsou nutné pro přímé posílání zpráv (paketů) stanicím funkcí OMNeT++ sendDirect().

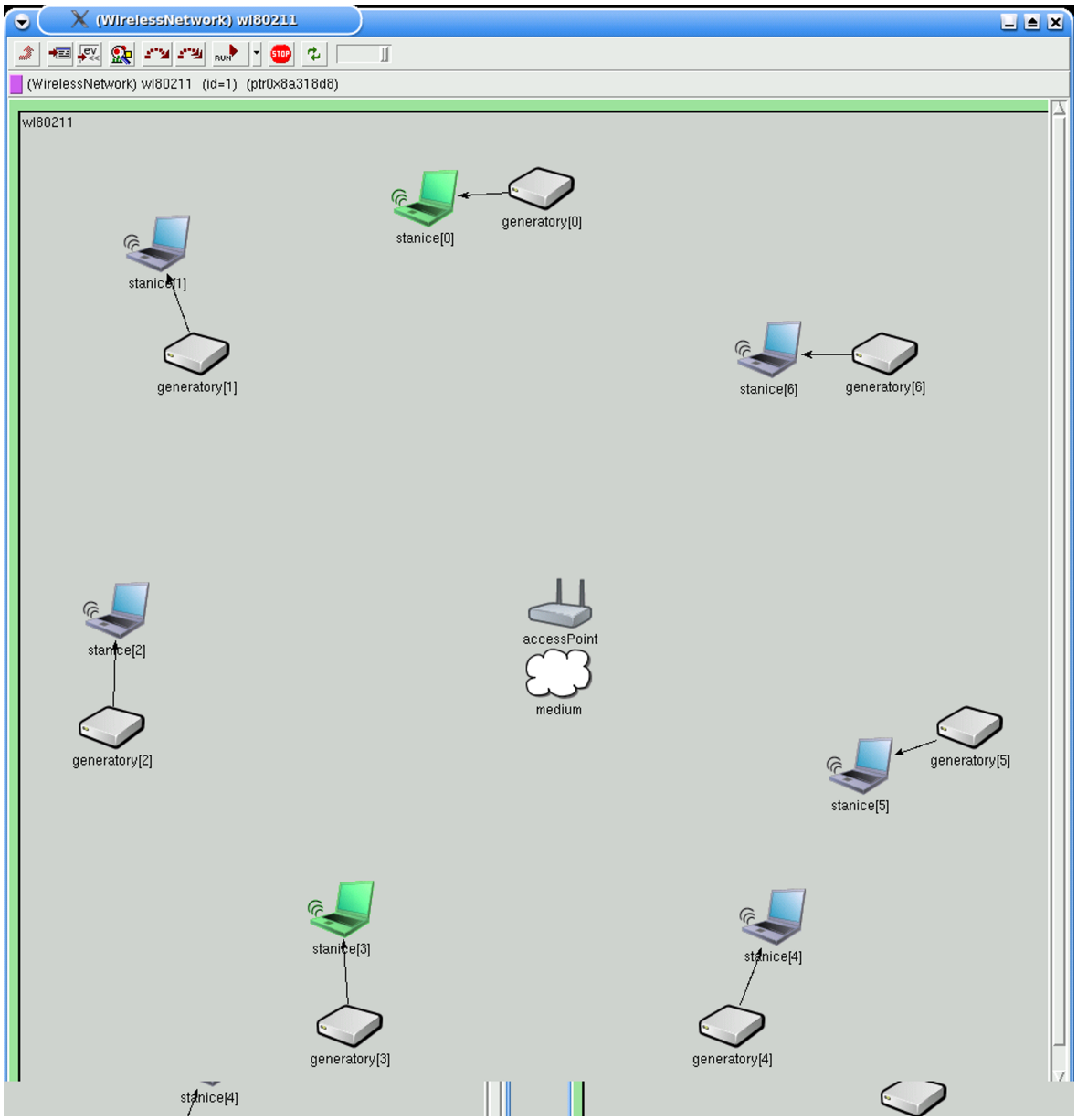
Třída AccessPoint je také odvozena od třídy Host80211. AccessPoint také předává události, související s potokolem MAC protokolem 802.11 třídě Host80211. Vlastní činnost třídy AccessPoint spočívá v přidání odesílatele do tabulky při zachycení asociačního paketu a v předávání paketů konečným příjemcům správnou přenosovou rychlostí na základě této tabulky. Třída AccessPoint také ukládá informace o celkově prošlém provozu stejným způsobem jako třída Station.

6 POUŽITÍ MODELU

Model je tvořen aplikací *wlsim* (wlsim.exe ve Windows). Popis modelu v jazyce NED je uložen v souboru wlsim.ned. Tento soubor obsahuje popis struktury modelu, parametry jednotlivých modulů a částečně vzhled modelu v grafickém prostředí pro běh simulace.

Vlastní konfigurace modelu je uložena v souboru omnetpp.ini. Tento soubor se skládá ze tří základních částí. V sekci [General] jsou parametry pro prostředí běhu, v sekci [Parameters] jsou hodnoty jednotlivých parametrů pro každou stanici a pro celý model a na konci souboru v sekcích [Run 1], [Run 2] a [Run 3] jsou parametry fyzické vrstvy pro jednotlivé verze protokolu 802.11a, 802.11b a 802.11g.

