

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra řídicí techniky

## Řízení topologie bezdrátových adhoc sítí s využitím bezpilotních prostředků

**Pavel Elis**

Vedoucí: Ing. Milan Rollo Ph.D.  
Obor: Kybernetika a Robotika  
Studijní program: Systémy a řízení  
Květen 2016



## Poděkování

Rád bych poděkoval svému vedoucímu práce panu Ing. Milanu Rollovi, Ph.D. za cenné připomínky a rady při vypracování této práce.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 27. května 2016

## Abstrakt

Cílem této práce je seznámit se s problematikou mobilních ad hoc sítí (MANET), s problematikou řízení jejich topologie a s frameworky, které se pro simulaci komunikace v takových sítích používají. Na to následně navázat navržením modulu pro simulaci takovéto komunikace s ohledem na využití bezpilotních prostředků. Navržený modul poté implementovat a experimentálně otestovat.

**Klíčová slova:** MANET, řízení topologie, síťové simulátory, OmNet++

**Vedoucí:** Ing. Milan Rollo Ph.D.

## Abstract

The aim of this bachelor thesis is to study mobile ad hoc networks (MANET), simulation frameworks for MANET and topology control. Furthermore, design and implement a new simulation modul for MANET with using of UAVs. In the end create an experiment to test the modul and compare the results.

**Keywords:** MANET, topology control, network simulator, OmNet++

**Title translation:** Mobile ad-hoc network topology control using UAVs



## Obrázky

2.1 Vrstvy ISO/OSI modelu . . . . .	7
2.2 Rozdělení řízení topologie . . . . .	10
3.1 Hierarchie modulů v OmNet++ . . . . .	16
4.1 Architektura aplikace . . . . .	22
4.2 Architektura uzlu . . . . .	23
4.3 Struktura sítě . . . . .	25
4.4 Struktura uzlu MyWirelessNode . . . . .	26
5.1 Experiment č. 1 - Počáteční pozice uzlů . . . . .	31
5.2 Experiment č. 1 - Průběh simulace . . . . .	32
5.3 Experiment č. 1 - Graf závislosti poměru ztracených zpráv na intervalu dotazování . . . . .	33
5.4 Experiment č. 2 - Počáteční pozice uzlů . . . . .	34
5.5 Experiment č. 2 - Průběh simulace . . . . .	35
5.6 Experiment č. 2 - Graf závislosti poměru ztracených zpráv na intervalu dotazování . . . . .	36

## Tabulky

5.1 Experiment č. 1 - bez řízeného uzlu . . . . .	32
5.2 Experiment č. 1 - Host1 . . . . .	33
5.3 Experiment č. 1 - Host2 . . . . .	33
5.4 Experiment č. 2 - bez řízeného uzlu . . . . .	34
5.5 Experiment č. 2 - Počet doručených zpráv . . . . .	35
5.6 Experiment č. 2 - Poměr ztracených zpráv . . . . .	35

# Kapitola 1

## Úvod

V dnešní době se čím dál tím masivněji využívají bezdrátové sítě. Bezdrátové vysílače můžeme nalézt téměř ve veškeré elektronice, která nás obklopuje, jejich význam tedy stále roste. Velmi rychle se rozvíjející i obor bezdrátových sítí bez pevné infrastruktury - mobilních ad hoc sítí (MANETs).

Pokud uvažujeme MANET síť tvořenou uzly, které nejsou kontrolovány člověkem, nabízí se otázka, jak je co možná nejefektivněji využít. Řešením by mohlo být zapojení uzlů, které si uvědomují umístění ostatních a s touto informací pracují. Například jak co nejrychleji sebrat data od skupiny uzlů, které se pohybují po předem známých trasách na určeném území. Nebo se nabízí otázka, jak uzly umístit tak, aby byly schopny navázat spojení a pokrýt přitom co největší plochu. Další možný případ je komunikace při mapování nesusnadno dostupného prostředí.

Při vývoji takových aplikací je nutné tyto aplikace testovat. Zkoušení na reálných sítích je velmi finančně nákladné a náročné na realizaci. Proto je vhodné využít síťových simulací.

Cílem této práce je nalézt takový síťový simulátor, ve kterém by bylo možné vytvořit simulační model právě pro takovéto sítě. Model poté otestovat a experimentálně ověřit.

Tato práce je členěna na teoretickou a praktickou část, celkově na 6 kapitol. V kapitole 2 je úvod do problematiky bezdrátových sítí a sítí obecně. Kapitola 3 je věnována porovnání existujících síťových simulátorů a výběru simulátoru, který pro návrh aplikaci využijeme. V kapitole 4 je uveden návrh simulačního modelu a také jeho implementace. Experiment si navrženou aplikací je popsán v kapitole 5 a v kapitole 6 je poté celá práce shrnutá.







# Část I

## Teoretická část



## Kapitola 2

### Bezdrátové sítě

#### 2.1 Úvod

Bezdrátové sítě jsou takové sítě, ve kterých není přenos signálů zajišťován drátovým spojením. Drátovým spojením je v tomto případě myšleno spojení jak po metalickém kabelu, tak optickými kabely. Mezi odesílatelem a příjemcem tedy neexistuje žádné fyzické spojení, není mezi nimi natažen žádný drát. Bezdrátový přenos je možné vytvořit v mnoha různých prostředích včetně vody, vzduchu nebo vakua. Bezdrátové sítě jsou často topologicky flexibilní a snadno měnitelné (jednotlivé uzly mohou být pohyblivé), díky čemuž jejich celosvětové využití stále roste. Denně se s nimi setkáváme v mobilních telefonech, přenosných počítačích, tabletech a v mnohých dalších zařízeních.

Absence fyzického propojení však přináší jisté nevýhody. Přenášený signál je více náchylný k vnějšímu rušení a šumu. Výsledkem je méně kvalitní a méně stabilní a hůře předvídatelný přijímaný signál. Kapacita kanálu je taky výrazně menší.

Aby bylo možné bezdrátové spojení mezi vysílačem  $x$  a přijímačem  $y$  vůbec možné navázat, musí být výkon signálu přijímaného uzlem  $y$  vyšší, než je mez citlivosti. Velikost meze citlivosti je individuální pro každý přijímač a také závisí na rychlosti komunikace. Výkon přijatého signálu  $P_p$  závisí na vysílacím výkonu  $P_v$  a na tom, jak velký je útlum prostředí, jak rychle ztrácí signál při prostupu prostředím svůj výkon. Za předpokladu, že útlum prostředí mezi uzly  $x$  a  $y$  je  $UP(x, y)$  můžeme vyjádřit výkon přijatého signálu pomocí rovnice 2.1.

$$P_p = \frac{P_v}{UP(x, y)} \quad (2.1)$$

Bezdrátový přenos je také, oproti klasickému po kabelu, obtížnější zabezpečit. Pro odposlech stačí použít dostupný přijímač, zatímco u metalické přenosové cesty je třeba fyzického propojení.

## 2.2 Mobilní Ad hoc sítě (MANET)

Mobilní ad hoc sítě jsou takové bezdrátové počítačové sítě, jejichž součástí je alespoň jeden uzel, který není statický tj. může se pohybovat. Díky možnému přesouvání uzlů dochází u těchto sítí k časté změně parametrů sítě. S měnící se vzdáleností se mění i minimální potřebný vysílací výkon, to je minimální výkon, který je potřeba k přenosu signálu. Spojení mezi uzly mohou tedy často vznikat i zanikat. Stejně tak má vzdálenost vliv na zpoždění, které tedy není konstantní. Dalším problémem je napájení uzlů, které je kvůli jejich mobilitě často řešeno přes baterie. Schopnost energetických úspor je tedy u tohoto druhu sítí důležitým kritériem.

Tento typ sítí má tedy vysoce dynamickou topologii a vysoký počet změn může být špatně předvídatelný. Uzly jsou většinou schopny přeposílat informaci, která pro ně není určena, na své sousedy, mohou tedy plnit také funkci routeru.

## 2.3 ISO/OSI model

Referenční model ISO/OSI (často nazýván jen OSI model) je pokus o standardizaci počítačových sítí od společnosti ISO z roku 1984. Tento model slouží k tomu, aby zařízení vytvořena za jiným účelem spolu mohla komunikovat. Je vytvořen z hierarchických vrstev, které spolu komunikují a jsou na sobě nezávislé. Model nepředepisuje konkrétní realizaci systému, ale pouze obecné principy vrstev a typy jejich služeb. [10]

To, z kterých vrstev se model skládá, můžete vidět na obrázku 2.1. Každá z vrstev využívá pro svoje služby služby vrstvy pod sebou a svoje služby poskytuje vrstvě vyšší.

### 2.3.1 Fyzická vrstva

Fyzická vrstva zajišťuje fyzickou komunikaci, tedy přenos jednotlivých bitů. Komunikaci nijak neinterpretuje, nepřikládá bitům žádný specifický význam. Jediné služby, které vyšším vrstvám nabízí, jsou služby zajišťující přijetí a odeslání jednoho bitu.

Na fyzické vrstvě můžeme rozlišovat sériový a paralelní přenos, podle časování můžeme rozlišit synchronní, asynchronní a arytmičtý přenos. Přenos může probíhat různými způsoby, například po sériové dvojince, podle standardu IEEE 802.11 (Wi-Fi), podle standardu 802.3 (Ethernet), po optických kabelech a mnohé další.



