

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018

Bc. Martin Pecina



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## NÁVRH LABORATORNÍ ÚLOHY ZAMĚŘENÉ NA PROTOKOL OLSR

DESIGN OF LABORATORY TASKS FOCUSED ON THE OLSR PROTOCOL.

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Martin Pecina

### VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Anna Kubánková, Ph.D.

BRNO 2018



# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Martin Pecina

**ID:** 156905

**Ročník:** 2

**Akademický rok:** 2017/18

**NÁZEV TÉMATU:**

## Návrh laboratorní úlohy zaměřené na protokol OLSR

**POKYNY PRO VYPRACOVÁNÍ:**

Nastudujte principy sítí MANET a směrovací protokoly používané v těchto sítích. Zaměřte se na protokol OLSR a podrobně prostudujte jeho principy. Seznamte se s prostředím NS-3. Navrhněte, vytvořte v prostředí NS-3 a popište scénář, který umožní pochopit a vyzkoušet vlastnosti zvoleného směrovacího protokolu. Analyzujte chování směrovacího protokolu v různých síťových podmínkách. Vytvořte podrobný návod k laboratorní úloze s doplňujícími otázkami a vzorové řešení. Časová náročnost laboratorní úlohy by měla být přibližně 2 hodiny.

**DOPORUČENÁ LITERATURA:**

1] RFC 3626, „Optimized Link State Routing Protocol (OLSR)“, 2003.

[2] ns-3. [online]. [cit. 2017-09-11]. Dostupné z: <https://www.nsnam.org/>

**Termín zadání:** 5.2.2018

**Termín odevzdání:** 21.5.2018

**Vedoucí práce:** Ing. Anna Kubánková, Ph.D.

**Konzultant:**

**prof. Ing. Jiří Mišurec, CSc.**  
*předseda oborové rady*

**UPOZORNĚNÍ:**

Autor diplomové práce nesmí při vytváření diplomové 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 části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Ad hoc mobilní síť je kolekcí mobilních uzlů, které se dynamicky a svévolně pohybují takovým způsobem, že se spojení mezi uzly nepřetržitě mění. Za účelem usnadnění komunikace v síti se k nalezení přenosových cest mezi uzly využívá směrovací protokol. Primární úkolem takového ad hoc směrovacího protokolu je přesné a efektivní sestavení tras mezi dvojicí uzlů tak, aby poslané zprávy byly doručeny včas. Sestavení tras by mělo být splněno při minimální režii a minimálním využití přenosového pásma. Tato práce popisuje stavový směrovací protokol optimalizovaný pro mobilní ad hoc síť OLSR. Klíčový koncept tohoto protokolu spočívá ve vícebodovém spojení (MPR), které poskytuje efektivní mechanismus výměny kontrolních zpráv při nižším počtu potřebných přenosů.

## KLÍČOVÁ SLOVA

OLSR, MANET, MPR, Směrování, NS-3

## ABSTRACT

An ad hoc mobile network is a collection of mobile nodes that are dynamically and arbitrarily located in such a manner that the interconnections between nodes are capable of changing on a continual basis. In order to simplify communication within the network, a routing protocol is used to discover routes between nodes. The primary goal of such an ad hoc network routing protocol is correct and efficient route establishment between a pair of nodes so that messages may be delivered on time. Route construction should be done with a minimum of overhead and bandwidth consumption. This document describes the Optimized Link State Routing (OLSR) protocol for mobile ad hoc networks. The key concept used in the protocol is that of multipoint relays (MPRs). MPRs provide an efficient mechanism for flooding control traffic by reducing the number of transmissions required.

## KEYWORDS

OLSR, MANET, MPR, Routing, NS-3

PECINA, Martin. *Návrh laboratorní úlohy zaměřené na protokol OLSR*. Brno, 2018, 72 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Anna Kubánková, Ph.D.



## PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Návrh laboratorní úlohy zaměřené na protokol OLSR“ jsem vypracoval samostatně pod vedením vedoucího diplomové 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 diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

podpis autora(-ky)

Výzkum popsany v této diplomové práci byl realizovaný v laboratořích podpořených projektem Centrum senzorických, informačních a komunikačních systémů (SIX); registrační číslo CZ.1.05/2.1.00/03.0072, operačního programu Výzkum a vývoj pro inovace.

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucí diplomové práce paní Ing. Anně Kubánkové, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno .....

.....

podpis autora(-ky)

# OBSAH

Úvod	11
<b>1 MANET sítě</b>	<b>12</b>
1.1 Vývoj	12
1.2 Cíle	12
1.3 Obecný popis	13
1.3.1 BAN	14
1.3.2 PAN	14
1.3.3 WLAN	14
1.4 Směrování v MANET sítích	15
1.4.1 Unicast	16
1.4.2 Broadcast	16
1.4.3 Multicast	16
1.4.4 Geocast	16
<b>2 Směrování</b>	<b>17</b>
2.1 Popis směrování	17
2.1.1 Logika směrování z pohledu stanice	17
2.1.2 Logika směrování z pohledu směrovače	18
2.2 Způsoby směrování	18
2.3 Algoritmy směrování	19
2.3.1 Vektorové algoritmy	20
2.3.2 Stavové algoritmy	20
<b>3 Směrovací protokoly pro mobilní sítě</b>	<b>21</b>
3.1 Proaktivní	22
3.1.1 DSDV	22
3.1.2 WRP	23
3.2 Reaktivní	24
3.2.1 AODV	24
<b>4 OLSR</b>	<b>25</b>
4.1 Použitelnost	26
4.2 Obecný popis	26
4.3 MPR	27
4.4 Struktura protokolu	27
4.4.1 Struktura jádra protokolu	28
4.4.2 Podpůrné funkce	29

4.4.3	Formát paketu a přeposílání . . . . .	30
4.4.4	Informační schránka . . . . .	31
4.5	OLSRv2 . . . . .	32
<b>5</b>	<b>NS-3</b>	<b>33</b>
<b>6</b>	<b>Výsledky studentské práce</b>	<b>35</b>
6.1	Programové řešení . . . . .	36
6.1.1	Model sítě . . . . .	36
6.1.2	Nastavení komunikace . . . . .	37
6.1.3	Generace datového provozu . . . . .	39
6.2	Výstupy simulace . . . . .	40
6.2.1	Výběr MPR . . . . .	40
6.2.2	Sestavení směrovací tabulky . . . . .	44
6.2.3	Statistiky protokolu . . . . .	49
6.3	Úkoly laboratorní úlohy . . . . .	53
<b>7</b>	<b>Závěr</b>	<b>57</b>
	<b>Literatura</b>	<b>58</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>59</b>
	<b>Seznam příloh</b>	<b>61</b>
<b>A</b>	<b>Přílohy</b>	<b>62</b>
A.1	Přehled metod pro rozmístění uzlů . . . . .	62
A.2	Přehled metod pro pohyb uzlů . . . . .	63
A.3	Přehled parametrů třídy YansWifiPhyHelper . . . . .	64
A.4	Laboratorní úloha . . . . .	64

# SEZNAM OBRÁZKŮ

1.1	Dosah pokrytí ad hoc sítí . . . . .	13
3.1	Rozdělení ad hoc směrovacích protokolů dle kategorií . . . . .	22
4.1	Formát paketu protokolu OLSR . . . . .	31
5.1	Technologie softwaru NS-3 . . . . .	33
5.2	Prostředí Eclipse s otevřeným příkladem NS-3 . . . . .	34
6.1	Příklad topologie OLSR . . . . .	35
6.2	Detekce sousedních uzlů pomocí zpráv HELLO . . . . .	40
6.3	Náhodná topologie pro ukázkou směrování OLSR protokolu . . . . .	41
6.4	Úvodní zpráva HELLO . . . . .	41
6.5	Zpráva HELLO informující o asymetrických linkách . . . . .	42
6.6	Sekvenční zpráva HELLO č. 2 nastavující symetrické linky . . . . .	42
6.7	Sekvenční zpráva HELLO č. 3 - zvolení MPR . . . . .	43
6.8	Dosah uzlu 9 . . . . .	43
6.9	Volba MPR uzlu 9 . . . . .	44
6.10	Směrovací tabulka uzlu 9 v čase 1 s . . . . .	45
6.11	Směrovací tabulka uzlu 9 v čase 12 s . . . . .	45
6.12	Kontrolní zpráva protokolu OLSR . . . . .	46
6.13	UDP přenos zobrazený v programu Wireshark . . . . .	47
6.14	Porovnání statistik v programu NetAnim . . . . .	48
6.15	Statistika propustnosti v závislosti na počtu uzlů v síti . . . . .	49
6.16	Statistika počtu přeskoků v závislosti na počtu uzlů v síti . . . . .	50
6.17	Statistika zpoždění v závislosti na počtu uzlů . . . . .	51
6.18	Statistika ztrátovosti v závislosti na rychlosti pohybu uzlů . . . . .	52

# SEZNAM TABULEK

2.1	Porovnání statického a dynamického směrování . . . . .	19
4.1	Tabulka základní komponent jádra OLSR protokolu . . . . .	28
4.2	Tabulka podpůrných funkcí jádra protokolu OLSR . . . . .	29
6.1	Typ zpráv OLSR protokolu a jejich popis . . . . .	46
A.1	Metody rozmístění uzlů [9] . . . . .	62
A.2	Metody pohybu uzlů [9] . . . . .	63
A.3	Parametry třídy YansWifiPhyHelper [9] . . . . .	64

# ÚVOD

V posledních letech se v oblasti komunikací dostává do popředí trend, kdy síť není závislá na jakémkoli centrálním prvku a bude přesto schopna sebe sama konfigurovat a organizovat. V tomto ohledu je vhodným příkladem mobilní ad hoc síť (MANET). Ta je díky své decentralizované povaze vhodná pro různé druhy aplikací od senzorných sítí, přes dopravní aplikace, k obecné komunikaci bod-bod(peer-to-peer) či armádnímu průmyslu, díky kterému tyto sítě původně vznikly. Sítím MANET je věnována kapitola 1, která popisuje vývoj těchto sítí, jejich obecný popis, využití z hlediska pokrytí a následně směrování využívané uvnitř těchto sítí.

Druhá kapitola 2 popisuje směrování jako takové, logiku směrování z pohledu stanice a směrovače, způsoby směrování a nakonec algoritmy využívané ve směrování. Na tuto kapitolu navazují směrovací protokoly mobilních sítí 3. Zde je popsáno rozdělení těchto protokolů z pohledu přístupu ke směrování s uvedenými příklady.

Následuje kapitola 4 s hlavní částí této práce, popis optimalizovaného protokolu pro mobilní sítě ad hoc OLSR. Je zde popsán klíčový prvek protokolu, vícebodové spojení MPR, struktura protokolu, jeho funkčnost a vylepšení v podobě jeho druhé verze.

Předposlední kapitola 5 se zabývá popisem využitého simulačního prostředí NS-3, na kterou navazuje praktická část, kapitola 6 diplomové práce - simulace sítě MANET s využitím protokolu OLSR. Simulace proběhly v náhodné topologii o odlišném počtu uzlů, kde se směrovací informace stanovily v průběhu simulace protokolem OLSR. V simulacích byl vyzkoušen přenos UDP paketů mezi danými uzly k otestování funkčnosti nastaveného směrování. Na základě těchto simulací a sepsané teorie byl vytvořen návrh laboratorní úlohy, tedy hlavní cíl diplomové práce.



# 1 MANET SÍTĚ

Mobilní ad hoc sítě, neboli MANET (Mobile Ad hoc Network), reprezentují systém mobilních uzlů, které se mohou volně a dynamicky seskupovat do dočasných síťových topologií. Dovolují lidem i zařízením bezproblémově komunikovat v prostoru bez jakéhokoli předchozí síťové konfigurace. Ačkoliv k nasazení velkoplošných MANET sítí zbývá dořešit několik problémů, malé (maloplošné) sítě jsou již plně funkční. Síťové karty pro jedno-skokové (single-hop) ad hoc sítě jsou volně dostupné a dosahují vzdáleností několika kilometrů[1] [2].

## 1.1 Vývoj

V posledních letech rozšíření mobilních výpočetních zařízení (laptopy, mobilní telefony, "wearable"<sup>1</sup> zařízení) způsobilo revoluci ve výpočetním světě. Společnost se posunula z doby osobních počítačů do doby všudypřítomných zařízení, které uživatelé využívají současně, na několika platformách s výhodou přístupu k potřebným datům kdykoliv a kdekoliv. Přirozenost těchto zařízení vytvořila z bezdrátových sítí nejjednodušší řešení pro komunikaci. To vedlo k rapidnímu vzrůstu využití bezdrátových technologií v prostředí lokálních sítí - LAN (Local Area Network), následně tedy WLAN (Wireless Local Area Network). WLAN zařízení ovšem spotřebovávají velké množství energie kvůli potřebné vzdálenosti (desítky m) a konektivě se spoustou ostatních zařízení. Vznikla tedy nová velikostní třída sítí - PAN (Personal Area Network). PAN síť umožňuje zařízením v rámci malého prostoru dynamicky sdílet informace při nízké energetické náročnosti[2].