Pro splnění požadavku nastavitelného počtu stanic v síti jsem zvolil jejich uložení v jednorozměrném poli (vektoru). Od toho se odvíjí jména stanic. Stejný způsob uložení i pojmenování jsem zavedl pro generátory provozu. Parametry jsou v konfiguračním souboru tedy zadány ve tvaru: wl80211.stanice[0].rate=11000000. V simulacích s OMNeT++ musí být moduly sdruženy do sítě. Ta je potom hlavní objekt simulace. V případě mého modelu se tato síť jmenuje wl80211, proto kompletní název každého parametru začíná tímto řetězcem. Další část názvu, stanice[0], je vlastní název konkrétního modulu. V hranatých závorkách je index. Modul potom vystupuje v simulaci pod jménem stanice[0], v řetězci jsou uloženy i hranaté závorky s indexem. Proto je v konfiguraci generátorů provozu uveden název (adresa, viz kap. 5.2.2) cílové stanice také v tomto tvaru. Poslední část kompletního názvu parametru tvoří název konkrétního parametru modulu, který musí být definován v NED popisu.



Obrázek 8.: Grafické zobrazení modelu bezdrátové sítě

Po spuštění aplikace modelu se zobrazí hlavní okno prostředí Tkenv (kap. 4.1.4 a Obrázek 7 na straně 27) a okno s grafickým zobrazením modelu (Obrázek 8). Ve středu modelu je umístěn přístupový bod spolu s modulem bezdrátového přenosového média. Kolem nich jsou rozloženy stanice a generátory provozu. Prostředí Tkenv se je snaží rozložit do kruhu, při příliš velkém počtu stanic hrozí nepřehlednost zobrazení. Skryté stanice jsou pro přehlednost zobrazeny jako zeleně podbarvené.

Spuštění a ovládání simulace viz kapitola 4.1.4, zobrazení výsledků je popsáno v kapitole 4.1.2 na straně 24. Po ukončení simulace je nutné na otázku prostředí, jestli se mají zavolat funkce finish(), odpovědět kladně. Volání těchto funkcí je důležité ke korektnímu ukončení simulace a k uložení všech výstupních dat.

7 EXPERIMENTY SE SIMULAČNÍM MODELEM

Podle zadání budu ověřovat chování simulované sítě v různých konfiguracích modelu. Zaměřím se na vliv skrytých stanic a nastavení RTS prahu. Budu sledovat propustnost, odezvu (ping) a odchylka odezvy (jitter).

Aktuální propustnost pro oba směry jednotlivých stanic je modelem ukládána do souboru omnetpp.vec způsobem, popsaným v kapitole 5.2.5. K zobrazení grafů propustnosti slouží aplikace Plove (kap. 4.1.2).

Odezva je při generování požadavků na ping vypisována do hlavního okna prostředí a také zaznamenávána k zobrazení, stejně jako propustnost. Na rozdíl od ní však při každém přijetí odezvy. Velikost paketu pro měření použijeme 64 bytů, což je běžná velikost.

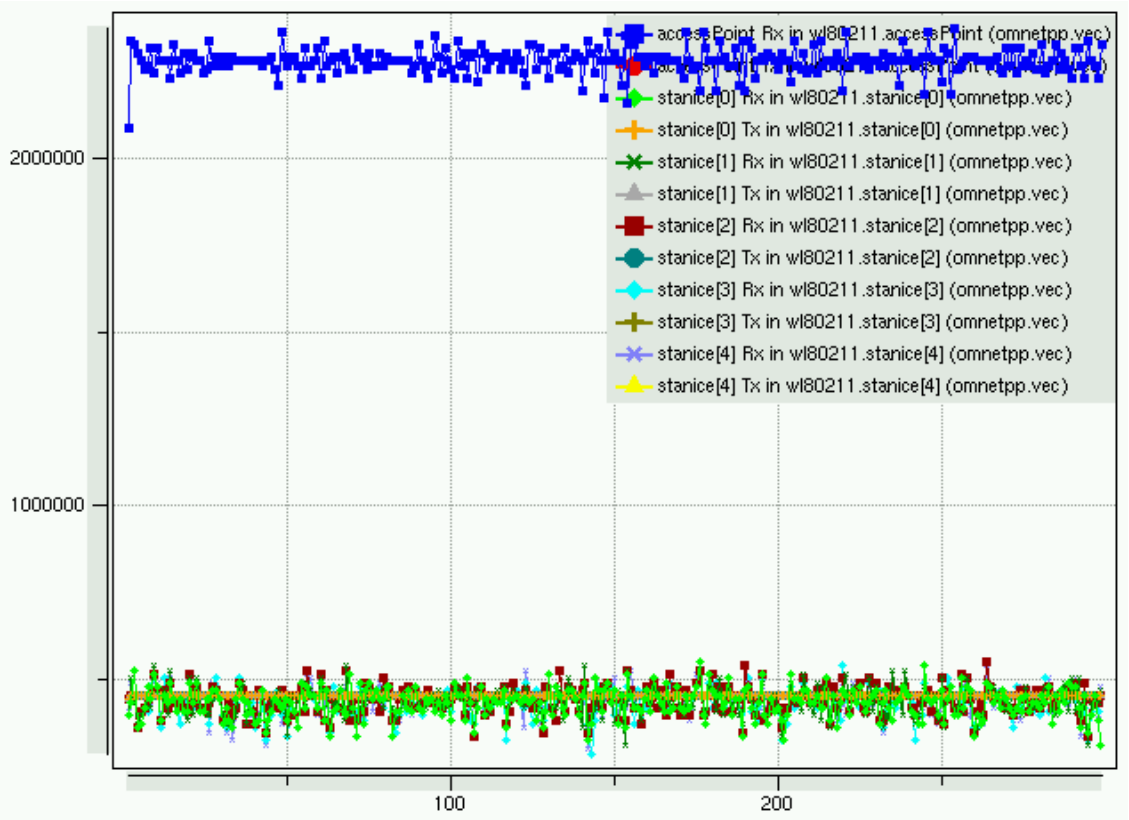
Jitter je podle [6] definován jako rozptyl naměřených hodnot odezvy za určitý čas. Pro jeho zjištění využijeme funkcí tříd pro sběr a zpracování statistických dat knihovny OMNeT++.