Obrázek 2.1: Vrstvy ISO/OSI modelu

### ■ 2.3.2 Linková vrstva

Linková vrstva, někdy nazývána také jako Spojová, spojuje data z Fyzické vrstvy do logických celků, rámců. Přenášené rámce seřazuje, nastavuje parametry linky, zajišťuje spolehlivost (detekuje chyby) a přidává rámcům fyzickou adresu. Poskytuje také synchronizaci pro Fyzickou vrstvu, brání tomu, aby vysílající nezahltil příjemce.

### ■ 2.3.3 Síťová vrstva

Tato vrstva přenáší větší bloky dat, označované jako pakety. Zajišťuje doručení paketu až ke konečnému adresátovi. Pokud neexistuje přímá cesta, hledá vhodnou cestu až k cíli, provádí tedy směrování (routing). Musí si tedy uvědomovat topologii sítě. Nejrozšířenější implementací síťového protokolu je protokol IP (Internet Protocol) verze 4 a nebo verze 6.

### ■ 2.3.4 Transportní vrstva

Úkolem Transportní vrstvy je zajistit takovou kvalitu přenosu, jakou vyžadují vyšší vrstvy. Jelikož nižší vrstvy, nemusí být na všech zařízeních stejné, tato vrstva musí provést příslušné přizpůsobení. Vrstva podporuje několik Transportních protokolů.

## ■ TCP

TCP (Transmission Control Protocol) je nejpoužívanější protokol transportní vrstvy. Tento protokol garantuje spolehlivé doručování, a to ve správném pořadí. Při nedoručení nebo poškození paketu dochází k jeho opakovanému zaslání.

## ■ UDP

UDP (User Datagram Protocol) je, stejně jako TCP, protokol Transportní vrstvy. Na rozdíl od něj ale nemusí být spolehlivý. Při nedoručení paketu nebo při jeho poškození, se odeslání neopakuje a odesílání pokračuje dál. Výhodou tohoto přístupu je jednoduchost a také rychlost komunikace. To jsou například online hry, a nebo přenosy videa a audia.

## ■ SCTP

SCTP (Stream Control Transmission Protocol) je protokol pro navázání několika komunikačních kanálů (stream) ve stejném okamžiku. Tyto kanály jsou na sobě nezávislé. Při navázání spojení jsou tedy paralelně přenášeny proudy (stream) dat a u každého z nich je možné zaručit jejich doručení, a to ve správném pořadí. Případný výpadek jednoho proudu se nijak netýká ostatních.

### ■ 2.3.5 Relační vrstva

Zajišťuje sestavení a řízení relací mezi relačními vrstvami dvou systémů. Může zajišťovat šifrování a kompresi dat nebo také synchronizaci z více datových toků.

### ■ 2.3.6 Prezentační vrstva

Zatímco nižší vrstvy se snaží doručit každý bit tak, jak byl odeslán, prezentační vrstva se věnuje situaci, kdy přenesený bit nemá pro příjemce stejný význam jako pro odesílatele. To se může stát například kvůli jinému kódování, jinému formátu čísel, kvůli formátu struktur a polí atd. Prezentační vrstva zajišťuje potřebné konverze.

### ■ 2.3.7 Aplikační vrstva

V době, kdy OSI model vznikl, panovala představa, že aplikační vrstva bude obsahovat všechny možné aplikace. Z důvodu velkého množství aplikací bylo od této představy upuštěno a aplikační vrstva nyní obsahuje jen části aplikací, které má smysl standardizovat. Jedná se například o mechanismy elektronické pošty.

## ■ 2.4 Řízení topologie

Posílání informace mezi dvěma uzly v síti probíhá na základě síťové topologie, struktury z uzlů, kterými je přenos zajišťován. O řízení topologie můžeme mluvit, pokud takových cest, které spojují dva uzly je více, a nebo také, pokud jsou jednotlivé uzly pohyblivé. Úkolem řízení topologie je zajistit, aby spojení bylo možné vytvořit (všechny uzly v síti byly propojitelné) a přitom optimalizovat některou vlastnost, například minimalizovat spotřebu energie.

Řízení topologie můžeme také dělit podle schopnosti uzlů sítě se pohybovat. Stacionární síť, kde všechny uzly sítě zůstávají nehybně na svém místě a mobilní síť, kde se alespoň jeden uzel může pohybovat.

Na řízení topologie [3] lze nahlížet několika způsoby, jak je ukázáno na obrázku 2.2. V případě homogenního řízení předpokládáme, že všechny uzly používají stejnou technologii a mají tedy stejný vysílací výkon. V tomto případě se celý problém redukuje na hledání minimální hodnoty parametru  $r$ , kde  $r$  je kritická přenosová vzdálenost (critical transmitting range - CTR). Pokud mluvíme o nehomogenním řízení, máme na mysli použití uzlů, u kterých lze vysílací výkon různě nastavit.

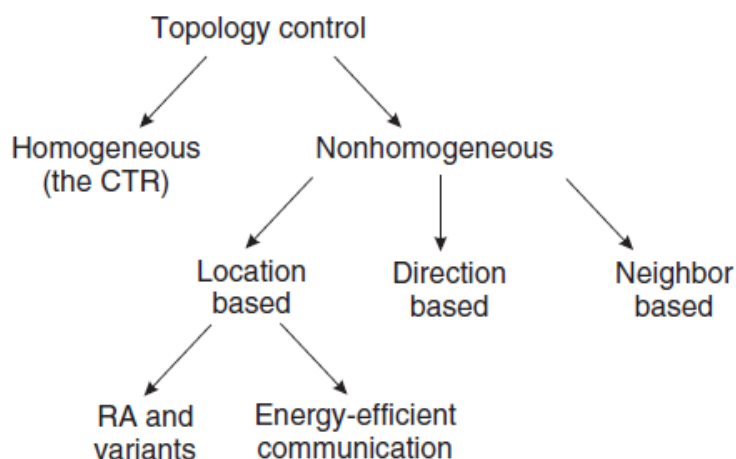
Nehomogenní řízení lze rozdělit podle tří rozdílných přístupů. V případě řízení podle pozice (location-based control) považujeme pozici všech uzlů za známou. Jednotlivé uzly jsou proto často vybaveny GPS přijímačem. V případě řízení podle směru (direction-based control) předpokládáme, že pozice uzlů není známá, uzly si jsou však vědomy relativního směru k jejím sousedům. V posledním případě, řízení podle sousedů (neighbor-based control) předpokládáme, že si je uzel vědom některých základních dat o svých sousedech, jako je třeba jejich identifikátor a je schopen sousedy seřadit podle určitých kritérií. Těmi může být například vzdálenost nebo kvalita spojení. Toto řízení je nejčastěji použito u mobilních ad hoc sítí.

### ■ 2.4.1 Kritická přenosová vzdálenost

Jak již bylo řečeno, kritická přenosová vzdálenost je využita v homogenním řízení topologie, kde mají všechny uzly stejné vysílací parametry. Jedná se o takovou vysílací vzdálenost, při které je možné všechny uzly sítě propojit. Při znalosti rozmístění uzlů se tato úloha zjednodušuje a kritická přenosová vzdálenost odpovídá délce nejdelší hrany sítě. Pokud rozmístění uzlů známe není, tak se často předpokládá, že jsou uzly v síti rozmístěny náhodně.

### ■ 2.4.2 Řízení podle pozice (Location-based)

Při použití tohoto typu řízení je nutná velmi přesná znalost polohy všech uzlů. Pro tyto účely jsou uzly nejčastěji vybaveny GPS přijímačem, je ale také možné použít jiné algoritmy pro určení polohy. Lze použít i alternativní



**Obrázek 2.2:** Rozdělení řízení topologie

přístup, kde svou přesnou polohu znají jen některé uzly a ostatní jsou určeny podle vzdálenosti k uzlu.

### ■ 2.4.3 Řízení podle směru (Direction-based)

Tento typ řízení spoléhá na schopnost jednotlivých uzlů odhadnout směr k jejich sousedům. Z tohoto důvodu jsou uzly často vybaveny přídatnou směrovou anténou. Hlavní výhodou tohoto přístupu je možnost použití v situacích, kdy použití GPS není možné (například uzavřené prostory, budovy). Nevýhodou je, že získaná informace o směru není při zjišťování přesné polohy uzly stejně kvalitní, jako informace o přesné poloze při řízení podle pozice.

### ■ 2.4.4 Řízení podle souseda (Neighbor-based)

Tento typ řízení závisí na schopnosti uzlu zjistit počet sousedů, identifikovat je a zjistit jejich maximální vysílací vzdálenost, což je vzdálenost, kterou jsou schopny vysílat signál u maximálního vysílacího výkonu. Pokud toho uzel schopen není, je nutné nastavit vysílací výkon náhodně.



## Kapitola 3

# Simulace komunikace v bezdrátových sítích

### 3.1 Úvod

Jak již bylo řečeno, mobilní ad hoc sítě (MANET) jsou sítě uzlů, mezi kterými je vždy alespoň jeden mobilní. Jako uzly jsou nejčastěji použity notebooky, mobilní telefony, tablety a podobná zařízení, která spolu komunikují pomocí Bluetooth a WiFi. Kvůli značné ceně těchto prvků a jejich malé flexibilitě v oblasti nastavení parametrů bezdrátových sítí, je vhodné s takovými sítěmi experimentovat pomocí simulací.

