

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SIMULAČNÍ MODEL BEZDRÁTOVÉ SÍTĚ 802.11 V NS 2

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

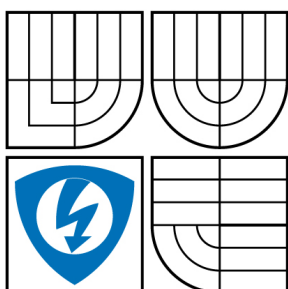
AUTOR PRÁCE
AUTHOR

MIROSLAV BOTTA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

SIMULAČNÍ MODEL BEZDRÁTOVÉ SÍTĚ 802.11 V NS 2

WIRELESS MODEL 802.11 IN NS2

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

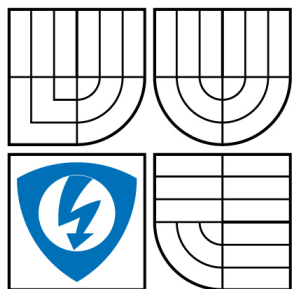
AUTOR PRÁCE
AUTHOR

MIROSLAV BOTTA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MILAN ŠIMEK

BRNO 2008



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor

Teleinformatika

Student: Botta Miroslav

ID: 78092

Ročník: 3

Akademický rok: 2007/2008

NÁZEV TÉMATU:

Simulační model bezdrátové sítě 802.11 v NS 2

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se se simulačním nástrojem Network Simulator 2 a prozkoumejte jeho možnosti pro návrh bezdrátových IP sítí. Prostudujte standard pro bezdrátové sítě 802.11 a v nástroji Network Simulator vytvořte základní model reprezentující tento standard. Na základě získaných zkušeností navrhnete dvě laboratorní úlohy zabývající se problematikou bezdrátových sítí standardu 802.11.

DOPORUČENÁ LITERATURA:

- [1] Information Sciences Institute, "The Network Simulator - ns-2", June 2004, online, dostupné <<http://www.isi.edu/nsnam/ns/>>
- [2] P. ZANDL, Bezdrátové sítě WiFi, ComputerPress, a.s, 2004, ISBN: 80-722-6632

Termín zadání: 11.2.2008

Termín odevzdání: 4.6.2008

Vedoucí práce: Ing. Milan Šimek

prof. Ing. Kamil Vrba, CSc.

předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

LICENČNÍ SMLOUVA

POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

1. Pan/paní

Jméno a příjmení: Miroslav Botta
Bytem: Nábřežná 38, 94064, Nové Zámky
Narozen/a (datum a místo): 20.8.1986, Nové Zámky

(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta elektrotechniky a komunikačních technologií
se sídlem Údolní 244/53, 60200 Brno 2
jejímž jménem jedná na základě písemného pověření děkanem fakulty:
prof. Ing. Kamil Vrba, CSc.

(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

- ☐ disertační práce
- ☐ diplomová práce
- ☒ bakalářská práce

jiná práce, jejíž druh je specifikován jako

(dále jen VŠKP nebo dílo)

Název VŠKP: Simulační model bezdrátové sítě 802.11 v NS 2

Vedoucí/školicel VŠKP: Ing. Milan Šimek

Ústav: Ústav telekomunikací

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

- ☒ tištěné formě - počet exemplářů 1
- ☒ elektronické formě - počet exemplářů 1

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.

3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.

4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
 - ☒ ihned po uzavření této smlouvy
 - ☐ 1 rok po uzavření této smlouvy
 - ☐ 3 roky po uzavření této smlouvy
 - ☐ 5 let po uzavření této smlouvy
 - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Anotácia

Práca sa vo všeobecnosti zaoberá simuláciou jednoduchých bezdrôtových sietí v nástroji Network Simulator 2. V prvej časti pojednáva o Wi-Fi sieťach, približuje jednotlivé typy týchto sietí ako aj ich štandard 802.11 a jeho možnosti pre praktické využitie v bežnom živote. V druhej časti je popísaný úvod do programovania TCL a možnosti, ktoré poskytuje Network simulator 2 pre simuláciu bezdrôtových sietí. Táto časť je ukončená vytvorením jednoduchej simulácie. Posledná tretia kapitola sa venuje vytvoreniu dvoch komplexných laboratórnych úloh, ktoré čerpajú z poznatkov získaných z predchádzajúcich dvoch kapitol.

Kľúčové slová

Network simulator 2, simulácia, TCL, bezdrôtové siete, komponenty, OSI, skript, protokol, 802.11, laboratórna úloha

Abstract

Theme of this work is simple wireless simulation Network Simulation 2 tool. The work consists of more parts. In the first section is basic theory about Wi-Fi networks, they types and standards and describes how useful can it be in practical life. In the second section there is a short annotation for programming in TCL language and possibilities of NS2 for simulating wireless networks. In the end there is a complex simulation of this network. In the third part there are two complex tasks where the information's are drawn from the previous two parts.

Key words

Network simulator 2, simulation, TCL, wireless networks, components, OSI, script, protocol, 802.11, laboratory exercise

BOTTA, M. *Simulační model bezdrátové sítě 802.11 v NS 2*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2008. 55 s. Vedoucí bakalářské práce Ing. Milan Šimek.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Simulační model bezdrátové sítě 802.11 v NS2“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.“

V Brně dne

.....
(podpis autora)

OBSAH

ÚVOD	12
1 ZÁKLADY SIETÍ 802.11	13
1.1 HISTÓRIA.....	13
1.2 KOMPONENTY SIETE.....	14
1.3 TYPY SIETÍ.....	14
1.3.1 <i>Ad-hoc siete</i>	15
1.3.2 <i>Infraštruktúrne siete</i>	15
1.4 ROZŠÍRENÁ OBLASŤ SLUŽIEB (ESS)	16
1.5 OSI – SEDEMVRSTVOVÝ MODEL PRE POČÍTAČE KOMUNIKUJÚCE V SIETI	16
1.6 SPOJOVÁ VRSTVA A MAC PODVRSTVA	16
1.7 FYZICKÁ VRSTVA PHY	18
1.8 ŠTANDARD 802.11.....	18
2 ÚVOD DO NS2.....	20
2.1 APLIKÁCIE OBSAHUJÚCE NS2.....	20
2.2 TCL.....	22
2.2.1 <i>Niekoľko základov pre prácu s TCL a OTCL jazykom</i>	22
2.3 INICIALIZOVANIE A UKONČENIE SKRIPTU	24
2.4 MOŽNOSTI SIMULÁCIE BEZDRÔTOVÝCH 802.11 SIETÍ V NS2	26
2.4.1 <i>Agenti</i>	26
2.4.1.1 Generovanie náhodného scenára pre vytvorenie dátovej komunikácie	27
2.4.2 <i>Simulovanie pohybovania uzlov</i>	27
2.4.2.1 Generovanie automatického pohybovania uzlov	28
2.4.3 <i>Smerovacie protokoly</i>	28
2.4.4 <i>Hierarchické smerovanie</i>	30
2.5 VYTVORENIE JEDNODUCHEJ BEZDRÔTOVEJ SIMULÁCIE	31
2.5.1 <i>Popis trasovacieho súboru pre bezdrôtové uzly:</i>	35
2.6 SIMULÁCIA HYBRIDNÝCH SIETÍ.....	36
3. POSTUP TVORBY LABORATÓRNYCH ÚLOH.....	39
3.1 POPIS A VÝBER KAPITOL	39
3.2 ZÁKLADNÉ POŽIADAVKY	39

3.3 DOPORUČENIE	40
3.4 ROZVRHNU Tie LABORATÓRNEJ ÚLOHY	40
3.4.1 Praktická časť laboratórnej úlohy	40
3.4.2 Samostatná práca	41
3.5 ČASOVÝ PLÁN ÚLOHY	41
ZÁVER.....	42
ZOZNAM POUŽITEJ LITERATÚRY	43
ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK	44
ZOZNAM PRÍLOH	45
PRÍLOHA A – 1. LABORATÓRNA ÚLOHA - SIMULÁCIA JEDNODUCHEJ 802.11	
SIETI	46
A.1 ÚVOD	46
A.2 TEORETICKÝ ÚVOD	46
A.3 PRAKTICKÁ ČASŤ	48
A.4 SAMOSTATNÁ PRÁCA	49
A.5 ZHRNUTIE	50
A.6 POUŽITÁ LITERATÚRA	50
PRÍLOHA B – 2. LABORATÓRNA ÚLOHA - SIMULÁCIA HYBRIDNÝCH	
BEZDRÔTOVÝCH 802.11 SIETÍ.....	51
B.1 ÚVOD.....	51
B.2 TEORETICKÝ ÚVOD	51
B.3 PRAKTICKÁ ČASŤ	53
B.4 SAMOSTATNÁ PRÁCA	54
B.5 ZHRNUTIE	55
B.6 POUŽITÁ LITERATÚRA	55

Zoznam Obrázkov

Obr. 1.1: Komponenty siete 802.11	14
Obr. 1.2: Ad-hoc sieť	15
Obr. 1.3: Infraštruktúrna sieť	15
Obr. 1.4: CSMA/CA.....	17
Obr. 2.1: Network animator	21
Obr. 2.2: xgraph	22
Obr. 2.3: Príklad hybridnej siete	36
Obr. A.1: Vizualizácia simulácie pomocou nam.....	47
Obr. A.2: Časové priebehy zobrazené pomocou programu xgraph	47
Obr. A.3: Topológia siete definovaná v súbore wireless1.tc	48
Obr. B.1: Vizualizácia simulácie pomocou nam.....	53
Obr. B.2: Časové priebehy zobrazené pomocou programu xgraph	53
Obr. B.3: Topológia siete definovaná v súbore wired-cum-wireless.tcl	54
Obr. B.4: Rozšírená topológia siete	55

Zoznam tabuliek

Tab. 1.1: Základné charakteristické hodnoty 802.11	19
Tab. 2.1 Tabuľka popisu jednotlivých polí trasovacích dát paketov	35
Tab. 2.2 Tabuľka popisu jednotlivých polí trasovacích dát pohybov	35

ÚVOD

Len pred niekoľkými rokmi sa celosvetová počítačová sieť stala bežným denne využívaným nástrojom. Technický pokrok sa čoraz ďalej vyvíjal ale až v poslednej dobe sa zisťuje, že počítač nemusí byť pripútaný na jednom mieste aby mohol byť využívaný na použitie v sieti.

V dnešnej dobe je mobilita zariadení veľmi dôležitá a možnosti, ktoré poskytuje nemôžu ostať nepovšimnuté. Keďže rádiové frekvencie, ktoré umožňujú taktiež bezdrôtový prenos dát sú okupované mnohými ďalšími zariadeniami a službami z ktorých väčšina je licencovaná, bolo nutné vyhradiť určité miesto pre bežné používanie, na ktoré nie je potrebná licencia. Preto vzniklo pásmo ISM teda pásmo vyhradené pre priemyselné a vedecké potreby.

Bezdrôtové dátové siete založené na štandarde 802.11 pracujúce v pásme ISM sú dnes veľmi rozšírené, veľkou výhodou týchto sietí je, že nie je potreba budovať nákladnú káblovú infraštruktúru, tieto siete sa veľmi ľahko budujú a dávajú užívateľovi výhodu mobility.

Táto práca sa venuje problematike simulácie bezdrôtových sietí Wi-Fi štandardu 802.11 v nástroji Network Simulator 2. Celá tematika je delená na jednotlivé časti. V prvej časti sa práca pokúša vysvetliť základné znalosti z oblasti samotných Wi-Fi sietí a v druhej časti vysvetľuje jednotlivé prostriedky pre simuláciu danej topológie, základy programovania v jazyku TCL a končí vytvorením komplexných simulácií zahrnutých v dvoch laboratórnych úlohách ktoré sa nachádzajú v prílohe.

1 Základy bezdrôtových sietí štandardu 802.11

1.1 História

Každým výrobcom bola na začiatku vyrábaná vlastná technológia no nakoniec kvôli veľkej výhode spoločného štandardu v roku 1997 sa medzinárodným inštitútom IEEE špecifikoval štandard bezdrôtovej siete pracujúcej v pásme ISM pod názvom 802.11. Tento štandard definoval dve rýchlosti 1 a 2 Mb/s, využívajúce infračervený prenos a rádiový prenos v ISM pásme 2,4 GHz používajúci prenos v rozprestretom spektre (priamo rozprestreté spektrum) alebo FHSS (rozprestreté spektrum s preskakovaním kmitočtov). Tento štandard definoval základnú prístupovú metódu CSMA/CA. V roku 1999 bola skupinou spoločností (3COM, CISCO, LUCENT, NONA a ďalší) založená nezisková spoločnosť Wireless Ethernet Compatibility Alliance WECA, ktorá nebola spokojná s nedostatkami štandardu 802.11. Na základe ich práce vzniká nový štandard IEEE 802.11b, ktorý pracuje s vyššími rýchlosťami a je spätne kompatibilný. IEEE 802.11b teda definícia bezdrôtovej siete pracujúcej v pásme 2,4 GHz s rýchlosťou 11 Mb/s. Neskôr v roku 1999 bol definovaný štandard 802.11a, ktorý pracuje vo frekvenčnom pásme 5 GHz s vyššími rýchlosťami, teda až 54 Mb/s. Posledný podstatný štandard bol definovaný v roku 2003 a to 802.11g, ktorý priniesol zväčšenie prenosovej rýchlosti v pásme 2,4GHz na 54Mb/s. 802.11 obsahuje ešte niekoľko ďalších revízií, ktoré odstraňujú nedostatky predošlých.

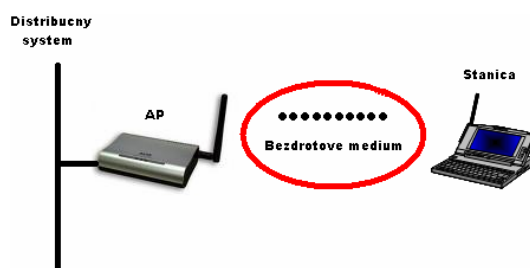
O certifikáciu kompatibility zariadenia sa stará certifikačná autorita WECA neskôr premenovaná na Wi-Fi alianciu, ktorá testuje interoperabilitu jednotlivých zariadení hlásiacich sa k štandardu 802.11. Výrobkom, ktoré vyhovujú prepožičiava logo Wi-Fi, ktoré informuje kupujúceho o prepojitelnosti týchto zariadení.

Bezdrôtové siete Wi-Fi sú veľmi podobné sieťam typu Ethernet, avšak len z globálneho hľadiska. Je nutné, aby sa bezdrôtové siete dokázali vysporiadať s rozličnými problémami ako problematika nestability rádiového prostredia. Medzi ich hlavné základne vlastnosti patrí dynamická topológia, médium je oproti drôtovým sieťam menej spoľahlivé a menej chránené voči vonkajším nežiaducim signálom. Komunikačné médium nemá pevné hranice.

1.2 Komponenty siete

Medzi základné komponenty Wi-Fi siete patrí:

- **Stanica** - môže ňou byť akékoľvek zariadenie používané pre komunikáciu v sieti, takéto zariadenia nemusia byť vôbec mobilné.
- **Bezdrôtové médium** - znamená pre sieť WLAN to isté čo pre pevné káblové siete samotná kabeláž. 802.11 sa zaoberá dvoma rádiovými frekvenciami 2.4 a 5GHz, a samozrejme aj infračerveným prenosom dát.
- **Prístupový bod** - nie je nutnou súčasťou bezdrôtovej siete. Jeho hlavné využitie spočíva v tom, že pracuje ako most medzi káblovou a bezdrôtovou sieťou, ale poskytuje aj veľa ďalších funkcií.
- **Distribučný systém** - nie je nutnou podmienkou fungujúcej bezdrôtovej siete samozrejme ak od takejto siete nie je očakávané pripojenie do inej siete. Takýto systém je nutný v okamžiku keď má medzi sebou komunikovať viac prístupových bodov a tvoriť rozsiahlejšiu sieť. Je najviac využívaný v komerčnej oblasti. Pozostáva zo sieťového mostu a z distribučného média, ktorým je väčšinou chrbticová sieť, ktorá je skoro vždy riešená ako Ethernet.