7.1 Sledování propustnosti, odezvy a jitteru

Tyto veličiny budeme sledovat na síti, ve které bude k přístupovému bodu připojeno 5 stanic. Provoz mezi stanicemi bude rovnoměrně rozložen tak, abychom se přiblížili saturaci [7]. Vyhodnocovat budeme průměrné hodnoty sledovaných veličin po nasimulování 5 minutového provozu. Simulace budeme provádět pro protokoly 802.11a, 802.11b a 802.11g, stanice budou komunikovat v jednom případě na nejvyšší přenosové rychlosti daného protokolu.

7.1.1 Protokol 802.11b

Experimenty začneme bez výskytu skrytých uzlů, s vypnutým RTS/CTS. Experimentálně jsem došel k tabulce Tabulka 1, která popisuje konfiguraci modelu při dosažení saturace sítě. Výsledný simulovaný provoz je zobrazen na obrázku číslo 9. Na obrázku je modře zobrazen provoz přes přístupový bod v jednom směru. K němu patří ještě provoz v druhém směru, jehož křivka je překryta modrou. Takový provoz přes přístupový bod je proto, že každá stanice odešle data nejprve jemu, přístupový bod je potom odesílá dál. V modelu podle Tabulky 1 je provoz přes přístupový bod v každém směru asi 2,3 Mbps, což je celkem 4,6 Mbps. Na obrázku 9 je dále vidět provoz generovaný jednotlivými generátory, několik rovných čar přes sebe v úrovni 445 kbps (tak jako je požadovaný provoz v Tabulce 1). Ten je částečně překryt skutečně přijímaným provozem jednotlivých stanic.



Obrázek 9.: Provoz při dosažení saturace

V síti 802.11b v režimu infrastructure tedy reálně přeneseme celkový provoz asi 4,6 Mbps při komunikaci všech stanic přenosovou rychlostí 11 Mbps, při stejné velikosti paketů a stejné rychlosti odesílání požadavků.

Při zapnutí mechanismu RTS/CTS pro všechny stanice i přístupový bod klesne průměrná hodnota přijímaného provozu pro jednotlivé stanice na 325 až 170 kbps ze 445 kbps v předchozím případě.

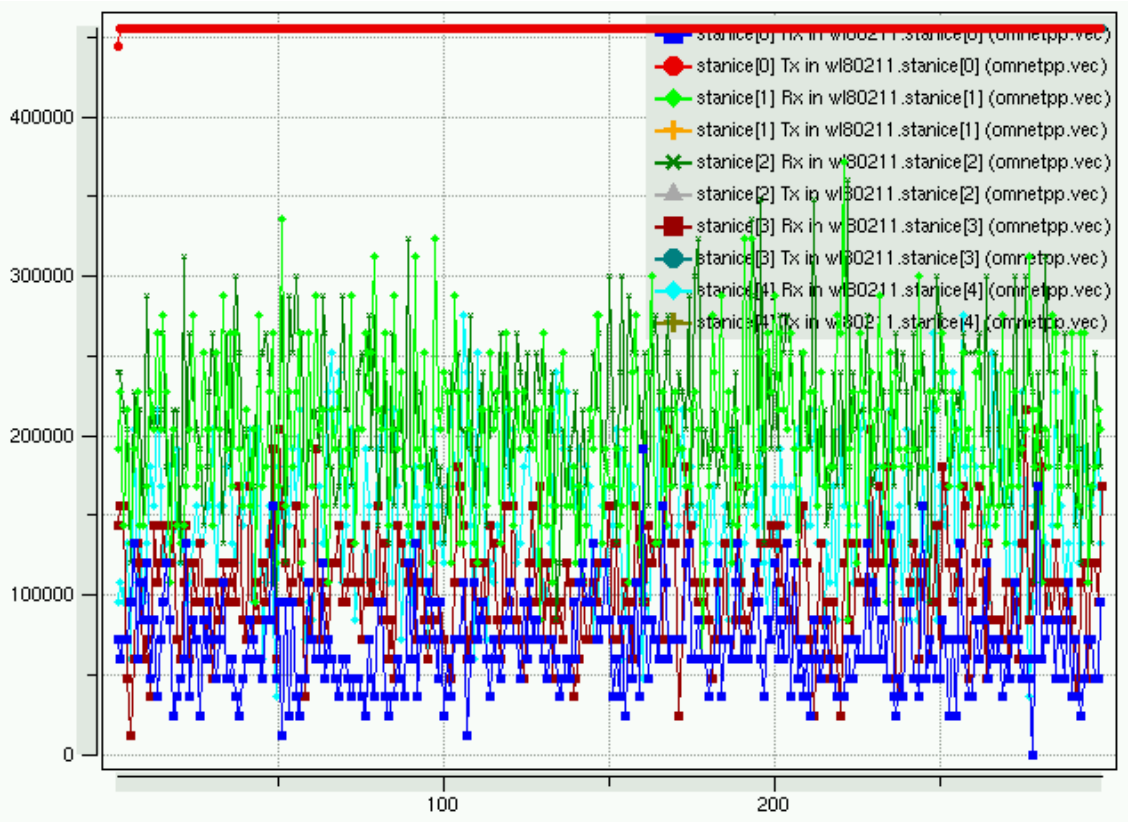
Stanice	Přen. rychl.	Vel. pkt.	Pkt/sec	Traffic	Cíl st.
stanice[0]	11 Mbps	1500	38	445 kbit	stanice[1]
stanice[1]	11 Mbps	1500	38	445 kbit	stanice[2]
stanice[2]	11 Mbps	1500	38	445 kbit	stanice[3]
stanice[3]	11 Mbps	1500	38	445 kbit	stanice[4]
stanice[4]	11 Mbps	1500	38	445 kbit	stanice[0]