## 1.2 Cíle

Jedním z cílů moderních zařízení je možnost adaptace technologie uživateli aniž by uživatel musel změnit způsob práce nebo zařízení rozumět. Osobní počítače nabízí uživatelům velké množství aplikací a služeb, ale ke skutečnému využití těchto výhod se musí uživatelé přizpůsobit počítačovým standardům. Vzniká tedy trend, kdy zkoumáním technologie a infrastruktury jednotlivá zařízení pomáhají uživateli v libovolném prostředí bez nutnosti změny přístupu či chování. Tato filozofie byla základem pro současný koncept "okolní inteligence". Cílem této okolní inteligence je integrace digitálních zařízení a sítí do každodenního života. Vzniká nové paradigma, kdy uživatelská mobilní zařízení vytvářející síť, nejsou pouze konečným přístupovým bodem, ale zároveň kooperativně poskytují funkcionalitu, kterou tradičně zastarávají

---

<sup>1</sup>wearable - nositelná elektronika (chytré hodinky, MP3 přehrávače, fitness náramky)

prvky síťové infrastruktury (server, směrovač, přepínač). Tento systém představuje ad hoc síť, neboli neinfrastrukturní (decentralizovanou) bezdrátovou síť[2].

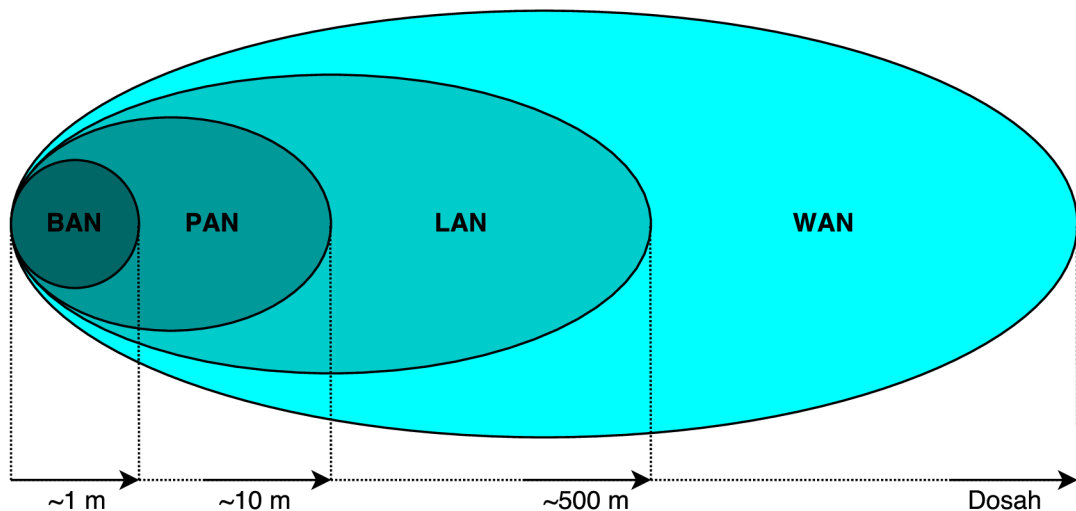
### 1.3 Obecný popis

Mobilní ad hoc síť je systém bezdrátových mobilních uzlů, které dynamicky utváří dočasné síťové topologie. Lidé a vozidla (dopravní prostředky) mohou být tedy propojeny bez předchozí konfigurace infrastruktury nebo v případě, že stávající infrastruktura vyžaduje bezdrátové rozšíření.

Sítě ad hoc můžeme dle jejich pokrytí rozdělit do známých tříd[2]:

- BAN (Body Area Network),
- PAN (Personal Area Network),
- LAN (Local Area Network),
- WAN (Wide Area Network).

Dosah jednotlivých tříd pokrytí je možné vidět na Obr.1.1 [2].



Obr. 1.1: Dosah pokrytí ad hoc sítí

Prvky širokopásmové ad hoc sítě jsou tzv. více-skokové (z anglického multi-hop). V rámci WAN bezdrátových sítí je nutné dořešit určitá specifika - adresování, směrování, bezpečnost, přizpůsobení polohy atd.

V následujících podsekcích budou popsány využívané třídy pokrytí ad hoc sítí.

### 1.3.1 BAN

Sítě BAN jsou spojeny s "wearable" zařízeními. Tyto komponenty jsou umístěny na našem těle k usnadnění práce (sluchátka, chytré hodinky, fitness měřiče atd) a BAN síť poskytuje propojení mezi těmito zařízeními. Propojení může být drátové (kabelové) i bezdrátové.

Hlavní požadavky pro BAN síť jsou[2]:

- možnost propojit heterogenní zařízení, od komplexních zařízení (mobilní telefon) po součásti zařízení (sluchátka, fitness měřič),
- schopnost autokonfigurace - připojení/odpojení zařízení do/z BAN sítě musí být pro uživatele transparentní,
- integrace služeb (izochronní dostupnost dat jako je provoz mobilních dat (i většího množství současně),
- možnost propojení s ostatními BAN nebo PAN sítěmi (výměna dat mezi lidmi/aplikacemi).

### 1.3.2 PAN

Osobní sítě propojují mobilní zařízení, nesené uživateli, s ostatními zařízeními i stacionárními. Zatímco BAN sítě zajišťují propojení zařízení na lidském těle, PAN jsou sítě v okolí člověka, ať už se jedná o propojení BAN sítí mezi sebou nebo spojení BAN sítě s okolním prostředím. Nejrozšířenější z PAN sítí jsou WPAN - bezdrátové PAN v ISM<sup>2</sup> pásmu 2,4 GHz. Technologie WPAN nabízí široké uplatnění pro inovativní řešení a aplikace, které způsobily radikální změnu v každodenním životě. Dle předpokladů se WPAN rozhraní rozšířilo nejen u mobilních telefonů, laptopů a PDA (Personal Digital Assistant), ale u každého digitálního zařízení. Dnes se začíná využívat síť PAN i pro tzv. chytrou domácnost, kdy přes telefon (tablet či "chytré" hodinky) můžeme řídit osvětlení pokojů, odemykání/zamykání dveří, ovládání klimatizace atd[2].

### 1.3.3 WLAN

V posledních letech se využití bezdrátových technologií v prostředí lokálních sítí stalo více a více důležité obzvláště díky ohromné flexibilitě, kterou oproti kabelovým LAN sítím nabízí. Tyto sítě jsou dnes základem automatizace každé kanceláře a pomalu také domácnosti. Stejně jako kabelová, tak i bezdrátová, LAN síť má dosah 100-500 m. WLAN splňuje stejné požadavky jako LAN - vysoká kapacita, plná konektivita a možnost broadcast vysílání. Při plnění těchto požadavků se setkáváme s

---

<sup>2</sup>Pásma pro rádiové vysílání, bez licenčních poplatků, avšak bez garance proti rušení - industrial, scientific and medical)

určitými problémy bezdrátového prostředí jako je bezpečnost v atmosféře, spotřeba elektrické energie a limitace šířky pásma ve volném prostředí[2].

U implementace WLAN sítí se setkáváme s dvěma přístupy (konfiguracemi) - infrastrukturní a bez infrastrukturní (ad hoc). Infrastrukturní architektura stanovuje existenci pevného centrální prvku pro danou síť či podsít, nazvaného přístupový bod (Access Point). Ten je připojen ke kabelové síti a mobilním zařízením poskytuje přístup k Internetu. V konfiguraci ad hoc není žádný pevný centrální uzel vyžadován, ale je dynamicky zvolen stanicemi zúčastňujícími se komunikace[2].

## 1.4 Směrování v MANET sítích

Směrování jako takovému je věnována následující kapitola 2. Tato sekce se věnuje popisu směrování v sítích MANET.

Směrování v sítích MANET je díky častým změnám v topologii, kdy aktivní linky mohou být posunem uzlů přerušeny, velmi náročné. Směrovací schéma v ad hoc sítích musí zahrnovat mechanismy, které si poradí s obtížemi způsobenými mobilitou uzlů a změnami v topologii při nízké energetické náročnosti, šířce pásma a nízkými výpočetními prostředky. Tradiční stavové a vektorové algoritmy využívané v klasických sítích nejsou pro MANET sítě vhodné kvůli obrovské režii směrovaných informací (aktualizací). K vyřešení těchto problémů, obzvláště u velkých sítí, byla navržena spousta protokolů. Některá z nich jsou popsány v následujících kapitolách 3, 4.

Tyto protokoly využívají určité typy (způsoby) směrování zpráv:

- unicast,
- broadcast,
- multicast,
- geocast.

Jednotlivé typy směrování jsou krátce popsány v následujících podsekcích.

### 1.4.1 Unicast

Pomocí unicast směrování jsou směrovací zprávy přenášeny způsobem bod-bod (one-to-one), neboli je přesně specifikovaný zdroj i adresát. U běžných sítí se jedná o nejpoužívanější způsob přenosu dat. V případě více příjemců stejných dat je tento způsob neefektivní[6].

### 1.4.2 Broadcast

Broadcast je využíváný několika způsoby pro procesy jako objevení sousedních uzlů, sestavení trasy a kontrolní funkce unicast, multicast i geocast relací v mobilních ad hoc sítích. Broadcast je tedy jednou ze základních operací v MANET sítích[3]. Jedná se o způsob přenosu bod-okolí (one-to-all). Odeslaný paket se šíří na (zatěžuje) všechny uzly v okolí. Mezi výhody broadcast směrování patří úspora přenosového pásma a snadná dostupnost všech stanic[6].

### 1.4.3 Multicast

Směrování multicast se využívá v případě, kdy mobilní uzel sítě MANET chce doručit zprávy dané skupině destinací. Jedná se tedy o směrování bod-skupina, skupina-skupina (one-to-many, many-to-many). Vždy se přenáší pouze jedna kopie zprávy, větvení k adresátům se řeší vytvářením kopií. Multicast směrování dosahuje vysoké efektivity při nízkém zatížení linek[6].

### 1.4.4 Geocast

Geocast je variantou multicast směrování založenou na lokaci, tedy cílová skupina je definována svojí lokací. Geocast skupina je tvořena uzly, které sídlí v dané geocast oblasti, získávající stejnou informaci od mobilního zdroje této nebo jiné oblasti. Obecně, kdykoli uzel v geocast oblasti obdrží geocast paket, multicast směrováním nebo algoritmem zaplavení tento paket směruje svým sousedním uzlům geocast skupiny či oblasti[3].

## 2 SMĚROVÁNÍ

Směrování je proces, při kterém se vybírá trasa přenosu informací v síti. Probíhá na síťové vrstvě modelu OSI<sup>1</sup>. Úkolem síťové vrstvy je nalezení vhodné cesty od adresáta k příjemci (příjemcům) přes několik mezilehlých uzlů. Při směrování se setkáváme s dvěma pojmy[5]:

- doručování paketů - přímé doručení paketů v případě, kdy zdrojová i cílová stanice jsou ve stejné síti,
- předávání paketů - postupné přeposílání paketu v řetězci směrovačů až k cílové stanici, což je považováno za směrování jako takové.

Hlavním prvkem pro směrování v síti je tzv. směrovač (z anglického router), který propojuje jednu síť s další odlišnou. Je tedy zodpovědný za doručení dat mezi různými sítěmi co nejvíce efektivním způsobem[4].

### 2.1 Popis směrování

Směrování vyžaduje spolupráci stanic a směrovačů. Stanice na začátku vytvoří pakety a ty posílá k nejbližšímu směrovači (nastavený směrovač nebo výchozí brána). Směrovač pak rozhodne o směrování, k jaké stanici či dalšímu směrovači bude tyto pakety přeposílat. Každý další směrovač provádí stejnou operaci dokud pakety nejsou doručeny k cílové stanici[4].

#### 2.1.1 Logika směrování z pohledu stanice

Po vytvoření paketu daná stanice prvně zkoumá zda cílová IP<sup>2</sup> adresa nepatří do jeho sítě (podsítě). Zkontroluje vlastní adresu a masku své sítě, vypočte rozsah sítě, následně porovná s adresou a rozsahem sítě cílového uzlu a rozhodne dle situace[4]:

1. Destinace je lokální, dojde k přímému doručení - Zjistí cílovou MAC<sup>3</sup> adresu pomocí ARP<sup>4</sup> protokolu (dotaz nebo záznam v ARP tabulce. IP paket vloží do linkového rámce s MAC adresou cílové stanice.
2. Destinace není lokální, dojde k předání paketu výchozímu směrovači - Zjistí MAC adresu výchozího směrovače pomocí ARP dotazu/záznamu. Následně

---

<sup>1</sup>Referenční model stanovující řešení komunikace v počítačových a telekomunikačních sítích pomocí vrstevnatého modelu

<sup>2</sup>Internet Protocol - jeden ze základních protokolů v informačních technologiích pracující na síťové vrstvě modelu ISO

<sup>3</sup>Unikátní fyzická adresa zařízení linkové vrstvy modelu OSI - Media Access Control

<sup>4</sup>Komunikační protokol pro zjištění linkové adresy přidružené k IP adrese - Address Resolution Protocol

vloží IP paket do linkového rámce s cílovou linkovou adresou a předá rámec výchozímu směrovači.

