

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ
Fakulta elektrotechnická
katedra řídicí techniky

Analýza bezdrátových sítí IEEE 802.11e
Bakalářská práce

Vedoucí práce: Ing. Michal Sojka
Student: Milan Švarc

červenec 2008

Abstrakt

Cílem této práce je, seznámit vás se standardem IEEE 802.11, jeho doplňkem IEEE 802.11e a jeho možné implementace do ns-2 s následnou simulací. V této práci naleznete základní popis ns-2 (network simulator), stručný přehled základních modulů a jejich ukázky. Dále je zde uveden podrobný popis instalace ns-2 a jeho rozšíření od TKN, které rozšiřuje základní standard IEEE 802.11. V práci je popsán skriptovací jazyk Tcl/Tk. Jedná se o stručný úvod do programování pomocí tohoto jazyku.

Abstract

The goal of this work is introduce you with IEEE 802.11, his addition IEEE 802.11e and his possible implementation into the ns-2 with simulation. In this work find basic description of ns-2 (network simulator), conspectus moduls and their preview. The next part describe installation ns-2 and TKN modul, which extend IEEE 802.11. In this work find description Tcl/Tk and introduction to programming.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

V Praze, dne

.....

podpis

Obsah

Seznam ilustrací.....	6
Seznam tabulek.....	7
Seznam použitých zkratk.....	8
Úvod.....	10
1 Úvod do bezdrátových sítí.....	10
1.1 Wi-Fi aliance	10
1.2 IEEE(Institute of Electrical and Electronics Engineers).....	10
2 IEEE 802.11.....	11
2.1 Přehled standardů IEEE 802.11.....	11
2.2 Stručný popis jednotlivých standardů.....	11
3. Popis základních termínů.....	15
3.1 Obecné termíny.....	15
3.2 CSMA.....	15
3.3 Vyhledání Wi-Fi sítí.....	16
3.4 Zabezpečení Wi-Fi sítí.....	16
4. Vrstvy a modulace.....	17
4.1 OSI Model.....	17
4.2 IEEE 802.11 vrstvy.....	17
4.3 Modulace.....	17
4.4 MAC vrstva.....	18
4.5 Problém skryté stanice.....	19
4.6 Časování.....	20
5 IEEE 802.11e.....	22
5.1 Metody přístupu k médium.....	22
5.2 QoS.....	23
5.3 Formát rámců.....	24
6 Network Simulator NS-2.....	27
6.1 Instalce NS-2.....	27
6.2 Součásti NS-2.....	29
6.3 NAM.....	29
6.4 Záznam dat.....	31
6.5 Rozšíření NS-2 od TKN.....	31
6.5.1 Instalace rozšíření od TKN.....	31
6.5.2 Popis změn v rozšíření TKN.....	33
7 TCL/Tk.....	34
7.1 Vlastnosti jazyka Tcl.....	34
7.2 Přednosti Tcl.....	34
7.3 Zápory Tcl.....	35
7.4 Rozšíření Tk.....	35
7.5 Syntaxe.....	35
7.5.1 Komentáře.....	35
7.5.2 Proměnné.....	36
7.5.3 Matematické funkce.....	37
7.5.4 Logické a bitové funkce.....	38
7.5.5 Funkce.....	38
7.5.6 Podmíněné příkazy.....	39
7.5.7 Cykly.....	40
7.5.8 Strukturované datové typy.....	40

7.5.9 Práce se soubory.....	42
8 Simulace IEEE 802.11 a IEEE 802.11e.....	43
8.1 Test IEEE 802.11.....	43
8.2 Popis základních parametrů simulace.....	43
8.3 Kreslení grafů.....	44
8.4 Testování.....	45
8.5 Popis Trace-file souboru.....	46
9 Závěr.....	48
Seznam použité literatury.....	49

Seznam ilustrací

Ilustrace 1: Logo Wi-Fi.....	10
Ilustrace 2: Ukázka částečného překryvu nosných frekvencí.....	18
Ilustrace 3: Problém skryté stanice.....	19
Ilustrace 4: Ukázka RTS a CTS.....	20
Ilustrace 5: Ukázka exponencionálního nárůstu CW.....	21
Ilustrace 6: Ukázka backoff procedury.....	21
Ilustrace 7: Ukázka využití AIFS.....	24
Ilustrace 8: Ukázka zpoždění beaconu.....	24
Ilustrace 9: Obecný formát rámce.....	25
Ilustrace 10: Ukázka položek Frame Control.....	25
Ilustrace 11: Základní okno NAM.....	30
Ilustrace 12: Ukázka navržené sítě pomocí NAM.....	30
Ilustrace 13: graf simulace wlan_complex IEEE 802.11.....	43
Ilustrace 14: Ukázka výstupu z prvni.tcl.....	45

Seznam tabulek

Tabulka 1: Přehled standardů.....	11
Tabulka 2: OSI Model.....	17
Tabulka 3: IEEE 802.11 vrstvy.....	17
Tabulka 4: Třídy provozu.....	22
Tabulka 5: Základní časové parametry EDCF.....	23
Tabulka 6: Význam hodnot ToDS/FromDS.....	25
Tabulka 7: Substituce.....	37
Tabulka 8: Podporované funkce.....	37
Tabulka 9: Logické a bitové funkce.....	38
Tabulka 10: Funkce pro pole.....	41
Tabulka 11: Funkce pro práci se seznamy.....	41

Seznam použitých zkratk

AC – Access Category
ACK – Acknowledge
AES – Advanced Encryption Standard
AIFS – Arbitration Inter Frame Space
AIFSN – Arbitration Inter Frame Spacing Number
AP – Acces Point (přístupový bod)
APSD – Automatic Power Save Delivery (automatická úspora energie)
BA – Block Acknowledgments (blokové potvrzování)
BSS – Basic Service Set
CCK – Complementary Code Keying (doplňkové kódové klíčování)
CSMA – Carrier Sense Multiple Access
CSMA/CA – Carrier Sense Multiple Access With Collision Avoidance (CSMA s předcházením kolizí)
CSMA/CD – Carrier Sense Multiple Access With Collion Detection (CSMA s detekcí kolize)
CTS – Clear To Send
CW – Contention Window
DCF – Distributed Coordination Fuction
DFIR – Diffused Infrared
DIFS – Distributed IFS
DLS – Direct Link Seup
DSSS – Direct Sequence Spread Spectrum
EDCA – Enhanced DCF Channel Access (vylepšená DCF přístupová metoda ke kanálům)
EDCF – Enhanced Distribution Coordination Function
EIFS – Extended IFS
ESS – Extended Service Set (rozšířený soubor služeb)
FHSS – Frequency Hopping Spread Spectrum
HCCA – HCF Controlled Channel Access (přístupová metoda ke kanálům řízeným HCF)
HCF – Hybrid Coordination Function
IBSS – Independent Basic Service Set
IEEE – Institute of Electrical and Electronics Engineers (Institut pro elektrotechnické a elektronické inženýrství)
IFS – Inter Frame Spaces
LAN – Local Area Network
MAC – Media Access Control
MIMO – Multiple Input Multiple Output (více vstupů více výstupů)
NAV – Network Allocation Vector
OFDM – Orthogonal Frequency Division Multiplex (ortogonální multiplex s kmitočtovým dělením)
OSI – Open Systems Interconnection
PCF – Point Coordination function
PIFS – Point Coordination IFS
QoS – Quality of Service
QPSK – Quadrature Phgase Shift Keying
QSTA – QoS Station
RTS – Request To Send
SIFS – Short Inter Frame Space
SSID – Service Set Identifer (identifikátor souboru služeb)

STA – Station
TC – Traffic Classes (třídý provozu)
TKIP – Temporal Key Integrity Protocol
TSID – Traffic Stream Identifier
TXOP – Transmit Opportunity
VoIP – Voice over Internet Protocol
WEP – Wired Equivalent Privacy
WMM – Wireless MultiMedia
WPA – Wi-Fi Protected Access

Úvod

V první kapitole stručně popisují pojmy Wi-Fi a IEEE.

V druhé kapitole nalezneme popis standardu IEEE 802.11 a jejich doplňky.

Třetí kapitola obsahuje základní informace o fungování Wi-Fi sítí a metodě CSMA.

Ve čtvrté kapitole vás seznámím s základním OSI modelem a následně s MAC vrstvou. V kapitole též nalezneme popis modulací a nakonec kapitoly i popis základních časových údajů.

Pátá kapitola se zabývá popisem IEEE 802.11e a obsahuje též základní popis přenášeného rámce.

V šesté kapitole podrobně popisují základní vlastnosti NS-2 (Network Simulator). Je zde podrobně popsán způsob instalace samotného NS a následně rozšíření od TKN včetně popisu změn, které toto rozšíření přináší.

Sedmá kapitola obsahuje základní popis Tcl/Tk, jeho klady a zápory. Dále je zde uveden stručný přehled příkazů a jejich praktické použití.

V osmé kapitole je uveden popis trace-file souboru a dále výsledky testů získaných pomocí NS-2.

V poslední kapitole shrnuji výsledky práce.

1 Úvod do bezdrátových sítí

Bezdrátové sítě se dnes nacházejí na mnoha místech. Existuje mnoho poskytovatelů Wi-Fi připojení k internetu, ať již placené nebo bezplatné. Mezi hlavní výhody těchto sítí patří mobilita nebo-li možnost volného pohybu. Funkčnost Wi-Fi sítí zaručuje standard IEEE802.11. Představím vám proto význam těchto dvou výrazů.

1.1 Wi-Fi aliance

- sdružuje výrobce Wi-Fi
- uděluje certifikáty o kompatibilitě



Ilustrace 1: Logo Wi-Fi

1.2 IEEE(Institute of Electrical and Electronics Engineers)

Institut byl vytvořen v roce 1963 spojením Institute of Radio Engineers(IRE) a American Institute of Electrical Engineers(AIEE). Jedná se o mezinárodní organizaci, která má v současnosti více než 375 000 členů po celém světě. Sídlo institutu je ve státě New York, USA.

2 IEEE 802.11

V této kapitole vám postupně popíší jednotlivé standardy a jejich doplňky. U standardů je udána jejich stručná vlastnost. Tato část je citována z [9].

2.1 Přehled standardů IEEE 802.11:

Původní norma IEEE 802.11 se stala základem pro rozvoj v komunikace v bezdrátových sítích. V současnosti existuje již několik modifikací této původní normy. Rozšíření této normy jsou využívány pro současné Wi-Fi sítě. Wi-Fi sítě byli zprvu navrženy pro vzájemné připojení bezdrátových zařízení, ale později byli připojovány do lokálních sítí LAN a dnes jsou využívány pro připojování k internetu. Wi-Fi zařízení jsou dnes více méně ve všech přenosných počítačích.

<i>Standard</i>	<i>Rok vydání</i>	<i>Pásmo[GHz]</i>	<i>Maximální rychlost[Mbit/s]</i>	<i>Fyzická vrstva</i>
<i>Původní IEEE 802.11</i>	<i>1997</i>		<i>2</i>	
<i>IEEE 802.11a</i>	<i>1999</i>	<i>5</i>	<i>54</i>	<i>OFDM</i>
<i>IEEE 802.11b</i>	<i>1999</i>	<i>2,4</i>	<i>11</i>	<i>DSSS</i>
<i>IEEE 802.11g</i>	<i>2003</i>	<i>2,4</i>	<i>54</i>	<i>OFDM</i>
<i>IEEE 802.11n</i>	<i>2009</i>	<i>2,4 nebo 5</i>	<i>248</i>	<i>MIMO</i>
<i>IEEE 802.11y</i>	<i>2008</i>	<i>3,7</i>	<i>54</i>	

Tabulka 1: Přehled standardů

2.2 Stručný popis jednotlivých standardů

Seznam všech IEEE 802.11 standardů:

- IEEE 802.11 – Původní standard pro 1 a 2 Mbit/s rychlost s frekvencí 2.4 GHz (1999)
- IEEE 802.11a – 54 Mbit/s, 5 GHz standard (1999, produkty od 2001)
- IEEE 802.11b – Vylepšení 802.11 s podporou 5.5 a 11 Mbit/s (1999)
- IEEE 802.11c – Bezdrátové přemostění (bridge); obsaženo v IEEE 802.1D standardu (2001)
- IEEE 802.11d – Mezinárodní roamingový dodatek (2001)
- IEEE 802.11e – Vylepšení QoS, včetně skupin (burst) paketů (2005)
- IEEE 802.11F – Komunikace mezi bezdrátovými přístupovými body (2003) Stažen v březnu 2006.
- IEEE 802.11g – 54 Mbit/s, 2.4 GHz standard (zpětně kompatibilní s 802.11b) (2003)