Tabulka 1: Maximální propustnost pro 5 stanic

Zavedením jediné skryté stanice do sítě se ukázalo, že v takto zatížené síti při vypnutém RTS/CTS poklesl provoz skryté stanice (stanice[2]) na 722 bps při příjmu a 40 bps je příjem cílové stanice, které skrytá stanice odesílá. Provoz ostatních stanic se pohybuje od 445 kbps,

přes 392 kbps až po 63 kbps.

Zapnutí RTS/CTS má v tomto případě výrazný vliv hlavně na provoz skryté stanice, který se upravil na 213 kbps na příjmu a 105 kbps na odesílání (Obrázek 10).



Obrázek 10.: Provoz s jednou skrytou stanicí a zapnutým RTS/CTS

Pro sledování odezvy a jitteru nahradíme v Tabulce 1 u stanice[0] velikost paketu hodnotou 64 a v konfiguračním souboru modelu změníme typ dat z 11 (data) na 12 (požadavek na ping). Změníme také počet paketů za sekundu na 10. Zbytek konfigurace modelu zůstává beze změny.

Bez skrytých stanic, s vypnutým RTS/CTS: min ping 1,6 ms, max ping 232,5 ms, průměr 11,2 ms, jitter 20,2 ms.

Se skrytou stanicí, RTS/CTS zapnuto: min ping 209 ms, max ping 14425 ms, průměr 3579 ms, jitter 2130 ms.

7.1.2 Protokol 802.11g

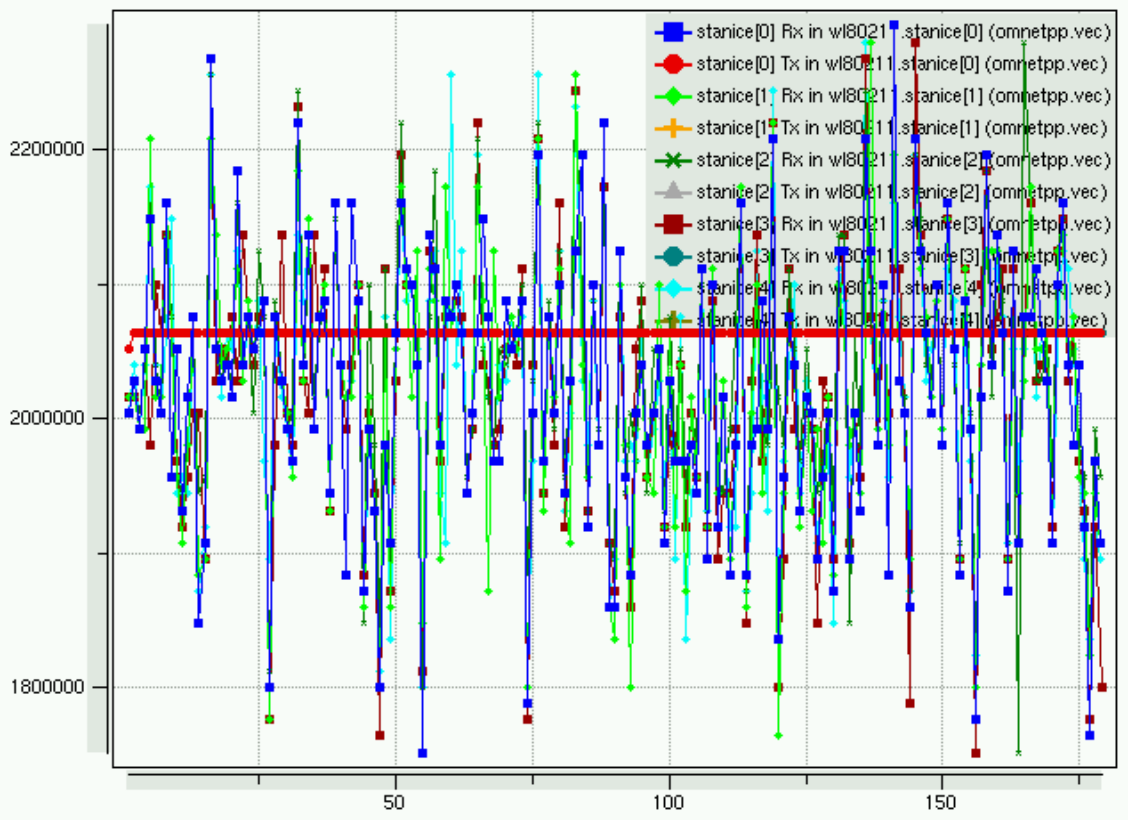
Vzhledem k časové náročnosti simulací vyšších přenosových rychlostí jsem se rozhodl zkrátit simulovaný časový úsek na 3 minuty.