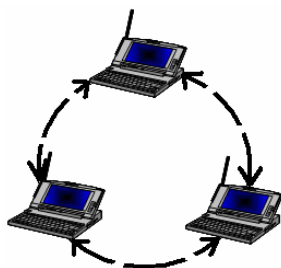
Obr. 1.1: Komponenty siete 802.11

1.3 Typy sietí

Základný stavebný blok sietí 802.11 tvoria minimálne 2 stanice, ktoré navzájom komunikujú. Sú označované ako **Basic Service Set (BSS)**, teda *základný súbor služieb*. Takéto stanice komunikujú v určitom vyhradenom území závisiacich na dosahu týchto staníc a toto územie sa nazýva **Basic Service Area (BSA)**. Ak je viacero BSS prepojených medzi sebou do väčšieho celku, vzniká takzvaný rozšírený súbor služieb (ESS – Extended Service Set). Existujú 2 základne typy BSS:

1.3.1 Ad-hoc siete

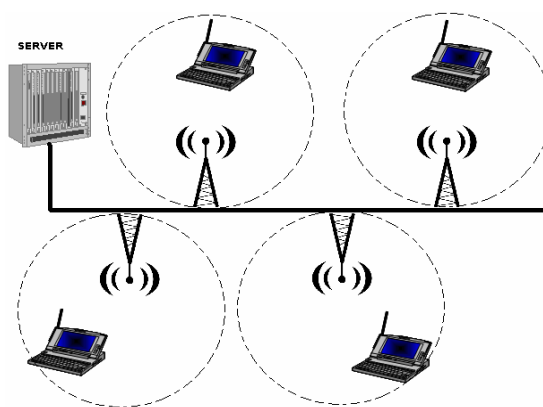
Sú to nezávislé siete. Stanice v tejto sieti spolu komunikujú priamo, nezávislé na nejakom prostredníkovi. Nenachádza sa tu žiadny centrálny bod. Pre komunikáciu týchto staníc je nevyhnutné aby boli vo vzájomnom dosahu. Tieto siete sú vhodné pre komunikáciu malého počtu počítačov, ktoré sa nachádzajú v menších priestoroch, kde je možné docieľiť vzájomného rádiového dosahu staníc.



Obr. 1.2: Ad-hoc sieť

1.3.2 Infraštruktúrne siete

Majú svoju presne vymedzenú infraštruktúru kde rola spojovacieho komponentu spočíva v prístupovom bode (AP).



Obr. 1.3: Infraštruktúrna sieť

Prístupový bod môže komunikovať s viac ako jednou stanicou. Pokiaľ chcú dve stanice v takejto infraštruktúrnej sieti spolu komunikovať musia tieto dáta postupovať cez AP teda minimálne dvoma skokmi, najprv na prístupový bod a z neho na cieľovú stanicu. V infraštruktúrnej sieti na rozdiel od sietí typu Ad-hoc stačí udržiavať iba jedno spojenie a navyše AP rozpozná keď stanica prejde do úsporného režimu a môže pre ňu ukladať dáta a v prípade prebudenia z úsporného režimu je schopný odoslať tieto dáta. Tento typ sietí má veľké výhody, keďže ponúka centrálnu správu a pri pripojení ďalšieho sieťového komponentu sa nemusí meniť celá topológia siete. Pre pripojenie k AP je dôležitá asociácia stanice, čo

znamená, že stanica má záujem o pripojenie. Takáto asociácia je exkluzívna, teda nemôže byť asociovaná k viacerým prístupovým bodom.

1.4 Rozšírená oblasť služieb (ESS)

ESS sa vytvára prepojením jednotlivých BSS cez chrbticovú sieť. Stanice vo vnútri ESS môžu medzi sebou komunikovať, aj keď sú v rozdielnych BSS a môžu sa medzi nimi i pohybovať.

1.5 OSI – sedemvrstvový model pre počítače komunikujúce v sieti

OSI (Open System Interconnection) je obecným modelom pre počítače komunikujúce v sieti. Skladá sa zo siedmych vrstiev, čím sa jedná o nižšiu úroveň, tým je táto úroveň viac skrytá užívateľovi. Tieto vrstvy sú: aplikačná, prezentačná, relačná, transportná, sieťová, spojová a fyzická. Samotný štandard 802.11 definuje ako vlastné iba dve najnižšie vrstvy OSI, teda spojovú a fyzickú, všetky ostatné vrstvy ostávajú pôvodné.

Podvrstva spojovej vrstvy označovaná ako MAC (Media Access Control) ovládanie prístupu k médiu, predstavuje súbor pravidiel určujúcich ako pristupovať k prostriedkom siete pre prenos dát. Na fyzickej vrstve PHY sú ponechané samotné detaily prenosu dát.

1.6 Spojová vrstva a MAC podvrstva

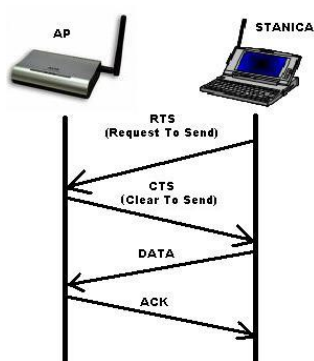
MAC podvrstva je dôležitou vrstvou pre prácu bezdrôtovej siete 802.11. Táto podvrstva slúži ako rozhranie medzi fyzickou vrstvou a hostiteľským zariadením a tiež vytvára podporu Ad-hoc a infraštruktúrneho zapojenia siete. Dvomi hlavnými vlastnosťami MAC podvrstvy sú CRC (Cyclic Redundancy Check) čo je cyklický kontrolný súčet, a fragmentácia paketov. Aby bolo možné zistiť, či paket bol respektíve nebol poškodený behom prenosu, sú tieto pakety opatrené pripojeným kontrolným súčtom CRC.

Fragmentácia paketov rozdeľuje pakety do menších kúskov, ktoré sú postupne prenášané sieťovým médiom. Oproti ethernetu je oveľa vyššia možnosť chyby prenášaného paketu a keďže opakovaný prenos by zbytočne sieť zahlcoval, sú prenášané len jeho časti, čo menej vyťažuje sieť.

Dôležitú úlohu pri bezdrôtových sieťach zohrávajú takzvané medzirámkové medzery (Inter Frame Spacing IFS). Tieto slúžia pre koordináciu prístupu k prenosovému médiu.

Koordinácia prístupu k prenosovému médiu je dôležitá aby nedošlo ku kolíziám s ostatnými stanicami, keďže v jednej chvíli môže byť prijímaný signál iba z jednej stanice. Pokiaľ by toto nebolo vyriešené a nedochádzalo ku koordinácii staníc, viacero by ich mohlo vysielat' naraz a cieľová stanica by zachytila poškodený signál.

Pre zaistenie predchádzania kolíziám sa v 802.11 používa protokol CSMA/CA. Tento protokol minimalizuje kolízie použitím štyroch rámcov: RTS (Request to send), CTS (Clear to send), ACK (Acknowledge) a NAV (Network allocation vector).



Obr. 1.4: CSMA/CA

Predchádzajúci obrázok popisuje mechanizmus predchádzania kolíziám spolu s kladným potvrdzovaním čo znamená, že stanica naslúcha a pokiaľ je médium voľné, počká ešte určitý čas (DIFS) a až potom začne vysielat'. Prijímacia stanica najskôr skontroluje kontrolný súčet CRC prijatého paketu a ak je paket v poriadku odošle potvrdenie ACK. Prijatie potvrdzujúceho paketu pre odosieláciu stanicu znamená, že nedošlo ku kolízii, pokiaľ nedostane ACK, opakuje vysielanie. K zníženiu pravdepodobnosti kolízií, ktoré vznikajú práve tým, že stanice nie sú v dosahu definuje štandard virtuálny naslúchajúci mechanizmus. Ak chce stanica vysielat', vyšle najprv riadiaci paket RTS, ktorý obsahuje zdroj, cieľ a trvanie nasledujúceho prenosu. Cieľová stanica odpovie odoslaním riadiaceho paketu CTS, ktorý obsahuje dobu trvania nasledujúceho prenosu. Ostatné stanice, ktoré registrujú paket RTS alebo CTS si nastaví indikátor virtuálneho naslúchania NAV na dobu trvania prenosu oznámenú v tomto pakete. Po celú túto dobu budú stanice považovať médium za obsadené. Týmto sa riziko kolízií minimalizuje.

1.7 Fyzická vrstva PHY

Pod týmto pojmom sa skrýva fyzické rozhranie medzi jednotlivými zariadeniami v sieti. PHY rieši problematiku prenosu dát pomocou rádiového signálu.

Pre dosiahnutie rýchlych dátových prenosov v pásme ISM sa používa technológia rozprestreného spektra (SS – spread spectrum). Rozprestrené spektrum používa matematické funkcie pre rozptýlenie sily signálu do širokého frekvenčného bloku. Na strane prijímača sa prevedie jednoducho opačná operácia a zloží sa takýto rozprestrený signál do úzkopásmového signálu s ktorým sa ďalej pracuje. Rozprestrené spektrum je požiadavkou pre prevádzku zariadení v pásme ISM, teda iný typ prenosu sa nesmie používať.

Fyzická vrstva štandardu 802.11 definuje tri techniky rozprestreného spektra a to:

- **Frekvenčné preskoky (Frequency hopping – FH, FHSS)** – táto technika má vojenský pôvod. Vysielač preskakuje v pseudonáhodnom poradí po jednotlivých frekvenčných pásmach a na každom vysielá krátky dátový prúd. Podstatnou výhodou týchto frekvenčných preskokov je väčší počet systémov pracujúcich naraz v pásme 2,4GHz.
- **Priama sekvencia (Direct Sequence – DS, DSSS)** – vysielaná informácia je za použitia matematického kódovania rozprestrená po 22MHz širokom frekvenčnom pásme. Takéto široké pásma sú k dispozícii celkom tri. Prijímač opačným postupom signály dekoduje. Štandard 802.11b priamou sekvenciou dosahuje vysoké rýchlosti až do 11Mb/s.

1.8 Štandard 802.11

Tento štandard nazývaný aj Wi-Fi štandard poukazuje na súbor bezdrôtových LAN štandardov vyvinutých pracovnou skupinou 11 štandardizačného výboru IEEE. Bol to prvý štandard ktorý sa zameriaval na problematiku bezdrôtových lokálnych počítačových sietí. Taktiež tento pôvodný štandard IEEE 802.11, ktorý bol vydaný v roku 1997, definuje dve prenosové rýchlosti a to 1 a 2 Mb/s prenášaných infračerveným signálom alebo na ISM rádiových frekvenciách 2,4 – 2,4835 GHz využívajúcej DSSS a FHSS. Infračervené signály sú stála súčasťou tohto štandardu, ale sa nevyužívajú. Ďalej štandard definoval prístupovú metódu k médiu nazývanú CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). Takže značné percento dostupnej kapacity prenosového kanálu je za týmto

účelom obetovaná, ale za cenu zvýšenia spoľahlivosti pri prenose údajov v rôznych heterogénnych a znečistených prostrediach. Štandard 802.11 v súčasnosti obsahuje rodinu viacerých revízií, z ktorých každá obsahuje úpravu techník komunikácie po médiu, ale využívajú ten istý protokol. Hlavnou nevýhodou tohto štandardu bolo že ponúkal viacero realizovateľných možností prepojenia. Preto bol tento štandard nahradený jeho revíziou 802.11b.

Zhrnutie štandardu 802.11	
Protokol	802.11
Dátum vydania	1997
Frekvencia	2,4 GHz
Priepustnosť	0,9 Mbit/s
Prenosová rýchlosť	2 Mbit/s
Modulácia	DSSS, FHSS
Dosah vo vnútri	~20 Metrov
Dosah vonku	~100 Metrov

Tab. 1.1: Základné charakteristické hodnoty 802.11

2 Úvod do NS2

Network Simulator 2 (ďalej len NS2) je simulačné prostredie pracujúce v operačných systémoch typu Unix. Pokrýva veľké množstvo aplikácií, protokolov, typov sietí a sieťových článkov. V roku 1989 sa NS2 začal vyvíjať ako varianta programu REAL Network Simulator a bol v podstate vyvinutý až o pár rokov neskôr. Vývoj NS bol v roku 1995 podporovaný organizáciou DARPA, ktorá je známa vyvíjaním nových technológií, predovšetkým pre armádne účely. V súčasnosti sa na vyvíjaní zúčastňuje viacero organizácií. NS2 nie je konečným produktom a na jeho vývoji sa stále intenzívne pracuje. Nasledujúcim produktom bude NS3, ktorý bude mať väčšie možnosti a vylepšenia oproti predchádzajúcej verzii. Tak ako v ostatných veľkých projektoch, tak aj v NS2 sú stále nachádzané chyby, ktoré sú nasledovne zafixované. NS2 je založený na dvoch programovacích jazykoch a to objektovo orientovanom simulátore, napísanom v C++ a OTCL (objektovo orientovanom rozšírení TCL) interpretéry, používaného pre vykonávanie užívateľských príkazových skriptov. V OTCL skripte môžu byť definované jednotlivé sieťové topológie, špecifické protokoly a aplikácie, ktoré si prajeme simulovať. C++ je používaný pre manipuláciu s dátami napríklad paketové udalosti a OTCL pre kontrolu napríklad periodických udalostí alebo spúšťacích udalostí.

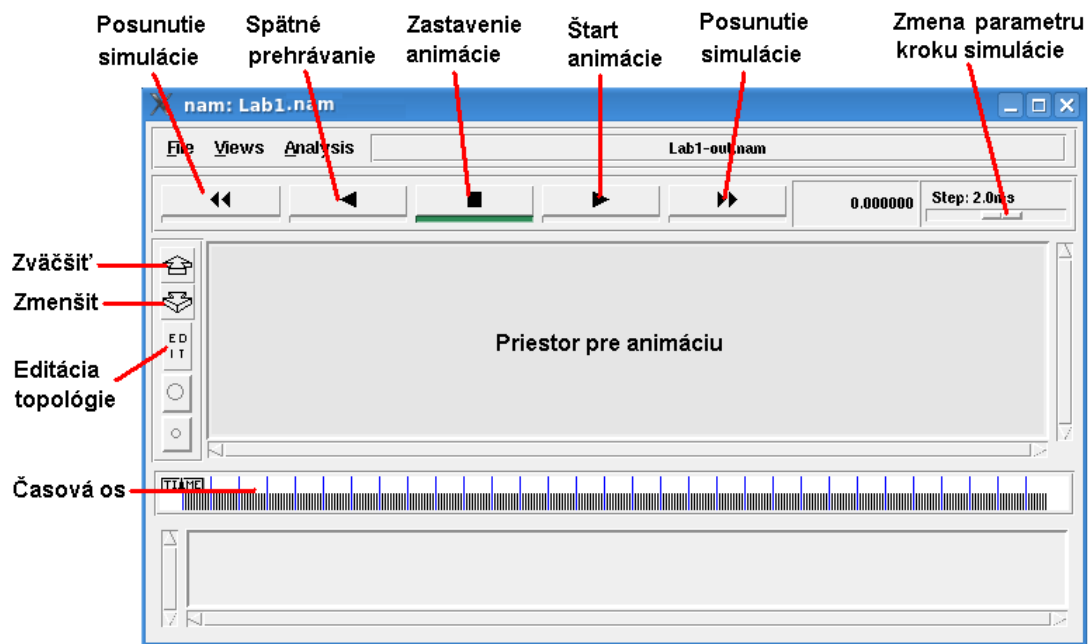
NS sa spúšťa príkazom „ns <tcl_skript>“, samozrejme je potrebné sa nachádzať v tej istej zložke kde je tcl skript prípadne zadať celú cestu k tomuto skriptu. Názov „tcl_skript“ je vlastné meno požadovaného súboru obsahujúceho daný tcl skript, ktorý obsahuje požadovaný simulačný scenár. Tento argument nie je povinný, je možné písať tento skript aj do samostatného shellu, ale týmto spôsobom je práca nekomfortná a zdĺhavá.