- IEEE 802.11h – Správa spektra 802.11a (5 GHz) pro Evropu (2004)
- IEEE 802.11i – Vylepšený autorizační a šifrovací algoritmus (WPA2) (2004)
- IEEE 802.11j – Dodatek pro Japonsko; nová frekvenční pásma pro multimedia (2004)
- IEEE 802.11k – Vylepšení správy rádiových zdrojů pro vysoké frekvence. (Navazuje na IEEE 802.11j)
- IEEE 802.11l – (rezervováno a nebude použito)
- IEEE 802.11m – Správa standardu: kombinace existujících dodatků do jednoho dokumentu
- IEEE 802.11n – Vylepšení pro vyšší datovou propustnost
- IEEE 802.11o – (rezervováno a nebude použito)
- IEEE 802.11p – Bezdrátový přístup pro pohyblivé prostředí (auta, vlaky, sanitky)
- IEEE 802.11q – (rezervováno a nebude použito, aby se nepletlo s 802.1Q)
- IEEE 802.11r – Rychlé přesuny mezi přístupovými body (roaming)
- IEEE 802.11s – Samoorganizující se bezdrátové sítě. (ESS Mesh Networking)
- IEEE 802.11T – Předpověď bezdrátového výkonu - testovací metody
- IEEE 802.11u – Spolupráce se sítěmi mimo 802 standardy (například s mobilními sítěmi)
- IEEE 802.11v – Správa bezdrátových sítí (konfigurace klientských zařízení během připojení)
- IEEE 802.11w – Chráněné servisní rámce
- IEEE 802.11x – (rezervováno a nebude použito)
- IEEE 802.11y – Pro běh ve frekvenčním pásmu 3650 - 3700 MHz (veřejné pásmo v USA)
- IEEE 802.11z – rozšíření Direct Link Setup (DLS) (Srpen 2007 - Prosinec 2011)

IEEE 802.11a

- schválena v roce 1999
- využívá pásmo 5 Ghz
- používá modulaci OFDM
- větší povolený vyřazovací výkon oproti IEEE 802.11b,g
- možnost použití na delší vzdálenost
- dosahuje rychlosti až 54 Mbit/s
- až 8 nezávislých, nepřekrývajících kanálů

IEEE 802.11b

- jedná se o doplněk k IEEE 802.11, jenž byl schválen v roce 1999
- známá jako Wi-Fi
- zvyšuje přenosovou rychlost na 11Mbit/s v přenosovém pásmu 2,4 Ghz
- pro dosažení této rychlosti je využit nový způsob kódování tzv. CCK(Complementary Code Keying – doplňkové kódové klíčování) v rámci DSSS

- tato norma specifikuje, že podle rušivosti prostředí se mění rychlost na nižší nebo na vyšší
- jestliže se bavíme o rychlosti 11Mbit/s je potřeba uvést, že se jedná o rychlost na fyzické vrstvě a ne efektivní rychlost

IEEE 802.11c

- doplňuje standard IEEE 802.11D
- přidává požadavky na přemostění MAC
- IEEE 802.11D upravuje základní LAN standard pro 802.11 rámce, zejména klauzule 2.5 Support of the Internal Sublayer Service

IEEE 802.11d

- nazývaný též jako globální harmonizační standard
- používá se zřejmě v zemích kde nejsou povoleny jiné dodatky k IEEE 802.11
- definuje požadavky na fyzickou vrstvu k uspokojení regulačních domén nepokrytých existujícími standardy
- liší se v povolených frekvencích, vyzařovacích výkonech a propustnosti signálu
- eliminuje nutnost vývoje a výroby specifických produktů pro různé země.
- při zapnutí podpory pro IEEE 802.11d v přístupovém bodě způsobí, že zařízení začne vysílat do celé sítě (broadcastovat) ISO kód země ve které se nachází jako součást svých beacon paketů a požadavků na odpověď. Pokud je zapnut, klient přispůsobí své frekvence, vyzařovací výkon a propustnost. Standard je tak vhodný pro systémy, které chtějí poskytovat globální roaming.

IEEE 802.11e

- doplněk IEEE 802.11
- vylepšuje MAC vrstvu rozšířením o QoS(Quality of Service – podporu kvality lužeb)
- tento standard je důležitý pro aplikace citlivé na zpoždění(například VoIP a proudové multimédia)

IEEE 802.11g

- rozšiřuje IEEE 802.11b
- je zpětně kompatibilní
- vysílá ve stejném frekvenčním spektru
- zvyšuje maximální rychlost na 54Mbit/s
- pro rychlosti 6,9,12,18,24,36,48 a 54 je použita OFDM modulace, pro ostatní rychlosti je modulace stejná jako v IEEE 802.11b

IEEE 802.11h

- doplňuje IEEE 802.11a
- řeší problémy s rušením od ostatních zařízení, které vysílají na 5 Ghz
- upravuje MAC vrstvu
- dynamický výběr kanálů

IEEE 802.11n

- cíl je upravit MAC vrstvu, aby se dosáhlo rychlostí přes 100 Mbit/s a zvýšení dosahu
- využití technologie MIMO

Pojem 802.11x je neformálně používán k označení libovolného 802.11 standardu. (Standard IEEE 802.1X pro řízení přístupu k síti založený na autentifikaci a filtrování portů, je běžně nesprávně označován jako 802.11x.)

802.11F a 802.11T jsou samostatné dokumenty, a nejsou to tedy dodatky k IEEE 802.11 standardu. Proto obsahují velké písmeno.

3. Popis základních termínů

Pro lepší porozumění funkce IEEE 802.11 je dobré rozebrat si pár základních definic. Poté si popíšeme základní metodu CSMA jak funguje a její modifikace. Na závěr této kapitoly uvádím princip vyhledávání Wi-Fi sítí a jejich zabezpečení.

3.1 Obecné termíny

Ad-Hoc Mode(„Ad-Hoc režim“). Jedná se o síť, která nemá žádný AP (Access Point). Každé zařízení, které by se chtělo připojit musí být v Ad-Hoc režimu. Takováto síť je někdy uváděna pod názvy jako nezávislá síť(IBSS – Independent Basic Service Set), síť s rovnocennými uzly(peer to peer).

AP(Access Point). AP řídí a zprostředkovává komunikaci mezi Wi-Fi zařízeními, které jsou připojeny do sítě.

BSS(Basic Service Set) – Wi-Fi síť s jedním AP a klienty.

DLS(Direct Link Setup) – umožňuje přímý přenos STA-STA v rámci BSS

ESS (Extended Service Set). ESS se skládá z několika sítí BSS, které jsou spojeny dohromady.

Channel(„kanál“). Frekvenční pásmo, ve kterém síť pracuje je rozděleno na několik kanálů.

MAC adresa – 48-bitová jedinečná adresa

Policing – jeden z výrazů pro zahazování příchozích paketů jejichž pozdržení už překročilo stanovenou dobu

QSTA – QoS Station. Stanice poskytující QoS službu

Roaming – jedná se o přechod jedné stanice z jedné BSS do druhé bez ztráty spojení. IEEE 802.11 nedefinuje jak by měl probíhat, pouze služby pro jeho podporu.

TXOP(Transmit Opportunity) – stanice mají možnost poslat několik paketů najednou v časovém intervalu určeném koordinátorem

WMM(Wireless MultiMedia) – podporuje pouze některé prvky IEEE 802.11e, například označování rámců podle priorit a tomu odpovídající řazení do front.

3.2 CSMA

CSMA(Carrier Sense Multiple Access) – jedná se o protokol umožňující přístup k médiu. Každý uzel ověřuje před vysláním, zda-li někdo jiný nevysílá(Carrier Sense - „naslouchání nosné“). Multiple Access(„vícenásobný přístup“) zase vyjadřuje, že na médiu přijímá a vysílá více uzlů. Z toho faktu je zřejmé, že dochází ke kolizím. Více vysílání současně způsobí, že rámce vysílané mezi sebou interferují a stanou se tudíž nečitelné pro přijímače. V CSMA sítích není možné zamezit kolizím úplně avšak je možné omezit. V CSMA je proti kolizím použito naslouchání nosné. Může však stát, že dva vysílače začnou vysílat téměř současně a tudíž ani jeden nezaznamená, že druhá stanice chce taky vysílat. Vyšlou tedy celý rámeček. Přijímače však nejsou schopny detekovat jestli je chyba v rámci nebo došlo ke kolizi. Tedy obnova po kolizi závisí na schopnosti detekování chyby rámce a vyvolání procedury pro obnovení. Například přijímač nevyšle potvrzení o přijetí, což vysílač přinutí vyslat znovu rámeček.

CSMA/CA. V tomto systému je základní ochranou fakt, že každý uzel, který chce vysílat musí nejdříve všechny uzly informovat, že se chystá vysílat. Když jsou všechny uzly informovány zahájí vysílání. Toto by mělo zamezit kolizím, nicméně však kolize jsou stále možné a nejsou detekovány, takže mají stejné následky jako v klasickém CSMA. CSMA/CA se využívá především v bezdrátovém přenosu, protože účastníci nejsou schopni vysílat a přijímat současně. Tento protokol se používá v Wi-Fi sítích.

CSMA/CD. Vysílací uzly jsou v tomto případě schopni detekovat kolizi a zastavit vysílání, poté počkají určitou dobu a pokusí se znovu o vysílání. Tento přístup je mnohem efektivnější než předchozí, protože se neztrácí čas vysíláním celých kolidujících rámců. CSMA/CD však nelze použít pro všechna média.

3.3 Vyhledání Wi-Fi sítí

1. Pasivní – vyčkáváme na beacon (maják). AP vysílá periodicky tzv. maják, ten slouží k synchronizaci času na stanicích, ale obsahuje i další informace. Nejpodstatnější údaj je SSID (Service Set Identifier).
2. Aktivní – Stanice se snaží najít nějaký AP pomocí paketů Probe Request. Dalo by se to popsat jako dotaz “je tu nějaká síť“. Když toto AP zachytí vyšle Probe Response v němž najdem jeho SSID.

SSID (Service Set Identifier). SSID je řetězec, který označuje síť. Každý AP vysílá pravidelně svůj SSID. SSID se skládá z maximálně 32 znaků. Toto je klíč, který si předávají zařízení, které se snaží spolu komunikovat. Pokud by se klíč klienta neshodoval s klíčem AP, přístup by byl zamítnut. Proto mluvíme o 2 typech sítě:

1. Ad-Hoc síť
2. Infrastrukturní síť – obsahuje AP(BSS) nebo ESS.

3.4 Zabezpečení Wi-Fi sítí

WEP(Wired Equivalent Privacy) – zabezpečovací mechanismus. Existují varianty WEP64, WEP128(i 256). Číslo udává počet bitů tzv. RC4 klíče(RC4 podle šifrovací metody). Například WEP64 se skládá z 40 bitového klíče a 24 bitového inicializačního vektoru. Bohužel WEP neposkytuje úplnou ochranu, kvůli několika slabostem. Ať již délkou šifrovacího klíče nebo možností útoků pomocí pozměněných paketů. Novější zabezpečení je WPA a jeho novější verze WPA2.

WPA2 – vzhledem k možnosti relativně snadného prolomení ochrany pomocí WEP klíče, byl vytvořen WPA jako jeho náhrada. WPA používá AES šifru a šifrování dynamickým klíčem TKIP.

4. Vrstvy a modulace

V této kapitole vás nejprve seznámím s OSI modelem, aby bylo lépe patrné na jaké vrstvě operují IEEE standardy, konkrétně kde je umístěna MAC vrstva. Dále zde uvedu několik používaných modulací a jak dané modulační funkce fungují. Poté vás seznámím blíže s MAC vrstvou. Ukáži tzv. problém skryté stanice. Na závěr kapitoly je stručně popsáno časování.

4.1 OSI Model

– model síťové hierarchie, představuje strukturu síťových protokolů jako řadu vertikálních vrstev.