Stanice	Přen. rychl.	Vel. pkt.	Pkt/sec	Traffic	Cíl st.
stanice[0]	54 Mbps	1500	172	1,96 Mbps	stanice[1]
stanice[1]	54 Mbps	1500	172	1,96 Mbps	stanice[2]
stanice[2]	54 Mbps	1500	172	1,96 Mbps	stanice[3]
stanice[3]	54 Mbps	1500	172	1,96 Mbps	stanice[4]
stanice[4]	54 Mbps	1500	172	1,96 Mbps	stanice[0]

Tabulka 2: Konfigurace modelu

Tabulka 2 ukazuje konfiguraci modelu, při které se přiblížíme saturaci při protokolu 802.11g. Grafické znázornění provozu je na obrázku číslo 11. Přístupovým bodem v tomto případě prochází provoz průměrně 9,8 Mbps v každém směru, celkový provoz je v tomto případě tedy asi 19,6 Mbps.

Zapnutím RTS/CTS poklesne průměrná hodnota přijímaného provozu z 1,96 Mbps na 0,96 Mbps až 306 kbps.



Obrázek 11.: Dosažení saturace protokolem 802.11g

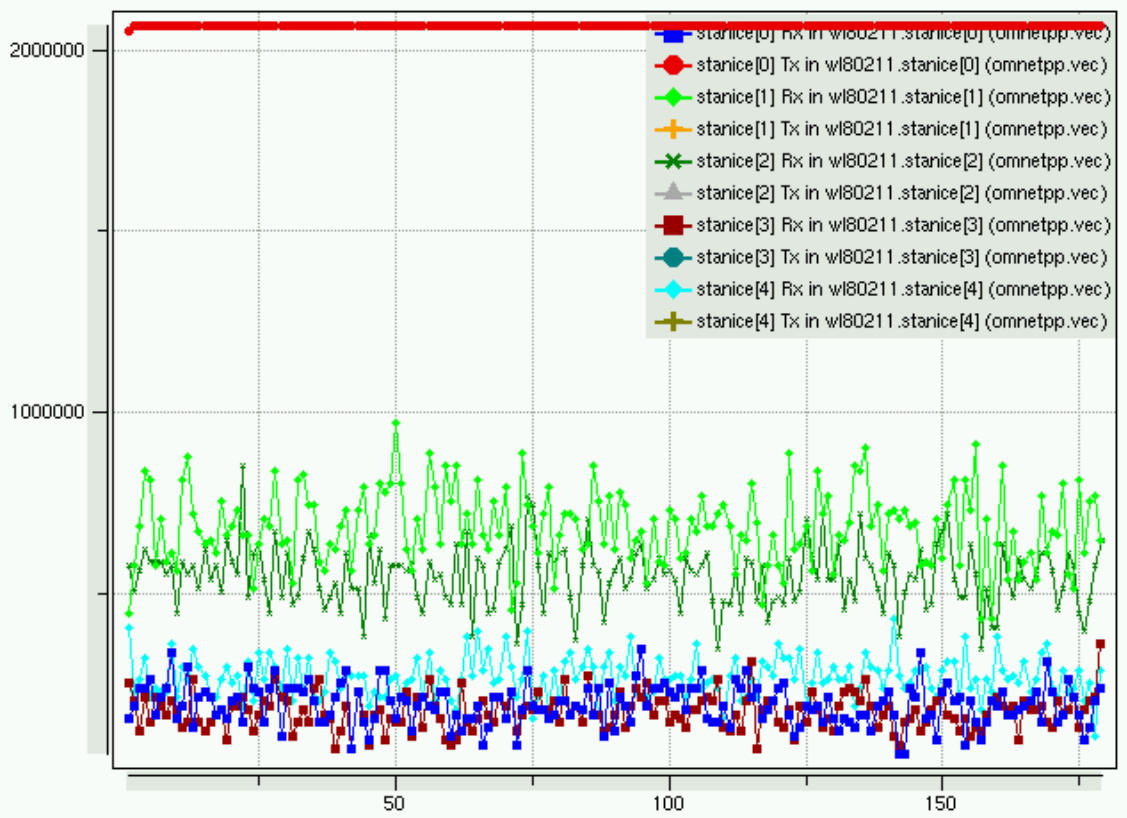
Zavedením jedné skryté stanice výrazně poklesne, prakticky zmizí, provoz této skryté stanice. Provoz ostatních stanic se pohybuje od 1,96 Mbps po 293 kbps.

Stejně jako u protokolu 802.11b má zapnutí RTS/CTS vliv zejména na provoz skryté stanice.

Ten vzrostl na 535 kbps na příjmu a 170 kbps na odesílání. Provoz ostatních stanic je mezi 188 kbps a 667 kbps. Provoz v tomto případě je na obrázku číslo 12.

Měření odezvy a jitteru provedeme stejně jako u protokolu 802.11b. Bez skrytých stanic, bez RTS/CTS: min ping: 0,35 ms, max ping: 66,3 ms, jitter: 5.47 ms.