2.1 APLIKÁCIE obsahujúce NS2

NS2 poskytuje viacero aplikácií (rozšírení), v tejto práci budú použité len dve takéto rozšírenia, akým je Network Animator (ďalej len NAM) a XGRAPH.

NAM je vizualizačný nástroj založený na Tcl/Tk pre prehliadanie naskriptovanej sieťovej simulácie a reálnej paketovej výmeny jednotlivých uzlov siete. Slúži pre zobrazenie jednoduchých animácií, ktoré číta ako príkazy z trasovacieho súboru. Ako prvý krok pre použitie NAM je vytvorenie NAM trasovacieho súboru, ktorý obsahuje informácie o topológii, uzloch, prepojeniach atd. Jeho výhodou je, že v pamäti uchováva len dôležité

údaje a všetky ostatné číta zo súboru uloženom na disku. Pri štarte NAM-u sa najprv prečíta trasovací súbor, vytvorí topológiu, objaví sa okno a pauzne simuláciu v čase 0. Cez samotný interface aplikácie, je možné ovládať veľa rôznych aspektov samotnej animácie. NAM-u pri spúšťaní z príkazového riadku je možné odovzdať ďalšie upresňujúce parametre pre vývoj a zobrazenie danej animácie. Pri jeho štarte sa vytvorí okno NAM konzoly. Je možné simulovať viacero animácií naraz pod rovnakou inštanciou animátora.



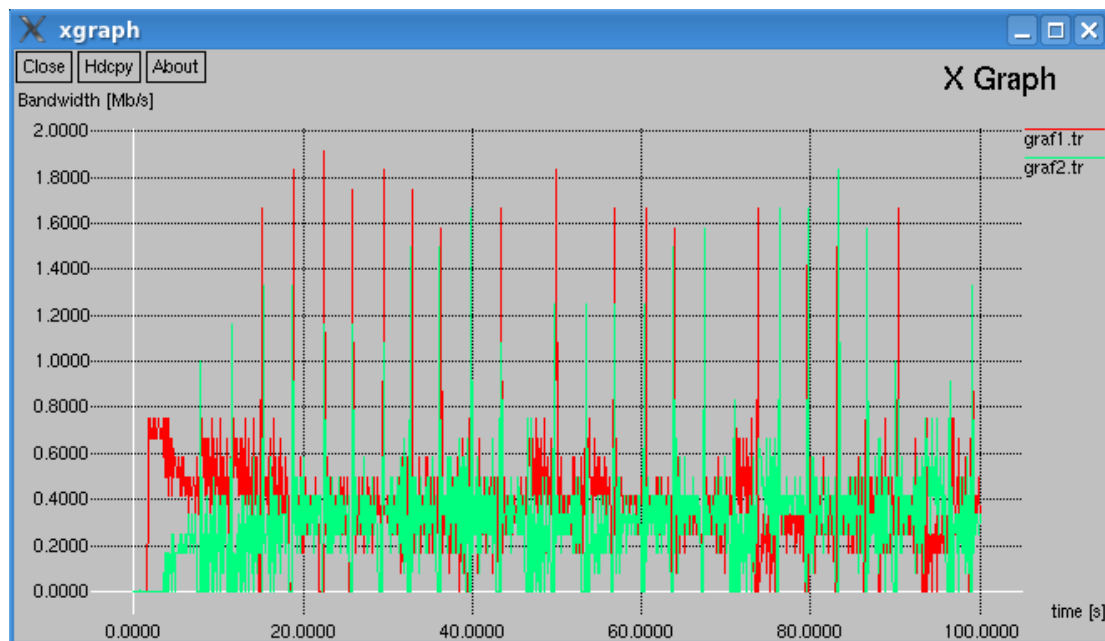
Obr. 2.1: Network animator

XGRAPH je aplikácia určená na vykresľovanie grafov priebehu komunikácie medzi jednotlivými uzlami. Táto aplikácia tiež umožňuje vytvorenie výstupného priebehu do obrazového súboru typu Tgif alebo vytvorenie postscript súboru, kliknutím na ikonku „HRDCPY“ a výberom vhodných parametrov. Ako vstup pre XGRAPH je očakávaný jeden alebo viac ASCII súborov obsahujúcich x,y súradnice pre vykresľovanie grafu. Ak je v parametri tejto aplikácie zadaných viacero ASCII súborov, sú tieto priebehy vykreslené do jedného grafu. XGRAPH obsahuje množstvo možností pre úpravu vykresľovaného grafu ako napríklad vypísanie titulku grafu, zmenu jeho rozmerov, alebo jednotlivé titulky pre vykresľované osi. Aby sme toho mohli dosiahnuť je nutné mu predat ďalšie upresňujúce parametre pri jeho spúšťaní. Medzi pre nás najzaujímavejšie a najpotrebnejšie patrí:

titulok grafu: -t "názov grafu"

titulky pre osi: -x "názov osi x" -y "názov osi y"

rozмеры grafu: -geometry <rozmer x> <rozmer y>



Obr. 2.2: xgraph

2.2 TCL

Tcl (Tool Command Language) je jazyk s veľmi jednoduchou syntaxou a umožňuje veľmi jednoduchú integráciu s ostatnými programovacími jazykmi. TCL bolo vytvorené Johnom Ousterhoutom. Charakteristické znaky pre tento jazyk sú napríklad rýchly vývoj, poskytovanie grafického rozhrania, kompatibilita pre veľa platforiem a flexibilita pre integráciu.

2.2.1 Niekoľko základov pre prácu s TCL a OTCL jazykom

Priradenie hodnoty premennej sa vykonáva cez príkaz „**set**“, napríklad „**set a 5**“ priradí premennej „**a**“ hodnotu 5. Ekvivalentný výraz v jazyku C by vyzeral „**a = 5**“. V prípade že chceme priradiť do premennej hodnotu inej premennej musíme pred ňou použiť znak „**\$**“. Napríklad „**set b \$a**“. Pri matematických operáciách musíme použiť výrazový príkaz. Napríklad keď chceme získať súčet premenných „**a**“ a „**b**“ a priradiť ho do premennej „**c**“, zapíšeme to nasledovne „**set c [expr \$a + \$b]**“. Dôležitou poznámkou je že premenné v TCL nie sú definované, takže premenné môžu byť reťazce alebo čísla závisiace na hodnote ktorú jej priradíme. Napríklad chceme vypísať výsledok „**1/100**“. Teda ak zapíšeme „**[expr 1/100]**“, bude výsledok v tomto prípade 0. Ak chceme získať správny výsledok

musíme nejakým spôsobom naznačiť, že sa nejedná o celočíselné číslo a musíme preto tento výraz zapísať ako „*[expr 1.0/100.0]*“.

Znakom „*#*“ označujeme jednoriadkovú poznámku, ktorá začína práve za týmto znakom. Poznámka nie je súčasťou programu, takže TCL interpreter nevykoná tento riadok.

Pre vytvorenie súboru, ktorého názov bude napríklad „*nazovsaboru1*“, a pre priradenie ukazateľa na tento súbor napríklad „*subor1*“, zapíšeme príkaz „*set subor1 [open nazovsaboru1 w]*“.

Spúšťanie unixovských príkazov sa vykonáva zadaním príkazu „*exec*“ a za nim daný príkaz. Napríklad, ak chceme vykresliť graf priebehu simulácie pomocou programu *xgraph*, použijeme príkaz „*exec xgraph subor &*“, kde „*subor*“ je súbor, ktorý obsahuje všetky dáta pre vykreslenie grafu a „*&*“ je znak ktorý určí že sa príkaz vykoná na pozadí.

Podmienky sa zadávajú podobne ako v programovacom jazyku C. Ich štruktúra je nasledujúca.

```
if {podmienka}
{
    prikazy
}
else
{
    prikazy
}
```

V predchádzajúcom príklade môže byť použitých viacero podmienok „*if*“ nasledujúcich za sebou.

Slučky v programe majú nasledujúcu syntax.

```
for {set j 0} {$j < 10} {incr j}
{
    prikazy
}
```

Najprv sa inicializuje premenná, ktorá slúži ako počítadlo, potom sa určuje podmienka do kedy sa budú príkazy v slučke vykonávať a nakoniec sa vykoná samotná operácia s premennou, ktorá zaistí počet vykonaní slučky. V tomto prípade sa slučka vykoná 10 krát teda pokiaľ hodnota premennej „*j*“ je menšia ako 10.

V Tcl jazyku podporuje písanie procedúr, ktoré sa môžu používať nie len k zjednodušeniu práce.

```
proc pokus { par1 par2 ... }
{
    global prem1 prem2
```

```

prikazy
return $nieco
}

```

Procedúra môže obdržať niekoľko parametrov, v našom prípade sú to „*par1*“ a „*par2*“, ktoré môžu byť napríklad objekty, súbory, alebo premenné. Procedúra je volaná s týmito parametrami napríklad „*pokus a b*“, kde „a“ a „b“ budú použité v procedúre pre premenné „*par1*“ a „*par2*“. V priebehu procedúry sa „*par1*“ a „*par2*“ môžu meniť čo ale neovplyvní premenné „a“ a „b“. Ale však ako to vyžadujeme je nutné nadefinovať globálne premenné v našom prípade „*preml*“ a „*prem2*“ aby sme vedeli pracovať s premennými mimo tejto procedúry. Nakoniec táto procedúra môže vrátiť hodnotu zadanú príkazom „*return*“, čo však nie je nutnou podmienkou.

2.3 Inicializovanie a ukončenie skriptu

V nasledujúcom texte bude popísaná predloha ktorá môže byť použitá v tcl skripte. Ide o jednoduchú ukážku začiatkov práce s nástrojom NS2 pre uvedenie do tematiky práce.

Na začiatok je potreba vytvoriť objekt simulátora. Toto je možné pomocou príkazu

```

set ns [new Simulator]

```

Je to väčšinou prvý riadok každého NS2 skriptu. Takto deklarujeme premennú „*ns*“, ktorá je po vykonaní tohoto príkazu novou inštanciou *triedy Simulatora*, teda objekt. Tuto premennú môžeme pomenovať ľubovoľne, ale tento názov je najviac preferovaný medzi užívateľmi. Používaním premennej „*ns*“ môžeme takto využívať všetky metódy triedy Simulator. Podľa potreby je následne nutné vytvoriť súbory kde budú uložené trasovacie dáta dôležité pre ďalšiu analýzu simulácie.

```

set tracesubor [open vystup.tr w]
$ns trace-all $tracesubor

```

```

set namsubor [open vystup.nam w]
$ns namtrace-all $namsubor

```

Prvý riadok vytvorí a následne otvorí súbor „**vystup.tr**“ a pre zapisovanie mu priradí na súbor ukazovateľ nazvaný „**tracesubor**“. Druhý riadok je nutný pre objekt simulátoru, ktorý mu hovorí že sme vytvorili súbor na zapisovanie všetkých simulačných údajov ktoré sú podstatné pre analýzu. V druhom prípade je postup obdobný avšak tu vytvárame trasovací súbor pre vizuálnu podporu simuláciu, teda pre program NAM.

V nasledujúcom kroku vytvoríme procedúru „**finish**“ ktorá má na starosti ukončenie nášho programu a následné spustenie programu NAM.

```
proc finish {} {  
    global ns tracesubor namsubor  
    $ns flush-trace  
    close $tracesubor  
    close $namsubor  
    exec nam vystup.nam &  
    exit 0  
}
```

Ďalší riadok prikáže objektu simulátoru vykonať procedúru „**finish**“ po 10.0 sekundách od začiatku simulácie.

```
$ns at 10.0 "finish"
```

Nasledujúci riadok konečne spusti simuláciu.

```
$ns run
```

Predchádzajúcu predlohu môžeme uložiť napríklad ako „**pokus.tcl**“. Pokiaľ by sme sa rozhodli predchádzajúcu simuláciu spustiť pomocou príkazu „**ns pokus.tcl**“, obdržíme chybové hlásenie, že trace file je prázdny, pretože sme doteraz ešte nedefinovali objekty a udalosti ktoré by mal skript obsahovať.

2.4 Možnosti simulácie bezdrôtových 802.11 sietí v NS2

Network simulator 2 ponúka veľké množstvo možností pre simuláciu WI-FI 802.11 sietí ako napríklad rôzne druhy smerovania paketov, aplikácií, dynamiku siete, vytvárania topológií atď. Keďže týchto možností je naozaj veľmi veľa, v tejto kapitole budú stručne popísané len niektoré pre nás významnejšie možnosti.

2.4.1 Agenti

V prípade požiadavku pre vytváranie simulácie dátového toku medzi jednotlivými uzlami sa používa trieda „Agent“ (agenti). Agenti predstavujú koncové body, kde sú jednotlivé pakety vytvárané, respektíve konzumované a sú využívané pri implementácii protokolov na rôznych sieťových vrstvách. Trieda „Agent“ obsahuje ďalšie premenné, ktoré priradujú hodnotu rôznym poliam simulovaného paketu predtým, než je odoslaný.

Tieto parametre sú nasledovné:

- addr_ - adresa zdrojového uzlu
- dst_ - adresa cieľového uzlu
- size_ - veľkosť paketu v bajtoch
- type_ - typ paketu
- fid_ - identifikátor
- prio_ - priorita paketu
- flags_ - príznaky paketu
- defttl_ - defaultná hodnota TTL (time to live)

Následne je ukázaných niekoľko Agentov podporovaných simulátorom:

- TCP – „Tahoe“ TCP (cwnd = 1 pri každej strate)
- TCP/Reno – „Reno“ TCP (s rýchlym obnovením)
- TCP/Newreno – modifikované Reno TCP
- TCP/FullTcp – viac plnefunkčné TCP s dvojcestným prenosom
- TCP/Vegas – „Vegas“ TCP
- TCPSink – „Reno“ alebo „Tahoe“ TCP prijímač (nepoužiteľné pre FullTcp)
- UDP – základný UDP agent
- RTP – RTP vysielateľ a prijímač

RTCP – RTCP vysielateľ a prijímač

LossMonitor – prijímač paketov, ktorý kontroluje chyby

Null – degenerovaný agent rušiaci pakety

2.4.1.1 Generovanie náhodného scenára pre vytvorenie dátovej komunikácie

Táto časť pojednáva o možnostiach využitia generovania náhodného scenára pre dátovú komunikáciu medzi jednotlivými uzlami v danej topológii. K tomuto účelu distribúcia NS2 obsahuje skript s názvom „cbrgen.tcl“, ktorý je umiestnený v zložke s adresou *~ns/indep_utils/cmu-scen-gen*. Tento skript môže byť použitý k vytvoreniu UDP alebo TCP dátových prenosov medzi jednotlivými uzlami siete. Pre vytvorenie takéhoto scenára musíme zadať typ dátového toku (UDP alebo TCP), počet uzlov, maximálny počet jednotlivých prepojení dátového toku medzi nimi atď. Skript sa spúšťa s nasledovnými parametrami:

```
ns cbrgen.tcl -type <CBR/TCP> -nn <počet uzlov> -seed <seed>  
-mc <max. prepojení> -rate <rýchlosť> > <názov súboru>
```

Čas začiatku generovania TCP/UDP dátových prenosov medzi uzlami je náhodne generované číslo, ktorého maximálna hodnota je 180 sekúnd. V prípade, že požadujeme vytvoriť náhodný dátový tok medzi 15 uzlami s maximálnym počtom prepojení 10 so „seed“ hodnotou 1 a rýchlosťou 5 Mb/s, tak bude príkaz vyzeráť nasledovne:

```
ns cbrgen.tcl -type CBR -nn 15 -seed 1.0 -mc 10 -rate 5.0 > cbr-test.tcl
```

2.4.2 Simulovanie pohybovania uzlov

Na rozdiel od simulovania pevných sietí je možné pri simulácii 802.11 sietí implementovať do skriptu pohyby jednotlivých uzlov čo môže byť prevedené ručne zadaním jednotlivých súradníc a rýchlosťou pohybu, alebo je využitý modul, ktorý je integrovaný v NS2. Súbor, ktoré definujú pohyby jednotlivých uzlov sú generované použitím CMU's node-movement generator.