Aplikační vrstva
Prezentační vrstva
Relační vrstva
Transportní vrstva
Síťová vrstva
Spojovací vrstva
Fyzická vrstva

Tabulka 2: OSI Model

4.2 IEEE 802.11 vrstvy

– Wi-Fi sítě pracují na fyzické a spojovací vrstvě

802.2			Spojovací vrstva
802.11 MAC			
FHSS	DSSS	DFIR	Fyzická vrstva

Tabulka 3: IEEE 802.11 vrstvy

4.3 Modulace

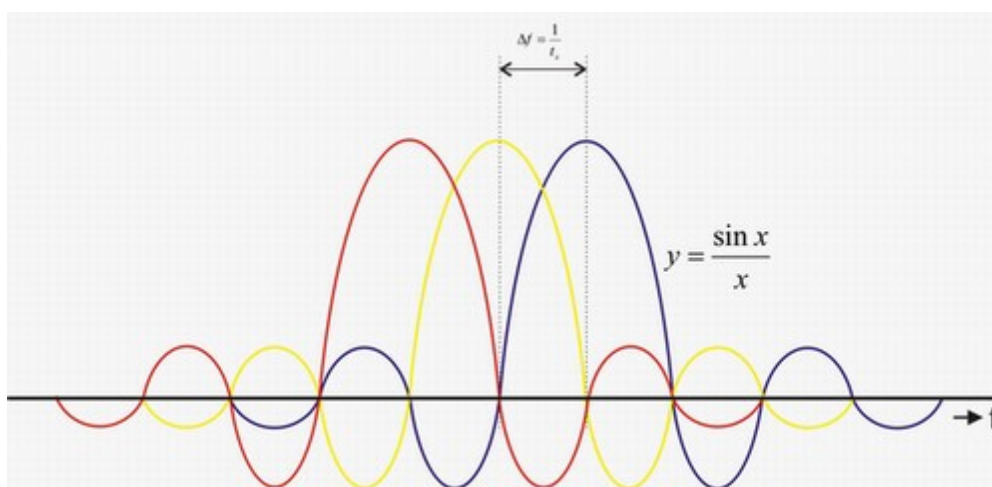
Tři základní typy modulace fyzické vrstvy IEEE 802.11:

1. **DSSS (Direct Sequence Spread Spectrum)** – princip této metody spočívá v tom, že vysílač moduluje symboly pseudonáhodnou šumovou frekvencí, poté vysílač přemění bity určené k vysílání na tok symbolů, kde jeden symbol může být jeden nebo více bitů. Použitím modulační techniky například QPSK (Quadrature Phase Shift Keying). Tím uměle zvýšíme použitou šířku pásma. Zavádíme tak vlastně redundanci, která nám zde způsobuje, že signál je rozprostřen do větší části spektra. Tím se stává signál více odolný proti rušení a bez znalosti mechanismu vytváření je velmi obtížné získat přenášená data. DSSS dělí pásmo na 14 kanálů po 22 Mhz, které se částečně překrývají.

2. **FHSS(Frequency Hopping Spread Spectrum)** – tato metoda využívá tzv. „přeskakování kanálů“. Pásmo se rozdělí na 75 podkanálů, každý po 1 Mhz. Při vysílání jsou pak podle pseudonáhodné posloupnosti přepínány jednotlivé kanály. Tuto posloupnost znají jak vysílač tak přijímač, v obou musí být synchronizována. Nevýhodou FHSS je tvoření rušení pro okolní systémy. Skoky mezi kmitočty totiž způsobují impulsní rušení.
3. **DFIR(Diffused Infrared)** – přenos infračerveným zařízením. Tato metoda je velmi zřídka používána

– všechny tyto principy pracují na rychlostech 1-2 Mbit/s

OFDM(Orthogonal Frequency Division Multiplex) – Jedná se o přenosovou techniku pracující s tzv. rozprostřeným spektrem, kdy je signál vyslán na více nezávislých frekvencích, což zvyšuje odolnost vůči interferenci. Modulační metoda OFDM spočívá v použití několika stovek až tisíců nosných kmitočtů. V případě Wi-Fi se jedná přibližně asi o 600. Nosné kmitočty jsou dále modulovány podle potřeby různě robustními modulacemi QPSK, 16QAM nebo 64QAM. Jednotlivé nosné jsou vzájemně ortogonální, takže maximum každé nosné by se mělo překrývat s minimy ostatních. Datový tok celého kanálu se tak dělí na stovky dílčích datových toků jednotlivých nosných. Poněvadž jsou ve výsledku toky na jednotlivých nosných malé, je možné vkládat ochranný interval – čas, kdy se nevysílá žádná nová informace. Na přijímací straně je tak možné nerušeně přijmout(právě) vysílaný symbol, i když přichází k přijímači více cestami s různým zpožděním. Stejný symbol přijatý vícekrát s různým zpožděním tak může odpovídat i více vysílačům. Přijímané výkonové úrovně více vysílačů resp. odrazů se tak na přijímací straně do jisté míry sčítají.



Ilustrace 2: Ukázka částečného překryvu nosných frekvencí

MIMO(Multiple Input Multiple Output – více vstupů více výstupů) – jedná se o multi-anténní komunikační systém. Jedná se tedy o model, který se snaží využít tzv. vícecestnou propagaci ke zvýšení propustnosti a dosahu nebo k snížení počtu přenosových bitových chyb. Tato technologie se využívá se spojením s OFDM. MIMO využívá více antén k posílání paralelních signálů.

4.4 MAC vrstva

standard IEEE 802.11 definuje dvě přístupové metody:

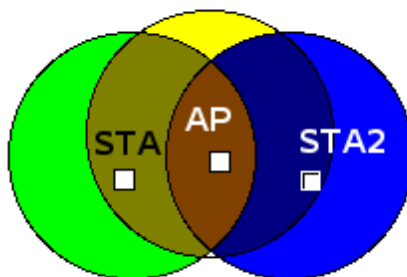
1. DCF(Distributed Coordination Function):

Je základní přístupová metoda v IEEE 802.11 sítích. Umožňuje sdílet komunikační médium na fyzické vrstvě mezi kompatibilními zařízeními pomocí CSMA/CA mechanismu. Veškerá přímá komunikace používá pozitivní potvrzování. Poslech nosné je zde realizován pomocí příposlechu na fyzické vrstvě a také pomocí virtuálního mechanismu. Virtuální mechanismus využívá řídicí rámce RTS/CTS.

2. PCF(Point Coordination function):

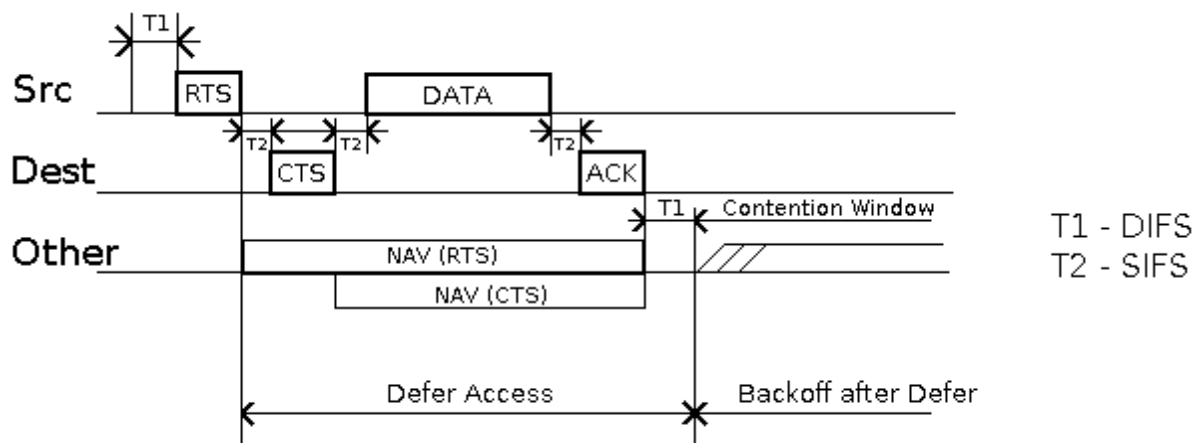
U této metody není rozlišován typ přenášených dat. AP přiřazuje prioritu každé stanici pro určený přenosový rámec. PCF slouží pouze jako volitelný mechanismus pro přenos aplikací citlivých z hlediska času, a proto ho téměř žádný výrobce neimplementuje.

4.5 Problém skryté stanice



Ilustrace 3: Problém skryté stanice

Jelikož u Wi-Fi sítí probíhá komunikace přes AP, tak jednotlivé stanice nemusí detekovat vysílání jiné stanice nebo vůbec detekovat jinou stanici. Proto se používá spolu s mechanismem předcházení kolize, režim s kladným potvrzováním. To znamená, že stanice, která chce vysílat naslouchá jestli je médium volné, poté počká ještě určený čas DIFS(Distributed Inter Frame Space) a teprve pak začne vysílat. Příjemci stanice zkontroluje CRC přijatého paketu a odešle potvrzení ACK. Přijetí potvrzujícího paketu ACK znamená, že nedošlo ke kolizi. V případě, že ACK nedorazí vysílací stanice opakuje vysílání. Pro snížení kolizí je definován tzv. virtuální naslouchací mechanismus. Tento mechanismus spočívá v tom, že stanice, která chce vysílat musí nejdříve vyslat krátký řídicí signál RTS(Request To Send), ten obsahuje mimo zdroje a cíle také trvání přenosu. AP odpovídá na RTS signálem CTS(Clear To Send), který rovněž obsahuje délku trvání přenosu. Ostatní stanice, které uslyší RTS nebo CTS si pak nastaví tzv. NAV(Network Allocation Vector - indikátor virtuálního naslouchání) na dobu trvání přenosu. Na tuto dobu považují ostatní stanice médium jako obsazené. Tím se snižuje možnost kolize. Efektivita tohoto mechanismu je hlavně pro dlouhé pakety, proto standard umožňuje také přenos bez RTS/CTS. Průběh přenosu viz Obr 4. RTS/CTS se používá pouze pro pakety od určité délky.



Ilustrace 4: Ukázka RTS a CTS

MAC vrstva vykonává detekci kolize a čeká na příjem potvrzení od jakéhokoliv vyslaného fragmentu. Výjimkou jsou pakety, které mají více než jeden cíl, například Broadcast a Multicast se nepotvrzují.

Typický LAN protokol používá pakety o velikosti několika stovek bytů, u Ethernetu až 1518 bytů. Pro bezdrátový přenos preferujeme spíše kratší pakety. A to z několika důvodů. S rostoucí délkou paketu roste i možnost jeho poškození nebo při opětovném vysílání poškozeného paketu se sníží vytíženost sítě nebo při Frequency Hopping Systému, kdy je médium periodicky přerušováno pro změnu frekvence. Má menší paket menší šanci, že bude odložen. Využívá se zde mechanismu „pošli a čekej“ nebo-li vysílací stanice nevyšle další paket dokud neobdrží potvrzení o přijetí, jinak stále vysílá stejný paket. Avšak jen do určitého počtu neúspěšných vysílání, pak je paket zahozen.

4.6 Časování

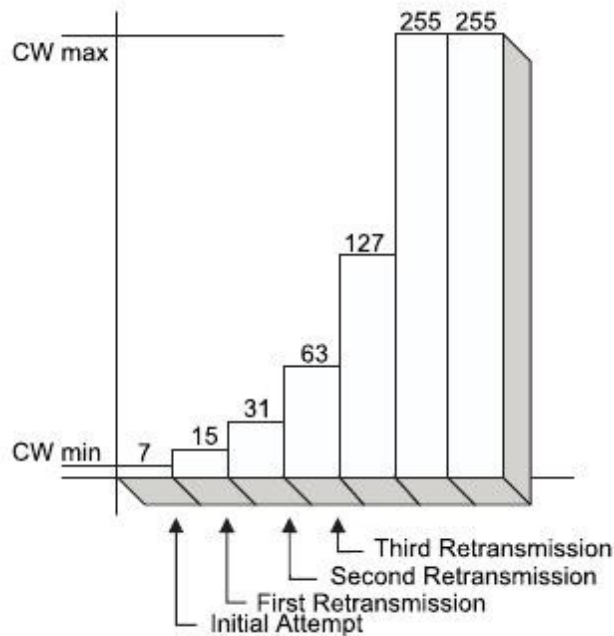
Inter Frame Spaces – jsou definovány 4 typy:

1. SIFS(Short Inter Frame Space) – používá se k oddělení signálů v rámci přenosu(například mezi Fragment-ACK). Je to nejkratší interval.
2. PIFS(Point Coordination IFS) – používán AP(nebo též Point Coordinator) k získání přístupu k médium před jinou stanicí. Jeho velikost je definována jako SIFS + Slot Time.
3. DIFS(Distributed IFS) – využíván mezi jednotlivými přenosy. Je definován jako PIFS + Slot Time.
4. EIFS(Extended IFS) – Používá se místo DIFS, pokud přenos, který začel nebyl úspěšný

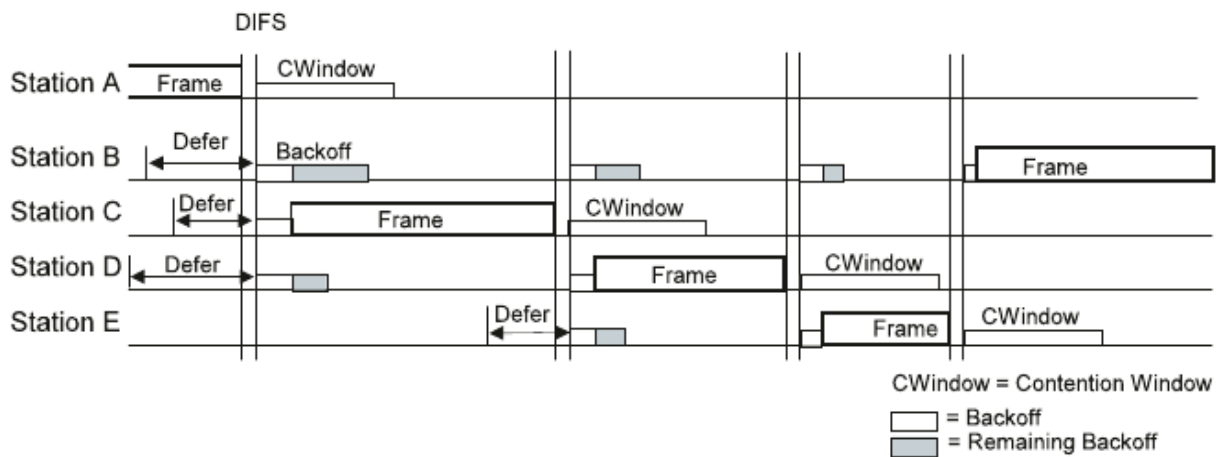
Slot Time – základní časová jednotka definovaná fyzickou vrstvou

Backoff algoritmus – Je algoritmus, který vygeneruje pseudonáhodně velký časový interval. Tento interval je generován před každým přenosem datového nebo management rámce. Vygenerovaný interval se během nečinnosti na médium zmenšuje. Vysílání začíná v okamžiku dosažení nulové hodnoty. Pokud by se během odpočtu změnil stav na médium, odpočet se zastaví. Velikost je

generována v rozmezí 0 až CW(Contention Window). Stále je však možné, že dojde ke kolizi. Pokud tedy dojde ke kolizi je CW zdvojnásobeno a začíná další pokus o odeslání(proto se též uvádí jako exponential backoff).