### 2.1.2 Logika směrování z pohledu směrovače

Směrovače jednají nezávisle, tedy směrování každého paketu vyžaduje, aby si směrovač prošel vlastní logikou bez pomoci jakéhokoliv jiného směrovače. Zároveň spolu směrovače vzájemně spolupracují ve smyslu předávání paketu dalšímu směrovači dokud nedojde k doručení cílové stanici. Oproti stanicím mají směrovače více práce při směrování paketů. Před načtením IP paketu do paměti má směrovač za úkol nejdůležitější část celého směrování - kontrola směrovací tabulky.

Pak následují kroky[4]:

1. Při každém obdržení linkového rámce se rozhodne zda daný rámec zpracuje nebo ne. Ke zpracování dojde pokud rámec neobsahuje žádné chyby (kontrola sekvence FCS<sup>5</sup> pole) a pokud cílová linková adresa není shodná s adresou směrovače (nevyžádaný broadcast či multicast).
2. Při splnění kroku 1 dojde k vypouzdření paketu z linkového rámce.
3. Uskutečnění směrovacího rozhodnutí, k čemuž se porovná cílová IP adresa paketu se směrovací tabulkou k nalezení trasy, která se shoduje s cílovou adresou. Taková trasa identifikuje rozhraní směrovače k dalšímu (sousednímu) směrovači.
4. Zapouzdření paketu do vhodného linkového rámce výstupního rozhraní. V případě předání paketu do neznáme sítě může být vyžadováno použití ARP dotazu, pokud MAC adresa dalšího směrovače není známa na aktuálním směrovači.
5. Předání rámce výstupnímu rozhraní dle vhodné trasy.

Předchozí kroky detailně popisují logiku směrování z pohledu směrovače, kde každý krok je velmi důležitý. Směrování lze ovšem popsat i jednodušeji - směrovač obdrží rámec, z něj vypouzdří paket, rozhodne kterému rozhraní paket předá, vloží paket do jiného rámce a ten odešle dále přes vybrané rozhraní[4].

## 2.2 Způsoby směrování

Tato kapitola se zabývá způsoby směrování. Konkrétních způsobů směrování existuje celá řada, od jednoduchých až po adaptivní, které se umí přizpůsobit provozu v síti (zatížení, výpadky spojů a další). Mechanismus směrování paketů je závislý zejména na topologii sítě z hlediska redundance přenosových linek.[5] Z pohledu topologie existují dva extrémy - stromové sítě a úplný polygon. V případě úplného

---

<sup>5</sup>Detekční kód chyb komunikačních protokolů - Frame Check Sequence.

polygonu existuje pro každé dva uzly přímé spojení. U stromových sítí se pro různou dvojici uzlů definuje právě jedna cesta. Směrovací funkce určují cestu mezi síťovými adresami. Základní atributy směrovacích technik a protokolů jsou[5]:

- výkonnostní kritéria (množství uzlů, náklady, zpoždění a propustnost),
- rozhodovací čas (pro datagramy, virtuální obvody),
- rozhodovací místo (každý uzel, tj. distribuovaně; centrální uzel, tj. centralizovaně),
- zdroje informací o síti (žádné, místní, připojené uzly, všechny uzly),
- směrovací techniky (pevné, lavinovité, nahodilé, adaptivní),
- časová aktualizace adaptivního směrování (průběžné, periodické, hlavní změny zátěže, změny topologie).

Jednotlivé způsoby (strategie) směrování jsou potom[5]:

- statické - manuálně nastavené linky,
- dynamické (náhodné) - ke směrování využívají určitý směrovací protokol.

Výhody a nevýhody obou způsobů jsou shrnuty v Tab.2.1 [6].

Tab. 2.1: Porovnání statického a dynamického směrování

Charakteristiky	Statické	Dynamické
Konfigurace	Směrovací informace nakonfigurovány ručně	Automatická výměna směrovacích informací mezi směrovači (směrovací protokoly)
Využití	Vhodné pro malé sítě, koncové sítě (gateway of last resort)	Vhodné pro větší a velké sítě, průchozí (tranzitní) sítě
Výhody	Žádný režijní provoz na linkách	Automaticky reaguje na změny v síti, výpadky linek
Nevýhody	Při každé změně nutný zásah administrátora	Režijní provoz na linkách (výměna směrovacích informací), zatížení směrovačů

## 2.3 Algoritmy směrování

Jak bylo zmíněno v předchozí kapitole 2.2, jsou definovány dva základní způsoby směrování - statické a dynamické. Statické směrování oproti dynamickému nevyžaduje žádný algoritmus. U dynamického směrování rozlišujeme algoritmy - vektorové a stavové. Jejich popisu se věnují následující podsekcce[6].



### 2.3.1 Vektorové algoritmy

Vektorové algoritmy jsou založeny na Bellman-Fordovu<sup>6</sup> algoritmu. Tento přístup určuje tzv. "cenu cesty" každé linky mezi uzly v síti. Při komunikaci mezi dvěma uzly je vždy zvolena linka s nejnižším počtem skoků, tj. nejnižší cenou cesty. Algoritmus funguje velmi jednoduše. Každý uzel zná pouze přímé sousední uzly a cenu cesty ke každému z nich. Tento seznam vzdáleností utváří směrovací (vzdálenostní) tabulku. Každý uzel pravidelně odesílá sousedním stanicím jeho aktuální ohodnocení ceny cest ke každé jeho známe destinaci. Sousední uzly tyto informace porovnají s jejich dosavadními a případně upraví své směrovací tabulky. Během krátké doby objeví všechny stanice v síti nejkratší cesty ke všem potřebným destinacím. Vektorové protokoly jsou vhodné pro použití uvnitř sítě (autonomního systému<sup>7</sup>). Každý směrovač odesílá svým sousedům svoji směrovací tabulku při každé aktualizaci. Algoritmus může být náchylný k vytváření smyček, ale je výpočetně jednodušší než stavový algoritmus. Příkladem vektorových algoritmů jsou protokoly RIP<sup>8</sup> a IGRP<sup>9</sup>, jejich popis je ale nad rámec této práce[4].

### 2.3.2 Stavové algoritmy

V případě stavových algoritmů je pro každý uzel nejdůležitější znalost topologie jeho sítě. K dosažení této znalosti zahrnuje každý uzel celou síť informací, ke kterým uzlům se může připojit. Každý uzel na základě těchto informací nezávisle utváří mapu sítě. S využitím této mapy může každý směrovač zvolit cestu o nejnižší ceně k jakémukoli uzlu v síti. Tento algoritmus, určující nejkratší cestu, je založen na Dijkstrově algoritmu<sup>10</sup>. Výsledkem je stromový graf s kořenem daného uzlu (směrovače) a dostupnými všemi destinacemi v síti o nejnižší ceně cesty. Na základě tohoto stromu (grafu) je vytvořena směrovací tabulka. Vytváří detailní databázi linek (podsítí) a jejich stavu (dostupnost), ze které kalkulují nejvhodnější trasy. Příkladem stavového algoritmu je shortest path first - algoritmus používaný protokolem OSPF<sup>11</sup> k nalezení všech možných tras a následné vybrání/zvolení trasy s nejnižší cenou cesty pro každou podsít[4].

---

<sup>6</sup> Algoritmus počítající nejkratší trasu mezi uzly v grafu.

<sup>7</sup> Autonomní systém představuje souhrn sítí pod společnou správou, kde se používá společná (vnitřní) směrovací strategie.

<sup>8</sup> Vektorový směrovací protokol - Routing Information Protocol

<sup>9</sup> Vektorový směrovací protokol firmy Cisco pro užití uvnitř autonomních systémů - Interior Gateway Routing Protocol

<sup>10</sup> Algoritmus počítající nejkratší trasu mezi uzly v grafu.

<sup>11</sup> Stavový směrovací protokol určený pro užití uvnitř autonomního systému - Open Shortest Path First

### 3 SMĚROVACÍ PROTOKOLY PRO MOBILNÍ SÍTĚ

V posledních několika dekádách se ve výpočetním průmyslu zvyšuje oblíbenost bezdrátových sítí. Platí to obzvláště v posledním desetiletí, kdy se počítačový uživatelé přesunuli od stolních počítačů k mobilním laptopům[7].

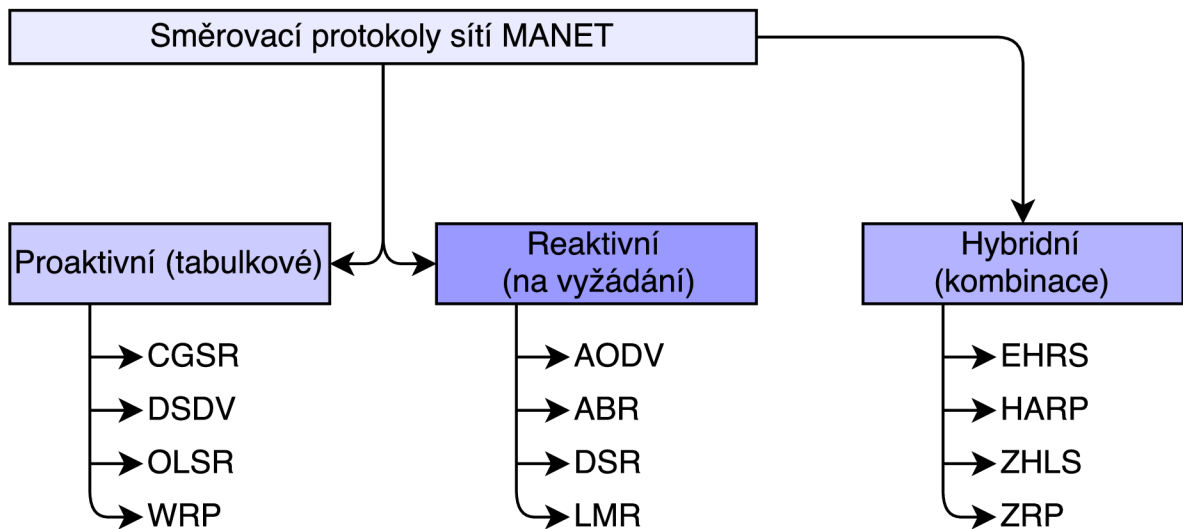
Aktuálně rozlišujeme dvě varianty mobilních bezdrátových sítí. První z nich je infrastrukturní síť, tedy síť s pevnou a kabelovou bránou. Mosty mezi těmito sítěmi jsou známy jako základny (z anglického base station). Mobilní stanice se uvnitř sítě připojuje a komunikuje vždy s nejbližší základnou ve svém okolí. V případě pohybu stanice mimo dosah aktuální základny k jiné dojde k přepojení od předchozí základny k "nové" a stanice je schopna pokračovat v komunikaci skrz síť. Typickým příkladem této varianty je bezdrátová lokální síť (WLAN) v kancelářích a domácnostech[7].

Druhý typ mobilní bezdrátové sítě je bez infrastrukturní (decentralizovaná) mobilní síť, známá jako ad hoc síť (MANET). Bez infrastrukturní sítě nemají stanovené žádné pevné směrovače, všechny uzly jsou schopny pohybu a mohou být dynamicky propojeny libovolným způsobem. Uzly těchto sítí pracují jako směrovače, objevují a spravují trasy k jiným uzlům v síti. Příkladem bez infrastrukturních mobilních sítí jsou aplikace pro konference k rychlému sdílení informací či sběr informací v nepřístupném terénu[7].

Od začátku rozvoje paketových rádiových sítí (70. léta 20. století) bylo vyvinuto několik protokolů pro ad hoc mobilní sítě. Tyto protokoly musí operovat s typickými limitacemi těchto sítí jako je vysoká spotřeba elektrické energie, nízká šířka pásma a vysoká míra chyb. Jak popisuje Obr.3.1 [7], protokoly jsou rozděleny do dvou hlavních kategorií[7]:

- proaktivní (tabulkové),
- reaktivní (na vyžádání).

Dále existuje ještě třetí kategorie směrovacích protokolů pro mobilní sítě - hybridní. Jedná se o kombinaci proaktivních a reaktivních protokolů. Následující sekce budou věnovány těmto kategoriím s uvedenými příklady některých protokolů.