Tyto experimenty lze provádět dvojným způsobem, pomocí softwarových simulátorů, anebo pomocí experimentálních sítí. I když je většina pokusů prováděna v reálné velikosti na experimentálních sítích, softwarové simulátory jsou jejich levnější a flexibilnější alternativa.

#### 3.1.1 Experimentální síť

Jedná se o laboratorní síť navržené a využívané kvůli výzkumu. Jejich hlavní nevýhodou je vysoká cena uzlů a složitější pozorování stavu jednotlivých uzlů a ovládání sítě. V praxi se nepoužívají experimentální sítě o více než 50 uzlech.

#### 3.1.2 Softwarové simulátory

Simulační programy tyto problémy nemají. Síť lze ovládat jako celek, její pozorování je jednodušší a její velikost je pouze otázkou výpočetního výkonu počítače. Nasimulování mobilní ad hoc sítě je ale velmi složitý úkol. Je třeba si uvědomit, že simulace nikdy nebude dokonale přesná. Je ale možné ji vytvořit velmi realistickou.

Jedním ze znaků, které u simulátorů můžeme porovnávat je detailnost simulace. Nikdy není možné nasimulovat reálný svět až do úplných detailů, detailnost modelu by měla vždy odpovídat potřebám simulované aplikace. Další znak, který můžeme u simulátorů rozlišit, je model pohyblivosti uzlů. Kromě náhodné pohyblivosti lze například použít model založený na pohybu



Jedná se o diskretní simulátor, ve kterém je síťová vrstva OSI modelu reprezentována pomocí objektů nazvaných entity. Události jsou vytvořeny jako zprávy s časovou značkou, které jsou vykonávané právě entitami. V tomto simulátoru není simulován každý uzel jednotlivě (kvůli jejich velkému počtu), místo toho jsou pomocí entit simulovány celé vrstvy OSI modelu sítě. Zprávy, které představují události v síti jsou mezi entitami předávány tak, jako by byly předávány mezi jednotlivými vrstvami. Díky tomu může GloMoSim simulovat síť s velkým počtem uzlů (až desítky tisíc).

Stejně jako v ns-2 simulátoru lze dosáhnout reálných simulací pomocí modelu překážek v prostředí. Existuje také dostupný vizualizační model v Javě. I když je to dobrý simulátor pro MANET sítě, hlavním nedostatkem GloMoSimu je špatná dokumentace. Další z velkých nevýhod je aktualizace systému, který vychází velmi nepravidelně.

### ■ OPNet

OPNet je diskretní simulátor napsaný v C++. Jedná se pravděpodobně o nejrozšířenější komerční simulátor. Obsahuje interaktivní vývojové prostředí, ve kterém lze pozorovat a simulovat aplikace, jednotlivé uzly, ale i celé sítě.

### ■ OMNet++

OMNet++ je stejně jako OMNet napsán v C++. V základu se jedná o simulátor schopný simulovat jakékoli prostředí se zařízeními, které spolu interagují. Při využití rozšíření o mobilitu uzlů se jedná o velmi dobrý MANET simulátor. Je open-source, často využívaný pro výukové a akademické účely.

### ■ NAB

NAB je diskretní simulátor přímo určený pro simulaci MANET sítí. Vznikl kvůli neschopnosti simulace velkých ad hoc sítí pomocí ostatních simulátorů. Je objektově orientován, jednotlivými objekty jsou uzly sítě. Jedná se o open-source napsaný v jazyce OCaml.

### ■ J-Sim

J-sim, někdy také označován jako JavaSim, je původně simulátor navržený pro klasické matalické sítě. Obsahuje rozšíření pro bezdrátové sítě, které implementuje IEEE 802.11 MAC. Jedná se prozatím o jediný simulátor, který toto podporuje. Kromě toho obsahuje J-sim i několik modelů radiových sítí a modelů pohyblivosti. Díky tomu je J-sim vhodný simulátor pro MANET sítě. Je napsán v Javě a jedná se o open-source.



pro akademické účely je bezplatné. OmNet++ je využitelný při simulaci de facto libovolných sítí a je pravidelně aktualizovaný. Tento simulátor je podrobně popsán v následující kapitole.

## ■ 3.2 Podrobný popis OmNet++

OmNet++ ([4]) je simulační knihovna a framework určený k tvorbě síťových simulací a to nejen drátových a bezdrátových sítí, ale i sítí na čipech, sítě složených z front propojených směrovacími protokoly a tak dále. OmNet++ poskytuje podporu například pro sensorové sítě, bezdrátové ad hoc sítě a internetové protokoly. Uživatelské rozhraní je založené na vývojovém prostředí Eclipse. S ním je možná grafická tvorba topologie sítě a jednoduché nastavení průběhu simulace. Existuje celá řada rozšíření simulátoru. Za zmínku jistě stojí rozšíření pro simulace v reálném čase, integrace SystemC (soubor knihoven a maker v C++ zajišťující rozhraní událostmi řízeného simulátoru), INET, o kterém se podrobněji zmiňuji v další kapitole a nebo propojení s databázemi.

Jelikož se jedná o událostmi řízený simulátor, veškeré plánované změny simulovaného systému jsou ukládány do seznamu událostí. Každá přidaná událost má svou časovou značku (timestamp), podle které jsou poté zpracovávány. Jak již bylo řečeno, OmNet++ je napsán v C++, pro popis struktury jednotlivých modulů, základních prvků systému, se používá jazyk NED (Network Description).

Pro simulaci tohoto projektu byla použita nejnovější verze simulátoru OmNet++ 5.0. Tato verze vyšla 15.4.2016 téměř 2 roky po verzi 4. Obsahuje několik vylepšení, nejvýraznější rozdíly jsou ve 2D a 3D zobrazení.

### ■ 3.2.1 Moduly

Simulovaný systém se v simulátoru OmNet++ skládá z modulů, které spolu komunikují pomocí bran. Nejjednodušší datová struktura se nazývá *Simple module*. Z nich jsou pak složeny složitější moduly *Compound module*. Tyto moduly ze sebe potom dědí, jedná se vlastně o objektově orientovaný program v C++.

Veškeré moduly je možné spojovat, a nebo do sebe vnořovat, hloubka zanoření není v tomto simulátoru omezená. Nejvyšší úroveň hierarchie se nazývá *System module*. Možná struktura modulů je znázorněna na obrázku 3.1.

### ■ 3.2.2 Brány a linky

Komunikace mezi moduly je zajištěna posíláním zpráv, jednoduchých, ale i složitějších datových struktur. Zprávy jsou posílány přes brány, které jsou mezi sebou propojeny linkami. Brány mohou být trojího typu: vstupní brány (input), výstupní brány (output) a obousměrné brány (inout).



### ■ Komunikace po soketech

Tento způsob propojení vyžaduje, aby se v simulaci nacházel modul, který se chová jako proxy. Ten poté zajišťuje komunikaci s externí aplikací. Hlavní výhodou této metody je to, že není potřeba externí aplikaci měnit a implementace metody je velmi jednoduchá. Je vhodné ji využít v případě, že v aplikaci nepotřebujeme simulovat nižší vrstvy, ale pouze aplikační.

Při použití této metody musíme počítat s některými problémy. Simulace v OmNet++ běží obvykle tak rychle, jak jen může a používá k tomu veškerý možný čas procesoru. Externí aplikace tímto může být zatížena a nemusí na ni zbýt dostatek výpočetního času procesoru. Abychom se tomuto vyhnuli, je vhodné, aby procesy reálné aplikace s vysokou prioritou běžely na jiném procesoru.

Další problém, se kterým musíme počítat je rozdíl mezi simulačním a reálným časem. To, že se tyto časy liší, může vést k nepřesné a nebo dokonce chybné simulaci. Tento problém je možné řešit pomocí speciálního simulačního plánovače, který srovná čas simulace a čas externí aplikace. Problémem tohoto řešení je fakt, že při srovnání časů simulaci velmi zpomalíme. Simulace, která je schopna běžet i mnohonásobně rychleji nyní poběží maximálně rychlostí externího programu.

### ■ Přímé propojení zdrojového kódu

Přímé propojení se zdrojovým kódem je asi nejpřirozenější způsob vytvoření komunikace. Spočívá ve vytvoření modulu v simulátoru tak, aby se choval jako externí aplikace. Je tedy nutné ho napsat v C++ a přeložit simulátorem. Tento přístup se nejvíce hodí pro malé simulace. Jelikož je veškerý kód součástí OmNet++, odpadají problémy s různými časy.

Musíme ale počítat s tím, že při přenesení veškerého zdrojového kódu do OmNet++ musíme přenést i veškeré programy a části, se kterými aplikace pracuje, a na které odkazuje. Další věc, na kterou je potřeba brát zřetel je to, že v takto vloženém kódu není možné využívat možnosti klasického programu jako jsou časovače a vlákna. Nahrazení časovače je například možné pomocí *selfMessage*, zpráv, které modul zašle sám sobě.

### ■ Sdílené knihovny

Propojení pomocí sdílených knihoven spočívá, podobně jako předchozí možnost, v implementaci binárního kódu. Tím vzniká velká výhoda nemuset řešit překlad projektu v simulátoru. Tuto možnost je tedy nejlepší využít v případě, kdy integrace externího programu do OmNet++ zahrnuje jeho velké úpravy. Simulace v simulátoru má tedy jiný kompilátor než externí program. Změna výstupu externího programu na sdílenou knihovnou je jen malý zásah do aplikace a může být proveden pomocí flagů v příkazové řádce.