2.4.2.1 Generovanie automatického pohybovania uzlov

V tejto časti budeme hovoriť o generovaní náhodného pohybovania uzlov pomocou programu „setdest“ obsiahnutom priamo v distribúcii ns2. Tento program môžeme teda štandardne nájsť v zložke s adresou *~ns/indep-utils/cmu-scen-gen-setdest*

Program sa spúšťa s nasledovnými parametrami:

```
./setdest -n <počet uzlov> -p <pauza> -M <max. rýchlosť> -t <dĺžka simulácie>  
-x <rozmer osi x> -y <rozmer osi y> > <súbor pre uloženie generovaných dát>
```

Napr. ak by sme chceli vytvoriť scenár pohybu pre 15 uzlov, ktoré sa pohybujú maximálnou rýchlosťou 10.0 m/s s priemernou pauzou medzi pohybmi 1s a simulácia sa má ukončiť po 120 sekundách v celkovej rozlohe 200x200 m, tak zadané parametre programu budú nasledovné: **./setdest -n 15 -p 1.0 -M 10.0 -t 120 -x 200 -y 200 > gen-15-test.tcl**

Pomocou operátora presmerovania „>“ je možné presmerovať štandardný výstup, teda výpis na obrazovku do iného nami zadaného súboru. Vygenerovaný súbor obsahuje na začiatku prvotné pozície jednotlivých uzlov a ďalej jednotlivé náhodne vygenerované udalosti pohybu uzlov. Ďalej obsahuje informácie pre GOD objekt. Tento objekt GOD (General Operations Director) je využívaný na ukladanie globálnych informácií o aktuálnom stave jednotlivých zariadení v danej simulácii. V našom prípade je GOD použitý iba na ukladanie informácií o najmenšom počte preskokov nutných na dosiahnutie jedného uzla ďalším. Tieto informácie nie sú počítané počas behu simulácie z dôvodu veľkej konzumácie času pre samotný výpočet. Tieto informácie sú pevne zadávané a to buď priamo v simulácii, alebo z vygenerovaného súboru pohybového scenára. GOD objekt môže byť definovaný nasledovne:

```
$god_ set-dist 5 10 2
```

Čo znamená, že najmenší počet preskokov pre vzájomnú komunikáciu uzlov 5 a 10 sú 2 preskoky.

2.4.3 Smerovacie protokoly

V NS2 sú implementované nasledujúce smerovacie protokoly:

- **DSDV (Destination Sequenced Distance Vector)** – je úsekový vektorový smerovací protokol. Každý uzol obsahuje smerovaciu tabuľku, ktorá indikuje všetky miesta určenia, ktoré sú nasledujúcim skokom a obsahuje počet skokov na dosiahnutie cieľového miesta. Každý uzol periodicky obnovuje svoju smerovaciu tabuľku. Ďalej sú tu používané sekvenčné čísla na označovanie každej cesty. Cesta s vyšším

sekvenčným číslom je viac uprednostňovaná. Ak tabuľka obsahuje dve cesty s rovnakým sekvenčným číslom, je uprednostnená tá, ktorá má menší počet skokov. Keď uzol zistí, že cesta k cieľu je prerušená, nastaví počet skokov na nekonečno.

- **AODV (Ad-hoc On Demand Distance Vector)** – je tiež úsekový vektorový typ smerovania. Nevyžaduje používanie uzlov pre udržiavanie tabuľky ciest k cieľovému miestu, ktoré nie sú aktívne používané. Pokiaľ koncové body danej komunikácie majú funkčné spojenie tak AODV nevykonáva žiadne iné operácie. Protokol používa rôzne správy pre objavovanie a udržiavanie ciest. Tieto správy sú posielané cez UDP protokol. AODV používa cieľové smerové číslo pre každý záznam cesty. Toto číslo je vytvárané cieľom pre všetky smerovacie informácie, ktoré sú poslané uzlu, ktorý si ich vyžiadal. Sekvenčné číslo predchádza vytváraniu slučiek a určuje, ktorá z ciest je najvhodnejšia. Zo všetkých možných ciest je vybraná tá s väčším sekvenčným číslom. Ak uzol chce nájsť cestu k inému uzlu odošle broadcastovú správu RREQ (Route Requests) do všetkých sietí pokiaľ nenájde cieľ, alebo iný uzol ktorý obsahuje hľadanú cestu. V tom prípade správa RREP (Route Replies) je poslaná späť k zdroju a objavená cesta je pripravená k použitiu. Ak uzol rozpozná, že cesta k susednému uzlu je prerušená, tak vymaže smerovací záznam a pošle správu RERR (Route Errors) susedom, ktorý sú aktívny a použije danú cestu. Vďaka tomuto je možné udržiavať tabuľku aktívnych susedných uzlov. AODV neumožňuje ukladanie záznamov, ktoré nevedú k cieľu.
- **DSR (Dynamic Source Routing)** - tento typ smerovania je navrhnutý pre mobilné Ad-hoc siete, ktoré obsahujú viac ako 200 uzlov. Protokol pracuje „na vyžiadanie“ teda bez periodického aktualizovania smerovacích tabuliek. Jednotlivé pakety uchovávajú kompletnú cestu. Toto znižuje nároky pre veľké množstvo smerovacích aktualizácií prebiehajúcich v sieti. Všetky uzly obsahujú vo svojej pamäti všetky známe cesty. Pri objavovaní ciest zdroj odosielaťci paket cieľu odošle najprv RREQ paket. Uzol, ktorý prijal RREQ hľadá vo svojej smerovacej tabuľke cestu k cieľu, v prípade, že cesta nebola nájdená je RREQ paket ďalej odoslaný a uzol si pridá svoju adresu do ukladanej skokovej sekvencie. Toto sa opakuje pokiaľ nie je dosiahnutý cieľ. Cesta k zdroju je umožnená vďaka opačnému čítaniu skokovej sekvencie.

- **TORA (Temporally Ordered Routing Algorithm)** – tento protokol môže poskytovať viacero ciest medzi zdrojom a cieľom. TORA pozostáva z troch etáp: vytváranie, udržiavanie a vymazanie ciest. Na každom uzle je kópia protokolu TORA spustená pre každý cieľ oddelene. TORA vytvára riadený acyklický graf zameraný na cieľ. Týmto sa priradzuje váha každému uzlu v sieti. Dáta sú prenášané vzhľadom na váhu jednotlivých uzlov od uzla s väčšou váhou k uzlu s menšou váhou. Cesty sú objavované používaním QRY (Query) a UPD (Update) paketov. Keď uzol bez ciest pre dátový tok vyžaduje cestu k cieľu tak opakovane vysiela paket QRY, pokiaľ nenájde cestu k cieľu alebo cieľ samotný. Cieľový uzol odpovedá odoslaním UPD paketu obsahujúceho váhu uzla. Uzol prijímajúci UPD paket obnoví primerane záznam o váhe a odošle ďalší UPD paket. Z toho vyplýva, že môže existovať viacero ciest od zdroja k cieľu. Ak uzol objaví, že je daný cieľ nedosiahnuteľný nastaví váhu na maximálnu hodnotu. V dôsledku tohto uzol nemusí nájsť suseda s konečnou váhou pre dosiahnutie cieľa a pokúsi sa nájsť novú cestu. Ak sa zistí, že neexistuje žiadna cesta k cieľu, uzol vyšle správu CLR, ktorá vynuluje všetky smerovacie stavy a vymaže všetky neplatné cesty.

2.4.4 Hierarchické smerovanie

Táto časť pojednáva o základoch hierarchického smerovania implementovaného v NS2. V prvej časti sa pokúsime stručne nahliadnuť do samotného hierarchického smerovania a v druhej časti budeme v skratke písať o jednotlivých dôležitých nastaveniach tohto smerovania.

Hierarchické smerovanie bolo popri ostatných požiadavkách vymyslené hlavne kvôli zredukovaniu pamäťových nárokov simulácie veľmi veľkých topológií. Myšlienkou tohto smerovania je rozdelenie topológie do niekoľkých vrstiev, ktoré znižujú veľkosti smerovacích tabuliek. Veľkosť tabuľky je zredukovaná z n^2 , použitej pre priame smerovanie na veľkosť $\log n$, použitej pri hierarchickom smerovaní. Optimálny počet úrovní pre toto smerovanie bol v súčasnej verzii NS stanovený maximálne na tri úrovne hierarchického smerovania. Aby sme mohli využívať hierarchické smerovanie pri simulácii, je nutné definovať hierarchiu danej topológie a taktiež k jednotlivým uzlom priradiť hierarchickú adresu. Pri použití priameho smerovania všetky uzly topológie o sebe vzájomne vedia, čo spôsobuje veľkosť smerovacej tabuľky n^2 . Naopak pri hierarchickom smerovaní každý uzol vie iba o uzloch v danej vrstve, kde sa nachádza a pri zasielaní dát uzlu mimo jeho úroveň

tieto dáta zašle na hraničný bod danej úrovne, odkiaľ dáta ďalej pokračujú k cieľu. Tento spôsob teda redukuje veľkosť smerovacej tabuľky na $\log n$.

Hierarchické smerovanie vyžaduje určité mechanizmy pre simuláciu. Napr. užívateľ musí najprv špecifikovať samotné hierarchické smerovanie pred vytvorením danej topológie a to nasledovným spôsobom:

```
set ns [new Simulator]
$ns node-config -addressType hierarchical
```

V takomto prípade pri vytváraní uzlu volaním metódy simulátoru „node“

```
$ns node 0.1.0
```

sa v podstate vytvorí uzol s adresou 0.1.0. Ďalej sa používa trieda „AddrParams“ pre ukladanie informácií o hierarchii topológie ako napr. počet jej úrovní, počet miest v každej jednotlivéj úrovni, teda počet domén, clusterov a počet uzlov v každom clusteri.

V prípade, ak by bolo nutné definovať topológiu s dvomi doménami, každá by obsahovala 2 cluster a počet uzlov v prvom clusteri by bol 1 a v druhom 2, v treťom 3 a štvrtom clusteri 4 uzly, tak definícia by vyzerala nasledovne.

```
AddrParams set domain_num_ 2
lappend cluster_num 2 2
AddrParams set cluster_num_ $cluster_num
lappend eilastlevel 1 2 3 4
AddrParams set nodes_num_ $eilastlevel
```

2.5 Vytvorenie jednoduchéj bezdrôtovej simulácie

V tejto časti bude vytvorená jednoduchá dvojbodová simulácia bezdrôtovej siete. Topológia bude pozostávať z dvoch mobilných uzlov „*node(1)*“ a „*node(2)*“. Jednotlivé stanice sa budú pohybovať v oblasti 100m x 100m, prvotná pozícia každého uzla bude na opačných koncoch tohto pola a budú sa pohybovať smerom k sebe v prvej polovici simulácie a v druhej polovici sa začnú pohybovať od seba. Ďalej medzi nimi bude vytvorené TCP spojenie, jednotlivé pakety budú vymieňané medzi nimi akonáhle stanice budú v dosahu prenosovej vzdialenosti. Keď sa budú pohybovať od seba budeme vidieť ako budú jednotlivé pakety zahadzované.

Mobilné siete pozostávajú zo sieťových komponent a na začiatku tejto simulácie je nutné nadefinovať ich typ.

```
set prem(chan)          Channel/WirelessChannel    ;# typ kanalu
set prem(prop)          Propagation/TwoRayGround    ;# model radioveho sirenia
```

```

set prem(ant)           Antenna/OmniAntenna    ;# typ anteny
set prem(ll)            LL                     ;# typ linkovej vrstvy
set prem(ifq)           Queue/DropTail/PriQueue ;# typ rozhrania fronty
set prem(ifqlen)        50                     ;#dĺžka fronty
set prem(netif)         Phy/WirelessPhy        ;# typ sieťového rozhrania
set prem(mac)           Mac/802_11             ;# MAC typ
set prem(rp)            DSR                     ;# smerovací protokol
set prem(pocet)         2                      ;#počet uzlov
set prem(x)             100.0                  ;#rozmer x
set prem(y)             100.0                  ;#rozmer y
set prem(end)           20.0                   ;#čas konca simulácie

```

Ďalej prejdeme na hlavnú časť programu a začneme ho vytvorením inštancie simulátora

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

Potom vytvoríme súbor kde sa budú ukladať všetky dôležité trasovacie údaje a zavoláme procedúru trace-all {}, ktorá to umožní.

```

set tracefd [open tracefile1.tr w]
set namfd [open namfile1.nam w]
$ns trace-all $tracefd
$ns namtrace-all-wireless $namfd $prem(x) $prem(y)

```

V nasledujúcom kroku bude vytvorený objekt topológie, ktorý určuje oblasť pohybov jednotlivých mobilných uzlov v danom topologickom ohraničení.

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

Ďalej bolo určené, že mobilné uzly sa budú pohybovať v poli o veľkosti 100 x 100 m čo bude dosiahnuté príkazom, ktorý určuje objektu topologie rozмеры X a Y, ktoré sa rovnajú veľkosti tohto pola (X=100, Y=100).

```
$topo load_flatgrid $prem(x) $prem(y)
```

Objekt God (General Operation Director) sa vytvára nasledovne.

```
create-god $prem(pocet)
```

Tento objekt slúži pre uchovávanie globálnych informácií o jednotlivých stavoch daného prostredia, siete alebo uzlov. V tomto prípade objekt GOD uchováva totálny počet uzlov a tabuľku, ktorá obsahuje najmenší počet skokov vyžadovaných pre dosiahnutie jednotlivých uzlov navzájom.

Pred vytvorením mobilných uzlov ich je najprv nutné nakonfigurovať čo uskutočňuje uzlová konfigurácia API. Uzlová konfigurácia API môže pozostávať z viacerých definícií, ako je napríklad typ adresovania, typ ad-hoc trasovacieho protokolu, spojová vrstva (link layer LL), MAC vrstva a ďalšie.

Pre všetky tieto premenné je implicitne priradená hodnota NULL okrem typu adresovania (addressingType: flat).