Ilustrace 5: Ukázka exponencionálního nárůstu CW



Ilustrace 6: Ukázka backoff procedury

5 IEEE 802.11e

V této kapitole se seznámíme blíže se změnami v IEEE 802.11e. Nejprve si popíšeme základní přístupové metody. Dále QoS (Quality of Service). Na závěr kapitoly popíšeme formát přenášeného rámce.

5.1 Metody přístupu k médiu

IEEE 802.11e vylepšuje DCF a PCF novou hybridní funkcí tzv. HCF (Hybrid Coordination Function).

HCF má dvě metody přístupu ke kanálu:

1. HCCA (HCF Controlled Channel Access – Přístupová metoda ke kanálům řízený HCF)
 - podobně jako u PCF, řídí AP komunikaci pomocí výzev jednotlivým stanicích. AP přiděluje právo vysílat na základě osmi indikátorů provozu (TSID – Traffic Stream Identifier).
 2. EDCA (Enhanced DCF Channel Access – Vylepšená DF přístupová metoda ke kanálům)
 - někdy též uváděno jako EDCF – Enhanced Distribution Coordination Function
 - jedná se o rozdělení šířky pásma na základě kategorií. Každá stanice má čtyři kategorie, na které se mapuje.
- obě metody definují TC (Traffic Classes - třídy provozu) viz Tab 4 a základní časování viz Tab 5

Priority	User Priority	Access Channel	Description
Lowest	0	AC_BK	Background
	1	AC_BK	Background
	2	AC_BE	Best Effort
	3	AC_BE	Best Effort
	4	AC_VI	Video
	5	AC_VI	Video
	6	AC_VO	Voice
Highest	7	AC_VO	Voice

Tabulka 4: Třídy provozu

AC	CW_{min}	CW_{max}	$AIFSN$
0	W_{min}	W_{max}	7
1	W_{min}	W_{max}	3
2	$\frac{wmin+1}{2}-1$	W_{min}	2
3	$\frac{wmin+1}{4}-1$	$\frac{wmin+1}{2}-1$	2

Tabulka 5: Základní časové parametry EDCF

– další přidané funkce:

- APSD (Automatic Power Save Delivery – automatická úspora energie) – jedná se o efektivnější metodu správy napájení než původní Power Save Polling
- BA (Block Acknowledgments – blokové potvrzování) – umožňuje potvrdit celý TXOP (Transmit Opportunity) v jediném rámci. To snižuje režii protokolu, zvláště při delších TXOP
- NoAck – služba pro posílání rámců v QoS módu může nabývat dvou hodnot: QoSAck a QoSNoAck. Rámce s QoSNoAck (QoS not acknowledged) nejsou potvrzovány

5.2 QoS

QoS (Quality of Service) – nám pomáhá řídit datové toky. Zajišťuje spolehlivé rozdělení a předchází tak zahlcování sítě. Například v síti, kde jeden uživatel začne stahovat například film a další uživatel by chtěl volat přes VoIP. Dojde k tomu, že druhý uživatel bude mít velké problémy s komunikací a také velkou latencí. Takováto situace nastane, pokud není uplatněn nějaký omezovač rychlosti nebo QoS.

U QoS se zájímáme o tyto parametry:

1. zpoždění – čas mezi odesláním a přijetím. Čím menší tím lepší
2. ztrátovost – poměr mezi počtem odeslaných a přijetých paketů
3. rozptyl zpoždění (jitter)
4. doručení mimo pořadí

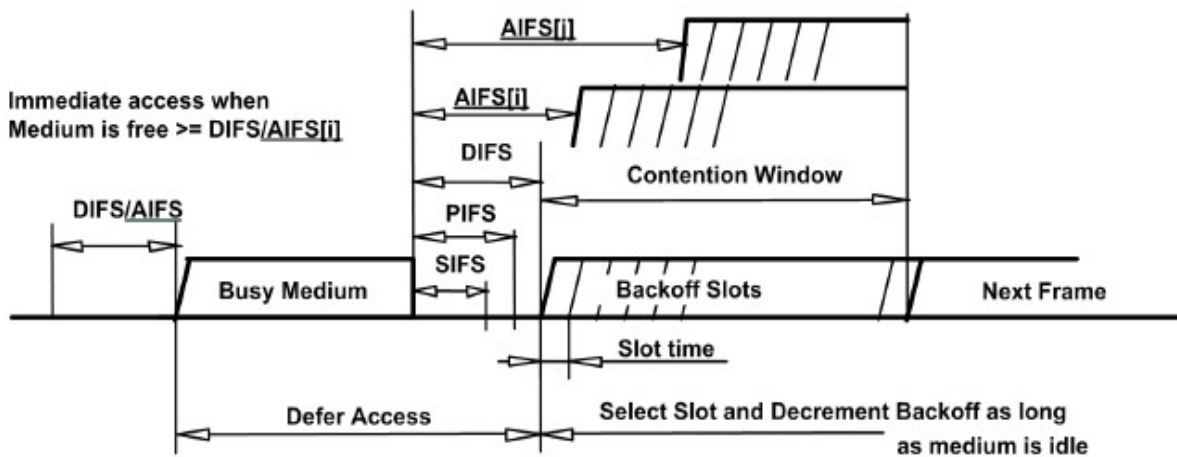
QoS limity pro DCF – DCF podporuje tzv. službu best-effort, ale nezaručuje QoS.

- Nepodporuje real-time aplikace
- Všechny stanice mohou přistupovat ke kanálu se stejnou prioritou

QoS limity pro PCF – třebaže byl PCF navržen pro časově závislé multimediální aplikace, má tři hlavní problémy, které vedou k malému výkonu QoS:

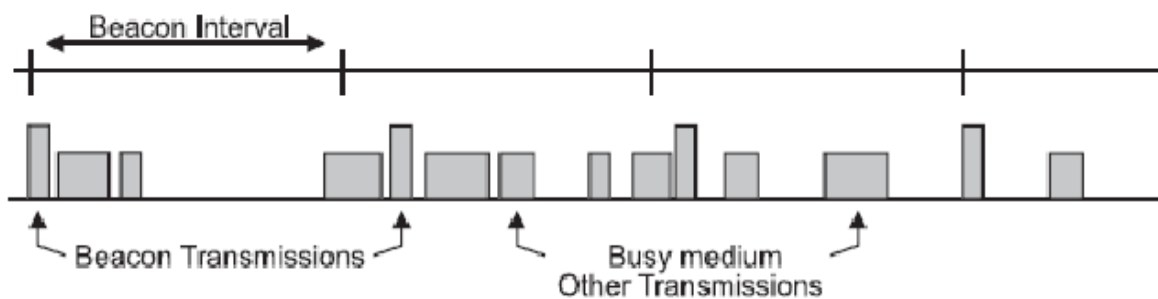
1. Celá komunikace mezi dvěma stanicemi v rámci jedné BSS vede skrz AP.
2. Stanicím je dovoleno začít vysílat svůj přenos dokonce když rámec přenosu nemůže být ukončen před příchodem TBTT(Target Beacon Transmission Time).
3. Měnící se délka rámce a stavu kanálu představuje změnu přenosového času.

AIFS(Arbitrate Inter Frame Space) – čas oddělující vysílání stanic, když je médium volné. AIFS se prodlužuje se snižující se prioritou. Tímto se zavádí řízeně neférový provoz. Je zavedeno v IEEE 802.11e. Ukázka AIFS viz Obr 7.



Ilustrace 7: Ukázka využití AIFS

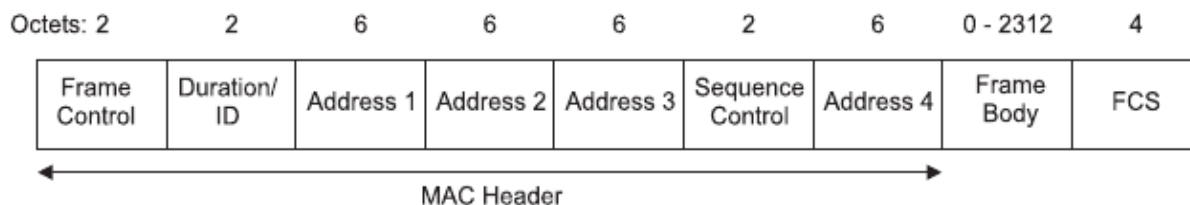
Synchronizace pomocí beaconu – Synchronizační signál může být vlivem komunikace odložen v DCF, jestliže v čase vysílání je médium obsazené. Viz Obr 8.



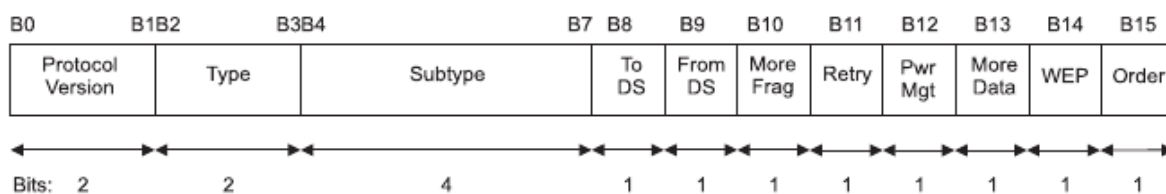
Ilustrace 8: Ukázka zpoždění beaconu

5.3 Formát rámců

- Každý rámec se skládá ze tří částí:
 1. MAC hlavička
 2. Data
 3. Kontrolní součet



Ilustrace 9: Obecný formát rámce



Ilustrace 10: Ukázka položek Frame Control

popis položek rámců:

Protocol Version – 2 bitové pole obsahující verzi použitého protokolu, nastaveno na 0, jiné hodnoty jsou rezervovány pro případné další použití

Type a Subtype – identifikují typ rámce(řídicí, management, datové) a jejich podtyp

ToDS/FromDS – význam je patrný následující tabulky

ToDS/FromDS	Význam
00	Rámce jsou posílány přímo mezi stanicemi IBSS
01	Data jsou určena pro DS(z STA do AP)
10	Data opouští DS(z AP do STA)
11	Data jsou přenášena v rámci DS(mezi AP)

Tabulka 6: Význam hodnot ToDS/FromDS

More Frag – identifikuje použití fragmentace dat při přenosu

Retry – je nastaveno na 1, pokud je přenos dat u datových a řídicích rámců opakován

Power Management – identifikuje použití režimu úspory energie stanicí

More Data – indikuje cílové stanici, zda vysílací stanice disponuje více než jedním rámcem určeným pro tuto stanici. Pokud cílová stanice používá úsporný režim, měla by vyčkat s přechodem do úsporného režimu na přenesení všech rámců(tj až do příjmu rámce s nastaveným bitem na 0)

6 Network Simulator NS-2

V této kapitole nejprve představím instalaci NS-2 a jeho součásti, dále instalaci rozšíření TKN a popis změn, které přináší.