## Část II

### Praktická část



## Kapitola 4

### Architektura a implementace modelu

#### 4.1 INET framework

Při tvorbě tohoto projektu byl využit INET framework ([5]), open-source knihovna, která rozšiřuje základní knihovny OmNet++. Tato knihovna byla pro tento projekt vybrána z důvodů velkého množství modulů, možnosti využití a také kvůli kvalitní dokumentaci.

Přidává mnoho bezdrátových i drátových uzlů, třídu Mobility, jejíž implementace umožňuje přesné nastavení pohybu jednotlivých uzlů, protokoly linkové vrstvy OSI modelu (Ethernet, PPP, IEEE802.11, atd.), protokoly síťové vrstvy (IPv4, IPv6), transportní vrstvy (TCP, UDP), MANET protokoly a i několik aplikací.

##### 4.1.1 Historie INET

INET vznikl v roce 2003 jako projekt Andrase Vargy, který převzal a pokračoval v projektu IPSuite studentů univerzity v Karlsruhe. Varga modely zdokumentoval, upravil a některé i celé přepsal.

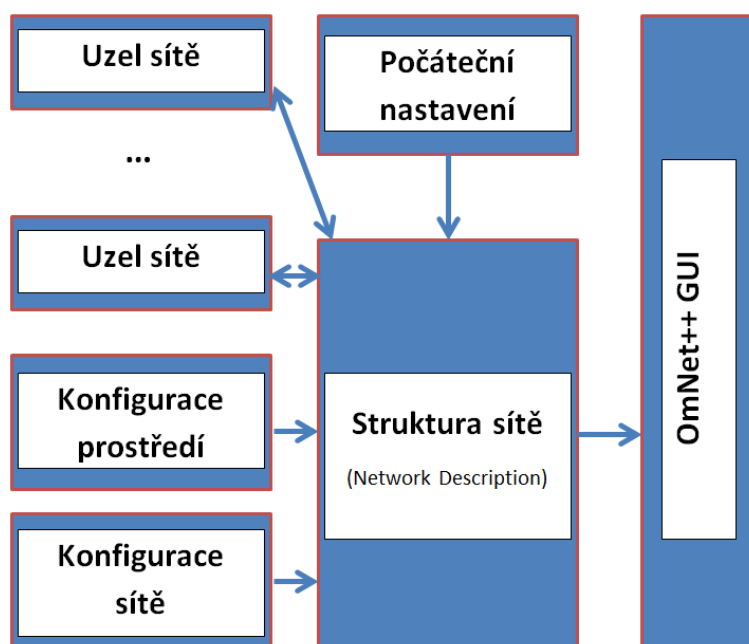
V následujícím roce framework INET doplnil o několik TCP implementací, o specifikaci Ethernet, IP a o několik aplikačních modelů. V současné době je INET spravován a doplňován studenty z celého světa, mnoho uživatelů pomáhá hledat chyby a podílí se na opravných balíčcích.

##### 4.1.2 Architektura INET

INET framework využívá stejné základní prvky jako jiné základní knihovny OmNet++. Komunikace mezi moduly je opět uskutečněna pomocí zpráv, které si přes brány a propojující linky posílají.

Struktura je taktéž založena na modulech. Ty zde mohou prezentovat síťová zařízení, rozhraní a protokoly v těchto zařízeních ale i komunikační aplikace. Komunikační aplikací je myšlena aplikace Aplikační vrstvy modelu OSI, která se stará o logiku komunikace. V této aplikaci vzniká zpráva a

následně postupuje do nižších vrstev až do Fyzické vrstvy, která zprávu odešle na jiné zařízení.



Obrázek 4.1: Architektura aplikace

## 4.2 Architektura aplikace

Vytvořená aplikace pro řízení topologie bezdrátových sítí obsahuje několik částí. Její struktura je znázorněna na obrázku č. 4.1.

Jak je z obrázku patrné, centrálním prvkem architektury aplikace je soubor ve formátu *Network Description*, který zajišťuje komunikaci mezi různými částmi sítě. Předává také aktuální stav sítě grafickému rozhraní simulátoru k vizualizaci. Tento soubor je povinný v každé aplikaci.

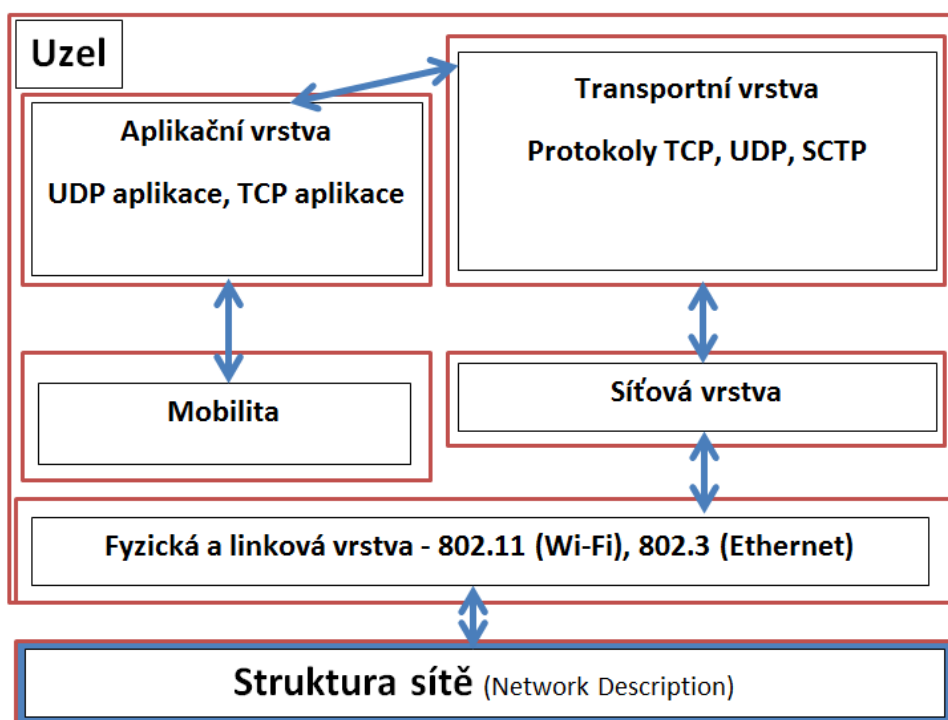
Dalším nezbytným prvkem je inicializační soubor ve formátu *.ini*. Pomocí něho jsou nastaveny vlastnosti a parametry všech modulů, ze kterých se síť skládá.

V síti může být umístěný různý počet libovolných uzlů. Jejich počet není omezen. Může se jednat o uzly drátové i bezdrátové, uzly se schopností se pohybovat i uzly statické. Uzly si předávají zprávy přes strukturu sítě, jsou přes ni nastaveny počátečními parametry, posílají zprávy o své poloze.

Další modulem, který je potřebný pouze pro síť s využitím bezdrátového přenosu je prvek, který zajišťuje konfiguraci prostředí, kterým se signál šíří. Předává strukturu sítě informace o vlastnostech signálu v průběhu prostupu

prostředím.

Posledním naznačeným prvkem je prvek konfigurace síťové vrstvy. Ten předá přes strukturu sítě nastavení uzlů pro komunikaci.



Obrázek 4.2: Architektura uzlu

## 4.3 Architektura uzlu sítě

Struktura uzlů, které jsou použité v mojí aplikaci je znázorněna na obrázku č. 4.2.

Uzel se skládá z několika částí, na které se dají podle své funkčnosti přiřadit jednotlivým vrstvám OSI modelu. Jak již bylo vidět v části 4.1.2, uzel je propojen se sítí simulovanou souborem NED. Přes toto rozhraní vykonává veškeré interakce s dalšími prvky sítě.

Komunikace přichází ze sítě do Fyzické vrstvy. ta může být v uzlu simulovaná mnoha způsoby. Při bezdrátové komunikaci použijeme modul reprezentující síťovou kartu, při komunikaci po drátové lince lze využít např. modul simulující ethernetové připojení.

Zpráva dále pokračuje do síťové vrstvy, která ji přiřadí příslušnému protokolu vrstvy Transportní. V implementaci mého vytvořeného uzlu pracuji s protokolem UDP, který využívá UDP aplikace. Funkčnost u ostatních proto-

kolů je obdobná. Aplikace zpracuje zprávu a může poté vykonat další kroky.

V mé implementaci uzlu je s UDP aplikací propojen prvek *Mobilita*, který zajišťuje pohyb uzlu. Tyto dva prvky spolu komunikují, aplikace je schopná zjistit svoji aktuální polohu.

## 4.4 Implementace aplikace

Vlastní navržený modul pro simulaci komunikace v MANET sítích je založen na modulech z INET frameworku. V nich ale chybí několik důležitých modulů, které jsou pro řízení topologie nezbytné. Neexistuje například modul řídicí pohyb uzlu rozšiřující třídu *IMobility*, jehož pohyb by byl nějakým způsobem závislý na okolních uzlech.

Tento modul bylo tedy nutné vytvořit. Na tento projekt se dá nahlížet jako na rozšíření knihovny INET implementující modul pro řízení pohybu, který je schopen pohyb uzlu ovládat pomocí informace o pozici uzlů ostatních. Tento modul, *MyMobility*, je umístěn mezi ostatní pohybové moduly knihovny INET.