Obr. 3.1: Rozdělení ad hoc směrovacích protokolů dle kategorií

## 3.1 Proaktivní

Proaktivní, nebo také tabulkové, směrovací protokoly se snaží udržovat konzistentní aktuální směrovací informace od každého uzlu ke všem ostatním v rámci sítě. Tyto protokoly vyžadují, aby si každý uzel udržoval jednu nebo více tabulek k uložení směrovacích informací, které budou reagovat na změny topologie sítě šířením aktualizací napříč sítí s cílem udržet konzistenci zobrazení sítě. Oblasti, kterými se liší, jsou číslem udávajícím počet nezbytných směrovacích tabulek a metod šíření změn v síťové struktuře[7]. Následující podsekcce popisují některé existující tabulkové ad hoc směrovací protokoly. Jedním z reaktivních protokolů je také OLSR<sup>1</sup>, tomu je ale věnována samostatná kapitola 4.

### 3.1.1 DSDV

Směrovací protokol DSDV<sup>2</sup> pracuje jako tabulkový algoritmus, je založený na klasickém Bellman-Fordovu směrovacím mechanismu. Vylepšení oproti Bellman-Fordovu algoritmu spočívá v osvobození od smyček ve směrovacích tabulkách. Každý mobilní uzel v síti udržuje směrovací tabulku, ve které jsou zaznamenány všechny možné destinace v síti s počtem skoků ke každé z nich. Každý záznam je označen sekvenčním číslem přiděleným cílovým uzlem. Sekvenční čísla dovoluují mobilním uzlům

<sup>1</sup>Proaktivní stavový směrovací protokol sítě MANET - Optimized Link State Routing Protocol

<sup>2</sup>Distančně-sekvenční vektorový směrovací protokol sítě MANET - Destination-Sequenced Distance-Vector Routing

rozlišit staré (neplatné) trasy od nových, tím se předchází utváření smyček. Aktualizace směrovací tabulky jsou přenášeny periodicky sítí k udržení konzistence[7].

Ke snížení případného zatížení sítě těmito směrovacími aktualizacemi mohou být zvoleny dva typy paketů. Prvním je tzv. plná sada (hromada). Takovýto paket nese všechny dostupné směrovací informace a může vyžadovat několikanásobné datové jednotky (NPDUs<sup>3</sup>). V případě občasného pohybu jsou tyto pakety vysílány jen zřídka. Druhým typem jsou inkrementální pakety. Ty jsou využity k přenosu pouze těch informací, které se změnilo od poslední "plné sady". Na každý z těchto přenosů by měla postačit standardní datová jednotka (NPDU). Tím se sníží rozsah generovaného provozu. Mobilní uzly udržují také dodatečnou tabulku, kde uchovávají data poslané inkrementálními pakety[7].

### 3.1.2 WRP

Cílem WRP<sup>4</sup> protokolu je udržování směrovacích informací mezi všemi uzly sítě. Každý uzel v síti je zodpovědný za správu čtyř tabulek[7]:

- tabulka vzdálenosti,
- tabulka směrování,
- tabulka cen jednotlivých cest,
- tabulka (seznam) opakovaných přenosů zpráv.

Uzly se navzájem informují o každé změně linky aktualizacími zprávami. Ty jsou posílány pouze mezi sousedícími uzly a obsahují seznam aktualizací (destinace, vzdálenost k ní a předchůdce této destinace) a seznam odpovědí, značící které uzly tuto zprávu mají potvrdit (ACK<sup>5</sup>). Uzly posílají aktualizací zprávy po zpracování aktualizací od sousedních uzlů nebo po zjištění změny linky k sousedním uzlům. V případě ztráty linky mezi dvěma uzly odesílají oba uzly aktualizací zprávy svým sousedům. Ti následně změni své tabulky vzdáleností a hledají nové možné trasy přes okolní stanice. Jakékoli nové trasy jsou poslány zpět oběma uzlům za účelem aktualizace jejich tabulek[7].

---

<sup>3</sup>Datová jednotka síťového protokolu - Network Protocol Data Unit

<sup>4</sup>Proaktivní vektorový směrovací protokol sítě MANET - Wireless Routing Protocol

<sup>5</sup>Informační zpráva telekomunikačních protokolů značící potvrzení - Acknowledgement

## 3.2 Reaktivní

Reaktivní protokoly se od proaktivních liší v přístupu ke směrování. Trasy jsou vytvořeny pouze na vyžádání určitého uzlu. Pokud uzel vyžaduje trasu do určité destinace, spustí proces vyhledávání tras v síti. Proces je dokončen v případě nalezení potřebné trasy nebo po prozkoumání všech možných tras. Pokud je nějaká trasa stanovena, je udržována speciální procedurou dokud je destinace dostupná anebo dokud přestane být potřebná.

### 3.2.1 AODV

Protokol AODV<sup>6</sup> je postaven na dříve popsaném algoritmu DSDV, v podsekcí 3.1.1. Mechanismus DSDV vylepšuje tím, že minimalizuje počet potřebných vysílání vytvářením přenosových cest na vyžádání, oproti udržování všech přenosových cest. Pokud chce uzel A komunikovat s uzlem B a nemá k němu platnou cestu, spustí proces vyhledávání tras k lokaci uzlu B. Začne tedy všesměrově vysílat požadavek trasy (RREQ<sup>7</sup>) sousedním uzlům dokud není lokalizován uzel B nebo uzel s platnou trasou k uzlu B. AODV stejně jako DSDV využívá sekvenční číslování destinací k zajištění toho, aby žádná přenosová cesta nevytvářela smyčky. Každý uzel si udržuje své sekvenční číslo a všesměrový identifikátor (ID). Toto ID je inkrementováno pro každý požadavek trasy (RREQ).

---

<sup>6</sup>Reaktivní vektorový směrovací protokol sítí MANET - Ad hoc On-Demand Distance Vector Routing

<sup>7</sup>Požadavek o nalezení trasy protokolu AODV - Route Request

## 4 OLSR

Tato kapitola popisuje optimalizovaný stavový směrovací protokol pro ad hoc sítě - OLSR. Protokol je optimalizací klasického stavového algoritmu přizpůsobenému požadavkům mobilních bezdrátových LAN sítí. Klíčový koncept protokolu spočívá ve vícebodovém spojení (MPR<sup>1</sup>). MPR jsou vybrané uzly, které přeposílají všesměrové zprávy během procesu zahlcení. Tato technika značně redukuje režii zpráv oproti klasickým mechanismům zahlcení, kde každý uzel přeposílá každou zprávu při obdržení její první kopie. U OLSR je stavová informace generována pouze uzly zvolenými jako MPR. Druhou optimalizací je dosažení minimalizace počtu kontrolních zpráv zahlcujících síť. Třetí optimalizací je to, že si MPR uzel může vybrat ohlašování pouze stavu linek mezi ním a uzly, které si jej jako MPR vybrali. Dále je oproti klasickému stavovému algoritmu distribuována pouze částečná stavová informace, která je později použita pro kalkulaci směrování. OLSR poskytuje optimální trasy ve smyslu nejnižšího počtu skoků. Protokol je vhodný i pro velké a husté sítě díky dobré funkcionalitě vícebodových přenosů (MPR)[8].

OLSR je vyvinut pro mobilní ad hoc sítě. Funguje jako proaktivní (tabulkový) protokol - dochází k pravidelné výměně informací o topologii s ostatními uzly sítě. Každý uzel si vybere sadu sousedních uzlů jako MPR. U OLSR jsou pouze uzly vybrané jako MPR zodpovědné za předávání kontrolních zpráv, určených k šíření po celé síti. MPR uzly tak poskytují efektivní mechanismus pro zahlcování sítě kontrolními zprávami snížením počtu potřebných přenosů[8].

Uzly MPR mají zároveň speciální zodpovědnost při odmítnutí/neposkytnutí stavové informace v síti. Ve skutečnosti jediný požadavek pro OLSR na poskytnutí nejkratších tras na všechny destinace je řešen neposkytnutím stavových informací uzlům určujícím MPR. Dodatečné stavové informace mohou být využity pro redundanci[8].

Uzly zvolené jako MPR svými sousedními uzly oznamují pravidelně tuto informaci v jejich kontrolních zprávách. Tímto je oznámeno síti, že MPR uzel má dostupné právě ty sousední uzly, které ho jako MPR zvolili. Při kalkulaci trasy se uzly MPR využívají k formování trasy od daného počátečního uzlu k jakékoliv destinaci v síti[8].

OLSR je vyvinut tak, aby pracoval nezávisle na rozdíl od jiných protokolů. OLSR zároveň neprovádí žádné předpoklady na základě nižší linkové vrstvy[8].

OLSR dědí koncept přeposílání a přenášení od protokolu HiperLAN<sup>2</sup>, který je standardizován ETSI<sup>3</sup>. Protokol je vyvíjen projekty IPANEMA (část programu EUCLID) a PRIMA (část programu RNRT)[8].

---

<sup>1</sup>Vícebodové spojení/přenos protokolu OLSR - Multipoint Relay

<sup>2</sup>WLAN standard, protokol podvrstvy MAC linkové vrstvy - High Performance Radio LAN

<sup>3</sup>Evropský ústav pro telekomunikační normy - European Telecommunications Standards Institute

## 4.1 Použitelnost

Protokol se dobře hodí k velkým a hustým mobilním sítím díky využití vícebodového spojení (MPR). Čím větší a hustější síť, tím lepší optimalizace lze dosáhnout oproti klasickým stavovým algoritmům. OLSR používá směrování "hop-by-hop" (krok po kroku). To znamená, že každý uzel využívá aktuální lokální informace ke směrování paketů[8].

OLSR je vhodný více pro síť, kde přenos je náhodný a sporadický mezi spoustou uzlů než pro malou topologii specifických uzlů. Jako proaktivní protokol je zároveň vhodný také ve scénáři, kde se komunikující dvojice uzlů časem mění. Nepřibývá žádné kontrolní dopravy, jelikož trasy jsou udržovány pro všechny distance v jakýkoliv čas[8].

## 4.2 Obecný popis

OLSR je proaktivní směrovací protokol pro mobilní ad hoc síť. Dědí stabilitu stavového algoritmu a díky své proaktivitě má výhodu okamžitě dostupných tras kdykoliv je potřeba. OLSR je oproti klasickým stavovým protokolům optimalizovaný pro mobilní ad hoc síť[8].

Protokol může vylepšit reaktivitu na změnu topologie snížením doby intervalu přenosu kontrolních zpráv. Krom toho, že OLSR neustále spravuje trasy do všech destinací v síti, je zároveň užitečný při konfiguraci, kde rozsáhlá skupina uzlů komunikuje s podobně rozsáhlou množinou a kde se páry zdroj-cíl pravidelně mění. Protokol je vhodný pro velké husté síť, kde skvěle pracují prvky MPR a čím je daná síť větší, tím lepší efektivity je možné dosáhnout oproti klasickému stavovému algoritmu[8].

Protokol je stavěn tak, aby pracoval kompletně distribuovaným způsobem a nebyl závislý na žádné centrální entitě. Protokol nevyžaduje spolehlivý přenos kontrolních zpráv. Každý uzel posílá kontrolní zprávy periodicky a tudíž může ustát přiměřenou ztrátovost kontrolních zpráv. Tyto zprávy probíhají často v rádiových sítích díky kolizím a dalším přenosovým problémům[8].

OLSR také nevyžaduje sekvenční doručení zpráv. Každá kontrolní zpráva obsahuje sekvenční číslo, které je pro každou zprávu inkrementováno. Adresát kontrolní zprávy může v případě nutnosti identifikovat, která informace je aktuální i kdy se pořadí zpráv během přenosu změní[8].

Dále pak OLSR nabízí rozšíření jako je režim spánku, multicast směrování a další. Podobná rozšíření mohou být uvedeny jako doplňky protokolu bez narušení zpětné kompatibility s předchozími verzemi[8].

Protokol nevyžaduje žádné změny formátu IP paketů, tudíž jakákoliv existující síť založená na IP protokolu může využít OLSR aniž by byla potřeba změna v její konfiguraci. Protokol řeší pouze náležitosti směrování[8].

### 4.3 MPR

Myšlenkou vícebodového spojení je minimalizovat režii zahlcení sítě zprávami zredukováním redundantního přeposílání (opětovného posílání) v dané části sítě. Každý uzel (U) sítě si vybírá skupinu (sadu) uzlů v jeho symetrickém jedno-skokovém okolí, která může jeho zprávy přeposlat. Tato skupina vybraných sousedů je označena jako vícebodové spojení tohoto uzlu - MPR(U). Sousedé tohoto uzlu, kteří nejsou v jeho skupině MPR obdržují a zpracovávají jeho broadcast zprávy, ale tyto zprávy nikam nepřeposílají[8].

Každý uzel si vybírá svoji sadu MPR z jeho jedno-skokového okolí. Tato sada je zvolena takovým způsobem, aby pokryla všechny symetrické dvou-skokové uzly (sousedy). MPR sada uzlu U, označena jako MPR(U), je tedy náhodná skupina jedno-skokově vzdálených sousedů uzlu U a musí splňovat následující podmínku - každý uzel dvou-skokového okolí uzlu U musí mít linku k MPR(U). Čím menší je sada MPR, tím menší je výsledná kontrolní režie směrovacího protokolu[8].