Konfigurácia API pre vytvorenie mobilných uzlov vyzerá nasledovne:

```
$ns node-config -adhocRouting $prem(rp) \           ;#Ad-hoc smerovací protokol
                -llType $prem (ll) \             ;#typ linkovej vrstvy
                -macType $prem (mac) \           ;#typ MAC
                -ifqType $prem (ifq) \           ;#typ rozhrania fronty
                -ifqLen $prem (ifqlen) \         ;#dĺžka rozhrania fronty
                -antType $prem (ant) \           ;#typ antény
                -propType $prem (prop) \         ;#model rádiového šírenia
                -phyType $prem (netif) \         ;#typ sieťového rozhrania
                -topoInstance $topo \           ;#inštancia topológie
                -channelType $prem (chan) \      ;#typ kanálu
                -agentTrace ON \                 ;#trasovanie na úrovni agenta
                -routerTrace ON \               ;#trasovanie na úrovni smerovania
                -macTrace OFF \                 ;#trasovanie na úrovni MAC
                -movementTrace OFF              ;#trasovanie pohybu uzlov
```

Ďalej vytvorenie dvoch mobilných uzlov a vypnutie náhodného pohybu pre uzly:

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

Nasleduje zadanie pozícií jednotlivých uzlov:

```
$node(0) set X_ 30.0
$node(0) set Y_ 50.0
$node(0) set Z_ 0.0

$node(1) set X_ 70.0
$node(1) set Y_ 50.0
$node(1) set Z_ 0.0
```

Ak je vyžadovaná zmena smeru a rýchlosti pohybu jednotlivých bezdrôtových uzlov, použijeme nasledujúce príkazy.

```
$ns at 0.0 "$node(1) setdest 30.0 50.0 2.0"
$ns at 0.0 "$node(0) setdest 70.0 50.0 2.0"
```

Prvé dve čísla za výrazom „setdest“ určujú cieľové miesto pohybu a tretie číslo vyjadruje rýchlosť pohybu v m/s.

Nasleduje definícia dátového prenosu. V tomto prípade bude použité TCP spojenie medzi uzlom 0 a uzlom 1 pričom zdroj TCP bude uzol 0.

```
set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
$ns attach-agent $node(0) $tcp
$ns attach-agent $node(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
```

Ďalej je nutné nadefinovať jednotlivé hodnoty časov kedy sa prevedú dané príkazy a nadefinovanie procedúry pre správne ukončenie celého skriptu.

```
proc koniec {} {
    global ns tracefd namfd
    $ns flush-trace
    close $tracefd
    close $namfd
    exit 0
}

$ns at 1.0 "$ftp start"
$ns at $prem(end) "$ns nam-end-wireless $prem(end)"
$ns at $prem(end) "koniec"
```

Týmto je dané, že v čase 1.0 sekúnd od začiatku simulácie sa spustí ftp prenos a v čase 20.0 sekúnd sa simulácia ukončí. Úplne na konci skriptu sa nachádza príkaz, ktorý spúšťa samotný network simulator:

```
$ns run
```

2.5.1 Popis trasovacieho súboru pre bezdrôtové uzly:

s 1.0000000000 _0_ AGT --- 0 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0
r 1.0000000000 _0_ RTR --- 0 tcp 40 [0 0 0 0] ----- [0:0 1:0 32 0] [0 0] 0 0

Pole	Význam
1	Toto pole môže nadobúdať hodnoty r, s, f, D (received, send, forwarded, dropped). Môže mať taktiež hodnotu M pre udanie polohy alebo pre zistenie pohybu.
2	Čas
3	Číslo uzla
4	Toto pole určuje, ako sa paket zaoberá MAC vrstvou. Ak je to AGT indikuje paket z transportnej vrstvy, ak RTR zaoberá sa smerovaným paketom a ak je to IFQ určuje udalosti súvisiace s prioritou zásobníka.
5	Globálne sekvenčné číslo paketu.
6	Typ paketu (tcp, ack, udp).
7	Veľkosť paketu v bytoch.
8	Prvé číslo tohto pola je hexadecimálne číslo určujúce očakávaný čas v sekundách pre poslanie tohto paketu cez bezdrôtový kanál. Druhé číslo určuje identifikačné číslo MAC vysielajúceho uzla. Tretie určuje MAC-id prijímajúceho uzla. Štvrté číslo určuje typ MAC.
9	IP adresa zdroja a cieľa a hodnota ttl (time to live) paketu.
10	TCP informácie: sekvenčné číslo a číslo potvrdenia (acknowledgement).

Tab. 2.1: Tabuľka popisu jednotlivých polí trasovacích dát paketov

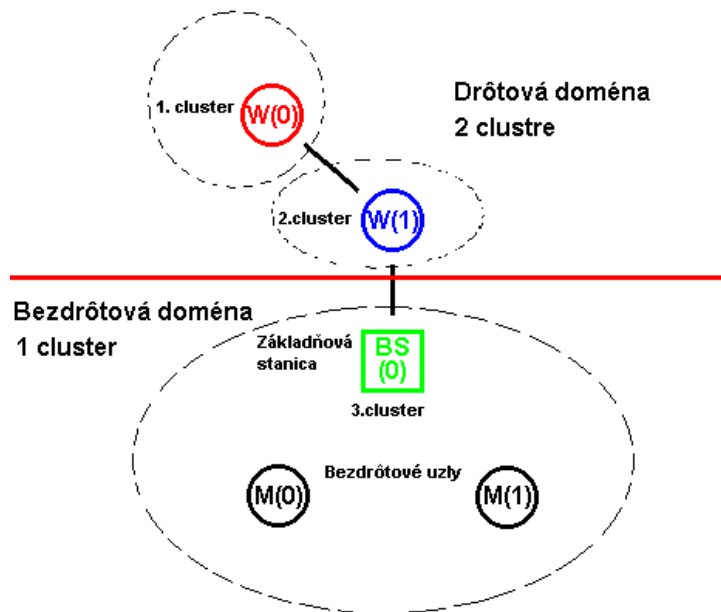
M 0.00000 0 (30.00, 50.00, 0.00), (30.00, 50.00), 0.00
M 0.00000 1 (70.00, 50.00, 0.00), (70.00, 50.00), 0.00

Pole	Význam
1	Paket určujúci polohu a pohyb paketu.
2	Čas
3	Číslo uzla
4	Pôvodné umiestnenie
5	Cieľové umiestnenie
6	Rýchlosť

Tab. 2.2: Tabuľka popisu jednotlivých polí trasovacích dát pohybov

2.6 Simulácia hybridných sietí

Pre simuláciu hybridných sietí, teda v našom prípade vzájomne prepojených pevných a mobilných sietí NS2 tiež ponúka rozsiahle možnosti. V nasledujúcom texte budú popísané základné kroky vytvorenia simulácie, ktorá simuluje takto vzájomne prepojené pevné a mobilné siete na základe hierarchického smerovania. Nasledujúca topológia zobrazená na obr. 2.3 obsahuje dve domény a to drôtovú a bezdrôtovú, v ktorých prebieha celý prenos dát.



Obr. 2.3: Príklad hybridnej siete

Hlavným problémom, ktorý zabraňuje takejto simulácii, je problematika smerovania. V nástroji NS2 sú smerovacie informácie generované na základe konektivity jednotlivých uzlov v topológii, tzn. ako sú jednotlivé uzly prepojené pomocou pevných spojov tzv. „Links“. Mobilné, teda bezdrôtové uzly takúto možnosť nemajú, čo vyplýva aj z ich podstaty. V takomto prípade sú jednotlivé pakety prenášané pomocou ich vlastných smerovacích protokolov, akými sú napr. DSR, DSDV, AODV a TORA, medzi nimi navzájom. Problém prepojenia rieši použitie základňovej stanice (BS), ktorú bude obsahovať bezdrôtová doména. Úlohou základňovej stanice je vytvorenie brány medzi doménami, ktorej funkciou je upraviť všetok tok dát tak, aby mohli byť distribuované ako mobilným bezdrôtovým, tak aj pevným drôtovým prostredím. Základňová stanica je v podstate hybrid medzi uzlom pevným a mobilným. Pre jej použitie je nutné použiť hierarchické smerovanie. Každá bezdrôtová doména so svojou základňovou stanicou má svoju vlastnú unikátnu doménovú adresu, ktorá jej je priradená.

DSDV agent pri smerovaní paketu najprv zistí, či je cieľ mimo danej domény. Ak áno, tak sa snaží smerovať paket k určenej základňovej stanici. V prípade, že cesta k BS nie je nájdená, je paket zahodený. V opačnom prípade, ak je paket smerovaný z inej domény k BS, je postup analogický. DSR agent pri prijatí paketu cieleného mimo danú doménu odošle požiadavku pre vyhľadanie cesty (route-query) pre jeho základňovú stanicu v prípade, že cesta k základňovej stanici nie je známa. Paket je dočasne uložený v pamäti do doby, kedy príde odpoveď na požiadavku (route-reply) zo základňovej stanice. V prípade príchodu odpovede na požiadavku je následne paket vybavený hlavičkou, ktorá obsahuje potrebné smerovacie informácie a takto pripravený paket je odoslaný k základňovej stanici. Základňová stanica spracuje paket a distribuuje ho daným prostredím k cieľu.

Simulácia je veľmi podobná simulácii v predchádzajúcej kapitole. Ako smerovací protokol je zvolený DSDV. Je nutné do skriptu pridať nasledujúce riadky, ktoré definujú počet pevných drôtových uzlov a počet základňových staníc použitých v simulácii.

```
set num_wired_nodes    2  
set num_bs_nodes      1
```

Následne je nutné nastavenie hierarchického smerovania, ktoré je potrebné, ako už bolo vysvetlené vyššie, pre simuláciu hybridných sietí. A to nasledujúcim nastavením:

```
$ns node-config -addressType hierarchical  
AddrParams set domain_num_ 2  
lappend cluster_num 2 1  
AddrParams set cluster_num_ $cluster_num  
lappend eilastlevel 1 1 3  
AddrParams set nodes_num_ $eilastlevel
```

Prvý riadok predchádzajúceho kódu definuje, že jednotlivé uzly budú využívať hierarchického smerovania. V nasledujúcich riadkoch sa definuje hierarchia topológie. Zo zdrojového kódu vyplýva, že simulácia bude obsahovať 2 domény, a to jedna pevná, teda drôtová a jedna bezdrôtová, obsahujúca mobilné uzly a samotnú základňovú stanicu. Počet clusterov v doménach je definovaný ako „2 1“, čo znamená, že prvá doména v našom prípade pevná drôtová bude obsahovať 2 cluster a druhá v poradí, teda bezdrôtová bude obsahovať 1 cluster. Ďalší riadok definuje počet uzlov v danom clusteri „1 1 3“. Znamená to, že v poradí prvý cluster bude obsahovať 1 uzol, druhý takisto 1 uzol a nasledujúci teda tretí cluster v poradí bude obsahovať 3 uzly, z čoho vyplýva, že hierarchia topológie je definovaná do troch úrovní. Je nutné upozorniť, že všetky trasovacie údaje pre hybridnú sieť sa generujú ako pre mobilnú bezdrôtovú, tak aj pre drôtovú doménu, ktoré sú následne ukladané do jedného trasovacieho súboru. Pre rozlíšenie takto ukladaných informácií všetky bezdrôtovo

prenášané pakety sú na začiatku označené skratkou „WL“. V ďalšom kroku je nutné vytvoriť jednotlivé uzly pevné, mobilné a základňovú stanicu. Keďže sme v predchádzajúcom kroku povolili a konfigurovali hierarchické smerovanie, tak toto platí na všetky následne vytvorené uzly, a teda je nutné im priradiť logické IP adresy a to nasledujúcim spôsobom:

```
set temp {0.0.0 0.1.0 0.2.0}
for {set i 0} {$i < $num_wired_nodes} {incr i} {
    set W($i) [$ns node [lindex $temp $i]]
}
```

Pre vytvorenie ďalších uzlov vrátane základňovej stanice je postup analogický. Je nutné, aby bola dodržaná aj hierarchická štruktúra jednotlivých IP adries. V tomto prípade máme 2 domény, a to doména 0 pre pevné drôtové uzly, a doména 1 pre mobilné uzly. Ďalej pevná drôtová doména obsahuje 2 cluster, z ktorých každý obsahuje 1 uzol, teda platí, že ich adresa má formát 0(doména).0(cluster v doméne).0(uzol v clusteri). Analogicky pre druhý uzol v tejto doméne platí adresa 0.1.0. Pre bezdrôtovú doménu sme definovali, že obsahuje 1 cluster, takže adresy jednotlivých uzlov sa budú vytvárať taktiež analogicky ako v nulte doméne. Základňová stanica bude mať adresu 1(doména).0(cluster v doméne).0(uzol v clusteri) a jednotlivým bezdrôtovým uzlom v tejto doméne sa bude líšiť iba posledné číslo adresy, ktoré určuje číslo uzlu v clusteri. Pri vytváraní základňovej stanice je nutné vedieť niekoľko faktov. Keďže zohráva úlohu brány, z jej podstaty vyplýva, že musí byť schopná komunikovať ako v pevnom drôtovom prostredí, tak aj v bezdrôtovom prostredí. To znamená, že v jej konfiguračnom API musí mať zapnutý drôtový smerovací mechanizmus, čo je možné pomocou príkazu:

\$ns node-config -wiredRouting ON

Po vytvorení a nakonfigurovaní základňovej stanice musíme pre ďalšie potreby toto nastavenie vypnúť nastavením príznaku na „OFF“. Všetky ostatné nastavenia pre bezdrôtové uzly ostávajú rovnaké ako pre základňovú stanicu. Pre každú základňovú stanicu priradenú k danej doméne platí, že všetky pakety pochádzajúce od bezdrôtových uzlov tejto domény smerujúcich mimo ich domovskú doménu budú smerované priamo k nej. Dôležitým faktom je, že všetky pakety smerované do bezdrôtovej domény, ktoré pochádzajú z inej domény, než do ktorej patrí daná BS budú smerované priamo k nej a ona na základe jej Ad-hoc smerovacieho protokolu presmeruje pakety do korektného uzla. Z predchádzajúceho textu pre simuláciu hybridných sietí teda vyplýva, že je nutné povoliť hierarchické smerovanie, ďalej vytvoriť separátne domény pre pevné a mobilné uzly, ktorých môže byť viacero pre simulácie viacnásobne prepojených sietí a na koniec priradiť ku každej bezdrôtovej doméne maximálne 1 základňovú stanicu, cez ktorú budú komunikovať jednotlivé uzly s uzlami ostatných domén.

3. Postup tvorby laboratórných úloh

Základnou časťou laboratórnej úlohy je nainštalovaný všetok potrebný software, čo v našom prípade znamená operačný systém unixového typu a distribúciu Network Simulator 2. Ale skôr než sa žiaci dostanú k samotnému simulovaniu bezdrôtových sietí, je nutné sa zoznámiť so samotným simulačným nástrojom.

3.1 Popis a výber kapitol

Kapitola 1 – Základy bezdrôtových sietí štandardu 802.11

- zaoberá sa základnou otázkou využitia bezdrôtových sietí, jednotlivými komponentami siete a stručne nahliada do dvoch vrstiev OSI modelu

Kapitola 2 – Úvod do NS2

- je popísaná základná otázka programovania v jazyku TCL
- možnosti pre simulácie 802.11 sietí v NS2
- bližšie priblíženie simulácií na jednotlivých príkladoch základných modelov topológií sietí

Do laboratórnej úlohy sme vybrali len stručný popis, to znamená len najnutnejšie témy, bez ktorých sa študent meriaci úlohu nezaobíde. Boli vytvorené dve laboratórne úlohy na tematiku simulácie bezdrôtových sietí štandardu 802.11. Pre každú laboratórnu úlohu boli vybrané iné témy simulácie a tak obsahujú rôzne podklady ktoré sme sa snažili čo najstručnejšie vybrať z jednotlivých kapitol, aj keď sa opierajú prakticky o ten istý základ.

3.2 Základné požiadavky

- ✓ Systém unixového typu s grafickým X rozhraním nainštalovaný na počítači
- ✓ Správne nainštalovanú distribúciu NS2 (najvhodnejšia je vždy najaktuálnejšia verzia)
- ✓ Zadané laboratórných úloh
- ✓ Vytlačený manuál k NS2 alebo aspoň v elektronickej podobe

3.3 Doporučenie

Hlavným doporučením je aby vedúci cvičenia bol dobre oboznámení s týmto simulačným nástrojom a aspoň so základnou prácou v Unixe, teda aby vedel dobre a spoľahlivo odpovedať na prípadné otázky vzniknuté behom merania. Ďalej by bolo vhodné tento nástroj spomenúť na prednáške aby študenti mali aspoň čiastočný prehľad, prípadne aby na začiatku vedúci cvičenia tento nástroj študentom merajúcim túto úlohu dokázal priblížiť, teda jeho základné črty a možnosti, vďaka ktorými môže študentov motivovať pre ďalšiu prácu s týmto systémom .