Skrytá stanice, RTS/CTS zapnuto: min ping: 222 ms, max ping: 3663 ms, průměr: 1102 ms, jitter: 658 ms



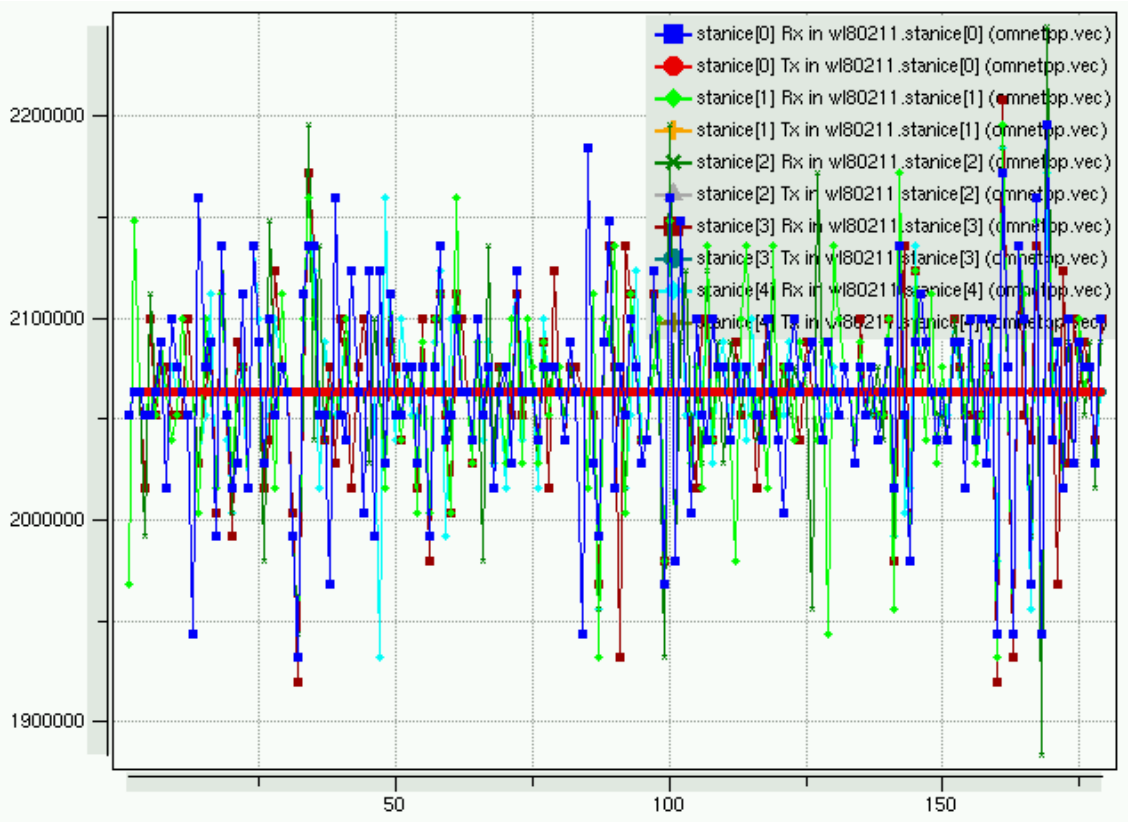
Obrázek 12.: Provoz 802.11g se skrytou stanicí a RTS/CTS

7.1.3 Protokol 802.11a

Protokol 802.11a budeme simulovat na stejné přenosové rychlosti jako 802.11g, proto použijeme stejnou konfiguraci modelu. Oba protokoly dosahují saturace při podobném provozu (Obrázek 13). Přístupovým bodem prochází provoz asi 9,8 Mbps v každém směru, celkem tedy 19,6 Mbps.

Stanice	Přen. rychl.	Vel. pkt.	Pkt/sec	Traffic	Cíl st.
stanice[0]	54 Mbps	1500	172	1,96 Mbps	stanice[1]
stanice[1]	54 Mbps	1500	172	1,96 Mbps	stanice[2]
stanice[2]	54 Mbps	1500	172	1,96 Mbps	stanice[3]
stanice[3]	54 Mbps	1500	172	1,96 Mbps	stanice[4]
stanice[4]	54 Mbps	1500	172	1,96 Mbps	stanice[0]

Tabulka 3: Konfigurace modelu pro simulace protokolu 802.11a



Obrázek 13.: Dosažení saturace protokolem 802.11a

Zapnutím RTS/CTS poklesne průměrný provoz, přijímaný stanicemi, na 1,13 Mbps až 399 kbps.

Zavedením jedné skryté stanice bez použití RTS/CTS podobně jako u protokolu 802.11g prakticky zmizí příchozí i odchozí provoz skryté stanice. Provoz ostatních stanic se pohybuje mezi 1,96 Mbps a 680 kbps.

Zapnutím RTS/CTS dojde, podobně jako u předchozích protokolů, zejména ke zlepšení u skryté stanice. Ta přijímá data rychlostí 731 kbps a odesílá rychlostí 282 kbps. Provoz ostatních stanic se pohybuje mezi 246 kbps a 753 kbps.

Odezva a jitter bez RTS/CTS a skrytých stanic: min ping: 0,39 ms, max ping: 60,8 ms,

průměr ping: 2,77 ms, jitter: 4.93 ms.