Dále bylo potřeba vytvořit vlastní uzel vhodný pro použití v bezdrátových ad hoc sítích. Modul *MyWirelessNode* je založen na modulu z knihovny INET *WirelessNode*. Jedinou změnou je vytvoření linky mezi interface pro ovládání pohybu *Mobility* a interface pro komunikační aplikace pomocí UDP protokolu. Toto propojení je využito při zjišťování aktuální polohy těchto uzlů a nastavování pohybu při běhu simulace.

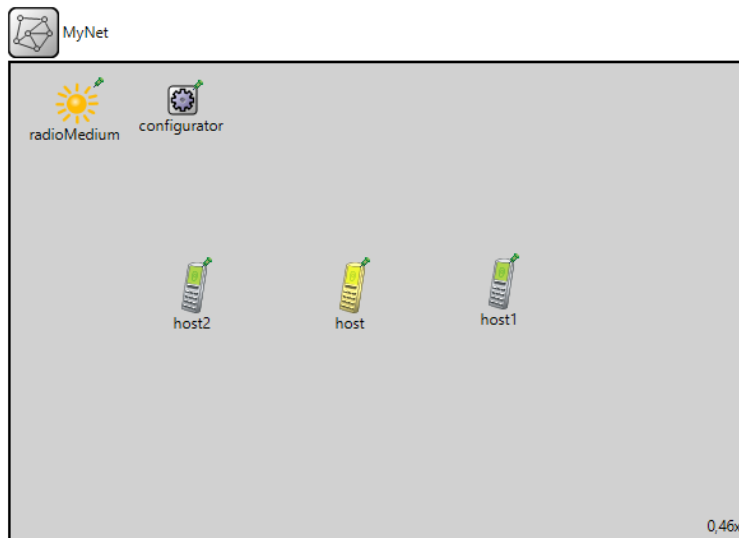
Nakonec byly vytvořeny komunikační aplikace, které zajišťují komunikaci nutnou pro řízení topologie. Tyto aplikace jsou dvě: *MyUDPApp* a *MyUDPAppControl*. První jmenovaná je určena pro uzly, nad jejichž pohybem nemá administrátor sítě kontrolu. Tato aplikace pracuje tak, že při dotazu na svoji polohu odpoví. Dále je schopná v parametrem nastaveném časovém intervalu odesílat data do druhého nekontrolovaného uzlu.

Druhá aplikace, *MyUDPAppControl*, je komplikovanější. Je určená pro uzly, jejichž pohyb je kontrolován. Má za úkol získat souřadnice polohy okolních uzlů, tyto informace zpracovat, určit souřadnice, na které se má kontrolovaný uzel přesunout.

Jméno všech implementovaných modulů má předponu *My* a to z důvodu, aby byly v knihovně INET snadno identifikovatelné.

### 4.4.1 MyNet

Tento NED soubor reprezentuje samotnou síť. Jeho součástí je možné vidět na obrázku 4.3. *MyNet* obsahuje tři uzly typu *MyWirelessHost*, modul pro síťovou konfiguraci a modul, který nastavuje okolní prostředí. Dva uzly



Obrázek 4.3: Struktura sítě

(klasické zbarvení) nejsou říditelné a jeden (zlatý) je. Síť má také několik parametrů nutných nastavit při startu a to velikosti pole, ve kterém se uzly mohou pohybovat.

#### 4.4.2 MyWirelessHost

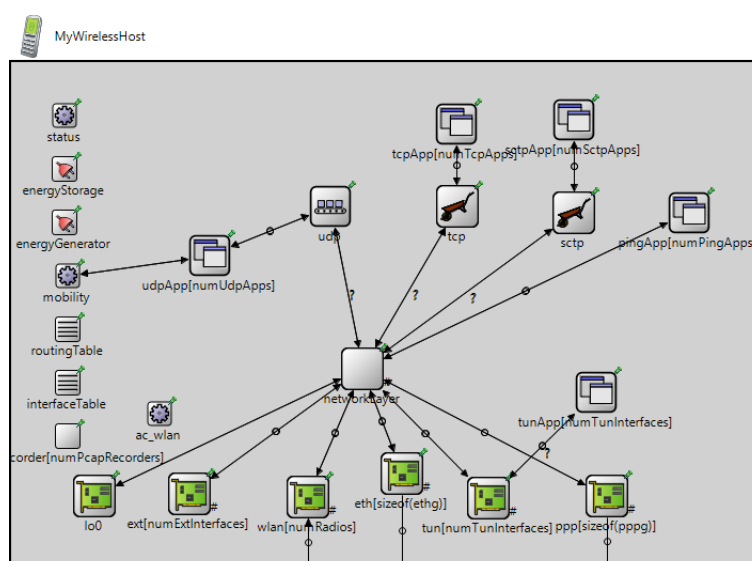
Tento modul vychází ze standardního uzlu knihovny INET *WirelessHost*, jehož strukturu můžete vidět na obrázku 4.4. Ten může reprezentovat libovolný bezdrátový uzel, který komunikuje pomocí TCP, UDP a nebo SCTP protokolů.

Tento modul může reprezentovat celou řadu reálných zařízení, a to jak řízených člověkem, tak bezpilotních. Může se jednat o bezdrátové zařízení schopné pohybu na zemi, ve vzduchu, ale i v jiných médiích. Jedná se tedy například o roboty, vozidla, letadla, a nebo drony vybavené schopností bezdrátově komunikovat.

Jak je patrné, tento modul se skládá hned z několika modulů. Ty, jenž jsou pro tuto aplikaci zásadní a při simulaci hojně použité budou podrobněji popsány, a to od nižších vrstev OSI modulu k vyšším:

**WLAN** Interface *WLAN* patří mezi prvky linkové vrstvy knihovny INET a reprezentuje interface ovladače bezdrátové sítě (Wireless Network Interface Controller - WNIC). Ten v tomto uzlu rozšiřuje modul taktéž z knihovny INET *Ieee80211Nic*. Jedná se o implementaci bezdrátové síťové karty, Wi-Fi karty podle standartu IEEE 802.11.

**NetworkLayer** Opět se jedná o interface z INET a to takový, který reprezentuje síťovou vrstvu uzlu. Tento interface rozšiřuje modul *IPv4NetworkLayer*, což je modul, který simuluje síťovou vrstvu pro *Internet Protocol* verze 4



**Obrázek 4.4:** Struktura uzlu MyWirelessNode

(IPv4). Stará se o přepojování paketů vyšším vrstvám transportní vrstvy TCP, UDP, echo/ping a RSVP.

**Ac\_wlan** Jendá se o *HostAutoConfigurator* z knihovny INET. Jeho funkcí je automatické přiřazení IP adres a nastavení routovací tabulky.

**UDP** UDP je modul z knihovny INET pro implementaci UDP protokolu pro IPv4 i pro IPv6. Ten poté předává paket do jedné a nebo více UDP aplikací.

**UdpApp** Interface, který rozšiřují UDP aplikace. V této aplikaci byly obě aplikace připsány a budou podrobně popsány později.

**Mobility** Mobility rozšiřují přidání interface MyIMobility. Tomu byly oproti standardnímu *IMobility* interface z knihovny INET přidány brány pro komunikaci s UDP aplikacemi. Použitý modul, *MyMobility* bude také popsán níže.

#### 4.4.3 RadioMedium

Interface *RadioMedium*, v aplikaci rozšířen modulem *Ieee80211ScalarRadioMedium* slouží k zajištění správného chování prostředí a jeho vlivu na posílané data.

Obsahuje několik pod-modulů. Modul *Propagation* popisuje, jak je rádiový signál propagován skrze prostředí v závislosti na čase. Modul šumu popisuje termální šum, kosmický šum a ostatní náhodné vlivy elektromagnetického pole a jejich vliv na kvalitu komunikačního kanálu. Dále obsahuje moduly, které simulují redukci síly signálu při průchodu prostředím a nebo i překážkami.



V tomto použitém modulu je šum v pozadí nastaven na -110dBm, prostředí na volné bez překážek, propagace signálu má konstantní rychlost a další nastavení pomocí modulu *FreeSpacePathLoss* knihovny INET.

#### ■ 4.4.4 Configurator

Interface *Configurator* rozšířen modulem *IPv4NetworkConfigurator* slouží ke nastavení IP adres pro všechny uzly sítě a k nastavení statického routování. Konfigurator není přímo propojen s některým uzlem (nemá žádné brány), v síti je vždy jen jeden.

Nastavení adres může být automatické, manuální a nebo je také možné tyto dva přístupy volně kombinovat. při automatickém nastavení není zvolená IP úplně libovolná. Přidělená adresa je vždy ve formátu 10.0.x.x

#### ■ 4.4.5 MyMobility

Modul *MyMobility* rozšiřuje interface *MyIMobility* a stará se o řízení pohybu uzlů typu *MyWirelessHost*. Možnosti a způsob pohybu byly uzpůsobeny pro potřeby simulace bezdrátové sítě s uzly, které se pohybují nekontrolovaně a uzly, které svůj pohyb přizpůsobují pohybu ostatních uzlů sítě. S uzlem samotným je nakládáno jako s nehmotným tělesem. Veškerá dynamika pohybu je zanedbána.