3.4 Rozvrhnutie laboratórnej úlohy

1. *Úvod* - zoznámenie sa s laboratórnou úlohou a základnými črtami programovacieho jazyka TCL
2. *Teoretický úvod* - stručný popis simulačného modelu a možností pre simuláciu bezdrôtových sietí v NS2.
3. *Praktická časť* - stručný popis vybranej simulácie, topológie siete a jednotlivých komponent, ktoré obsahuje simulácia
4. *Samostatná práca* - práca ktorú musia študenti urobiť sami, podľa zadaných bodov, podľa získaných poznatkov z predchádzajúcich troch kapitol a zo samotného skriptu ktorý je dostatočne okomentovaný a tak je práca s ním intuitívna
5. *Zhrnutie* - záver a ciele cvičenia
6. *Použitá literatúra* - odkazy na literatúru, ktorá môže prípadným záujemcom o tento nástroj pomôcť k budúcemu rozvoju

3.4.1 Praktická časť laboratórnej úlohy

Rozširuje poznatky získané z teoretického úvodu, stručne popisuje prvotný model simulovanej siete. Je písaný čo najstručnejšie aby laboratórna úloha nebola príliš obsiahla a snaží sa priblížiť jednotlivé mechanizmy či už komunikácie a dátových prenosov medzi uzlami, ale taktiež približuje ďalšie nástroje pre analýzu prenosu dát akými sú *nam* a *xgraph*.

3.4.2 Samostatná práca

V tejto časti sú študenti vedení k vlastnému uvažovaniu a mysleniu. Ich úlohou bude využiť poznatky získané z teoretického úvodu, praktickej časti a zo samotného skriptu, ktorý si je dôležité potrebné preštudovať pred podniknutím ďalších krokov uvedených v tejto časti. Táto kapitola je študentovi podaná tak aby nebol zbytočne zavádzaný nepodstatnými faktami pričom v určitých krokoch je mu podaný ďalší stručný výklad jednotlivých potrebných komponent tak aby bol vedený k správne výsledku merania.

Zmyslom týchto laboratórnych úloh nebolo naučiť študenta profesionálne programovať a ani využiť všetkých možností NS2, ktorých je nespočetne veľa, ale priblížiť tento nástroj a jeho základné možnosti ktoré poskytuje pre simuláciu bezdrôtových sietí a taktiež hybridných, teda drôtových a bezdrôtových sietí vzájomne prepojených. Taktiež sme sa nimi snažili zaujať a motivovať študentov pre ďalšie využitie tohto nástroja. Študent je taktiež na konci cvičenia odkázaný na správne riešenie laboratórnej úlohy v prípade neúspechu pochopenia tohto nástroja, ale taktiež k porovnaniu správnosti jeho vlastného merania.

3.5 Časový plán úlohy

Laboratórna úloha je navrhnutá na časové rozloženie do dvoch pracovných hodín. Tento čas by mal byť dostačujúci pre naštudovanie celej úlohy a pre vykonanie jednotlivých zadaných modifikácií modelov sietí. Keďže študenti majú rôzne počítačové zručnosti a schopnosti, je možné že niektorí vykonajú jednotlivé body rýchlejšie a niektorí pomalšie, v každom prípade bol pri vytváraní úlohy braný ohľad na tieto fakty a teda by nemal robiť väčší problém tieto úlohy zvládnuť aj pre úplného začiatníka.

ZÁVER

Počas práce sme sa stručne zoznámili s princípom fungovania bezdrôtových sietí a ich simuláciou v nástroji NS2 kde boli prebrané základy programovania TCL. Ďalej sa práca zaoberala možnosťami, ktoré NS2 poskytuje pre tieto simulácie ukázané na jednoduchých príkladoch. Obe simulácie sa snažili čo najviac približovať skutočnej prevádzke.

Cieľom práce nebolo naučiť užívateľa programovať na profesionálnej úrovni, ale iba priblížiť tento nástroj (resp. možnosti) a jeho jednotlivé možnosti pre simulácie bezdrôtových sietí na dvoch laboratórnych úlohách uvedených v prílohe. Keďže týchto možností je veľké množstvo, boli vybrané len tie najdôležitejšie a najbežnejšie využívané. Táto práca sa taktiež snažila mať motivačný charakter pre ďalšie využívanie tohto simulačného nástroja.

Riešenie laboratórnych úloh je časovo obmedzené a tak pri ich vytváraní bola venovaná hlavná snaha čo najúčinnnejšiemu výkladu jednotlivých možností pre užívateľa.

V prvej časti práce sme sa snažili priblížiť štandard 802.11 a jeho možnosti pre praktické využitie v bežnom živote. Druhá časť sa venovala samotnému priblíženiu simulačného nástroja Network Simulator 2, základy programovania v jazyku TCL, ďalej jednotlivé základné kroky pri vytváraní funkčného skriptu, samozrejmosťou boli možnosti ktoré tento nástroj ponúka pre simulovanie bezdrôtových sietí. Posledná teda tretia časť, čerpá poznatky získané z prvých dvoch kapitol a bola venovaná vytvoreniu dvoch komplexných laboratórnych úloh, ktorých cieľom je ďalšie približovanie toto nástroja bežnému užívateľovi.

Zoznam použitej literatúry

- [1] ZANDL, Patrick: *Bezdrátové sítě WiFi*. 1. vyd. Brno: ComputerPress a.s, 2006. 190s. ISBN 80-722-6632
- [2] Information Sciences Institute. "The Network Simulator - ns-2". [online]. Dostupné z <<http://www.isi.edu/nsnam/ns/>>.
- [3] ALTMAN, Eitan and JIMÉNEZ, Tania: *NS Simulator for beginners*. Mérida: Univ. de Los Andes, 2003. Dostupné z <<http://evanjones.ca/ns2/>>
- [4] FALL Kevin and VARADHAN, Kannan. The ns Manual : formerly ns Notes and Documentation. [online]. Dostupné z <<http://www.isi.edu>>.
- [5] JONES Evan. 802.11 Simulation within NS2. [online] posledná aktualizácia 16.4. 2006. Dostupné z <<http://evanjones.ca/ns2/>>.
- [6] GREIS Marc. Tutorial for the Network Simulator. [online] posledná aktualizácia 3.7. 2002. Dostupné z <<http://www.isi.edu/nsnam/ns/tutorial/index.html>>.
- [7] XIUCHAO Wu. Simulate 802.11 Channel within NS2. [online]. Dostupné z <http://www.comp.nus.edu.sg/~wuxiucha/research/reactive/report/80211ChannelinNS2_new.pdf>.
- [8] IEEE Computer Society. IEEE std 802.11, 2007. [online]. Dostupné z <<http://standards.ieee.org/getieee802/download/802.11-2007.pdf>>. ISBN: 0-7381-5656-6

Zoznam symbolov, veličín a skratiek

ACK Acknowledge

AP Access Point

BSA Basic Service Area

CCK Complementary Code Keying

CRC Ciclic Redundanty Check

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

CTS Clear To Send

DIFS Distributed Interframe Space

DS Distribution System

DSSS Direct Sequence Spread Spectrum

ESS Extended Service Set

FHSS Frequency Hopping Spread Spectrum

GHz Giga hertz

IEEE Institute of Electrical and Electronics Engineers

IFS Inter Frame Spacing

ISM Industrial Scientific Medical

MAC Medium Access Control protocol

Mb/s Megabit per second

MHz Mega hertz

NAM Network Animator

NAV Network Allocation Vector

NS2 Network Simulator

OFDM Orthogonal Frequency Division Multiplexing

OSI Open Systems Interconnection

OTCL Object Tool Command Language

PHY Physical Layer

RTS Request To Send

SS Spread Spectrum

TCL Tool Command Language

WECA Wireless Ethernet Compatibility Alliance

μs Mikrosekunda

Zoznam príloh

PRÍLOHA A – 1. LABORATÓRNA ÚLOHA - SIMULÁCIA JEDNODUCHEJ 802.11 SIETI..... 46

A.1 ÚVOD	46
A.2 TEORETICKÝ ÚVOD.....	46
A.3 PRAKTICKÁ ČASŤ	48
A.4 SAMOSTATNÁ PRÁCA	49
A.5 ZHRNUTIE.....	50
A.6 POUŽITÁ LITERATÚRA	50

PRÍLOHA B – 2. LABORATÓRNA ÚLOHA - SIMULÁCIA HYBRIDNÝCH BEZDRÔTOVÝCH

802.11 SIETÍ..... 51

B.1 ÚVOD	51
B.2 TEORETICKÝ ÚVOD.....	51
B.3 PRAKTICKÁ ČASŤ.....	53
B.4 SAMOSTATNÁ PRÁCA.....	54
B.5 ZHRNUTIE.....	55
B.6 POUŽITÁ LITERATÚRA	55

PRÍLOHA A – 1. Laboratórna úloha - Simulácia jednoduchej 802.11 sieti

A.1 Úvod

Len pred niekoľkými rokmi sa celosvetová počítačová sieť stala bežným denne využívaným nástrojom. Technický pokrok sa čoraz ďalej vyvíjal a vďaka rozšíreniu výpočtovej techniky bola vyvíjaná rada simulačných programov, ktoré mali jednoduchším a hlavne lacnejším spôsobom nahradiť experimentálne testovanie rôznych sietí, sieťových protokolov a jednotlivých sieťových prvkov. V súčasnosti existuje obrovské kvantum takýchto aplikácií umožňujúcich práve spomínané simulovanie a experimentovanie s virtuálnymi sieťami. Nasledujúci text je zameraný na simulačný nástroj *Network Simulátor 2*, označovaný skratkou *NS2*. *NS2* je založený na dvoch programovacích jazykoch a to objektovo orientovanom simulátore, napísanom v C++ a OTCL interpretovy, používaného pre vykonávanie užívateľských príkazových skriptov. V OTCL skripte môžu byť definované jednotlivé sieťové topológie, špecifické protokoly a aplikácie, ktoré si prajeme simulovať. C++ je používaný pre manipuláciu s dátami napríklad udalosti súvisiace s jednotlivými paketmi. a OTCL pre kontrolu napríklad jednotlivých periodických alebo spúšťacích a riadiacich nadchádzajúcich udalostí. Tento nástroj je veľmi výkonný a rozsiahly. Z tohto dôvodu budú nasledujúce kapitoly obsahovať čo najstručnejší popis a úvod do danej problematiky. Budúcich užívateľov a záujemcov tohto nástroja odkazujem preto na zdroje uvedené v zozname použitej literatúry.

A.2 Teoretický úvod

- Základné operácie, ktoré ns 2 umožňuje:
- Vybudovanie topologie siete
- Vybudovanie spojenia
- Generovanie prevádzky
- Simulovanie pohybovania uzlov
- Monitorovanie prevádzky

Budovanie určitej topológie siete sa prevádza vytvorením textového súboru v jazyku TCL. Prvým krokom je vytvorenie objektu (inštancie) simulátora, prevádza sa to príkazom **set ns [new simulator]**. Premenná **ns** je odkaz na vytvorenú inštanciu tohto objektu, vďaka ktorej je možné vytvárať uzly a následne prepojenie medzi nimi.

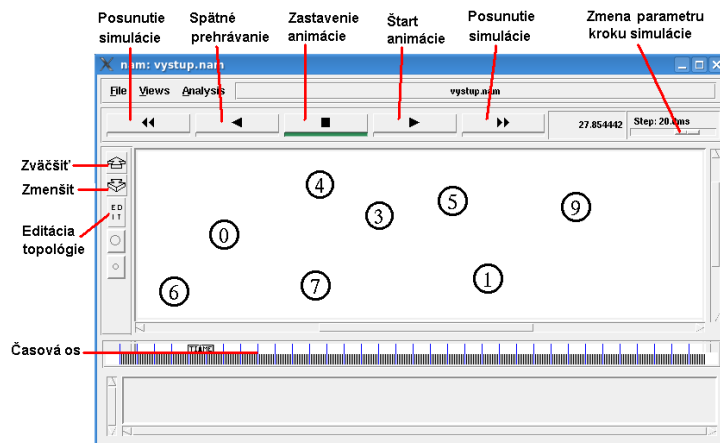
Pred samotným vytváraním jednotlivých bezdrôtových uzlov je nutné nastaviť konfiguráciu API, ktorá popisuje ich vlastnosti a správanie v sieti, teda v našom prípade zaistí že sa jedná o uzly komunikujúce pomocou štandardu 802.11, ktorý je implementovaný defaultne v NS2. Pre záujemcov je možné nadefinovaním ďalších parametrov vytvoriť simulácie rôznych revízií tohto štandardu. Všetky nastavenia konfigurácie API sa prevádzajú za príkazom **\$ns_ node config** a jednotlivé parametre sú dostatočne popísané v priloženom skripte. Keď je konfigurácia API dokončená, môžeme pristúpiť k samotnému vytváraniu bezdrôtových uzlov. To sa prevádza príkazom **set n0 [\$ns node]**, kde premenná **n0** je odkaz na vytvorenú inštanciu uzla. Je možné jednotlivým uzlom zadať ich prvotné pozície v topológii, ale taktiež je možné využiť ich náhodné rozmiestnenie pomocou nastavenia príznaku príkazu **random-motion** na **1**. Teda príkaz bude **\$node0 random-motion 1**.

Po nadefinovaní celej topológie nasleduje definícia charakteru prenosu dát v sieti. Pre tento účel sa v NS2 využíva trieda „Agent“. Agenti predstavujú koncové body, kde sú jednotlivé pakety vytvárané, respektíve konzumované a sú využívané pri implementácii

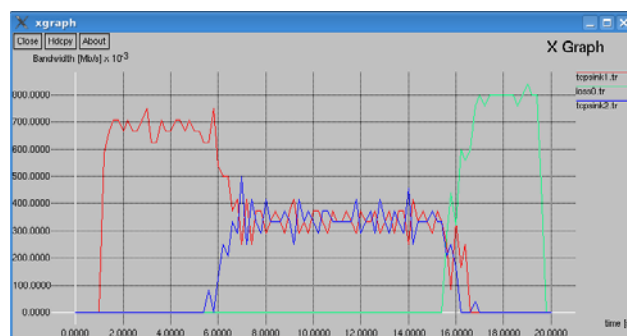
dátových protokolov na rôznych sieťových vrstvách. Príkladom definície dátového prenosu využívajúceho spoľahlivý transportný protokol (TCP) je **set tcp1 [new Agent/TCP]**, kde **tcp1** je odkaz na inštanciu tohto zdroja. Príkladom definície konzumenta takéhoto prenosu je **set sink1 [new Agent/TCPSink]**, reprezentovaného inštanciou **sink1**, ktorý je určený na jednoduché pohltenie dát s možnosťou ďalšej analýzy prijatých dát. Prepojenie vytvoreného zdroja a cieľa sa prevádza príkazom **\$ns connect \$tcp1 \$sink1**. V prípade nespoľahlivého transportného protokolu je situácia analogická, teda **set udp1[new Agent/UDP]**, **set udpsink1 [new Agent/Null]**, **\$ns connect \$udp1 \$udpsink1**, vytvorí zdroj, cieľ a vzájomne ich prepojí. Po definícii spôsobu prenosu je nutné nadefinovať samotný generátor dát(paketov). Príkladom definície takéhoto generátora je **set ftp1 [new Application/FTP]**, ktoré vytvorí generátor modelujúci prenos dát pomocou protokolu FTP. Príkazom **\$ftp1 attach-agent \$tcp1** naviažeme vytvorený generátor na vyššie definovaný transportný protokol **tcp1**. Takýchto generátorov NS2 podporuje veľké množstvo, pričom postup ich vytvorenia a prepojenia je analogický. Taktiež je možné ovplyvňovať ich jednotlivé parametre.