S jednou skrytou stanicí a RTS/CTS: min ping: 78,14 ms, max ping: 4333,83 ms, průměr ping: 1016,68 ms, jitter: 674,35 ms.

7.1.4 Dílčí shrnutí

Výsledky simulací ukazují, že použitím RTS/CTS klesá celková propustnost sítě. Je to dáno tím, že v síti přibude paketů, navíc vysílaných na nižší propustnosti takže přenosové médium je jimi déle obsazeno. Při výskytu skrytých uzlů pomáhá RTS/CTS hlavně provozu těchto uzlů. Mechanismus RTS/CTS má také velmi nepříznivý vliv na odezvu a jitter.

8 EXPERIMENTY S REÁLNÝM ZAŘÍZENÍM

Pro ověření výsledků modelu jsem zvolil experiment na zařízení, které je reálně vybudováno za účelem poskytování internetových služeb. Z technických důvodů budu provádět experiment pouze na síti 802.11b.

8.1 Konfigurace testovací sítě

Zvolil jsem přístupový bod Orinoco AP500, ke kterému je připojeno několik klientů v jednom areálu a blízkém okolí. Pět z těchto stanic jsou PC s OS Linux, čtyři s PCMCIA kartou Orinoco, jeden je připojen zařízením Compex WP11B+ v režimu ethernet adapter. Na všechny PC je možné přihlásit se pomocí SSH, mám tedy možnost provést test v reálném prostředí.

Pro generování testovacího provozu jsem zvolil linuxovou aplikaci *jtg*. Tento program umožňuje generovat IP pakety nastavitelné velikosti, nastavitelnou rychlostí a protokolem TCP nebo UDP. Aplikace funguje buď v režimu naslouchání, kdy pouze vyhodnocuje zachycené pakety, nebo v režimu odesílání. Test budu provádět tak, že se na každou stanici přihlásím 2x, jednou spustím aplikaci pro příjem dat, jednou pro odesílání tak, aby provoz odpovídal simulovanému provozu.

Zdroj	Cíl	Provoz [kbps]
station[0]	station[1]	228,3
station[1]	station[2]	137,8
station[2]	station[3]	170,8
station[3]	station[4]	405,3
station[4]	station[0]	276,5

Tabulka 4.RTS/CTS vypnuto

Odezva: min/avg/max = 3.5/5237.8/19843.7 ms

Tabulka 4 ukazuje výsledek reálného testu 802.11b s vypnutým RTS/CTS.

Zdroj	Cíl	Provoz [kbps]
station[0]	station[1]	180,7
station[1]	station[2]	153,2
station[2]	station[3]	146,5
station[3]	station[4]	405.2
station[4]	station[0]	286,1

Tabulka 5: RTS/CTS zapnuto

Odezva:min/avg/max = 3.5/2120.4/6147.9 ms

Experimenty na reálné síti dávají podobné výsledky jako simulační model. Rozdíly mohou být způsobeny tím, že i když jsem pro testování zvolil dobu s nejnižším provozem, nebylo možné veškerý cizí provoz úplně vyloučit. Další důvod je ten, že model je určitým zjednodušením modelovaného systému. Výsledky však ukazují, že chování modelu se dostatečně přibližuje reálné bezdrátové síti.

9 DOPORUČENÍ PRO VÝSTAVBU BEZDRÁTOVÝCH SÍTÍ

V dnešní době existuje nepřeberné množství prvků pro výstavbu bezdrátových sítí. Toto množství zařízení je založeno na několika málo čipových sadách. Různá zařízení bývají různě kvalitní, i když jsou založena na stejné čipové sadě.

Pro hodnocení kvality bezdrátových zařízení existuje mnoho kritérií. Pro jedno z nich byl zaveden pojem *bezdrátový fingerprint* [8], který charakterizuje dané zařízení. Tento pojem byl zaveden zejména pro hodnocení zařízení z hlediska bezpečnosti a možných útoků, charakteristika však může ukazovat také na výkon zařízení.

Jednak z praxe, jednak z vývoje simulačního modelu se ukazuje, že velkou roli při výkonnosti sítě hraje hardware i software jednotlivých zařízení. Jak už jsem ve své práci uvedl (kapitola 3.1.1 na straně 15), nejvíc vytíženým zařízením v režimu infrastructure je přístupový bod. Pokud nemá dostatečný výkon procesoru nebo dostatečné množství vyrovnávací paměti, může degradovat výkon celé sítě. Při hustším provozu jsou potom zahazovány pakety, které by jinak v pořádku dorazily k cíli.

Důležitou kapitolou při výstavbě bezdrátových sítí je také pečlivé dodržování montážních postupů, VF kabelů, jejich spojování, používání odpovídajících antén, dodržování podmínky přímé viditelnosti a neporušenosti Fresnelovy zóny minimálně na přístupový bod.

Po výstavbě sítě nastává fáze jejího provozu. Udržujeme síť na kanále, který je volný, nebo (hlavně v případě 2,4 GHz pásma) aspoň co nejméně obsazený. V závislosti na použitých zařízeních máme možnost ovlivnit některé jejich parametry (např. CTS timeout, práh fragmentace atd.). U téměř všech zařízení je nastavitelný mechanismus RTS/CTS. Pomocí něho můžeme u vysoce zatížených sítí zlepšit průchodnost dat od a ke skrytým stanicím, má však negativní vliv na odezvu v síti a jitter, způsobuje celkové rozkolísání provozu.