*MyMobility* poskytuje dva režimy pohybu v závislosti na tom, zda je uzel řízený a nebo není. ([1], [2]) Pokud je uzel kontrolovatelný, snaží se vždy co nejrychleji nejkratší cestou dostat co cílových souřadnic. Ty jsou při startu aplikace nastaveny na stejnou hodnotu jako aktuální pozice, uzel se tedy nehýbe. Pokud se cílové souřadnice změní, uzel se začne pohybovat.

V případě, že uzel nelze ovládat, je pohyb řízen automaticky. Uzel se pohybuje náhodným směrem, po předem určený čas, poté se zastaví, změní svůj směr na jiný náhodný úhel a pokračuje v pohybu. V obou případech se uzel, když narazí na okraj vymezené plochy, odrazí pod stejným úhlem jako pod kterým dopadl.

Tento modul obsahuje kromě metod zděděných z tříd pro zajištění mobility ještě dvě metody další. Jedná se o metodu *posliSouradnice*, která při zavolání odešle aktuální x-ovu a y-ovu souřadnici branou *out* rovnou do UDP aplikace. Druhá metoda, *setSouradnice* je volána z metody *handleMessage*, která je zavolána v případě, že vstupní branou *in* přijde zpráva. Tato metoda má dva argumenty a ty uloží jako cílové souřadnice x a y.

Modul *MyMobility* má několik parametrů, které je možné nastavit při spuštění. Všechny se při nenastavení nastaví automaticky na určené hodnoty. Patří mezi ně počáteční souřadnice uzlu, rychlost pohybu uzlu, úhel, o který mění svůj směr a parametr *isControlled*, který určuje, zda je uzel možno kontrolovat.

V reálném světě může tento modul reprezentovat pohyb většiny mobilních zařízení. Změny pohybu mohou být ale velmi prudké, proto pro použití u letadel a nebo klasických aut by bylo vhodné řešit pohyb jinak s ohledem na jejich omezení při zatáčení.

#### ■ 4.4.6 MyUDPApp

*MyUDPApp* je první ze dvou UDP aplikací vytvořených pro tento projekt. Je to aplikace, která ovládá komunikaci v uzlech, které nejsou ovládatelné. Po spuštění aplikace dochází k nastavení socketu pomocí zadaných parametrů. Aplikace zajišťuje dvě funkce.

První funkce je simulace jednoduché UDP komunikace mezi neovladatelnými uzly. Tato komunikace probíhá v pravidelných intervalech, interval je nastaven pomocí počátečního parametru. Jeden z uzlů odešle jednu zprávu o určené velikosti (opět nastavené pomocí parametru) na náhodně zvolenou adresu jiného neovladatelného uzlu. Ten zprávu přijme, započítá a smaže. Této komunikaci dopomáhá i kontrolovaný uzel, který je při přijetí této zprávy, která není určena pro něj, schopen zprávu přeposlat.

Druhá funkce souvisí s předáváním informací o poloze řízenému uzlu. Při přijetí takovéto žádosti je zavolána metoda *getPosition*, která zašle zprávu branou *out* modulu *MyMobility*. Jak již bylo popsáno dříve, *MyMobility* tuto žádost vyhodnotí a odešle zpět aktuální souřadnice. Metoda *handleMessage* zprávu přijme, a zavolá metodu *sendPosition*, která vytvoří packet se souřadnicemi jako parametry a ten odešle na adresu řídicího uzlu.

Komunikace mezi metodami v uzlu je zajišťována pomocí *selfMessage*, zpráv, které se dají naplánovat na konkrétní čas simulace a po odeslání se chovají, jako klasická zpráva co přišla branou. Zprávy jsou poté rozděleny na *selfMessage* a klasické zprávy, s každými je nakládáno jinak.

#### ■ 4.4.7 MyUDPAppControl

Druhá použitá aplikace, *MyUDPAppControl*, je aplikace řídicí UDP komunikaci na řízeném uzlu. Stejně jako v předchozí aplikaci se hned při spuštění nastaví pomocí parametrů komunikační socket. Tato aplikace má jen jednu funkci.

Účel tohoto uzlu je řídit topologie sítě pomocí řízení své polohy tak, aby se nacházel na zvoleném místě, které je pro přeposílání komunikace mezi neříditelnými uzly příhodné. Komunikace mezi neřízenými uzly může být díky tomuto uzlu i přes nedostatečný vysílací výkon úspěšná.

Řízený uzel se v pravidelných intervalech dotazuje na polohu všech neřízených uzlů, interval je opět nastaven pomocí parametru. Pokud se mu podaří navázat kontakt s uzlem, zažádá o zaslání polohy pomocí metody *ask* a

počká na odpověď. Pokud chvíli zažádá o polohu další neřízený uzel. Odpovědi přijdou a jsou zpracovány metodou *handleMessage*. Všechny přijaté souřadnice se uloží a po dotázání posledního neřízeného uzlu a jeho eventuelní odpovědi se zpracují. Všechny souřadnice, které byly úspěšně přijaty se zprůměrují a výsledné souřadnice jsou odeslány modulu *MyMobility*. Tam jsou uloženy jako cílové a řízený uzel se co nejdříve přesune na zadané místo.

Komunikace uvnitř uzlu je opět zajišťována pomocí *selfMessage*. Možnosti plánovat tak události je využito u plánování příštího dotazu na polohu.



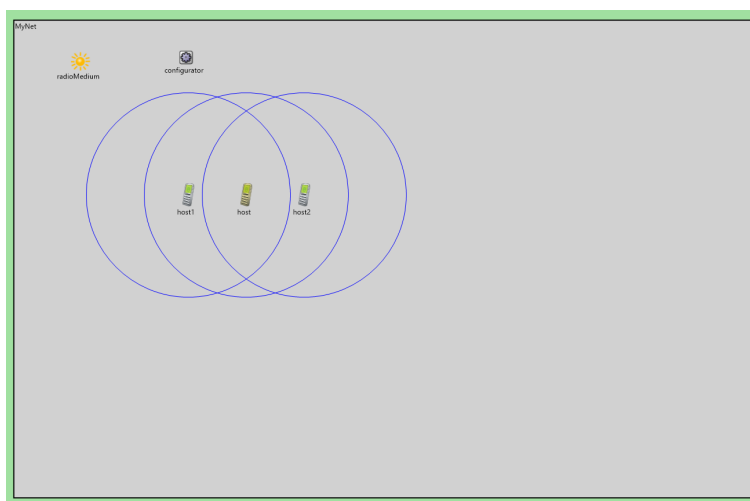
# Kapitola 5

## Experiment

V této části experimentálně ověříme vlastnosti a funkčnost vytvořeného modelu. Budeme porovnávat dvě nastavení simulace. První, kdy se komunikace účastní pouze neřízené uzly a v pravidelných intervalech si vyměňují zprávy. Druhé, ve kterém přibude řízený uzel, který s neřízenými uzly komunikuje a snaží se vylepšit úspěšnost jejich komunikaci.

Další nastavení, které budeme ovlivňovat je interval mezi dotazováním polohy.

Během simulace budeme pozorovat poměr ztracených zpráv vůči celkovému počtu odeslaných a také zpoždění, se kterým jsou jednotlivé zprávy doručeny.



**Obrázek 5.1:** Experiment č. 1 - Počáteční pozice uzlů

### 5.1 Nastavení simulace

Při počátečním nastavení jsou neřízené uzly umístěny 200m od sebe a spojnici mezi nimi je řízený uzel. Počáteční poloha je patrná na obrázku č. 5.1.

Jako model prostředí, ve kterém probíhá bezdrátový přenos je použit modul

z knihovny INET *Ieee80211ScalarRadioMedium*. Prostupnost je nastavena opět z knihovny INET jako *FreeSpacePathLoss*, jedná se tedy o prostředí, ve kterém nejsou žádné překážky.

Typ propagace signálu skrz prostředí je nastaven na typ, s konstantní rychlostí. Síla šumu v pozadí má hodnotu -110dBm. Signál je přenášen na frekvenci 2.4GHz, což je hodnota podle standardu 802.11 (Wi-Fi).

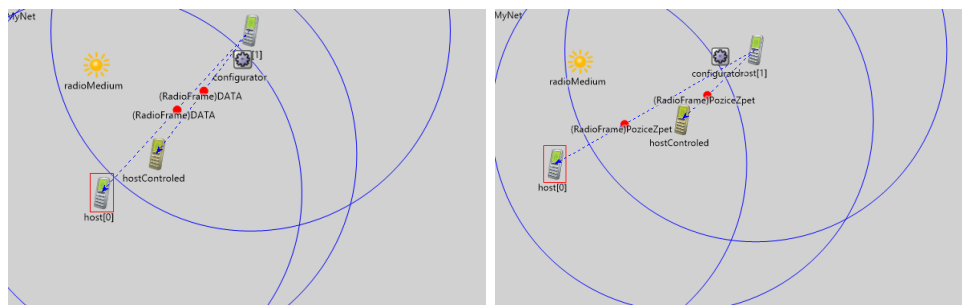
Výkon vysílačů všech uzlů má hodnotu 2mW a citlivost přijímačů u všech uzlů je nastavena na hodnotu -85dBm. To znamená, že přijímač není schopen signál přijmout, pokud je jeho výkon pod touto hodnotou.

Uzly si posílají 100 B zprávy a to jak v datové komunikaci, která probíhá mezi neřízenými uzly, tak v komunikaci k předání polohy. Délka simulace je nastavena na 30 minut.