Spúšťanie jednotlivých generátorov a rôznych udalostí simulácie sa prevádza pomocou príkazu **at**. Napríklad **\$ns at 10.0 "\$ftp1 start"**, čo spustí generátor **ftp1** v čase 10 sekúnd od začiatku simulácie. V úvodzovkách je uvedený príkaz ktorý sa má vykonať v zadanom čase.

Vizualizácia celej simulácie prebieha nezávisle na simulačnom nástroji NS2. Pre grafické zobrazenie jednotlivých udalostí akými sú pohyby uzlov, generovanie dát, udalosti spojené s paketmi ako vznik, strata, zahodenie, spracovanie, udalosti spojené s frontami atď. sa využíva program implementovaného priamo v distribúcii NS2 a to **nam** (Network animator) Obr. 1. Pre grafické zobrazenie jednotlivých priebehov simulácie ktoré sú generované do ASCII súboru je najčastejšie používaný **xgraph** Obr. 2.



Obr. A.1: Vizualizácia simulácie pomocou nam

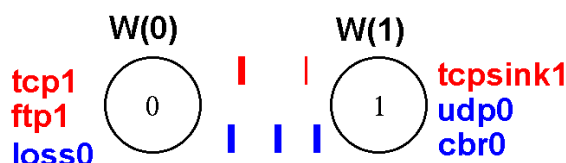


Obr. A.2: Časové priebehy zobrazené pomocou programu xgraph

A.3 Praktická časť

Táto časť sa skladá z overenia vybraných vlastností NS2 na dopredu pripravených modeloch siete, ktorých presné umiestnenie určí vedúci cvičenia a na ich čiastočnej modifikácii podľa pokynov uvedených v zadaní. Modely siete sú napísané v jazyku TCL a sú uložené v textových súboroch s príponou **.tcl**.

Otvorte v textovom editore súbor **wireless1.tcl** a podrobne preštudujte všetky kroky vytvorenia simulácie. Jednotlivé príkazy sú dostatočne okomentované a tak by ich pochopenie malo byť triviálne. Topológia siete je zobrazená na Obr. 3



Obr. A.3: Topológia siete definovaná v súbore **wireless1.tcl**

Tento model znázorňuje jednoduchú sieť obsahujúcu dva uzly **W(0)** a **W(1)** komunikujúce pomocou štandardu 802.11. Tieto dva uzly sú prepojené bezdrôtovo pomocou smerovacieho protokolu **DSR**. Tento protokol pracuje na vyžiadanie, teda bez periodického aktualizovania smerovacích tabuliek, vysiela pakety s požiadavkou pre nájdenie cesty k cieľu, vďaka ktorým sú potom jednotlivé dátové pakety vybavené celou cestou k cieľu. Tento spôsob smerovania je najvhodnejší pre simuláciu veľkých sietí, lebo znižuje nároky na pamäťové požiadavky jednotlivých uzlov. Existujú ďalšie smerovacie protokoly a to **DSDV**, **AODV** a **TORA**. K uzlu **W(0)** je priradený agent **tcp1**, ktorý je prepojený s objektom **tcpsink1** priradeným k uzlu **W(1)**. Objekt **tcpsink1** pohlcuje prichádzajúce pakety a odosiela potvrdzovacie pakety ACK. O generovanie prevádzky sa stará FTP generátor s názvom **ftp1** priradeného k agentovi **tcp1**. Dáta sú taktiež generované a prenášané od uzla **W(1)** smerom k uzlu **W(0)** a to pomocou agenta **udp1** ku ktorému je priradený objekt **cbr1** ktorý generuje dáta s konštantnou bitovou rýchlosťou, ktorú je možné nastaviť úpravou vhodných parametrov tohto objektu. Naopak k uzlu **W(0)** je priradený objekt **loss0**, ktorý zohráva úlohu konzumenta paketov generovaných pomocou **cbr1**. Prevádzka FTP je zahájená po 1 sekunde od začiatku simulácie a prevádzka CBR po 5 sekundách od začiatku simulácie.

Po preštudovaní modelu overte jeho funkčnosť v simulačnom prostredí NS2. Zmeňte pracovný adresár tak aby odpovedal adresáru, kde je uložený súbor **wireless1.tcl**. Zadaním príkazu **ns wireless1.tcl** do príkazového riadku spustíte simulátor, ktorý následne vygeneruje niekoľko súborov. V súbore **vystup.tr** sú uložené trasovacie údaje simulácie, v súbore **vystup.nam** sú uložené vygenerované údaje pre **nam**, ktoré slúžia na vizuálne prehliadanie simulácie a to zadaním príkazu **nam vystup.nam** do príkazového riadku. Sú vygenerované taktiež dva súbory pre grafické zobrazenie priebehov prenosu dát v sieti, ktoré sa prehliadajú programom **xgraph**, zadaním príkazu **xgraph graf1.tr**. Taktiež je mu možné predať viacero názvov súborov ktoré obsahujú dáta pre vykreslenie grafu, aby bolo možné vykresliť viacero priebehov do jedného grafu.

A.4 Samostatná práca

Po overení funkčnosti modelu preskúmajte vplyv zmeny jednotlivých parametrov, uvedených v skripte, na zmenu priebehov komunikácie.

V nasledujúcom kroku rozšírte model tak aby obsahoval 3 mobilné uzly, pričom uzol **W(1)** bude slúžiť ako konzument paketov generovaných uzlom **W(2)**, ku ktorému bude priradený agent **tcp2**, a bude generovať pakety pomocou objektu **ftp2**. Taktiež tomuto novému uzlu priradíte ľubovoľne vybranú prvotnú pozíciu a taktiež následný pohyb v topológii. Pokúste sa vygenerovať pomocou procedúry **record** graf k tomuto priebehu prenosu dát. Procedúra **record** používa dva parametre a to ako prvý názov konzumenta a ako druhý parameter odkaz na súbor kam sa budú ukladať jednotlivé dáta. Je dôležité aby ste tento súbor vytvorili na začiatku simulácie, ktorý sa vytvára rovnako ako ostatné súbory, ktoré sú popísané v skripte. Taktiež je nutné tento súbor po ukončení simulácie validne ukončiť a to pridaním vhodných údajov do procedúry stop. Overte funkčnosť rozšíreného skriptu. Sledujte ako ovplyvní prídanie nového uzla celkovú prenosovú rýchlosť medzi uzlami.

V ďalšej úlohe postupujte obdobne ako v prvej časti merania a rozšírte súbor **wireless2.tcl** ktorý má modelovať simuláciu 10 bezdrôtových uzlov a vytvorte dátový prenos medzi nimi. Je možné ručne nadefinovať jednotlivé uzly, ale tento spôsob je neefektívny pri vytváraní väčších simulácií. K tomuto účelu sú priamo v distribúcii NS2 obsiahnuté dva programy a to **setdest** pre generovanie pohybov uzlov a **cbrgen.tcl** pre vygenerovanie dátovej prevádzky medzi uzlami.

Program **setdest** môžeme teda štandardne nájsť v zložke s adresou **~ns/indep-utis/cmu-scen-gen-setdest**. Program sa spúšťa s nasledovnými parametrami:

```
./setdest -n <počet uzlov> -p <pauza> -M <max. rýchlosť> -t <dĺžka simulácie>  
-x <rozmer osi x> -y <rozmer osi y> > <súbor pre uloženie generovaných dát>
```

Napr. ak by sme chceli vytvoriť scenár pohybu pre 10 uzlov, ktoré sa pohybujú maximálnou rýchlosťou 10.0 m/s s priemernou pauzou medzi pohybmi 1s a simulácia sa má ukončiť po 120 sekundách v celkovej rozlohe 200x200 m, tak zadané parametre programu budú nasledovné: **./setdest -n 10 -p 1.0 -M 10.0 -t 120 -x 200 -y 200 > gen-10-test.tcl**

Pomocou operátora presmerovania „>“ je možné presmerovať štandardný výstup, teda výpis na obrazovku do iného nami zadaného súboru. Vygenerovaný súbor obsahuje na začiatku prvotné pozície jednotlivých uzlov a ďalej jednotlivé náhodne vygenerované udalosti pohybu uzlov. Ďalej obsahuje informácie pre GOD objekt. Tento objekt GOD (General Operations Director) je využívaný na ukladanie globálnych informácií o aktuálnom stave jednotlivých zariadení v danej simulácii. V našom prípade je GOD použitý iba na ukladanie informácií o najmenšom počte preskokov nutných na dosiahnutie jedného uzla ďalším. Tieto informácie nie sú počítané počas behu simulácie z dôvodu veľkej konzumácie času pre samotný výpočet. Tieto informácie sú pevne zadávané a to buď priamo v simulácii, alebo z vygenerovaného súboru pohybového scenára. GOD objekt môže byť definovaný **\$god_ set-dist 5 10 2**. Čo znamená, že najmenší počet preskokov pre vzájomnú komunikáciu uzlov 5 a 10 sú 2 preskoky.

Pre generovanie dátovej prevádzky nájdeme skript **cbrgen.tcl** na rovnakej adrese na ktorej sa nachádzal program **setdest**. Pre vytvorenie takéhoto scenára musíme zadať typ dátového toku (UDP alebo TCP), počet uzlov, maximálny počet jednotlivých prepojení dátového toku medzi nimi atď. Skript sa spúšťa s nasledovnými parametrami:

```
ns cbrgen.tcl -type <CBR/TCP> -nn <počet uzlov> -seed <seed>  
-mc <max. prepojení> -rate <rýchlosť> > <názov súboru>
```

Čas začiatku generovania TCP/UDP dátových prenosov medzi uzlami je náhodne generované číslo, ktorého maximálna hodnota je 180 sekúnd. V prípade, že požadujeme vytvoriť náhodný dátový tok medzi 10 uzlami s maximálnym počtom prepojení napríklad 8 so „seed“ hodnotou 1 a rýchlosťou 1 Mb/s, tak bude príkaz vyzerat' nasledovne:

```
ns cbrgen.tcl -type CBR -nn 9 -seed 1.0 -mc 10 -rate 1.0 > cbr-10-test.tcl
```

Obidva vygenerované súbory presuňte do pracovnej zložky kde sa nachádza skript, ktorých obsah následne preštudujte. Do jednotlivých premenných uvedených v hlavičke skriptu nadefinujte cestu k daným súborom. Spustite simuláciu zadaním **ns wireless2.tcl** a prehliadnite pomocou **nam**. Zaujímavcovia môžu prispôbiť skript tak aby vygeneroval patričné súbory pre vygenerovanie grafov jednotlivých priebehov. Celý postup je analogický ako v predchádzajúcom skripte.

A.5 Zhrnutie

Je možné skontrolovať správnosť merania pomocou výsledných skriptov, ktorých presné umiestnenie určí vedúci cvičenia. Cieľom tohto merania bolo priblížiť študentom niektoré z mnoho možností ktoré poskytuje NS2 pre simulácie bezdrôtových sietí. Bol ukázaný spôsob akým je možné generovať pohyby jednotlivých uzlov a taktiež generovať prenos dát medzi nimi manuálne a automaticky pomocou prídavných programov vďaka ktorým môžeme efektívnejšie a hlavne rýchlejšie vytvárať simulácie väčších rozmerov.

A.6 Použitá literatúra

- [1] Information Sciences Institute. "The Network Simulator - ns-2". [online]. Dostupné z < <http://www.isi.edu/nsnam/ns/> >.
- [2] ALTMAN, Eitan and JIMÉNEZ, Tania: *NS Simulator for beginners*. Mérida: Univ. de Los Andes, 2003. Dostupné z < <http://evanjones.ca/ns2/> >
- [3] FALL Kevin and VARADHAN, Kannan. The ns Manual : formerly ns Notes and Documentation. [online]. Dostupné z < <http://www.isi.edu> >.
- [4] GREIS Marc. Tutorial for the Network Simulator. [online] posledná aktualizácia 3.7. 2002. Dostupné z < <http://www.isi.edu/nsnam/ns/tutorial/index.html> >.

PRÍLOHA B – 2. Laboratórna úloha - Simulácia hybridných bezdrôtových 802.11 sietí

B.1 Úvod

Len pred niekoľkými rokmi sa celosvetová počítačová sieť stala bežným denne využívaným nástrojom. Technický pokrok sa čoraz ďalej vyvíjal a vďaka rozšíreniu výpočtovej techniky bola vyvíjaná rada simulačných programov, ktoré mali jednoduchším a hlavne lacnejším spôsobom nahradiť experimentálne testovanie rôznych sietí, sieťových protokolov a jednotlivých sieťových prvkov. V súčasnosti existuje obrovské kvantum takýchto aplikácií umožňujúcich práve spomínané simulovanie a experimentovanie s virtuálnymi sieťami. Nasledujúci text je zameraný na simulačný nástroj *Network Simulátor 2*, označovaný skratkou *NS2*. *NS2* je založený na dvoch programovacích jazykoch a to objektovo orientovanom simulátore, napísanom v C++ a OTCL interpretovy, používaného pre vykonávanie užívateľských príkazových skriptov. V OTCL skripte môžu byť definované jednotlivé sieťové topológie, špecifické protokoly a aplikácie, ktoré si prajeme simulovať. C++ je používaný pre manipuláciu s dátami napríklad udalosti súvisiace s jednotlivými paketmi. a OTCL pre kontrolu napríklad jednotlivých periodických alebo spúšťacích a riadiacich nadchádzajúcich udalostí. Tento nástroj je veľmi výkonný a rozsiahly. Z tohto dôvodu budú nasledujúce kapitoly obsahovať čo najstručnejší popis a úvod do danej problematiky. Budúcich užívateľov a záujemcov tohto nástroja odkazujem preto na zdroje uvedené v zozname použitej literatúry.

B.2 Teoretický úvod

- Základné operácie, ktoré ns 2 umožňuje:
- Vybudovanie topologie siete
- Vybudovanie spojenia
- Generovanie prevádzky
- Simulovanie pohybovania uzlov
- Monitorovanie prevádzky

Budovanie určitej topológie siete sa prevádza vytvorením textového súboru v jazyku TCL. Prvým krokom je vytvorenie objektu (inštancie) simulátora, prevádza sa to príkazom **set ns [new simulator]**. Premenná **ns** je odkaz na vytvorenú inštanciu tohto objektu, vďaka ktorej je možné vytvárať uzly a následne spoje medzi nimi. Vytvorenie uzlu sa prevádza príkazom **set n0 [\$ns node]**, kde premenná **n0** je odkaz na vytvorenú inštanciu uzla. Vytvorenie pevného spoja medzi uzlami **n0** a **n1** sa prevádza: **\$ns duplex-link \$n0 \$n1 <šírka pásma> <oneskorenie prenosu> <typ fronty>**. Najbežnejším typom fronty je DROPTAIL. Smerovanie paketov prebieha na základe topológie, teda závisí na tom ako sú jednotlivé uzly medzi sebou prepojené, a podľa toho sú aj ich smerovacie tabuľky vytvárané. Pri bezdrôtových uzloch toto nie je možné, a tak je uplatnený mechanizmus bezdrôtových smerovacích protokolov (DSDV, DSR, AODV, TORA). Pri simulácii hybridných sietí, teda drôtových a bezdrôtových vzájomne prepojených je nutné vykonať niekoľko základných krokov. Najdôležitejšie je použitie hierarchického smerovania, nadefinovanie jeho parametrov a topológie siete.