Každý uzel si udržuje informace o skupině sousedů, které si jej zvolili jako MPR. Skupina se nazývá "vybraná sada vícebodového spojení" daného uzlu[8].

### 4.4 Struktura protokolu

Tato sekce se věnuje popisu celkové funkčnosti protokolu OLSR. Protokol je modulární v jádru funkčnosti, což je vždy vyžadováno pro správnou funkčnost protokolu a jeho sadu podpůrných funkcí. Jádro specifikuje schopnost protokolu provádět směrování v rámci samostatné MANET sítě[8].

Každá podpůrná funkce poskytuje dodatečnou funkčnost, která je aplikovatelná ve specifických scénářích. Například v případě, že uzel poskytuje konektivitu mezi sítí MANET a odlišně směrovanou sítí.

Všechny podpůrné funkce jsou kompatibilní do té míry, kdy jakákoliv sada podpůrných funkcí může být implementována k jádru. Mimo jiné protokol povoluje heterogenní uzly. Tedy uzly, které implementují odlišné sady podpůrných funkcí ke společné koexistenci v síti[8].

Účel rozlišení/dělení fungování OLSR v jádru funkčnosti a sadě podpůrných funkcí je k poskytnutí jednoduchého, snadně pochopitelného protokolu a způsobu pouhého přidávání složitostí tam, kde je specifická dodatečná funkčnost vyžadována[8].



## 4.4.1 Struktura jádra protokolu

Jádro funkčnosti OLSR specifikuje chování uzlu, vybaveného OLSR rozhraním, který je součástí sítě MANET a využívá OLSR směrování. To zahrnuje univerzální specifikaci OLSR zpráv a jejich přenos přes síť stejně jako sledování stavu linky, rozptýlení topologie a kalkulace tras.

Jednotlivé komponenty, ze kterých se skládá jádro OLSR protokolu jsou v následující Tab.4.1 <sup>4 5</sup> [8]:

Tab. 4.1: Tabulka základní komponent jádra OLSR protokolu

Komponenty	Popis
Formát paketu a přeposílání	Specifikace paketového formátu a optimalizovaného mechanismu zahlcení sloužící jako transportní mechanismus pro veškerou kontrolní režii protokolu.
Naslouchání linky	Naslouchání linky je splněno při periodickém vysílání zprávy typu "HELLO" na všechna rozhraní uzlu, dochází k ověřování konektivity.
Detekce sousedního uzlu	V případě sítě se zařízeními o jednom rozhraní může uzel detekovat sousední stanici při naslouchání linky. U zařízení s více rozhraními je potřeba dodatečná informace k detekci sousedního uzlu hlavní adresou uzlu aktuálního. Dodatečnou informací je zpráva "MID".
Volba a signalizace MPR	Musí být splněna podmínka MPR pro každé rozhraní - každý uzel dvou-skokového okolí uzlu U musí mít linku k MPR(U). Ke zvolení sady MPR dojde po několika výměnách zpráv "HELLO".
Šíření kontrolních zpráv	Kontrolní zprávy se šíří za účelem poskytnutí každému uzlu v síti dostatečné stavové informace a následnému umožnění kalkulace cest.
Kalkulace cest	S danými stavovými informacemi dosaženými periodickou výměnou zpráv a vhodnou konfigurací rozhraní je uzel schopen sestavit směrovací tabulku.

Klíčovým prvkem těchto mechanismů je MPR vztah.

<sup>4</sup>HELLO - Zprávy vykonávající úlohu naslouchání linky, detekci sousedního uzlu a signalizace MPR spojení

<sup>5</sup>MID - Zprávy MID obsahují seznam všech jiných adres rozhraní uzlu než je to hlavní (OLSR)

## 4.4.2 Podpůrné funkce

Jako doplněk k funkčnosti jádra OLSR slouží situace, kdy je požadována dodatečná funkčnost. To zahrnuje situace, kdy má uzel více rozhraní, z nichž některé slouží ke komunikaci v odlišně směrované síti, kde programovací rozhraní síťovému hardwaru poskytuje dodatečnou informaci ve formě notifikace linkovou vrstvou a kde je požadováno poskytnutí redundantní topologické informace síti na úkor režie protokolu[8]. V následující Tab.4.2 <sup>6</sup> jsou popsány jednotlivé podpůrné funkce jádra protokolu OLSR [8].

Tab. 4.2: Tabulka podpůrných funkcí jádra protokolu OLSR

Podpůrné funkce	Popis
Jiná než OLSR rozhraní	Uzel může být vybaven více rozhraními, některé nemusí být OLSR. Takové rozhraní mohou mít spojení bod-bod (point-to-point) k jiným uzlům nebo jiným sítím. Za účelem poskytnutí spojení s OLSR MANET rozhraním by měl být uzel schopen poskytnout externí směrovací informace síti OLSR MANET síti.
Notifikace linkové vrstvy	OLSR byl vyvinut tak, aby neočekával žádné informace od linkové vrstvy. Nicméně, pokud je informace od linkové vrstvy dostupná (například porucha linky), uzel tuto informaci může využít.
Pokročilé naslouchání linky	Sestavené linky by měli být maximálně možně spolehlivé, aby nedocházelo ke ztrátě dat. To znamená, že naslouchání linky stabilní ve smyslu zjištění ztráty linky nebo její přechodné dostupnosti.
Redundantní topologie	Za účelem poskytnutí redundance topologie, oznamovaná sada linek k sousedním uzlům může obsahovat i ty uzly, které nejsou zvoleny jako MPR. S využitím parametru "TC_REDUNDANCY" může být minimální sada linek, kterou sdílí každý uzel ve svých kontrolních zprávách svým MPR rozšířena na kompletní seznam sousedních uzlů.
Redundantní zahlcení MPR	Schopnost uzlu zvolit si redundantní uzly MPR. Dochází k ovlivnění režie kontrolních zpráv, efektivity mechanismu zahlcení, ale k navýšení počtu poskytnutých linek.

<sup>6</sup>TC\_REDUNDANCY - kontrolní zpráva protokolu OLSR odesílaná zvoleným MPR rozšířená i o sousední uzly nezvolené jako MPR

### 4.4.3 Formát paketu a přeposílání

OLSR komunikuje použitím jednotného paketového formátu pro veškeré relace protokolu. Účelem sjednocení formátu paketů je usnadnění rozšiřitelnosti protokolu bez narušení zpětné kompatibility. To zároveň nabízí jednoduchý způsob tzv. "piggybackingu"<sup>7</sup> různých druhů informací do jednotného přenosu a tudíž pro danou implementaci dojde k optimalizaci ve využití maximální velikosti rámce nabízeného sítě. Tyto pakety jsou zakotvené v UDP<sup>8</sup> datagramech pro přenos po síti. OLSR v základu využívá IP protokol verze 4[8].

Každý paket zapouzdřuje jednu nebo více zpráv. Tyto zprávy následně sdílejí společný formát záhlaví, který dovoluje uzlům korektně přijímat a případně i přeposílat zprávy neznámého typu[8].

Zprávy mohou zaplavit celou síť, případně je zaplavení zprávami omezeno na uzly v rámci daného rozsahu (ve smyslu počtu skoků) od původního odesílatele. Odesílání zpráv okolí uzlu je tedy speciální případ zaplavení (zahlcení). Při záplavě kontrolní zprávou se přeposílání duplikátu eliminuje lokálně (tj. každý uzel spravuje duplicitní sadu, aby zabránil opětovnému přenosu stejné OLSR kontrolní zprávy) a tím se minimalizuje využití MPR spojení v celé síti[8].

Dále pak může uzel prozkoumat záhlaví zprávy k získání informace o vzdálenosti (počtu skoků) k odesílateli zprávy. Tato vlastnost může být užitečná například v situacích, kdy informace o čase v obdržené kontrolní zprávě uložená v uzlu závisí na vzdálenosti k odesílateli[8].

Pakety v OLSR jsou přenášeny za pomoci UDP protokolu. Protokolu OLSR byl organizací IANA<sup>9</sup> přiřazen port 698[8].

OLSR definuje, že pro uzel s jediným rozhraním musí být jako hlavní adresa nastavena adresa tohoto rozhraní[8].

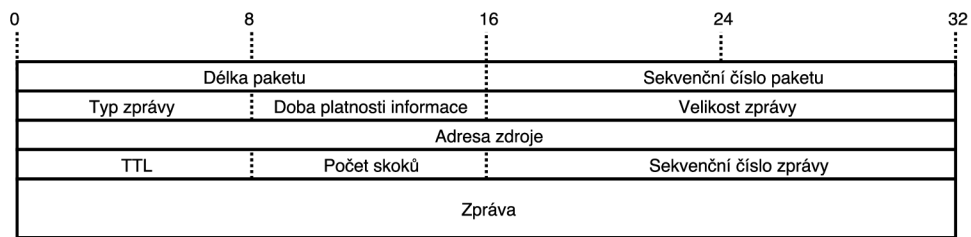
---

<sup>7</sup>Piggybacking - technika dočasného zdržení potvrzení obdrženého paketu dokud cílová stanice nebude potřebovat komunikovat s původním odesílatel. Obdoba posuvného okna v modelu OSI.

<sup>8</sup>Jednoduchý transportní protokol umožňující nespojovaný a nespolehlivý přenos dat - User Datagram Protocol

<sup>9</sup>Organizace dohlížející celosvětově na přidělování IP adres, správu kořenových zón DNS a další náležitosti internetových protokolů - Internet Assigned Numbers Authority

Základní rozložení OLSR paketu popisuje následující Obr.4.1.



Obr. 4.1: Formát paketu protokolu OLSR

Jednotlivá pole značí[8]:

- Délka paketu - délka paketu v bajtech.
- Sekvenční číslo paketu - inkrementované při každém odeslání (přeoslání) paketu.
- Typ zprávy - typ zprávy v datovém poli.
- Doba platnosti informace - značí jak dlouho od obdržení zprávy musí být informace v ní považována za platnou.
- Velikost zprávy - velikost zprávy v bajtech (měří se od pole velikosti zprávy daného paketu až po pole velikosti zprávy následujícího paketu, v případě posledního paketu jen do jeho konce).
- Adresa zdroje - hlavní adresa uzlu odesílatele zprávy.
- TTL (Time to live) - maximální počet skoků během přenosu paketu.
- Počet skoků - aktuální počet skoků, hodnota se zvyšuje před každým přeposláním paketu.
- Sekvenční číslo zprávy - unikátní identifikátor zprávy přidělen odesílatelem.

## IPv6

Veškeré operace a parametry protokolu OLSR pro verzi 4 IP protokolu jsou stejné jako pro verzi 6 protokolu IP. Jediná vyžadovaná změna je u IP adresy. Minimální velikost paketu a zpráv, pod kterou dochází k odmítnutí, by měla být přizpůsobena na základě větší velikosti IPv6 adresy oproti adrese IPv4[8].

### 4.4.4 Informační schránka

Na základě výměny OLSR kontrolních zpráv, každý uzel shromažďuje informace o síti a po určitou dobu si je ukládá. Každá destinace v síti je uložena pod trojicí[8]: [adresa rozhraní | hlavní adresa uzlu | čas do expirace záznamu].

Uzel tuto trojici informací označuje jako "sada připojených rozhraní".

## 4.5 OLSRv2

OLSR verze 2 je nástupce klasického OLSR protokolu, zachovává základní mechanismy a algoritmy rozšířené o schopnost využít jinou metriku než počet skoků při výběru nejkratší cesty. OLSRv2 taky využívá více flexibilní a efektivní signalizační rámce a také zahrnuje určitá zjednodušení výměny zpráv[10].

Každý uzel si vybírá dvě sady MPR, každá zahrnuje sousední uzly, které pokrývají všechny symetrické dvou-skokové sousedy. Tyto dvě sady jsou "zaplavující sada MPR" a "směrovací sada MPR", které se využívají pro redukci zaplavení sítě a redukci topologie, v tomto pořadí[10].

Redukce zaplavení sítě je dosaženo tím, že kontrolní provoz je posílán sítí přeposíláním hop-by-hop (skok po skoku) uzlem přeposílajícím kontrolní zprávu, která je obdržena přímo od uzlu zvoleného jako zaplavující MPR. Tento mechanismus poskytuje efektivní možnost distribuce informací v rámci sítě MANET redukcí počtu potřebných přeposílání[10].

Redukce topologie je dosaženo přidělením zvláštní zodpovědnosti uzlům zvoleným jako směrovací MPR při stanovení stavové informace. Dostačujícím požadavkem pro OLSRv2 k poskytnutí nejkratší směrovací trasy ke všem destinacím je, že uzly stanovují stavovou informaci k jejich směrovacím MPR, pokud nějaké má. Uzly, které nejsou vybrány jako směrovací MPR nepotřebují odesílat stavovou informaci. Na základě tohoto omezení stavových informací jsou směrovací MPR užity jako prostřední uzly ve víceskokových trasách[10].