## 5.2 Naměřená data

### 5.2.1 Experiment č. 1

V prvním experimentu budeme simulovat síť se dvěma neřízenými uzly. Budeme pozorovat počet přijatých zpráv s využitím řízeného uzlu při intervalu dotazování polohy 10s, 5s, 2s a při absenci řízeného uzlu. Při absenci ří-



Obrázek 5.2: Experiment č. 1 - Průběh simulace

zeného uzlu probíhá komunikace uzlů každých 10s. Výsledky jsou uvedené v tabulce č. 5.1. Druhé části experimentu byl přidán jeden řízený uzel. V

Jméno uzlu	Odeslaných zpráv	Přijatých zpráv	Ztracených zpráv [%]
Host 1	180	96	46.7
Host 2	180	102	46.3

Tabulka 5.1: Experiment č. 1 - bez řízeného uzlu

následujících tabulkách můžeme pozorovat, jak se procento ztracených zpráv

měnilo se změnou intervalu dotazování polohy a to zvláště pro *Host1* a pro *Host2* (5.2,5.3)

■ Host 1

Interval dotazování [s]	Odeslaných zpráv	Přijatých zpráv	Ztracených zpráv [%]
10	180	117	35
5	180	118	34.4
2	180	128	28.9

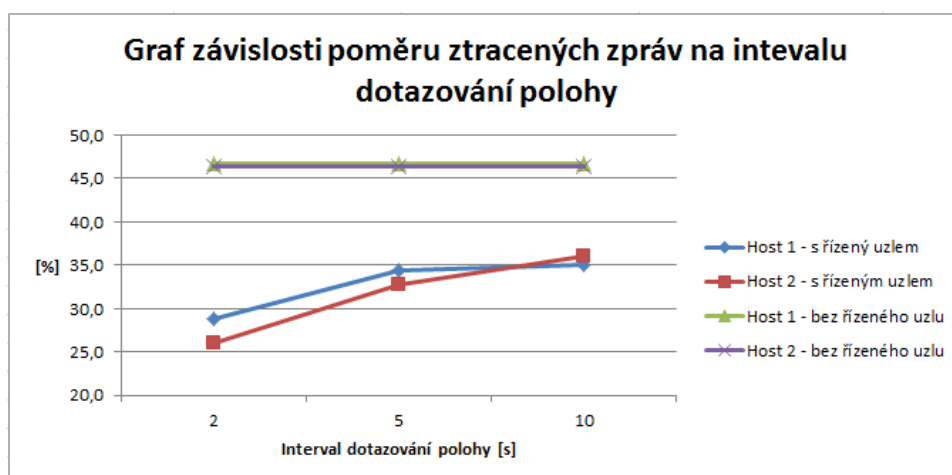
**Tabulka 5.2:** Experiment č. 1 - Host1

■ Host 2

Interval dotazování [s]	Odeslaných zpráv	Přijatých zpráv	Ztracených zpráv [%]
10	180	115	36.1
5	180	121	32.8
2	180	133	26.1

**Tabulka 5.3:** Experiment č. 1 - Host2

Průběhy závislostí poměru ztracených zpráv na čase dotazování polohy pro oba uzly jsou patrné na obrázku 5.3. Je z něj patrné, že při optimální implementaci řízeného uzlu je možné komunikaci sítě o dvou neřízených uzlech vylepšit o několik procent. Pokud je dotazovací interval příliš vysoká hodnota (v tomto experimentu 10s), řízený uzel není vždy přítomen na optimální pozici, a proto nemusí komunikaci optimálně vylepšit. Tento interval proto musí být dostatečně krátký, aby se řízený uzel ideálně přesouval.



**Obrázek 5.3:** Experiment č. 1 - Graf závislosti poměru ztracených zpráv na intervalu dotazování

## 5.2.2 Experiment č. 2

V druhém experimentu budeme uvažovat síť, která se skládá z jednoho řízeného a 4 neřízených uzlů. Počáteční umístění uzlů je pravidelné, neřízené uzly se nacházejí vždy 100m od řízeného uzlu ve směru nahoru, dolů, doleva a doprava. Toto umístění je patrné na obrázku 5.4. Zprávy byly odesílány pravidelně po 10 sekundách ze všech neřízených uzlů. Adresa příjemce byla náhodně vybrána.



Obrázek 5.4: Experiment č. 2 - Počáteční pozice uzlů

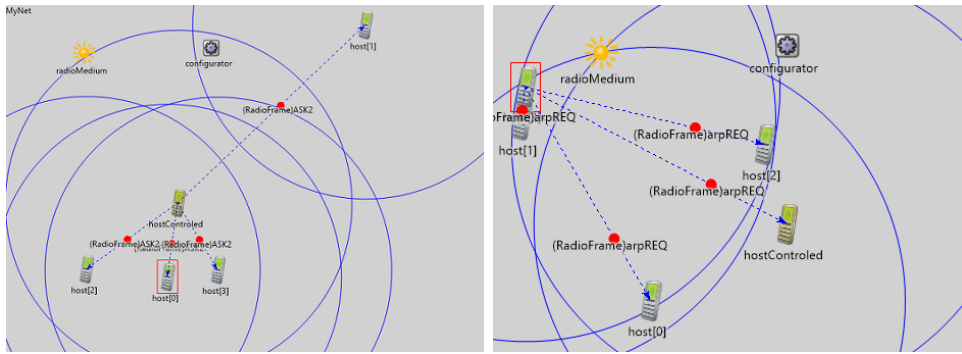
Při tomto experimentu budeme opět pozorovat počet přijatých zpráv s využitím řízeného uzlu při intervalu dotazování polohy 10s, 5s, 2s a při absenci řízeného uzlu. Jelikož je při odeslání zprávy náhodně vybírán příjemce z ostatních uzlů, pro větší přesnost měření jsem provedl každé měření 5x a výsledky zprůměroval. Všechny neřízené uzly vysílají zprávy v intervalu 10s. Délka simulace je nastavena na 30 minut, což znamená, že každý uzel vyšle za simulaci 180 zpráv. Průběh simulace je patrný na obrázcích 5.5.

Opět jako první provedu experiment bez přidání řízeného uzlu. Výsledky tohoto testu jsou patrné v tabulce číslo 5.4

Jméno uzlu	Odeslaných zpráv	Přijatých zpráv	Ztracených zpráv [%]
Host 1	180	41	77.2
Host 2	180	55	69.4
Host 3	180	61	66.1
Host 4	180	46	74.4

Tabulka 5.4: Experiment č. 2 - bez řízeného uzlu





Obrázek 5.5: Experiment č. 2 - Průběh simulace

Poté byl přidán jeden řízený uzel. V tabulce číslo 5.5 můžete vidět počet přijatých zpráv pro jednotlivé uzly v závislosti na intervalu dotazování polohy. Jak již bylo řečeno, jedná se o hodnoty vzniklé zprůměrováním pěti měření.

Interval dotazování [s]	Host 1	Host 2	Host 3	Host 4
10	43	56	58	47
5	64	57	59	60
2	67	53	67	64

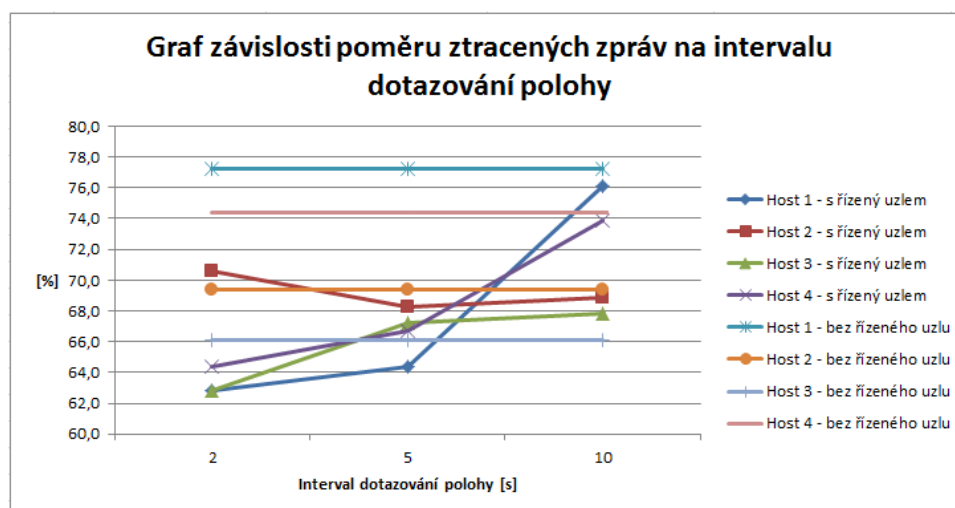
Tabulka 5.5: Experiment č. 2 - Počet doručených zpráv

V následující tabulce, tabulce číslo 5.6 jsou hodnoty z tabulky 5.5 přepočítány na poměr ztracených zpráv v procentech. Tyto poměry jsou také vyznačeny v obrázku číslo 5.6

Interval dotazování [s]	Host 1	Host 2	Host 3	Host 4
10	76.1	68.9	67.8	73.9
5	64.4	68.3	67.2	66.7
2	62.8	70.6	62.8	64.4

Tabulka 5.6: Experiment č. 2 - Poměr ztracených zpráv

Z naměřených hodnot je patrné, že přidání řízeného do sítě čtyř neřízených uzlů opět zvýší počet doručených zpráv a tím i vylepší vlastnosti sítě. Toto vylepšení není ale oproti síti s pouze dvěma uzly tak výrazné. Je patrné, že řízený uzel není schopen vylepšit komunikaci mezi všemi neřízenými uzly sítě, vylepšuje pouze tu, která probíhá mezi uzly, se kterými je schopen komunikovat. Tento problém lze předpokládám vyřešit přidáním dalších řízených uzlů.