Použití se může vyplatit v případě, že transakce RTS/CTS neblokuje médium déle, než následující datový paket. V případě protokolu 802.11b blokuje RTS/CTS transakce médium stejně dlouho, jako paket velikosti 374 bytů, přenášený rychlostí 11 Mbps. V případě 802.11a, za předpokladu přenosu RTS/CTS na 6 Mbps je datový paket na 54 Mbps časově stejně dlouhý při velikosti 306 bytů.

ZÁVĚR

Účelem této práce bylo vytvořit simulační model MAC vrstvy protokolů bezdrátových počítačových sítí 802.11. Každý model je zjednodušeným obrazem reality. Míra zjednodušení záleží na účelu modelu. Při tvorbě modelu jsem v maximální míře přihlížel k tomu, aby byly zachovány všechny důležité funkce MAC vrstvy simulované sítě. Méně už jsem přihlížel k fyzikálním vlastnostem přenosového kanálu nebo ke vlivu hardwaru.

Model jsem vytvořil v jazyce C++ s použitím knihovny pro diskrétní simulace OMNeT++. Z testovaných simulačních knihoven se pro řešený problém hodila nejlépe, i tak ale bylo třeba přidat řadu vlastností.

Na simulovaném modelu jsem provedl srovnání protokolů 802.11a, b, g při různých situacích. Zaměřil jsem se zejména na chování sítí při výskytu skryté stanice v síti a při použití mechanismu RTS/CTS. Ukázalo se, že RTS/CTS má kladný vliv na provoz skryté stanice, snižuje celkovou propustnost sítě, zhoršuje odezvu a jitter. Způsobuje celkové rozkolísání provozu. Z testovaných protokolů se nejlépe choval 802.11a.

Použití RTS/CTS má omezený význam u vysoce zatížených sítí s výskytem skrytých stanic a u provozu, který není citlivý na zpoždění a zejména na jeho kolísání. Podle mého názoru se od mechanismu RTS/CTS upouští a vývoj bezdrátových sítí směřuje spíše k použití zajištěné kvality služeb (802.11e).

ENDCLOSURE

Purpose of this work was to create simulation model of MAC layer of 802.11 wireless networks protocols. Every model is simplified view of reality. Scale of simplification depends on purpose of the model. Upon creating the model I maximally kept important functionf of MAC layer of simulated network. Less I reflected physical characteristisc of transmit channel or hadrware.

I created the model in C++ language with OMNeT++ discrete simulation library. OMNeT++ matches the best for solved problem.

On simulated model I performed comparsion of 802.11a, b and g protocols in different situations.

I focused hidden node problem and RTS/CTS use. It turned out positive influence of RTS/CTS to hidden node traffic and negative influence to cummulated troughtput, response time and jitter. It causes unstability of all traffic. From tested protocols 802.11a had the best output.

RTS/CTS have limited sense for highly loaded networks with hidden nodes and with no response and jitter critical traffic. My opinion is, QoS in 802.11e is better solution than RTS/CTS.

SEZNAM POUŽITÉ LITERATURY

[1] Georgi Kirov, Simulation Investigation of the Ethernet Network Performance [online]. [cit. 2007-03-11]. Dostupný z WWW:

<http://www.gcmarshall.bg/sfp981149/files/**Simulation_Investigation**_of_the_**Ethernet_Network_Performance**.doc>

[2] Jia Wang,Srinivasan Keshav, Efficient and Accurate Ethernet Simulation [online]. [cit. 2007-03-28]. Dostupný z WWW: <http://citeseer.ist.psu.edu/358110.html>

[3] ANSI/IEEE Std 802.11, 1999 Edition [online]. [cit 2007-02-25]. Dostupný z WWW: <http://standards.getieee.org/getieee802/download/802/download/802.11-1999.pdf>

[4] Ishwar Ramani, Stefan Savage, *SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks* [online]. [cit 2007-03-03]. Dostupný z WWW: <http://citeseer.ist.psu.edu/358110.html>

[5] OMNeT++ domovská stránka, [online]. Dostupný z WWW: <http://www.omnetpp.org>

[6] Network and Telecom Dictionary and Encyclopedia, [online]. [cit 2007-04-30]. Dostupný z WWW: <http://www.networkdictionary.com/>

[7] Hung-Huan Liu, Jean-Lien C. Wu, *A Scheme for Supporting Voice over IEEE 802.11 Wireless LocalArea Network*, [online]. [cit 2007-04-18].

[8] Johnny Cache, *Fingerprinting 802.11 Implementations via Statistical Analysis of the Duration field,* [online]. [cit 2007-05-14] Dostupný z WWW: <http://www.uninformed.org/?v=5&a=1&t=sumry>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MAC Medium Access Control, protokol přístupu ke sdílenému přenosovému médiu

ACK Potvrzené přijetí dat

BSS Basic Service Set, bezdrátová buňka

CTS Clear To Send, povolení vysílání

CW Contetion Window, interval pro výběr náhodného časového úseku pro čekání před vysíláním

DIFS Distributed Interframe Space, jeden z časových intervalů mezi rámci

NAV Network Allocation Vector, udává čas do předpokládaného uvolnění média při používání mechanismu RTS/CTS

RTS Request To Send, požadavek na vysílání s oznámením délky následujícího přenosu

Mbps megabit za sekundu

kbps kilobit za sekundu

SEZNAM OBRÁZKŮ

SEZNAM TABULEK