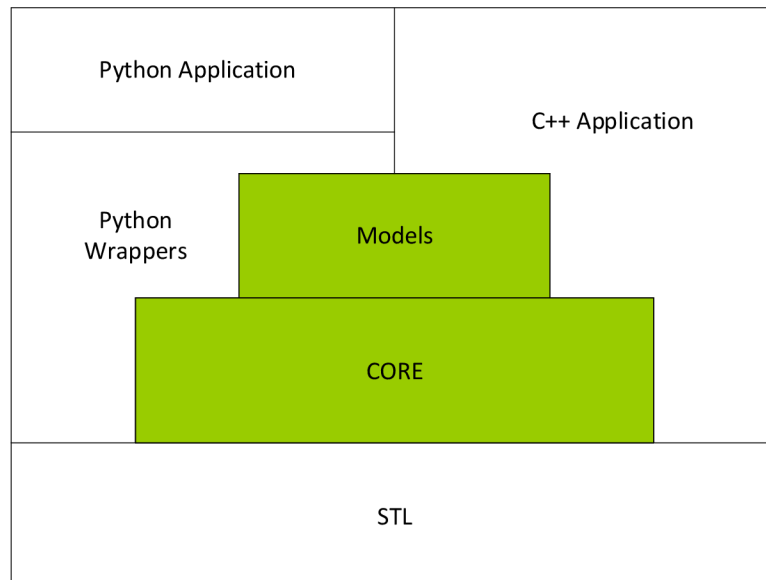
Využití MPR sad tedy umožní redukci počtu a velikosti stavových zpráv udržovaných každým uzlem. Existuje i možnost (v případě užití metriky počtu skoků), že budou zvoleny stejné uzly jako směrovací i zahlcovací MPR[10].

Uzel si vybere sadu směrovacích a zaplavujících MPR z okruhu svých symetrických jedno-skokových sousedů (oboustranné linky). Výběrem tras přes směrovací MPR se tedy vyhneme problémům spojeným s přenosem paketů přes jednosměrné linky (nepotvrzování na úrovni linkové vrstvy při každém skoku)[10].

OLSRv2 využívá a rozšiřuje NHDP (Neighbour Discovery Protocol - RFC6130) sítě MANET a také používá standardizovaný paketový formát MANET sítě (RFC5444). Dále využívá specifikaci TLV (RFC5497) a případně také kolísání zpoždění (jitter - RFC5148). Tyto 4 odlišné specifikace byly původně vytvořeny jako součást OLSRv2, ale následně specifikovány zvlášť pro širší využití[10].

## 5 NS-3

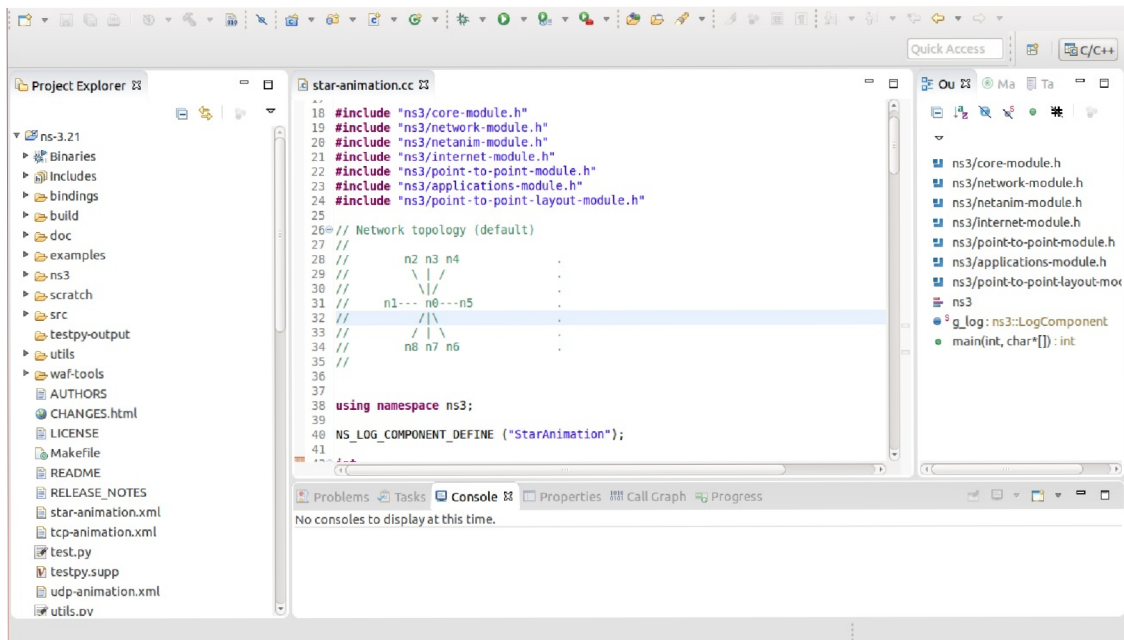
NS-3 (Network Simulator) je tzv. „discrete-event“ simulátor pro síťové systémy. Je volně šiřitelný, určený především pro akademické účely. Byl vytvořen za účelem vývoje a výzkumu různých sítí a jejich možností, které jsou tímto simulátorem ověřeny a následně využívány. Užitím simulátoru dosáhneme kvalitní simulace, s vytvořeným projektem se dá jednoduše pracovat, nastavovat určité parametry, jako používané standardu a technologie, prvky v síti a následným odsimulováním vyzkoušet případnou funkčnost. NS-3 podporuje oba síťové protokoly IPv4 i IPv6, ale také umožňuje síť bez využití IP protokolu. Simulátor je knihovnou C++, který poskytuje sadu síťových simulačních modelů implementovaných jako objekty C++ zabalených do Python [9].



Obr. 5.1: Technologie softwaru NS-3

Uživatel může standardně využívat knihovnu vyvíjením aplikace C++ nebo Python a používat sadu modelů k vytvoření simulačního scénáře. Knihovna je zabalena do Python díky knihovně pybindgen, kterou zastupuje a rozebírá hlavičku C++ do gccxml a pygccxml pak automaticky vygeneruje C++. Tyto C++ generované soubory jsou následně kompilovány do NS-3 Python modulu a povolí uživateli interakci s modely C++ NS-3 a jádrem pomocí skriptů v jazyce Python. Spousta simulátorů používá určitý svůj specifický jazyk k práci s modely a programem. NS-3 využívá C++ nebo Python, čímž má uživatel výhodu plné podpory každého z jazyků. Uzly vytvořené v NS-3 jsou odvozeny z Linuxové síťové architektury a rozhraní, objekty

jsou pak stejné jako ty v počítači s Linux operačním systémem. Simulace je potom možno považovat za reálnou a je následně jednodušší odvodit případné možnosti v praxi. Ve srovnání s jinými „discrete-event“ síťovými simulátory, jako například OMNET++, OPNET Modeler nebo NS-2, se NS-3 odlišuje výslednými cíli návrhu na vysoké úrovni [9].



Obr. 5.2: Prostředí Eclipse s otevřeným příkladem NS-3

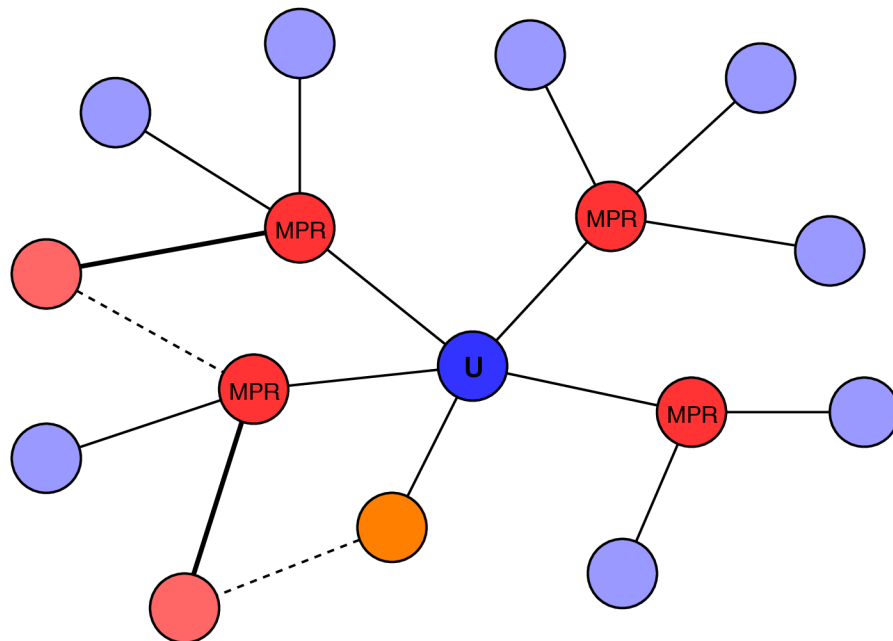
Obecný postup vytvoření simulace můžeme rozdělit do několika bodů [9]:

- definice topologie,
- použití modelu,
- konfigurace uzlů a linek,
- překlad,
- analýza výkonnosti,
- vizualizace.

## 6 VÝSLEDKY STUDENTSKÉ PRÁCE

V této kapitole bude detailně popsáno proramové řešení a výsledky simulace diplomové práce.

Diplomová práce je zaměřena na směrovací protokol OLSR optimalizovaný pro síť MANET. Hlavním konceptem tohoto směrovacího protokolu je využití vícebodových spojení MPR, díky kterým si uzel zajistí spojení s dvou-skokovými sousedy přes ty jedno-skokové. Při určení sady MPR uzlu  $U$  -  $MPR(U)$  je nutné dodržet následující podmínku: každý uzel dvou-skokového okolí uzlu  $U$  musí mít linku k  $MPR(U)$ . V rámci simulace ukázky funkčnosti protokolu OLSR byla sestaveno několik topologií, typický příklad OLSR topologie a jednotlivých MPR spojení je vidět na Obr.6.1.



Obr. 6.1: Příklad topologie OLSR

Na Obr.6.1 je naznačeno jakým způsobem dochází k pokrytí dvou-skokových sousedních uzlů. Nejprve se pomocí MPR spojení uzlu  $U$  pokryjí uzly "izolované", tedy uzly, ke kterým vede spojení pouze přes jeden daný MPR uzel. Tyto uzly jsou vyznačeny světle modrou barvou. Následují uzly, ke kterým vede spojení přes více MPR uzlů, vyznačené světle červenou barvou. MPR uzel vedoucí k nim se volí dle vyššího možného pokrytí dvou-skokových sousedů k minimalizaci celkového počtu MPR uzlů.



## 6.1 Programové řešení

Řešení simulace této práce probíhá v simulátoru NS-3. Vývojové prostředí zahrnuje softwarový nástroj Eclipse, ve kterém se definují části simulace a také možnost programu NetAnim, který průběh celé simulace, při vhodném nastavení kódu, zobrazí. Jednotlivé části programového řešení jsou popsány v následujících podsekcích.

### 6.1.1 Model sítě

Model sítě se skládá z několika uzlů (v jednotlivých simulacích 16 - 250), ty jsou rozmístěny buď v mřížce (4x4), případně náhodně rozmístěné v daném poli (mesh).

```
NodeContainer c;  
c.Create (16);
```

K rozmístění uzlů do slouží třída MobilityHelper a její funkce SetPositionAllocator. Pro tu se nastaví potřebné parametry pro metody rozmístění, ty jsou popsány v Tab.A.1

```
MobilityHelper mobility;  
mobility.SetPositionAllocator (  
    "ns3::GridPositionAllocator",  
    "MinX", DoubleValue (25.0),  
    "MinY", DoubleValue (25.0),  
    "DeltaX", DoubleValue (50.0),  
    "DeltaY", DoubleValue (50.0),  
    "GridWidth", UIntegerValue (4),  
    "LayoutType", StringValue ("RowFirst"));
```

Jak bylo zmíněno dříve, simulací bylo více, v předchozím výpisu kódu je ukázka přímého určení pozice uzlů do mřížky. Další simulace neměla pevně dané pozice uzlů, jedná se o náhodnou mesh topologii.

```
mobility.SetPositionAllocator (  
    "ns3::RandomDiscPositionAllocator",  
    "X", StringValue ("300.0"),  
    "Y", StringValue ("300.0"),  
    "Rho", StringValue (  
        "ns3::UniformRandomVariable [Min=0|Max=200] "));
```

Zde se pomocí souřadnic X a Y určí střed kruhu a následně pomocí limit parametru Rho, která stanoví radius náhodných pozic od středu kruhu.