**Obrázek 5.6:** Experiment č. 2 - Graf závislosti poměru ztracených zpráv na intervalu dotazování

## Kapitola 6

### Závěr

V této práci jsem zabýval problematikou řízení topologie bezdrátových ad hoc sítí s využitím bezpilotních prostředků. V první části jsem zpracoval teoretický základ k mobilním ad hoc sítím (MANET) a k možnostem řízení jejich topologie. Dále jsem se věnoval existujícím síťovým simulátorům.

Simulátory jsem porovnal podle jejich specifických vlastností a vybral z nich simulátor OmNet++ jako nástroj, který je pro simulaci takovýchto sítí vhodný, má velmi kvalitní dokumentaci, dostupné frameworky pro návrhy ad hoc bezdrátových sítí a jehož využití je bezplatné. Použil jsem nejnovější verzi OmNet++ 5.0, která je proti předchozím verzím vylepšena především ve 2D a 3D zobrazení. Na základě průběhů simulací bych tento simulátor doporučil.

V simulátoru OmNet++ jsem navrhl simulační model, který umožňuje simulovat sítě, ve kterých se využívá řízení pohybu uzlu pomocí informací o poloze ostatních uzlů sítě. Bylo nutné vytvořit moduly, které takovéto informace mohou předávat. Vytvořil jsem modul *MyWirelessNode*, který simuluje uzel komunikující pomocí bezdrátového spojení, u něhož je možné nastavení jeho mobility a také který je schopen s ostatními uzly komunikovat pomocí UDP protokolu. Modul *MyMobility*, který je schopen pohyb uzlu ovládat na základě informace o pozici ostatních uzlů, byl umístěn mezi pohybové moduly knihovny pro řízení pohybu. A konečně byly vytvořeny komunikační aplikace, které zajišťují komunikaci nutnou pro řízení topologie *MyUDPApp* a *MyUDPAppControl*.

Model, jeho funkčnost a vlastnosti jsem experimentálně ověřil. V experimentech byly porovnány dvě nastavení simulace, první, kdy se komunikace účastnily pouze neřízené uzly, které si v pravidelných intervalech vyměňují zprávy. A simulace, ve které přibyl řízený uzel komunikující s neřízenými uzly a snažící se vylepšit úspěšnost jejich komunikaci. V průběhu simulace byl vyhodnocen poměr ztracených zpráv vůči celkovému počtu odeslaných zpráv a také zpoždění, se kterým byly jednotlivé zprávy doručeny.

Z výsledků experimentu je patrné, že přidání řízeného uzlu do sítě za účelem

přeposílání zpráv komunikace má pozitivní vliv na celkový počet doručených zpráv. Pokud je však dotazovací interval příliš velký, řízený uzel není vždy přítomen na optimální pozici, a komunikaci nevylepší. Interval proto musí být dostatečně krátký, aby se řízený uzel ideálně přesouval. Čím přesněji je tedy tento uzel řízen, tím lépe udržuje ideální pozici po přeposílání zpráv a jeho vliv je tím tedy větší. Jedná se tedy o způsob, jak vylepšit komunikaci neovladatelných uzlů.

V případě větší sítě, sítě s více neřízenými uzly není řízení pomocí komunikační aplikace *MyUDPAppControl* příliš efektivní, protože se některé uzly dostávaly mimo dosah řízeného uzlu. Komunikaci u takových uzlů nemůže řízený uzel vylepšit. Možným řešením je zapojení více říditelných uzlů. Experiment však prokázal, že navržený modul *MyMobility* v takovýchto sítích je funkční a umožňuje další vývoj výše uvedené řídicí aplikace.

Jak bylo řečeno v úvodu, při návrhu MANET sítí je síť nutné testovat. Na základě popsaných výsledků jsem přesvědčen, že při rozšíření simulátoru OmNet++ o knihovnu INET s přidaným modulem *MyMobility* se jedná o vhodný simulátor sítí s možností řízení topologie.

Způsobem, jak navržené moduly využít v budoucnu je navázání dalších projektů v simulátoru OmNet++. Moduly umožňující předávání informací o poloze najdou uplatnění v simulaci sítí, ve kterých je nutné řídit roboty, drony, letadla a jiná bezpilotní zařízení podle polohy dalších uzlů sítě.



## Přílohy





## Příloha A

### Literatura

- [1] Ons Bouachir, Alineo Abrassart, Fabien Garcia, Nicolas Larrieu: A Mobility Model for UAV Ad hoc Network. ICUAS 2014, International Conference on Unmanned Aircraft Systems, May 2014, Orlando, United States. pp 383-388, 2014
- [2] JD. Biomo, T. Kunz, M. St-Hilaire, Y. Zhou: Unmanned Aerial ad Hoc Networks: Simulation-Based Evaluation of Entity Mobility Models Impact on Routing Performance. Aerospace - Open Access Aeronautics and Astronautics Journal, 2, 392-422, 2015
- [3] Paolo Santi: Topology Control in Wireless Ad Hoc and Sensor Networks. Wiley; 1 edition (August 12, 2005)
- [4] Andras, V., 2016. OMNET ++ User Manual version 5.0.  
<https://omnetpp.org/doc/omnetpp/manual/>
- [5] Group of contributors, 2016. INET Framework for OMNet++ Manual  
<https://omnetpp.org/doc/inet/api-current/inet-manual-draft.pdf>
- [6] Luc Hogue, Pacal Bouvry, Frédéric Guinand: An Overview of MANETs Simulation. Elektronik Notes in Theoretical Computer Science, Volume 150, Issue 1, 9 March 2006, pp 81-101
- [7] Varun Manchikalapudi, Sk. Khadar Babu: Simulation of Efficiency in Mobile Ad Hoc Networks using OMNet++. Research Journal of Applied Sciences, Engineering and Technology 10, 5 August 2015
- [8] Sajata V. Mallapur, Siddarama R. Patil: Survey on Simulation Tools for Mobile Ad-Hoc Networks. IRACST - International Journal of Computer Networks and Wireless Communications (IJCNWC), Vol. 2, No. 2, April 2012
- [9] Christoph P. Mayer, Thomas Gamer: Integrating real world applications into OMNeT++. TeleMatics Technical Reports, Institute of Telematics, University of Karlsruhe Zirkel 2, D-76128, Germany, 27 February 2008

- [10] Holub J., Novák J.: Přednášky předmětu A3B38DSY - Distribuované systémy a počítačové sítě, FEL, ČVUT. (2011)  
<http://measure.fel.cvut.cz/vyuka/predmety/A3B38DSY>





## Příloha B

### Obsah přiloženého CD

- BP.pdf

- inet

Tato složka obsahuje všechny vytvořené a nebo pozměněné moduly, které byly přidány do knihovny INET. Kompletní knihovna INET není kvůli svojí velikosti v příloze zahrnutá. INET je volně dostupný volně ke stažení na stránkách <https://inet.omnetpp.org/>

- BakalarskaPrace

Tato složka obsahuje projekt simulátoru OmNet++, který obsahuje samotnou síť a přidává počáteční nastavení.

- Literatura



eské vysoké u ení technické v Praze  
Fakulta elektrotechnická

katedra ídicí techniky

## ZADÁNÍ BAKALÁ SKÉ PRÁCE

Student: **Pavel Elis**

Studijní program: Kybernetika a robotika

Obor: Systémy a ízení

Název tématu: **ízení topologie bezdrátových adhoc sítí s využitím bezpilotních prost edk**

Pokyny pro vypracování:

1. Seznamte se s problematikou mobilních adhoc sítí (MANET) a ízením jejich topologií.
2. Seznamte se s frameworky pro simulaci komunikace v bezdrátových sítích.
3. Navrhn te modul pro simulaci komunikace v t chto sítích s ohledem na využití bezpilotních letoun .
4. Navržený modul implementujte a experimentáln ov te jeho vlastnosti.

Seznam odborné literatury:

- [1] Ons Bouachir, Alinee Abrassart, Fabien Garcia, Nicolas Larrieu: A Mobility Model for UAV Ad hoc Network. ICUAS 2014, International Conference on Unmanned Aircraft Systems, May 2014, Orlando, United States. pp 383-388, 2014.
- [2] JD. Biomo, T. Kunz, M. St-Hilaire, Y. Zhou: Unmanned Aerial ad Hoc Networks: Simulation-Based Evaluation of Entity Mobility Models Impact on Routing Performance. Aerospace - Open Access Aeronautics and Astronautics Journal, 2, 392-422, 2015.
- [3] Paolo Santi: Topology Control in Wireless Ad Hoc and Sensor Networks. Wiley; 1 edition (August 12, 2005).

Vedoucí: Ing. Milan Rollo, Ph.D.

Platnost zadání: do konce letního semestru 2016/2017

L.S.

prof. Ing. Michael Źebek, DrSc.  
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.  
d kan

V Praze dne 1. 2. 2016