NS-2 je nástroj na simulaci sítí, ať již drátových nebo bezdrátových. Jedná se o open-source projekt. Základní protokoly jsou naprogramovány pomocí C++ a jednotlivé skripty, se pak programují pomocí skriptovacího jazyka Tcl (viz následující kapitola). NS-2 je více platformový program. Dá se spustit i z Windows, avšak je potřeba použít Cygwin. Zvolil jsem tedy radši instalaci pod Linux, který je více vhodný pro tento program z pohledu uživatele. NS-2 obsahuje převážně základy všech sítí, ale neobsahuje úplně všechny normy. V tom spočívá výhoda i nevýhoda tohoto projektu. Pokud tedy potřebujeme nějakou síť simulovat a není implementována v NS-2 přímo, musíme si najít její základní normu a tu poté upravit. Základ NS-2 je napsán v jazyku C++ a proto, pokud se chystáme měnit nebo vylepšovat nějaké normy či protokoly je potřeba vždy znovu zkompileovat NS. V mé práci jsem používal verzi 2.32.

6.1 Instalce NS-2

Tento návod vychází z návodů získaných na stránkách [5][6]. Následující návod je psán pro instalaci pod Ubuntu. Pokud tedy používáte jiný systém, bude potřeba některé příkazy poupravit.

Postup instalace:

1. Nejprve je potřeba stáhnout NS2 z http://sourceforge.net/project/downloading.php?group_id=149743&use_mirror=puzzle&filename=ns-allinone-2.32.tar.gz&11742189
2. Vybalení archivu do domovského adresáře (může to být i jiný adresář)
3. Jelikož jsem použil novou instalaci Ubuntu, musel jsem pro úplnost doinstalovat ještě několik drobností, k čemuž slouží následující příkaz: `sudo apt-get install build-essential autoconf automake libxmu-dev`
4. Po spuštění vyžaduje předchozí příkaz ještě potvrzení "Y" pro doinstalování přibližně 14MB(záleží na konfiguraci systému – v mém případě čerstvě nainstalované Ubuntu 7.10) a vložení CD s Ubuntu 7.10 + stisk Enter
5. Po úspěšném doinstalování přichází samotná instalace NS-2. Tu spustíme příkazem: `./install`
6. Po spuštění následuje celkem dlouhá instalace
7. Pokud vše proběhne dobře, měl by se objevit následující výpis:

Nam has been installed successfully.

Please compile your gt-itm & sgb2ns separately.

Ns-allinone package has been installed successfully.

Here are the installation places:

tcl8.4.15: /home/milan/ns-allinone-2.32/{bin,include,lib}

tk8.4.15: /home/milan/ns-allinone-2.32/{bin,include,lib}

otcl: /home/milan/ns-allinone-2.32/otcl-1.13
tclcl: /home/milan/ns-allinone-2.32/tclcl-1.19
ns: /home/milan/ns-allinone-2.32/ns-2.32/ns
nam: /home/milan/ns-allinone-2.32/nam-1.13/nam
xgraph: /home/milan/ns-allinone-2.32/xgraph-12.1

Please put /home/milan/ns-allinone-2.32/bin:/home/milan/ns-allinone-2.32/tcl8.4.15/
unix:/home/milan/ns-allinone-2.32/tk8.4.15/unix

into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTICES:

(1) You MUST put /home/milan/ns-allinone-2.32/otcl-1.13, /home/milan/ns-allinone-2.32/lib,

into your LD_LIBRARY_PATH environment variable.

If it complains about X libraries, add path to your X libraries into LD_LIBRARY_PATH.

If you are using csh, you can set it like:

```
setenv LD_LIBRARY_PATH <paths>
```

If you are using sh, you can set it like:

```
export LD_LIBRARY_PATH=<paths>
```

(2) You MUST put /home/milan/ns-allinone-2.32/tcl8.4.15/library into your TCL_LIBRARY environmental variable. Otherwise ns/nam will complain during startup.

After these steps, you can now run the ns validation suite with `cd ns-2.32; ./validate`

For trouble shooting, please first read ns problems page

<http://www.isi.edu/nsnam/ns/ns-problems.html>. Also search the ns mailing list archive for related posts.

8. Nyní je potřeba upravit soubor `.bashrc` například příkazem: `gedit ~/.bashrc`
9. Otevře se okno se souborem. Na konec tohoto souboru je potřeba přidat následující řádky:
V mém případě jsem instaloval ns-2 do „/home/milan/“ tuto cestu je potřeba nahradit pokud

se neshoduje s vaší cestou.

```
# LD_LIBRARY_PATH
OTCL_LIB=/home/milan/ns-allinone-2.32/otcl-1.13
NS2_LIB=/home/milan/ns-allinone-2.32/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:
$X11_LIB:$USR_LOCAL_LIB
# TCL_LIBRARY
TCL_LIB=/home/milan/ns-allinone-2.32/tcl8.4.15/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB
# PATH
XGRAPH=/home/milan/ns-allinone-2.32/bin:/home/milan/ns-
allinone-2.32/tcl8.4.15/          unix:/home/milan/ns-allinone-2.32/tk8.4.15/unix
NS=/home/milan/ns-allinone-2.32/ns-2.32/
NAM=/home/milan/ns-allinone-2.32/nam-1.13/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

10. Nyní uložíme soubor a spustíme příkaz: `source ~/.bashrc`

11. Teď bychom měli spustit příkaz `./validate` pro kontrolu

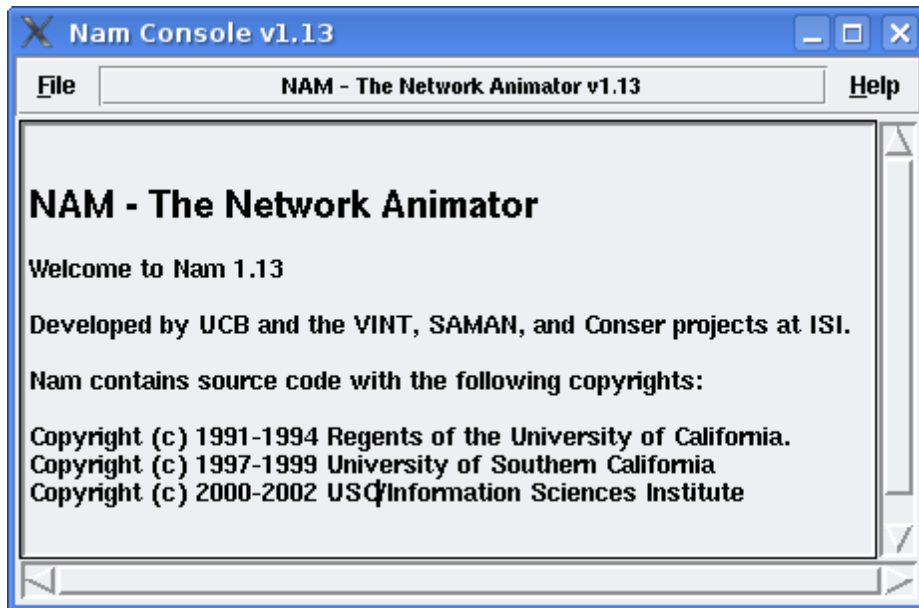
12. Pokud vše proběhne dobře můžeme spouštět NS-2 pomocí příkazu `./ns` nebo `ns`

6.2 Součásti NS-2

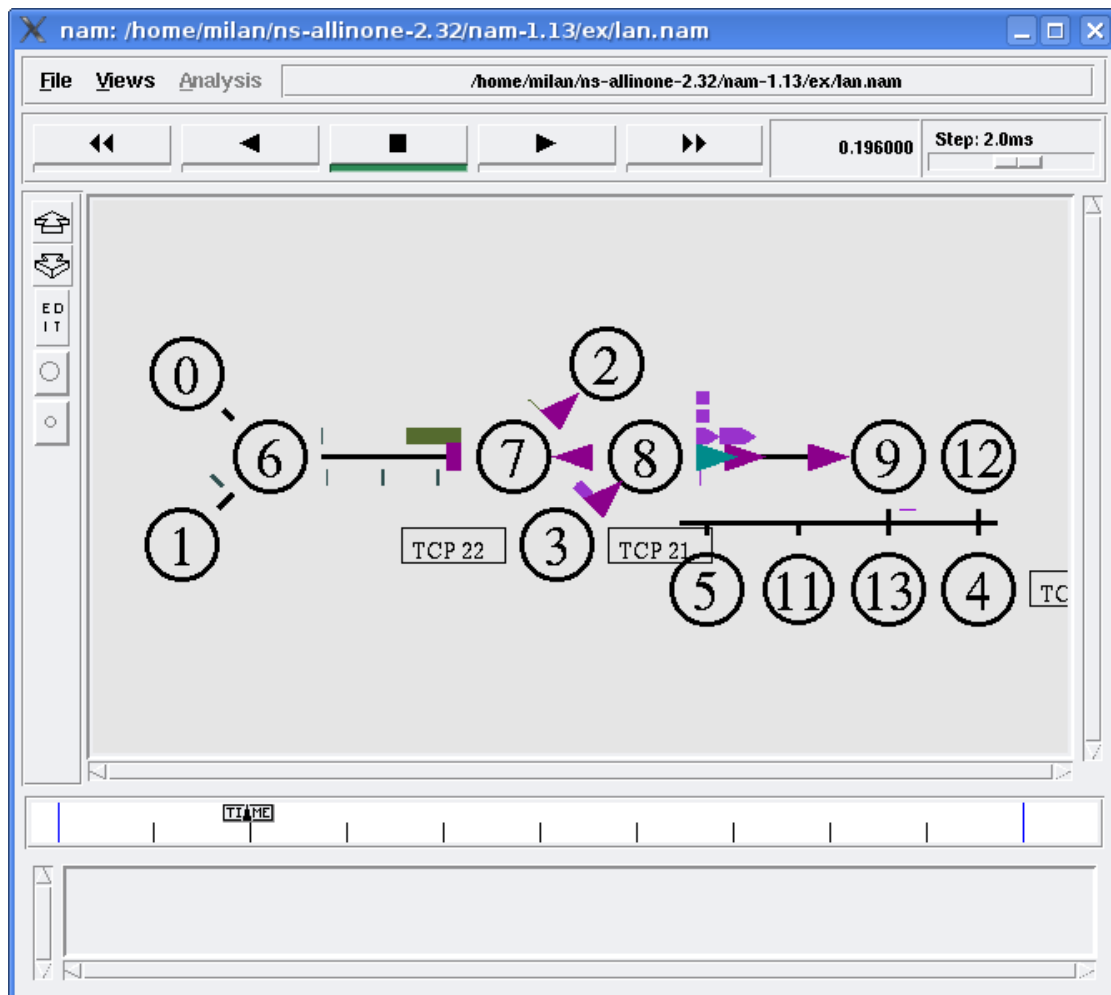
NS-2 se skládá ze dvou hlavních částí NS a NAM(The Network Animator). První jmenovaný slouží jako kompilátor napsaných kódů, nejčastěji námi vytvořených kódů pomocí skriptovacího jazyka Tcl. Druhý jmenovaný slouží, zejména pro vizualizaci chování sítě. Verze 2.32 se kterou sem pracoval obsahuje pouze základní IEEE 802.11. Abych tedy mohl testovat standard IEEE 802.11e, musel jsem využít kompilační balík od TKN TU Berlin.[7]

6.3 NAM

NAM slouží pro vizualizaci sítě. Má vlastní grafické prostředí, které může posloužit při základním návrhu sítě. Po vytvoření návrhu sítě můžeme využít vygenerovaný kód a usnadnit si tak práci, tím že již budeme mít předdefinovaný typ spojení uzlů, jejich rychlosti. Toto nám usnadní práci s následnou tvorbou scriptu pro NS.



Ilustrace 11: Základní okno NAM



Ilustrace 12: Ukázka navržené sítě pomocí NAM

6.4 Záznam dat

Abychom mohli jednotlivá data analyzovat potřebujeme získat přehled o tom co se děje během spuštění simulace. K tomu nám velmi dobře poslouží tzv. Trace-file(„trasování“). Z těchto dat je pak možné vygenerovat různé grafy. Zobrazování grafů probíhá převážně pomocí programu Xgraph. Dá se samozřejmě zvolit i jiný, ale vše záleží pouze na tom co nám vyhovuje lépe.

6.5 Rozšíření NS-2 od TKN

6.5.1 Instalace rozšíření od TKN:

1. Stáhnutý archiv rozbalíme do adresáře /home/milan/ns-allinone-2.32/ns-2.32/mac
2. Nyní je potřeba změnit obsah souboru v /home/milan/ns-allinone-2.32/ns-2.32/Makefile.in

přidáme do INCLUDES:

```
-I./mac/802_11e \
```

přidáme do OBJ_CC:

```
mac/802_11e/mac-802_11e.o mac/802_11e/priq.o \  
mac/802_11e/d-tail.o mac/802_11e/mac-timers_802_11e.o \
```

vymažeme v NS_TCL_LIB:

```
tcl/lib/ns-mobilenode.tcl \
```

přidáme do NS_TCL_LIB:

```
mac/802_11e/ns-mobilenode_EDCA.tcl \  
mac/802_11e/priority.tcl \
```

3. Nyní změním v /home/milan/ns-allinone-2.32/ns-2.32/tcl/lib/ns-lib.tcl

vymažeme z the source list:

```
source ns-mobilenode.tcl
```

přidáme do the source list:

```
source .././mac/802_11e/ns-mobilenode_EDCA.tcl  
source .././mac/802_11e/priority.tcl
```

4. Nyní změním v /home/milan/ns-allinone-2.32/ns-2.32/tcl/lib/ns-default.tcl

přidáme následující řádky:

```
Queue/DTail set drop_front_ false  
Queue/DTail set summarystats_ false  
Queue/DTail set queue_in_bytes_ false  
Queue/DTail set mean_pktsize_ 500  
Queue/DTail/PriQ set Prefer_Routing_Protocols 1  
Queue/DTail/PriQ set Max_Levels 4  
Queue/DTail/PriQ set Levels 4
```

```

Mac/802_11e set SlotTime_ 0.000020 ;# 20us
Mac/802_11e set SIFS_ 0.000010 ;# 10us
Mac/802_11e set PreambleLength_ 144 ;# 144 bit
Mac/802_11e set PLCPHeaderLength_ 48 ;# 48 bits
Mac/802_11e set PLCPDataRate_ 1.0e6 ;# 1Mbps

Mac/802_11e set RTSThreshold_ 3000 ;# bytes
Mac/802_11e set ShortRetryLimit_ 7 ;# retransmissions
Mac/802_11e set LongRetryLimit_ 4 ;# retransmissions

```

5. Nyní přidáme do add v tcl/lan/ns-mac.tcl

```

# IEEE 802.11e MAC settings
if [TclObject is-class Mac/802_11e] {
    Mac/802_11e set delay_ 64us
    Mac/802_11e set ifs_ 16us
    Mac/802_11e set slotTime_ 16us
    Mac/802_11e set cwmin_ 16
    Mac/802_11e set cwmax_ 1024
    Mac/802_11e set rtxLimit_ 16
    Mac/802_11e set bssId_ -1
    Mac/802_11e set sifs_ 8us
    Mac/802_11e set pifs_ 12us
    Mac/802_11e set difs_ 16us
    Mac/802_11e set rtxAckLimit_ 1
    Mac/802_11e set rtxRtsLimit_ 3
    Mac/802_11e set basicRate_ 1Mb ;# set this to 0 if want to use bandwidth_ for
    Mac/802_11e set dataRate_ 1Mb ;# both control and data pkts
    Mac/802_11e set cfb_ 0 ;# disables CFB
}

```

6. V souboru mac/wireless-phy.h vyměníme:

```
enum ChannelStatus { SLEEP, IDLE, RECV, SEND };
```

za

```
enum ChannelStatus { SLEEP, IDLE, RECVING, SENDING};
```

7. V souboru mac/wireless-phy.cc vyměníme všechny výskyty:

RECV a SEND (za RECVING a SENDING).

8. V souboru common/packet.h přidáme:

```
#define HDR_MAC802_11E(p) ((hdr_mac802_11e *)hdr_mac::access(p))
```

9. Nyní spustíme příkaz ./configure

10. Spustíme následující příkazy v našem NS adresáři: make clean, make depend, make

11. Nyní by již mělo být vše připraveno k simulaci.

– Přibyla nám teď možnost nastavit prioritu. V simulačním skriptu po na definování agenta nastavit prioritu

% \$your_transport_agent prio_x (př. \$src_udp0 set prio_0), kde x je v rozmezí mezi 0 a 3, 0 je nejvyšší a 3 nejnižší priorita).

– změnu 802.11e parametrů (CW_MIN, CW_MAX, AIFS, TXOPLimit, PF) můžeme udělat v souboru ../ns-x/mac/802_11e/priority.tcl. Avšak po každé takové to změně je potřeba znovu zkompileovat pomocí make.

6.5.2 Popis změn v rozšíření TKN

Kompilační balík by měl rozšířit základní standard IEEE 802.11 o EDCF kombinovaný s CFB(Contention Free Bursting) a opravit par chyb v IEEE 802.11 DCF modelu.

Pro úspěšné použití je potřeba v Tcl skriptu použít následující řádky:

```
set opt(mac) Mac/802_11e
```

```
set opt(ifq) Queue/DTail/PriQ
```

Pro nastavení jednotlivých parametrů pro každou prioritu je potřeba využít soubor priority.tcl

ukázka ze souboru:

```
#parameters for Queue 2
```

```
$ifq Prio 2 PF 2
```

```
$ifq Prio 2 AIFS 3
```

```
$ifq Prio 2 CW_MIN 31 ;# aCWmin
```

```
$ifq Prio 2 CW_MAX 1023 ;# aCWmax
```

```
$ifq Prio 2 TXOPLimit 0
```

Pro přístup k těmto parametrům pak můžeme využít funkce setCW_MIN() nebo get, atd. Tyto funkce jsou naprogramovány v d-tail.cc/h

7 TCL/Tk

V této kapitole si popíšem základní vlastnosti programovacího jazyku Tcl/Tk a ukážeme si základní příkazy pro práci s Tcl. Grafickou knihovnu Tk zde popisovat nebudu, jelikož není potřeba pro tuto práci. Tuto kapitolu jsem vytvořil pomocí dokumentace na stránkách tcl.tk [8]

TCL (Tool Command Language)/Tk(Toolkit)

Tcl je programovací jazyk, který se specializuje na rychlou tvorbu prototypových aplikací. Nejčastěji dodáván s knihovnou Tk, jenž slouží pro návrh grafických prostředí. Mezi nesporné výhody patří zejména možnost volat z programu napsaném v Tcl jiné programy nebo utility napsané v jiných programovacích jazycích. Tímto můžeme libovolně rozšiřovat Tcl o další funkce.

Tcl vychází z programovacího jazyk Lisp. Nicméně přináší pro uživatele zjednodušující syntaktické pravidla. Výsledný zdrojový kód je tak mnohem přehlednější a lépe pochopitelný.

Tcl byl vytvořen Johnem Ousterhoutem v osmdesátých letech dvacátého století.

7.1 Vlastnosti jazyka Tcl

Tcl se snaží vylézt nejvíce z faktu, že je interpretovaný. Samotný jazyk neobsahuje žádné smyčkové příkazy nebo podmínky a ani matematické či logické funkce. Ty jsou však implementovány z jiných programovacích jazyků (v našem případě C++).

Základní vlastnosti:

- vše je bráno jako řetězec, dokonce i samotný kód programu
- se vším zacházíme jako s řetězcem
- vše můžeme dynamicky přetížit nebo předefinovat
- snadná rozšiřitelnost
- změna kódu za běhu
- plná podpora Unicode
- nezávislost na platformě
- snadný zápis

Jelikož jde o interpretovaný jazyk, nemůžeme v průběhu programu pomocí překladače vytvořit spustitelný tvar programu. Nejprve se provádí převod do bytekódu a samotný převod do strojového kódu se provede až při spuštění či změně programu. Jedná se o tzv. překlad just-in-time.

7.2 Přednosti Tcl

Přednosti Tcl:

- jednoduchost tvorby aplikací
- snadné ladění

- návaznost na ostatní programovací jazyky
- přenositelnost mezi platformami
- podpora kódování řetězců v Unicode

Použitím Tcl můžeme využít například rychlosti C a dynamičnost Tcl. Aplikaci můžeme obecně vytvořit dvěma způsoby:

1. řídicí část programu je napsána v Tcl, složitější či časově náročné funkce jsou vytvořeny v jiném programovacím jazyce
2. řídicí část je napsána v jiném jazyce a Tcl používáme pro skriptování nebo práci s grafickým rozhraním. U tohoto přístupu je však trochu těžší návaznost na Tcl, zejména předávání parametrů.

7.3 Zápory Tcl

Ně největší nevýhodou je samozřejmě nižší rychlost provádění skriptů. Pokud Tcl používá například pro tvorbu grafického prostředí můžeme kritické části kódu přepsat například do C. K dalším nevýhodám patří některá omezení jazyka, například zde chybí možnost psaní čistě objektově orientovaných programů (OOP). Návrh jazyka s OOP primárně nepočítal, přesto však můžeme využít jiných programovacích jazyků a přidat tak OOP.

7.4 Rozšíření Tk

John Ousterhout vytvořil tuto knihovnu pro práci s grafickým uživatelským rozhraním. Knihovna umožňuje vytvářet okna dialogy spolu s umístováním jednotlivých ovládacích prvků. Samozřejmostí je přenositelnost celé knihovny na různé platformy. Grafické uživatelské rozhraní se přizpůsobí dané platformě a to včetně stylů oken, způsobu zaměrování widgetů apod.

7.5 Syntaxe

Na začátek jde potřeba říci, že v Tcl jsou od sebe jednotlivé příkazy odděleny buď koncem řádky nebo středníkem. Slova jsou brány jako řetězce, avšak nemusí se používat uvozovky pokud nejsou v zápisu úmyslné mezery. Pokud je na řádce pouze jeden příkaz nemusí se ukončovat středníkem. Příkaz je složen z jednoho nebo více slov, které od sebe oddělujeme „bílymi znaky“ mezera nebo tabulátor. První slovo je vždy chápáno jako jméno a další jako slova jsou parametry. Parametry se nevyhodnocují.

7.5.1 Komentáře

Každý komentář začíná symbolem #. Cokoliv je napsáno za # až do konce řádku je ignorováno.

Ukázka:

```
# ukázka komentáře
puts Hello World!; # vypíše Hello World!
```

7.5.2 Proměnné

Je zde jediné hlavní pravidlo pro práci s proměnnými a to, že každá proměnná musí být před prvním použitím inicializována. Inicializace se provádí příkazem **set**, po kterém následuje jméno a hodnota. Hodnota může být vyjádřena konstantou nebo výrazem, který se vyhodnotí.

Ukázka:

```
set objem 50
```

```
set nazev „V – objem“
```

Tcl nezná žádný datový typ a proto jsou obě proměnné považovány za řetězce. Uvozovky se musely použít u druhé proměnné, aby bylo použito „V - objem“, protože jsou v řetězci mezery.

Pokud bychom chtěli proměnou zrušit použijeme příkaz **unset**. Po provedení příkazu je proměnná uvolněna z paměti.

Ukázka:

```
unset objem
```

Pokud chceme vypsat proměnnou použijeme příkaz **puts**. Za puts vložíme jméno proměnné. Pokud potřebujeme získat, ale hodnotu proměnné vložíme před název **\$**.

Ukázka:

```
set objem 50
```

```
set nazev „V – objem“
```

```
puts objem; # vypíše slovo objem
```

```
puts $objem; # vypíše 50
```

Pokud bychom potřebovali zakázat substituci můžeme využít **{}**.

Ukázka:

```
set objem 50
```

```
puts “Vysledny objem je $objem”; # vypíše Vysledny objem je 50
```

```
puts {Vysledny objem je $objem}; # vypíše Vysledny objem je $objem
```

další možností zrušení substituce je vložení **** před **\$**

Ukázka:

```
puts “Vysledny objem je \$objem“
```

Složené závorky používáme například pro zápis těla smyček, protože nechceme aby se tělo smyčky provedlo již při jejím vytváření, ale až po spuštění.

Pro situaci kdy je potřeba vykonat nějaký příkaz a jeho návratovou hodnotu předat do dalšího příkazu použijeme **[]**. Kód umístěný do závorek se provede nejdříve a pak se výsledek vrátí nadřazenému příkazu. Počet takto vnořených příkazů není omezen.

Znak	Význam
\$	Nahradí jméno proměnné její hodnotou
[]	Vyhodnocení příkazu. Obsah závorek se vyhodnotí nejdříve.
“ “	Potlačí zobrazení mezer jako oddělovač příkazů.
{ }	Stejně jako uvozovky, jen substituce uvnitř se nenahrazují.
\	Ruší význam následujícího znaku

Tabulka 7: Substitute

7.5.3 Matematické funkce

Tcl nevlastní žádné speciální konstrukce pro zápis matematických operací. Tcl využívá příkaz **expr**, který vyhodnotí následný matematický zápis.

Ukázka:

expr 1 + 1

expr 10.0 / 5

expr 1 << 10

expr \$a + \$b

Příkaz **expr** tvoří výjimku, můžeme v něm používat jak konstanty(celočíselné nebo reálné), tak i proměnné. Všechny argumenty se musí normálně oddělovat bílým znakem, avšak **expr** tvoří výjimku. I když znaky neoddělíme mezerou, budou vyhodnoceny jako řetězec, který bude po zavolání **expr** rozložen na jednotlivé části. Příkaz **expr** můžeme používat společně s dalšími příkazy.

Ukázka:

set a [expr 1 + 1]

set b 3

set c "\$a + \$b se rovna [expr \$a + \$b]"

Přehled podporovaných operací:

Operace	Význam
+	Unární plus
-	Unární mínus
+	Součet
-	Rozdíl
*	Součin
/	Podíl
%	Zbytek po dělení

Tabulka 8: Podporované funkce

samořejmě jsou zde i další matematické operace:

acos, asin, atan, atan2, ceil, cos, floor, fmod, hypot, log, log10, pow, sin, sinh, sqrt, tan, tanh

7.5.4 Logické a bitové funkce

Funkce vychází z jazyka C. Využít je můžeme například v podmínkách.

Operace	Význam
<<	Bitový posun doleva
>>	Bitový posun doprava
~	Negace bit po bitu
!	Logická negace
&	Bitový operátor AND
	Bitový operátor OR
^	Bitový operátor XOR
&&	Logický součin
	Logický součet
?:	Ternární podmíněný výraz
<	Menší než
>	Větší než
<=	Menší nebo rovno
>=	Větší nebo rovno
==	Rovnost
!=	Nerovnost

Tabulka 9: Logické a bitové funkce

7.5.5 Funkce

Funkce Tcl se dělí na:

1. vestavěné funkce
2. vytvořené funkce pomocí klíčového slova **proc**
3. externí funkce

Pro vytvoření nové funkce se používá příkaz **proc**, za ním následuje jméno, poté parametry a následně tělo funkce. Funkce může být kdykoliv přerušena pomocí **return**(může vrátet hodnotu).

Ukázka:

```
proc pozdrav {} {  
    puts "Hello World"  
}
```

Ukázka funkce s parametry:

```
proc secti {a b} {  
    return [expr $a+$b]  
}  
  
proc log_funkce {operator} {  
    puts [expr 0 operator 0]  
    puts [expr 0 operator 1]  
    puts [expr 1 operator 0]  
    puts [expr 1 operator 1]  
}
```

Funkce log_funkce vypíše pravdivostní tabulku libovolné logické funkce.

Ukázka volání funkce:

```
log_funkce &  
log_funkce |
```

7.5.6 Podmíněné příkazy

Tcl má dvě možnosti větvení programu. Jednak pomocí **if-then-else** nebo pomocí **switch**. Podmínka if-then může být zapsána mnoha způsoby. Samotné slovo then/else není vůbec potřebné psát.

Ukázka:

```
if podmínka tělo_podmínky tělo_druhé_větve  
if podmínka tělo_podmínky else tělo_druhé_větve  
if podmínka then tělo_podmínky tělo_druhé_větve  
if podmínka then tělo_podmínky else tělo_druhé_větve
```

Pro lepší větvení můžeme též použít **elseif**(toto se musí uvádět na rozdíl od **then** a **else**).

Ukázka:

```
if podmínka tělo1 elseif podmínka2 tělo2 tělo_else  
if podmínka then tělo1 elseif podmínka2 then tělo2 else tělo_else
```

Vzhledem k tomu, že Tcl vyhodnocuje po jednotlivých řádkách nesmí else začínat na novém řádku.

Ukázka:

```
if {výraz} {  
    tělo_podmínky  
} else {  
    tělo_druhé_větve  
}
```

Jelikož u Tcl se porovnávají řetězce, tak příkaz **switch** může vyhodnocovat regulární výrazy.

Ukázka obecného zápisu:

switch řetězec *vzor1* větev1 *vzor2* větev2 ... *default* větev_n

Jednotlivé větve jsou pak uvedeny v složených závorkách.

7.5.7 Cykly

Základním typem smyčky je **while**.

Syntaxe:

while podmínka tělo_smyčky

Tento cykl funguje stejně jako například v jazyku C. Nejprve se vyhodnotí podmínka, jestliže podmínka je true provede se tělo cyklu, jinak program pokračuje dál.

Ukázka:

```
set i 1
while {i < 10} {
  puts $i
  incr i
}
```

Příkaz **incr** zvyšuje nebo snižuje proměnou a zadané číslo, pokud není zadáno implicitně se nastaví 1.

Ukázka:

```
incr i 2
incr i -1
```

Cyklus **for**

Syntaxe:

for start test interační_příkaz tělo_smyčky

Ukázka:

```
for {set i 5} {$i} {incr i -1} {
  tělo_smyčky
}
```

V obou typech smyček lze použít příkaz **break** a **continue**. **Break** způsobí, že se smyčka předčasně ukončí. **Continue** způsobí skok na začátek smyčky.

7.5.8 Strukturované datovém typy

V Tcl není problém pracovat i s asociativním polem nebo seznamem. Jelikož však Tcl převádí vše na řetězec můžeme toho využít a definovat pole pomocí klíče, který je také řetězec. Pole můžeme plnit i za pomoci smyčky.

Ukázka:

set pole(0) 50

set slovník(pocitac) computer

Název funkce	Význam funkce
array get pole	Vrací všechny hodnoty klíčů i prvků pole
array get pole vzor	Vrací všechny hodnoty klíčů i prvků pole, kde klíče odpovídají vzoru
array names pole	Vrací všechny klíče(indexy) pole
array names pole vzor	Vrací všechny klíče pole, které odpovídají vzoru
array set pole seznam	Vytváří pole ze seznamu
array exists pole	Provede ověření, zda existuje pole o zadaném názvu. Vrací 0 nebo 1
array size pole	Vrátí počet prvků v poli

Tabulka 10: Funkce pro pole

Seznamy se vytvoří pomocí příkazů set.

Ukázka:

set seznam1 { 1 2 3 4 5 }

Název	Význam
list	vytvoření seznamu z argumentů, které jsou tomuto příkazu zadány
concat	spojení dvou a více seznamů
llength	získání počtu prvků v seznamu
split	rozložení řetězce na seznam buď podle bílých znaků nebo podle specifikovaného oddělovače
join	vytvoření řetězce spojením prvků seznamu, mezi něž se může volitelně vložit oddělovač
lappend	přidání jednoho či více prvků do seznamu
linsert	vložení jednoho či více prvků na danou pozici (index)
lindex	získání prvku ze seznamu na dané pozici (indexu)
lreplace	nahrazení prvků v seznamu
lrange	vyjmutí více prvků ze seznamu (souvislá oblast)
lsearch	hledání prvků v seznamu (možné i podle regulárních výrazů!)
lsort	setřídění prvků v seznamu podle zadaných kritérií

Tabulka 11: Funkce pro práci se seznamy

Pro práci se seznamy můžeme sice použít cykly while a for, ale není to příliš efektivní, proto byla do Tcl přidána funkce **foreach**, která nám práci ulehčuje.

Syntaxe:

```
foreach proměnná seznam tělo_smyčky
```

Ukázka:

```
foreach i {jedna dve tri} {  
    puts $i  
}
```

7.5.9 Práce se soubory

Pro otevření se používá příkaz **open**.

Syntaxe:

```
open jménoSouboru přístup práva
```

přístup:

read – pro čtení

write – pro zápis

append – pro přidání

Pro zavření souboru používáme příkaz **close**.

Syntaxe:

```
close kanál
```

Kde kanál je interní hodnota, kterou vrací příkaz open.

Pro práci s obsahem souboru využijeme funkce gets a puts.

Syntaxe:

```
gets kanál proměnná
```

```
puts kanál řetězec
```

```
puts -nonewline kanál řetězec
```

parametr -nonewline pouze způsobí, že nebude zapsán znak pro odřádkování

Dále můžeme použít funkci read, která načítá pouze námi zvolený počet znaků. Pokud nezadáme počet znaků načte se celý obsah souboru.

Syntaxe:

```
read kanál početZnaků
```

Ukázka:

```
set soubor [open cetba.txt "r"]
```

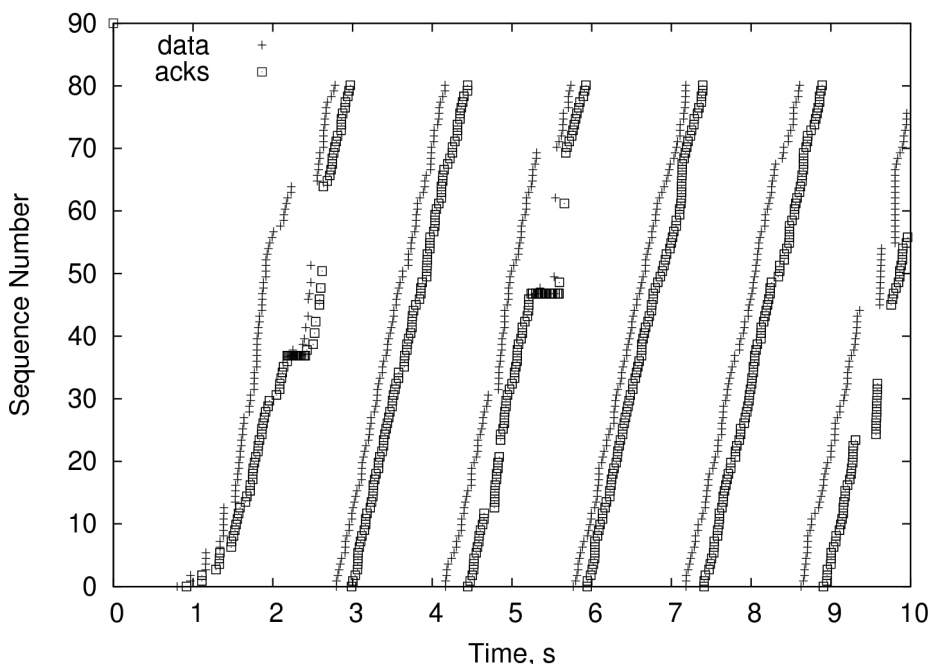
```
close $soubor
```

8 Simulace IEEE 802.11 a IEEE 802.11e

V této kapitoly ukáží vygenerované grafy a popíší základní nastavení node pro Wi-Fi sítě založené na normě IEEE 802.11.

8.1 Test IEEE 802.11

Pro vygenerování grafu využiji již předepsaný skript z adresáře v ns-2 /tcl/ex/wireless-scripts/, který je součástí instalace. Tento skript vygeneruje několik grafů. Zvolil jsem graf, který zobrazuje počty doručených paketů a ACK. Jedná se o graf, který byl vytvořen před instalací doplňku od TKN.



Ilustrace 13: graf simulace wlan_complex IEEE 802.11

8.2 Popis základních parametrů simulace

základní popis nastavení node:

Nastavení probíhá pomocí příkazu: `$ns_ node-config parametr nastavení_paramtru`

Parametry:

`-addressingType` *flat, hierarchical*

hierarchical se nejčastěji používá pro wireless sítě

`-adhocRouting` *DSDV, DSR, TORA, AODV*

Mezi podporované adhoc routing protokoly patří:

- Destination Sequence Distance Vector (DSDV)
- Dynamic Source Routing (DSR)
- Temporally ordered Routing Algorithm (TORA)

- Adhoc On-demand Distance Vector (AODV)

-llType *LL* – LinkLayer
-macType *Mac/802_11* – typ MAC vrstvy
-propType *Propagation/TwoRayGround*
-ifqType *Queue/DropTail/PriQueue* – představuje typ podporovaných front
-ifqLen *50* – velikost fronty
-phyType *Phy/WirelessPhy* – typ síťového rozhraní
-antType *Antenna/OmniAntenna* – typ antény
-channelType *Channel/WirelessChannel* – typ kanálu
-topoInstance *<the topography instance>* - typ topografie
-wiredRouting *ON* nebo *OFF* – zapíná nebo vypíná sledování na úrovni spojení
-mobileIP *ON* nebo *OFF* – zapíná nebo vypíná mobilní IP adresu
-energyModel *EnergyModel*
-initialEnergy hodnota – udává se v J
-rxPower hodnota – udává se ve W
-txPower hodnota – udává se ve W
-agentTrace *ON* nebo *OFF* – zapíná sledování na úrovni agenta
-routerTrace *ON* nebo *OFF* – zapíná sledování na úrovni routru
-macTrace *ON* nebo *OFF* – zapíná sledování na úrovni MAC
-movementTrace *ON* nebo *OFF* – zapíná záznam pohybu mobilních stanic

8.3 Kreslení grafů

Ke kreslení můžeme využít programy Xgraph nebo gnuplot. Já se zde budu zabývat převážně programem Xgraph, protože je součástí ns. Jeho ovládání není složité a umožňuje snadný export obrázků.

Základní parametry programu Xgraph:

-t “název grafu“
-geometry *XxY* – velikost rozlišení okna
-x “název osy x“
-y “název osy y“

Ukázka:

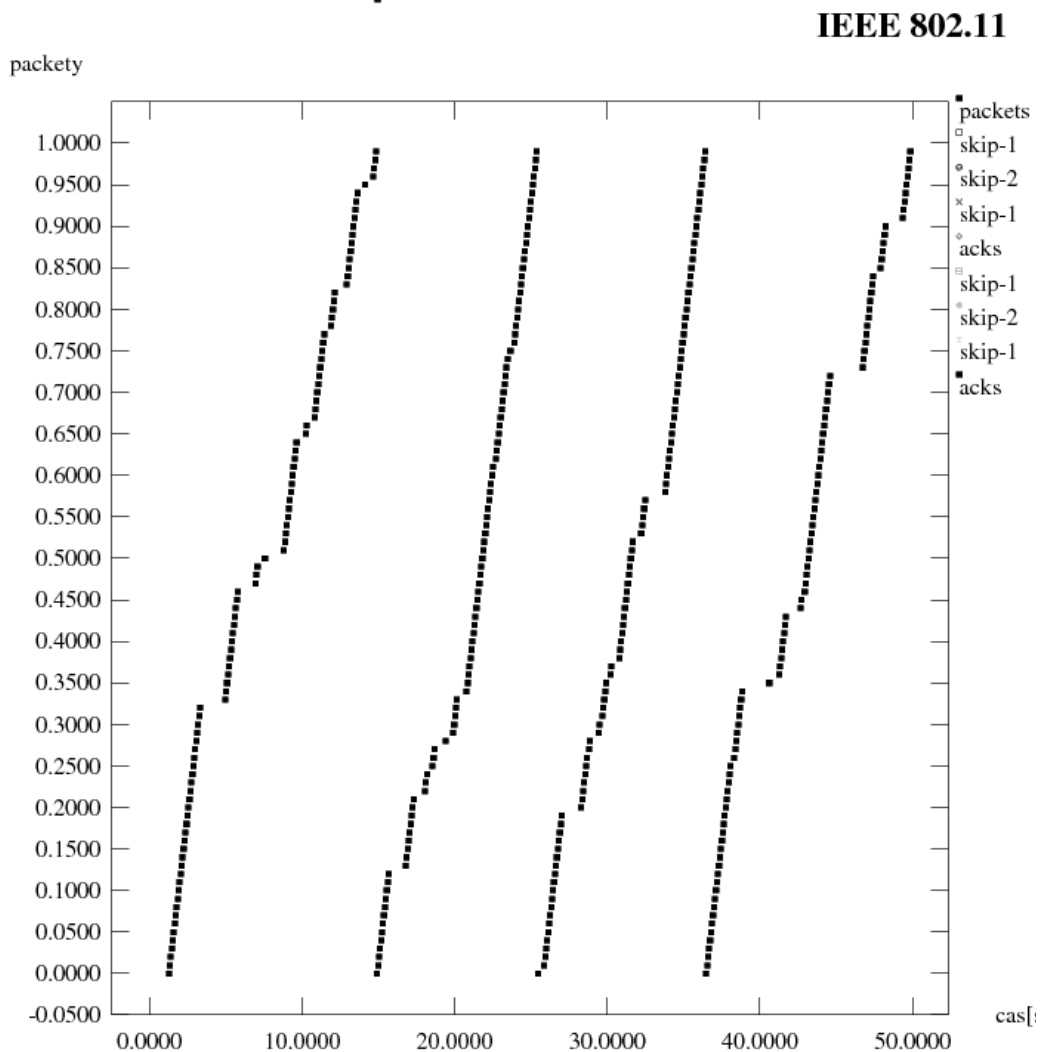
xgraph *nazev_souboru* **-geometry** 800x400 **-t** “Propustnost“ **-x** “cas“ **-y** “pocet paketu“

8.4 Testování

Předtím než začnu popisovat vygenerované grafy je potřeba vysvětlit ještě jeden pojem a to Agent. V rámci NS se jedná vlastně o takového „agenta“, který nám v místě kam ho přidáme buď vysílač nebo přijímač, vytvoří „sledovací bod“. Máme několik typů agenta např. TCP, UDP, které nám definují jaký přenos bude probíhat. Každý z těchto agentů má svůj přijímač např. Null, Sink, to jsou agenti na straně přijímače. Tyto agenti nám tak pomohou definovat typ paketu, který bude na daném spojení přenášen. Pro úspěšné použití agentů je ještě potřeba definovat jejich spojení a přidat jim typ datového toku například CBR (constant bit rate) nebo Exponential.

Skripty popsané níže se spouští příkazem: `ns nazevskriptu`

Pro základní test IEEE 802.11 využiji Agentu UDP a Exponential. Viz. zdrojový soubor *prvni.tcl*. Po spuštění tohoto skriptu jsou vygenerovány 2 soubory: *prvniout.nam* a *prvniout.tr*. Soubor *prvniout.nam* je pouze záznam probíhajících přenosů. Soubor *prvniout.tr* obsahuje data pro tvorbu grafů. V příloze jsou přidány soubory *IEEE 802.11.ps* a *IEEE 802.11_2.ps*. První graf znázorňuje jednosměrný přenos z node0 do node1 viz Obr. 14.



Ilustrace 14: Ukázka výstupu z prvni.tcl

Druhý graf znázorňuje obousměrný přenos mezi oběma. Viz příložený soubor *IEEE 802.11_2.ps*, jenž byl vygenerován pomocí skriptu *druhy.tcl*. Tento skript se liší od předchozího pouze přidáním dalších agentů a jejich spojení pro vysílání komunikace.

V dalším souboru *treti.tcl*. Aplikuji rozšíření od TKN. V základě se jedná o úpravu předchozího zdrojového kódu, tak aby vyhovoval rozšíření od TKN. Bylo tedy nutné změnit pár základních nastavení a to upravit předchozí kód:

```
set opt(mac)      Mac/802_11e
set opt(ifq)      Queue/DTail/PriQ
```

a poté ještě nastavení priority přenosu:

```
$udp2 set prio_ 0
```

V posledním skriptu *ctvrty.tcl* je zobrazen přenos paketů v závislosti na čase. Původní zdrojový kód *example4.tcl*, který se nachází v adresáři */tcl/ex/* jsem předělal z drátové sítě do wi-fi sítě a upravil záznam dat, tak aby zaznamenal počet obdržených paketů v závislosti na čase.

8.5 Popis Trace-file souboru

Pro následující sem využij popis v [15]. Jedná se o soubry s příponou *.tr. V těchto souborech jsou zaznamenány údaje celé komunikace.

Obecný popis obsahu souboru:

Event	Time	From node	To node	Pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id
-------	------	-----------	---------	----------	----------	-------	-----	----------	----------	---------	--------

Event – obsahuje symboly r, +, -, d.

r – receive(obdržení)

+ - enqueued(přijmutí do fronty)

- - dequeued(odchod z fronty)

d – dropped(zahození)

Time – čas, kdy daná událost nastala

From node – z jakého node přichází

To node – do jakého node je směřována

Pkt type – typ paketu(může být např. TCP, ACK apod.)

Pkt size – velikost paketu(udávána v bytech)

Flags - flags

Fig – flow id podle IPv6

Src addr – adresa zdroje

Dst addr – adresa cíle

Seq num – sekvence čísel pro určení konkrétního paketu

Pkt id – unikátní číslo paketu

Pokud bychom chtěli zjistit propustnost, je možné využít skriptu napsaného v perlu. Jméno skriptu je `propustnost.pl`. Pro jeho správnou funkci potřebujeme `trace-file`. Tento skript vlastně zjišťuje kolik bytů bylo posláno během určitého časového intervalu.

Ukázka:

```
perl nizev_skriptu nizev_tracefile cislo_node interval > vystupni_soubor
```

```
perl propustnost.txt out.tr 0 1 > vystup.txt
```

poté můžeme použít `xgraph` pro zobrazení dat

```
xgraph pokusny.txt -t Propustnost -x cas\[s] -y B/s
```

9 Závěr

Tato práce se zabývala popisem standardů IEEE 802.11, zejména IEEE 802.11e a jeho možné simulace pomocí programu NS-2. V prvních kapitolách jsme se seznámili s základními pojmy v Wi-Fi sítích. Dále jsme se seznámili s metodou CSMA/CA, která se v nich využívá. Popsali jsme si zde možnosti identifikace sítě, jejich zabezpečení. Dále jsme si podrobněji popsali MAC vrstvu, problém skryté stanice a základních časových údajů, se kterými se setkáváme při vysílání. V páté kapitole jsme si objasnili základní rozdíly, které přináší IEEE 802.11e.

V další kapitole jsme si popsali projekt Network Simulator NS-2. Jaké jsou jeho součásti a jak jej nainstalovat. Také jsme si zde popsali instalaci rozšíření ns-2 o normu IEEE 802.11e od TKN a jaké změny obsahuje. Obsahem práce je také stručný úvod do psaní skriptů pomocí Tcl.

V poslední kapitole je ukázána možnost generování grafů pomocí skriptů napsaných v Tcl a jejich export. Pokud se jedná o vytváření grafů, je zde potřeba uvést, že zde je nejen potřeba znát tvorbu skriptu v Tcl nebo-li vytvoření fungující sítě, ale poté je potřeba zpracovat trace-file soubor, aby mohli být potřebná data zobrazeny například pomocí programu Xgraph.

Vzhledem k tomu, že samotné nastudování standardu IEEE 802.11 včetně jejich dodatků a následně porozumění skriptovacího jazyku Tcl zabralo poměrně dost času nebyl jsem schopen vytvořit přímo skripty simulující vybrané scénáře a provést jejich přesnou simulaci a vygenerovat k nim odpovídající grafy.

Seznam použitých zdrojů

- [1] LASEK, Petr. *802.11 - MAC vrstva* [online]. 2001 [cit. 2008-07-25]. Dostupný z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=115&clanekID=121>>.
- [2] *Bezdrátové systémy v přístupové síti* [online]. 2008 [cit. 2008-02-03]. Dostupný z WWW: <<http://access.feld.cvut.cz/view.php?navezclanku=bezdratove-systemy-v-pristupove-siti&cisloclanku=2008020002>>.
- [3] BARTOŠ, Honza. *Quality of Service* [online]. 2007 [cit. 2008-07-25]. Dostupný z WWW: <<http://www.fi.muni.cz/~kas/p090/referaty/2007-jaro/ct/qos.html>>.
- [4] *Inter-operability 802.11 with 802.11e* [online]. 2006 [cit. 2008-07-25]. Dostupný z WWW: <<http://www.it.iitb.ac.in/~mukesh/mywebsite/Acads/MTP/firststage/MTPFirstStagePPT.pdf>>.
- [5] *Installing ns2.31 on Ubuntu7.04* [online]. 2008 [cit. 2008-07-25]. Dostupný z WWW: <http://nsgam.isi.edu/nsgam/index.php/Installing_ns2.31_on_Ubuntu7.04>.
- [6] *Network Simulator 2 (Ns2) - aggiornamento alla versione 2.32* [online]. 2007 [cit. 2008-07-25]. Dostupný z WWW: <<http://contalbodavide.blogspot.com/2007/10/network-simulator-2-ns2-aggiornamento.html>>.
- [7] *An IEEE 802.11e EDCA and CFB Simulation Model for ns-2* [online]. 2007 [cit. 2008-07-25]. Dostupný z WWW: <http://www.tkn.tu-berlin.de/research/802.11e_ns2/>.
- [8] *Tcl Commands* [online]. 2008 [cit. 2008-07-25]. Dostupný z WWW: <<http://www.tcl.tk/>>.
- [9] *Wikipedia* [online]. 2008 [cit. 2008-07-25]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/IEEE_802.11>.
- [10] DAŘÍLEK, Martin. *Standardy 802.11e a 802.11i*. [s.l.], c2006. 11 s. Seminární práce.
- [11] MOLNÁR, Martin, SOJKA Michal, HANZÁLEK Zdeněk. *Wireless Networks – Proof of Concept*. [s.l.], 2005. 48 s.
- [12] PATKA, Lukáš. *Bezdrátové sítě*. [s.l.], 2007. 45 s.
- [13] BERNHARD, František. *Přístupové metody WIFI*. [s.l.], 2007. 84 s. Diplomová práce.

- [14] WANG, Yue. *A Tutorial of 802.11 Implementation in ns-2*. [s.l.], c2005. 13 s.
- [15] ALTMAN, Eitan. *NS Simulator for beginners*. [s.l.], 2003. 146 s.
- [16] WU, Zhibin. *Ns-2 Wireless Simulation Tutorial*. [s.l.], 2003. 14 s.
- [17] FALL, Kevin. *The ns manual*. [s.l.], 2007. 418 s.
- [18] YU, Haobo. *Ns-2 Tutorial*. [s.l.], 2000. 15 s.
- [19] *The Network Simulator - ns-2* [online]. 2008 [cit. 2008-07-25]. Dostupný z WWW: <<http://www.isi.edu/nsnam/ns/>>.
- [20] IEEE 802.11—WG Reference number ISO/IEC 8802-11:1999(E) IEEE Std 802.11. International standard [for] information technology – telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, 1999.
- [21] IEEE 802.11—WG Reference number ISO/IEC 8802-11:1999(E) IEEE Std 802.11. International standard [for] information technology – telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, 2005.