Po rozmístění uzlů je možnost definice, zda uzly budou stacionární

```
mobility.SetMobilityModel (  
    "ns3::ConstantPositionMobilityModel");  
mobility.InstallAll ();
```

a nebo pohyblivé. K jejich mobilitě se nabízí několik metod podtřídy MobilityModel, ty jsou popsány v Tab.A.2.

```
mobility.SetMobilityModel (  
    "ns3::RandomWalk2dMobilityModel",  
    "Mode", StringValue ("Time"),  
    "Time", StringValue ("0.5s"),  
    "Distance", StringValue ("1.0"),  
    "Direction", StringValue (  
        "ns3::UniformRandomVariable [Min=0.0|Max=6.0]"), //gradient  
    "Speed", StringValue (  
        "ns3::UniformRandomVariable [Min=2.0|Max=10.0]"), //m/s  
    "Bounds", StringValue ("0|300|0|300"));
```

## 6.1.2 Nastavení komunikace

Mezi jednotlivými uzly je nyní potřeba vytvořit WiFi síť. K tomu se využilo několik tříd NS-3 - WifiHelper, YansWifiPhyHelper, YansWifiChannelHelper a WifiMacHelper.

```
WifiHelper wifi;  
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default();  
wifiPhy.Set("RxGain", DoubleValue(-32));  
wifiPhy.SetPcapDataLinkType(  
    ns3::YansWifiPhyHelper::DLT_IEEE802_11);  
YansWifiChannelHelper wifiChannel;  
wifiChannel.SetPropagationDelay(  
    "ns3::ConstantSpeedPropagationDelayModel");  
wifiChannel.AddPropagationLoss(  
    "ns3::FriisPropagationLossModel");  
wifiPhy.SetChannel(wifiChannel.Create());  
ns3::NqosWifiMacHelper wifiMac;  
wifi.SetStandard(WIFI_PHY_STANDARD_80211b);  
wifi.SetRemoteStationManager(  
    "ns3::ConstantRateWifiManager", "DataMode",  
    StringValue(phyMode), "ControlMode", StringValue(phyMode));
```

Třída YansWifiPhyHelper nám umožňuje nastavit fyzickou vrstvu modelu OSI, jednotlivé nastavitelné parametry jsou popsány v Tab.A.3. Každý parametr má výchozí hodnotu, pro zisk vysílání/přijímání je to 0 dB. Z hlediska zjednodušení tohoto nastavování byla upravena pouze hodnota RxGain - zisk přijímací antény. WiFi kanál byl nastaven pomocí třídy YansWifiChannelHelper, která doplňuje nastavení fyzické vrstvy (YansWifiPhyHelper) o ztrátovost a zpoždění, obě tyto třídy se standardně využívají současně.

```
wifiMac.SetType("ns3::AdhocWifiMac");  
NetDeviceContainer devices = wifi.Install(wifiPhy, wifiMac, c);
```

Následně se pomocí třídy WifiMacHelper určí využití kvality služeb (Qos/Nqos), zvolí se typ podvrstvy MAC, v našem případě AdhocWifiMac a pomocí třídy WifiHelper se aplikuje toto nastavení na kontejner zařízení spolu s využitím některého z WiFi standardů (802.11b), tedy dojde k vytvoření komunikačního WiFi prostředí.

Po nastavení komunikačního prostředí je nutné těmto uzlům nastavit IP adresy.

```
Ipv4AddressHelper ipv4;  
ipv4.SetBase("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer i = ipv4.Assign(devices);
```

Posledním bodem nastavení komunikace je určení směrování mezi uzly v topologii. K tomu slouží protokol OLSR.

```
OlsrHelper olsr;  
Ipv4StaticRoutingHelper staticRouting;  
Ipv4ListRoutingHelper list;  
list.Add(olsr, 10); // 10 - priorita  
InternetStackHelper internet;  
internet.SetRoutingHelper(list);  
internet.Install(c);
```

### 6.1.3 Generace datového provozu

Po nastavení komunikace už zbývá poslední část simulace a tou je generace datového provozu. K tomu byla využita aplikace `UdpEchoServer` jakožto jednoduchá aplikace stanovující zdroj komunikace a jeden nebo více cílových bodů - uzlů.

#### `UdpEchoServer`

Tento typ aplikace definuje řízení provozu podle UDP Echo serveru a jemu příslušným UDP Echo klientům. Je vytvořen server, který čeká na vstupní UDP pakety a následně je odešle zpět k původním odesílatelům. Obráceně toto platí pro klienty, kteří odesílají pakety a očekávají jejich návrat [9].

#### `On/Off` aplikace

Struktura aplikace vyplývá ze střídajících se stavů `On` a `Off`, které spouští, případně ukončí datový provoz. Trvání těchto stavů je definováno pomocí nahodilých proměnných `onTime` a `offTime`. Během stavu `Off` není generován žádný provoz, naopak u stavu `On` provoz generován je. Tento datový provoz je definován specifickými parametry míra dat (`data rate`) a velikost paketu (`packet size`).

Při startu `On/Off` aplikace nastane vysílání prvního paketu až po zpoždění, které je rovno poměru velikosti paketu a míře provozu. Pokud aplikace přechází do stavu `Off` při právě probíhajícím vysílání paketů, zbývající čas než nastane další vysílání se uloží (vyrovnávací paměť) a využije se opět, když se aplikace znovu spustí [9].

#### `PacketSink`

Tato aplikace byla vytvořena jako doplněk `On/Off` aplikace. Funkčně je důležitá při využití multicast situací, tudíž je možno přijímání důležitých rámců na 2. vrstvě modelu ISO/OSI, ale je také užitečná pro unicast jako příklad jak je možno poslat informaci pro jednoduché obdržení paketů na aplikační vrstvě [9].

Konstruktor aplikace specifikuje adresu zařízení (IP adresu a port) a transportní protokol, který bude využit. Adresu reprezentuje její číselné vyjádření a jedná se o adresu, na kterou se váže obdrženo sítě. Atribut protokol definuje transportní protokol užitý pro obdrženo sítě [9].

## 6.2 Výstupy simulace

Po spuštění simulace popsané v předchozí sekci 6.1 dojde ke komunikaci mezi jednotlivými uzly. OLSR protokol využívá pro vytvoření topologie směrování broadcast, to znamená, že každý jeden uzel začne do svého okolí, v rámci svého dosahu, vysílat zprávy typu HELLO k detekci sousední uzlů a ověřování konektivity, viz Obr.6.2.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.1.1.8	10.1.1.255	OLSR v1	84	OLSR
2	0.017064	10.1.1.9	10.1.1.255	OLSR v1	84	OLSR
3	0.023902	10.1.1.12	10.1.1.255	OLSR v1	84	OLSR
4	0.043837	10.1.1.10	10.1.1.255	OLSR v1	84	OLSR

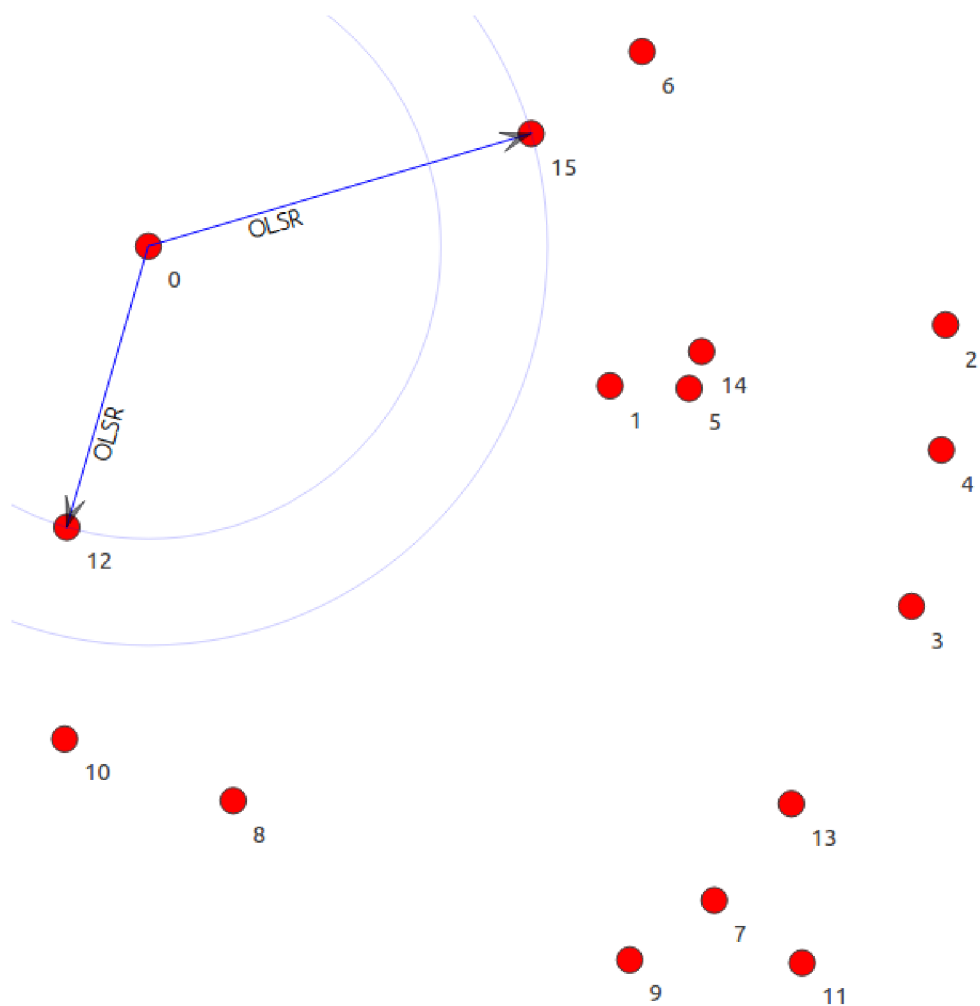
```
▶Frame 1: 84 bytes on wire (672 bits), 84 bytes captured (672 bits)
▶IEEE 802.11 Data, Flags: 0.....
▶Logical-Link Control
▶Internet Protocol Version 4, Src: 10.1.1.8 (10.1.1.8), Dst: 10.1.1.255 (10.1.1.255)
▶User Datagram Protocol, Src Port: olsr (698), Dst Port: olsr (698)
▼Optimized Link State Routing Protocol
  Packet Length: 20
  Packet Sequence Number: 0
  ▶Message: HELLO (1)
```

Obr. 6.2: Detekce sousedních uzlů pomocí zpráv HELLO

Pro jednotlivé zprávy protokolu OLSR jsou nastaveny výchozí intervaly, pro zprávu HELLO jsou to 2s, pro ostatní 5s. V rámci urychlení výběru MPR a zjištění všech destinací v síti byly intervaly zkráceny na 0,5s pro zprávu HELLO a 1s pro zprávu TC.

### 6.2.1 Výběr MPR

K volbě MPR slouží OLSR zprávy HELLO, první zprávy (se sekvenčním číslem 0) neobsahují žádnou informaci, slouží pouze ke zjištění sousedních uzlů v dosahu. Pro úvodní topologii vycházíme ze stacionární náhodné topologie dle Obr.6.3 a uzlu 0.