\$ns node-config -addressType hierarchical

```

AddrParams set domain_num 2
lappend cluster_num 2 1
AddrParams set cluster_num $cluster_num
lappend eilastlevel 1 1 3
AddrParams set nodes_num $eilastlevel

```

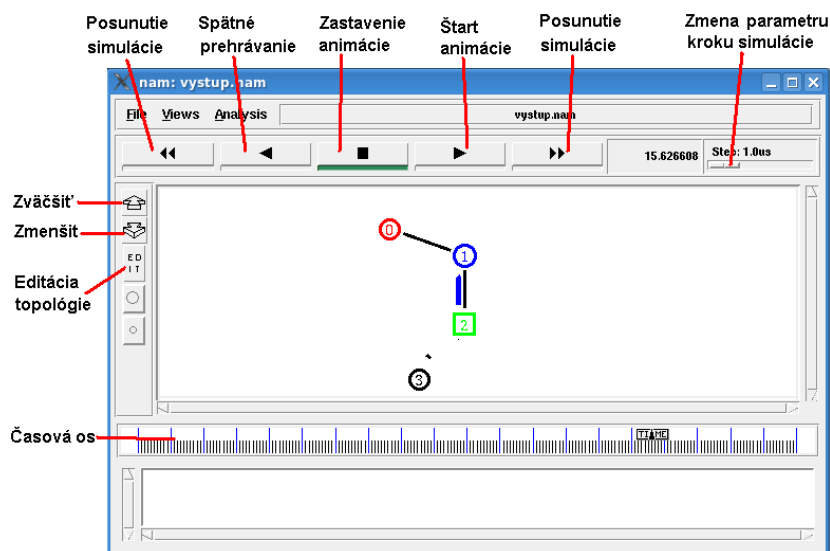
Prvý riadok predchádzajúceho kódu definuje, že jednotlivé uzly budú využívať hierarchického smerovania. V nasledujúcich riadkoch sa definuje hierarchia topológie. Zo zdrojového kódu vyplýva, že simulácia bude obsahovať 2 domény, a to jedna pevná, teda drôtová a jedna bezdrôtová, obsahujúca mobilné uzly a samotnú základňovú stanicu. Počet clusterov v doménach je definovaný ako „2 1“, čo znamená, že prvá doména v našom prípade pevná drôtová bude obsahovať 2 clusterly a druhá v poradí, teda bezdrôtová bude obsahovať 1 cluster. Ďalší riadok definuje počet uzlov v danom clusteri „1 1 3“. Znamená to, že v poradí prvý cluster bude obsahovať 1 uzol, druhý takisto 1 uzol a nasledujúci teda tretí cluster v poradí bude obsahovať 3 uzly, z čoho vyplýva, že hierarchia topológie je definovaná do troch úrovní.

Pre simuláciu hybridných sietí je teda nutné povoliť hierarchické smerovanie, ďalej vytvoriť separátne domény pre pevné a mobilné uzly, ktorých môže byť viacero pre simulácie viacnásobne prepojených sietí a na koniec je nutné priradiť ku každej bezdrôtovej doméne maximálne 1 základňovú stanicu, cez ktorú budú komunikovať jednotlivé uzly s uzlami ostatných domén. Všetkým uzlom taktiež musí byť priradená hierarchická adresa. Konfigurácia základňovej stanice a mobilných uzlov sa vykonáva pomocou konfigurácie API kde sa presne nadefinujú všetky parametre ako napríklad typ MAC vrstvy, smerovací protokol alebo typ kanálu. Pri vytváraní mobilných uzlov je taktiež dôležité vypnúť drôtový smerovací mechanizmus, ktorý používa **BS(0)** nastavením príkazu **\$ns node-config –wiredRouting OFF**. Všetky ostatné nastavenia pre bezdrôtové uzly ostávajú rovnaké ako pre základňovú stanicu. Ďalej je nutné všetky bezdrôtové uzly priradiť k danej základňovej stanici, keďže táto zohráva úlohu sieťovej brány medzi pevnou a mobilnou sieťou a prebieha ňou prakticky celá komunikácia.

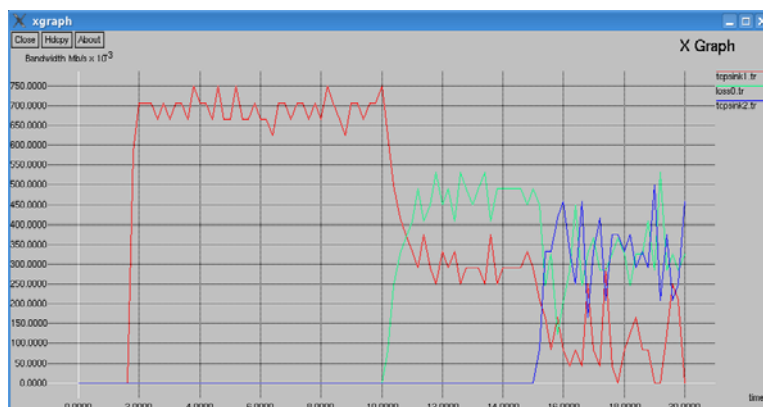
Po nadefinovaní celej topológie nasleduje definícia charakteru prenosu dát v sieti. Pre tento účel sa v NS2 využíva trieda „Agent“. Agenti predstavujú koncové body, kde sú jednotlivé pakety vytvárané, respektíve konzumované a sú využívané pri implementácii dátových protokolov na rôznych sieťových vrstvách. Príkladom definície dátového prenosu využívajúceho spoľahlivý transportný protokol (TCP) je **set tcp1 [new Agent/TCP]**, kde **tcp1** je odkaz na inštanciu tohto zdroja. Príkladom definície konzumenta takéhoto prenosu je **set sink1 [new Agent/TCPSink]**, reprezentovaného inštanciou **sink1**, ktorý je určený na jednoduché pohltienie dát s možnosťou ďalšej analýzy prijatých dát. Prepojenie vytvoreného zdroja a cieľa sa prevádza príkazom **\$ns connect \$tcp1 \$sink1**. V prípade nespoľahlivého transportného protokolu je situácia analogická, teda **set udp1[new Agent/UDP]**, **set udpsink1 [new Agent/Null]**, **\$ns connect \$udp1 \$udpsink1**, vytvorí zdroj, cieľ a vzájomne ich prepojí. Po definícii spôsobu prenosu je nutné nadefinovať samotný generátor dát (paketov). Príkladom definície takéhoto generátora je **set ftp1 [new Application/FTP]**, ktoré vytvorí generátor modelujúci prenos dát pomocou protokolu FTP. Príkazom **\$ftp1 attach-agent \$tcp1** naviažeme vytvorený generátor na vyššie definovaný transportný protokol **tcp1**. Takýchto generátorov NS2 podporuje veľké množstvo, pričom postup ich vytvorenia a prepojenia je analogický. Taktiež je možné ovplyvňovať ich jednotlivé parametre.

Spúšťanie jednotlivých generátorov a rôznych udalostí simulácie sa prevádza pomocou príkazu **at**. Napríklad **\$ns at 10.0 "\$ftp1 start"**, čo spustí generátor **ftp1** v čase 10 sekúnd od začiatku simulácie. V úvodzovkách je uvedený príkaz ktorý sa má vykonať v zadanom čase.

Vizualizácia celej simulácie prebieha nezávisle na simulačnom nástroji NS2. Pre grafické zobrazenie jednotlivých udalostí akými sú pohyby uzlov, generovanie dát, udalosti spojené s paketmi ako vznik, strata, zahodenie, spracovanie, udalosti spojené s frontami atd. sa využíva program implementovaný priamo v distribúcii NS2 a to **nam** (Network animator) Obr. 1. Pre grafické zobrazenie jednotlivých priebehov simulácie ktoré sú generované do ASCII súboru je najčastejšie používaný **xgraph** Obr. 2.



Obr. B.1: Vizualizácia simulácie pomocou nam

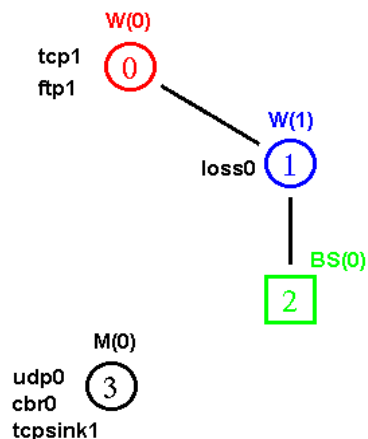


Obr. B.2: Časové priebehy zobrazené pomocou programu xgraph

B.3 Praktická časť

Táto časť sa skladá z overenia vybraných vlastností NS2 na dopredu pripravenom modeli siete, ktorého presné umiestnenie určí vedúci cvičenia a na jeho čiastočnej modifikácii podľa pokynov uvedených v zadaní. Model siete je napísaný v jazyku TCL a je uložený v textovom súbore s príponou **.tcl**.

Otvorte súbor **wired-cum-wireless.tcl** a podrobne preštudujte všetky kroky vytvorenia simulácie. Jednotlivé príkazy sú dostatočne okomentované a tak by ich pochopenie malo byť triviálne. Topológia siete je zobrazená na Obr. 3



Obr. B.3: Topológia siete definovaná v súbore `wired-cum-wireless.tcl`

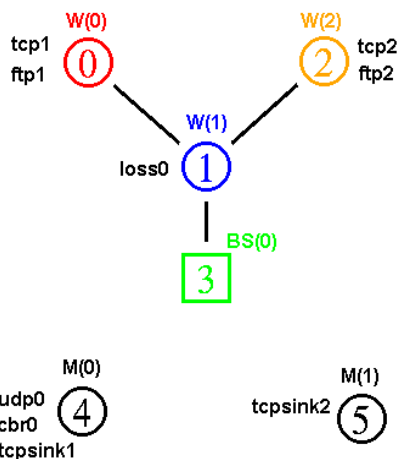
Tento model znázorňuje jednoduchú sieť obsahujúcu 4 uzly **W(0)**, **W(1)**, **BS(0)** a mobilný uzol **M(0)**, kde **BS(0)** a **M(0)** komunikujú pomocou štandardu 802.11. Tieto dva uzly sú prepojené bezdrôtovo pomocou smerovacieho protokolu **DSDV**. Tento protokol pracuje na princípe ukladania smerovacích tabuliek jednotlivých možných miest určenia, ktoré sú nasledujúcim skokom a obsahuje počet skokov na dosiahnutie cieľového miesta. Každý uzol periodicky obnovuje svoju smerovaciu tabuľku. Ďalej sú tu používané sekvenčné čísla na označovanie každej cesty. Cesta s vyšším sekvenčným číslom je viac uprednostňovaná. Ak tabuľka obsahuje dve cesty s rovnakým sekvenčným číslom, je uprednostnená tá, ktorá má menší počet skokov. Keď uzol zistí, že cesta k cieľu je prerušená, nastaví počet skokov na nekonečno. Existujú ďalšie smerovacie protokoly a to **DSR**, **AODV** a **TORA**.

K uzlu **W(0)** je priradený agent **tcp1**, ktorý je prepojený s objektom **tcpsink1** priradeným k uzlu **M(0)**. Objekt **tcpsink1** pohlcuje prichádzajúce pakety a odosiela potvrdzovacie pakety ACK. O generovanie prevádzky sa stará FTP generátor s názvom **ftp1** priradeného k agentovi **tcp1**. Dáta sú taktiež generované a prenášané od uzla **M(0)** smerom k uzlu **W(1)** a to pomocou agenta **udp1** ku ktorému je priradený objekt **cbr1** ktorý generuje dáta s konštantnou bitovou rýchlosťou, ktorú je možné nastaviť úpravou vhodných parametrov tohto objektu. Naopak k uzlu **W(1)** je priradený objekt **loss0**, ktorý zohráva úlohu konzumenta paketov generovaných pomocou **cbr1**. Prevádzka FTP je zahájená po 0 sekundách od začiatku simulácie a prevádzka CBR po 10 sekundách od začiatku simulácie.

Po preštudovaní modelu overte jeho funkčnosť v simulačnom prostredí **NS2**. Zmeňte pracovný adresár tak aby odpovedal adresáru, kde je uložený súbor **wired-cum-wireless.tcl**. Zadaním príkazu **ns wired-cum-wireless.tcl** do príkazového riadku spustíte simulátor, ktorý následne vygeneruje niekoľko súborov. V súbore **vystup.tr** sú uložené trasovacie údaje simulácie, v súbore **vystup.nam** sú uložené vygenerované údaje pre **nam**, ktoré slúžia na vizuálne prehliadanie simulácie a to zadaním príkazu **nam vystup.nam** do príkazového riadku. Sú vygenerované taktiež dva súbory pre grafické zobrazenie priebehov prenosu dát v sieti, ktoré sa prehliadajú programom **xgraph**, zadaním príkazu **xgraph graf1.tr**. Taktiež je mu možné predať viacero názvov súborov ktoré obsahujú dáta pre vykreslenie grafu, aby bolo možné vykresliť viacero priebehov do jedného grafu.

B.4 Samostatná práca

Po overení funkčnosti modelu preskúmajte vplyv zmeny jednotlivých parametrov, uvedených v skripte, na zmenu priebehov komunikácie.



Obr. B.4: Rozšírená topológia siete

V nasledujúcom kroku rozšírte model podľa Obr. 4, tak aby obsahoval 2 mobilné uzly a 4 drôtovo prepojené vrátane **BS(0)**. Pričom nový uzol **M(1)** bude slúžiť ako konzument paketov generovaných uzlom **W(2)**, ku ktorému bude priradený agent **tcp2**, a bude generovať pakety pomocou objektu **ftp2**. Taktiež tomuto novému mobilnému uzlu priradíte ľubovoľne vybranú prvotnú pozíciu a taktiež následný pohyb v topológii. Pokúste sa vygenerovať pomocou procedúry **record** graf k tomuto priebehu prenosu dát. Procedúra **record** používa dva parametre a to ako prvý názov konzumenta a ako druhý pramater odkaz na súbor kam sa budú ukladať jednotlivé dáta. Je dôležité aby ste tento súbor vytvorili na začiatku simulácie, ktorý sa vytvára rovnako ako ostatné súbory, ktoré sú popísané v skripte. Taktiež je nutné tento súbor po ukončení simulácie validne ukončiť a to pridaním vhodných údajov do procedúry **stop**. Overte funkčnosť rozšíreného skriptu. Sledujte ako ovplyvní pridanie nových uzlov celkovú prenosovú rýchlosť v sieti.

B.5 Zhrnutie

Je možné skontrolovať správnosť merania pomocou výsledných skriptov, ktorých presné umiestnenie určí vedúci cvičenia. Cieľom tohto merania bolo priblížiť študentom niektoré z mnoho možností ktoré poskytuje NS2 pre simulácie bezdrôtových sietí. Bol ukázaný spôsob akým je možné vytvoriť hybridné, teda drôtovo a bezdrôtovo prepojené siete. Ďalej sme si vyskúšali vytvorenie nových sieťových komponent a vytvorenie dátových prenosov medzi nimi.

B.6 Použitá literatúra

- [1] Information Sciences Institute. "The Network Simulator - ns-2". [online]. Dostupné z <<http://www.isi.edu/nsnam/ns/>>.
- [2] ALTMAN, Eitan and JIMÉNEZ, Tania: *NS Simulator for beginners*. Mérida: Univ. de Los Andes, 2003. Dostupné z <<http://evanjones.ca/ns2/>>
- [3] FALL Kevin and VARADHAN, Kannan. The ns Manual : formerly ns Notes and Documentation. [online]. Dostupné z <<http://www.isi.edu>>.
- [4] GREIS Marc. Tutorial for the Network Simulator. [online] posledná aktualizácia 3.7. 2002. Dostupné z <<http://www.isi.edu/nsnam/ns/tutorial/index.html>>.