Obr. 6.3: Náhodná topologie pro ukázkou směrování OLSR protokolu

Jde vidět, že tento uzel vidí v rámci svého dosahu uzly 12 (IP 10.1.1.13) a 15 (10.1.1.16). Podoba první zprávy HELLO je zobrazena na Obr.6.4.

```

▼Message: HELLO (1)
  Message Type: HELLO (1)
  Validity Time: 1,500 (in seconds)
  Message: 16
  Originator Address: 10.1.1.1 (10.1.1.1)
  TTL: 1
  Hop Count: 0
  Message Sequence Number: 0
  Hello Emission Interval: 0,500 (in seconds)
  Willingness to forward messages: Unknown (3)

```

Obr. 6.4: Úvodní zpráva HELLO

Lze z ní vyčíst dříve nastavený interval (Hello Emission Interval), hodnotu TTL (1 - určena pouze pro jedno-skokové sousedy), zdroj této zprávy (Originator Address) a další hodnoty. Po obdržení takové zprávy si sousední uzly (12 a 15) nastaví původce zprávy (0) jako asymetrickou linku (mohou od něj zprávy přijímat). Tuto informaci následně pošlou v další zprávě HELLO, jak lze vidět na Obr.6.5.

```

▼Message: HELLO (1)
  Message Type: HELLO (1)
  Validity Time: 1,500 (in seconds)
  Message: 40
  Originator Address: 10.1.1.13 (10.1.1.13)
  TTL: 1
  Hop Count: 0
  Message Sequence Number: 1
  Hello Emission Interval: 0,500 (in seconds)
  Willingness to forward messages: Unknown (3)
▶Link Type: Asymmetric Link (1)
▼Link Type: Asymmetric Link (1)
  Link Message Size: 8
  Neighbor Address: 10.1.1.1 (10.1.1.1)
▶Link Type: Asymmetric Link (1)

```

Obr. 6.5: Zpráva HELLO informující o asymetrických linkách

Uzel 0 již obdržel od svých sousedů úvodní zprávu HELLO (se sekvenčním číslem 0), má tedy nastavené linky k uzlům 12 a 15 jako asymetrické (přijímá od nich zprávy), nyní zjistí, že tyto uzly přímají zprávy od něj (sekvenční zpráva č. 1), změní tedy typ linky k nim na symetrický (oboustranný) - sekvenční zpráva č. 2, viz Obr.6.6.

```

Originator Address: 10.1.1.1 (10.1.1.1)
TTL: 1
Hop Count: 0
Message Sequence Number: 2
Hello Emission Interval: 0,500 (in seconds)
Willingness to forward messages: Unknown (3)
▼Link Type: Symmetric Link (6)
  Link Message Size: 8
  Neighbor Address: 10.1.1.13 (10.1.1.13)
▼Link Type: Symmetric Link (6)
  Link Message Size: 8
  Neighbor Address: 10.1.1.16 (10.1.1.16)

```

Obr. 6.6: Sekvenční zpráva HELLO č. 2 nastavující symetrické linky

Uzel 0 má tedy stanovené symetrické linky k uzlům 12 a 15, během další komunikace obdrží informace o symetrických linkách od těchto sousedů, díky kterým zjistí, že tyto uzly je mohou spojit s dalšími uzly v síti. Tyto uzly jsou vybrány jako MPR, viz sekvenční zpráva č. 3 na Obr.6.7.

```

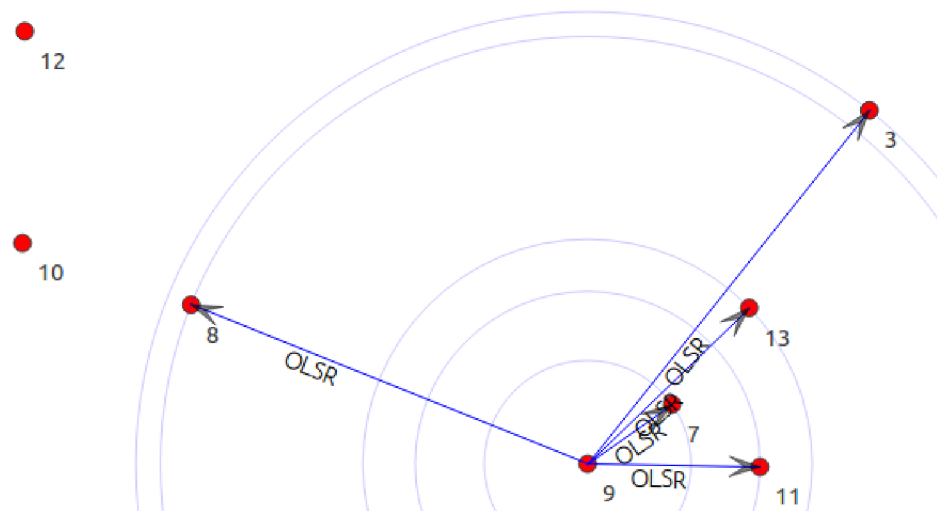
Originator Address: 10.1.1.1 (10.1.1.1)
TTL: 1
Hop Count: 0
Message Sequence Number: 3
Hello Emission Interval: 0,500 (in seconds)
Willingness to forward messages: Unknown (3)
▼Link Type: MPR Link (10)
  Link Message Size: 8
  Neighbor Address: 10.1.1.13 (10.1.1.13)
▼Link Type: MPR Link (10)
  Link Message Size: 8
  Neighbor Address: 10.1.1.16 (10.1.1.16)

```

Obr. 6.7: Sekvenční zpráva HELLO č. 3 - zvolení MPR

Pro jednoduchost vysvětlení základního výběru MPR sady byl v předchozí části vybrán uzel 0. Na dalších stranách bude popsán výběr MPR pro uzel 9, který má ve svém jedno-skokovém okolí více uzlů a výběr sady MPR je lehce komplikovanější. Pořád vycházíme z topologie dle Obr.6.1.

Dosah uzlu 9 je vidět na Obr.6.8, jeho jedno-skokový sousedé jsou uzly 3, 7, 8, 11 a 13.



Obr. 6.8: Dosah uzlu 9



Jako MPR mohou být zvoleny pouze ty uzly, které nabízí spojení se vzdálenějšími uzly v síti. Uzly 7 a 11 jsou tedy pro volbu vynechány, platí k nim symetrická linka, vztah MPR je ale odlišný, uzel 9 je zvolen jako jejich MPR vzhledem k jeho symetrické lince k uzlu 8.

Do sady MPR byly zvoleny uzly 3 a 8 jak lze vidět na Obr.6.9 v sekvenční zprávě HELLO č. 3. Jako MPR mohl být zvolen i uzel 13, který by uzel 9 spojil s uzly 4 a 5, ale stejné pokrytí má i uzel 3 a dle podmínky zmíněné na začátku této kapitoly 6 byl zvolen uzel ten, který nabízí více dalších spojení (dostupných uzlů).

```
Originator Address: 10.1.1.10 (10.1.1.10)
TTL: 1
Hop Count: 0
Message Sequence Number: 3
Hello Emission Interval: 0,500 (in seconds)
Willingness to forward messages: Unknown (3)
▼Link Type: Symmetric Link (6)
  Link Message Size: 8
  Neighbor Address: 10.1.1.8 (10.1.1.8)
▼Link Type: MPR Link (10)
  Link Message Size: 8
  Neighbor Address: 10.1.1.9 (10.1.1.9)
▼Link Type: Symmetric Link (6)
  Link Message Size: 8
  Neighbor Address: 10.1.1.12 (10.1.1.12)
▼Link Type: MPR Link (10)
  Link Message Size: 8
  Neighbor Address: 10.1.1.4 (10.1.1.4)
▼Link Type: Symmetric Link (6)
  Link Message Size: 8
  Neighbor Address: 10.1.1.14 (10.1.1.14)
```

Obr. 6.9: Volba MPR uzlu 9

## 6.2.2 Sestavení směrovací tabulky

Jedním z výstupů simulace je textový soubor, který pro jednotlivé uzly vypisuje jejich směrovací tabulku jak je možno vidět na Obr.6.10. S nastaveným intervalem kontrolních zpráv (TC - Topology Control) se tyto tabulky aktualizují. Na začátku aktualizace zná každý uzel pouze svoji localhost adresu (127.0.0.0) a adresu sítě (10.1.1.0). Na Obr.6.10 je vidět tabulka v čase simulace 1 s, tedy okolní uzly 3, 7, 8, 11 a 13 po přenesení zpráv HELLO se sekvenčním číslem 2 - po nastavení symetrických linek.

```

Node: 9 Time: 4s Ipv4ListRouting table
  Priority: 10 Protocol: ns3::olsr::RoutingProtocol
Destination      NextHop          Interface        Distance
10.1.1.4         10.1.1.4         1                1
10.1.1.8         10.1.1.8         1                1
10.1.1.9         10.1.1.9         1                1
10.1.1.12        10.1.1.12        1                1
10.1.1.14        10.1.1.14        1                1
HNA Routing Table:
  Priority: 0 Protocol: ns3::Ipv4StaticRouting
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
127.0.0.0        0.0.0.0         255.0.0.0       U      0    -    -    0
10.1.1.0         0.0.0.0         255.255.255.0   U      0    -    -    1

```

Obr. 6.10: Směrovací tabulka uzlu 9 v čase 1 s

S postupem času dochází ke zmíněné aktualizaci a každý uzel má následně přehled o všech dostupných zařízeních v síti s informací o jejich vzdálenosti (Distance), viz Obr.6.3.

```

Node: 9 Time: 12s Ipv4ListRouting table
  Priority: 10 Protocol: ns3::olsr::RoutingProtocol
Destination      NextHop          Interface        Distance
10.1.1.1         10.1.1.9         1                3
10.1.1.2         10.1.1.4         1                2
10.1.1.3         10.1.1.4         1                2
10.1.1.4         10.1.1.4         1                1
10.1.1.5         10.1.1.4         1                2
10.1.1.6         10.1.1.4         1                2
10.1.1.7         10.1.1.4         1                3
10.1.1.8         10.1.1.8         1                1
10.1.1.9         10.1.1.9         1                1
10.1.1.11        10.1.1.9         1                2
10.1.1.12        10.1.1.12        1                1
10.1.1.13        10.1.1.9         1                2
10.1.1.14        10.1.1.14        1                1
10.1.1.15        10.1.1.4         1                2
10.1.1.16        10.1.1.4         1                3
HNA Routing Table:
  Priority: 0 Protocol: ns3::Ipv4StaticRouting
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
127.0.0.0        0.0.0.0         255.0.0.0       U      0    -    -    0
10.1.1.0         0.0.0.0         255.255.255.0   U      0    -    -    1

```

Obr. 6.11: Směrovací tabulka uzlu 9 v čase 12 s

Simulace byla nastavena tak, aby se z uzlu 11 poslalo 20 paketů k uzlu 0. Pakety se začaly vysílat v čase 0s, první úspěšně doručený paket dorazil v čase 2s, tedy po 5 výměnách zpráv HELLO a jedné TC. Zprávy komunikace OLSR protokolu jsou v přílohách v Tab.6.1.

Tab. 6.1: Typ zpráv OLSR protokolu a jejich popis

Typ zprávy	Účel zprávy
HELLO	Zprávy HELLO slouží k ověřování konektivity a detekci sousedních uzlů.
TC	Kontrolní zprávy informují ostatní uzly o jedno-skokových sousedních uzlech uzlu U (tvorba směrovacích tabulek).
MID	Zprávy MID obsahují seznam všech jiných adres rozhraní uzlu než je to hlavní (OLSR).
HNA	Zprávami HNA informuje uzel své sousedy o k němu aktuálně připojených hostech a sítích (pár síťová adresa-masky).

Vždy záleží na účelu vytvořené sítě a k tomu příslušnému nastavení intervalů jednotlivých zpráv. Dále záleží zda jsou uzly sítě mobilní nebo stacionární. V případě mobilních uzlů může být vhodnější nastavení kratších intervalů, naopak vzhledem k energetické úspoře může být interval delší. Po několika výměnách zpráv typu HELLO tedy dojde k výměnám typu TC jak lze vidět na Obr.6.12. V těchto zprávách se uzly informují o svých sousedech a zároveň předávají informace o sousedních uzlech svým MPR, případně uzlům, kterým jako MPR slouží k sestavení směrovací tabulky, jak bylo znázorněno na Obr.6.3.

```

▼Message: TC (2)
  Message Type: TC (2)
  Validity Time: 15,000 (in seconds)
  Message: 36
  Originator Address: 10.1.1.10 (10.1.1.10)
  TTL: 255
  Hop Count: 0
  Message Sequence Number: 5
  Advertised Neighbor Sequence Number (ANSN): 9
  Neighbor Address: 10.1.1.9 (10.1.1.9)
  Neighbor Address: 10.1.1.12 (10.1.1.12)
  Neighbor Address: 10.1.1.8 (10.1.1.8)
  Neighbor Address: 10.1.1.14 (10.1.1.14)
  Neighbor Address: 10.1.1.4 (10.1.1.4)

```

Obr. 6.12: Kontrolní zpráva protokolu OLSR

Po sestavení směrovacích tabulek je možno generovat datový provoz, ten byl sestaven z uzlu 11 (10.1.1.12) k 0 pomocí UdpSocketFactory. V případě vyslání paketu před sestavením směrovacích tabulek dochází k ARP dotazům, ale bez odpovědi. V momentě sestavených směrovacích tabulek již probíhá doručování paketu a směrování, to u protokolu OLSR probíhá vždy lokálně. Uzel 11 vyšle paket vždy všem okolním uzlům - broadcast i když má paket danou cílovou adresu uzlu 0 (10.1.1.1), viz Obr.6.13.

No.	Time	Source	Destination	Protocol	Length
26	2.143671	10.1.1.12	10.1.1.1	UDP	1064

```

▶ Frame 26: 1064 bytes on wire (8512 bits), 1064 bytes captured (8512 bits)
▶ IEEE 802.11 Data, Flags: o.....
▶ Logical-Link Control
▶ Internet Protocol Version 4, Src: 10.1.1.12 (10.1.1.12), Dst: 10.1.1.1 (10.1.1.1)
▶ User Datagram Protocol, Src Port: 49153 (49153), Dst Port: http (80)
▶ Data (1000 bytes)

```

Obr. 6.13: UDP přenos zobrazený v programu Wireshark

Paket od uzlu 11 převezme uzel 9, který paket přepośle opět všem svým sousedům. Vzhledem k informaci ve směrovací tabulce s paketem pracuje pouze uzel 8, opakuje se broadcast vysílání, paket zpracuje pouze uzel 12 a předá cílové destinaci - uzlu 0.

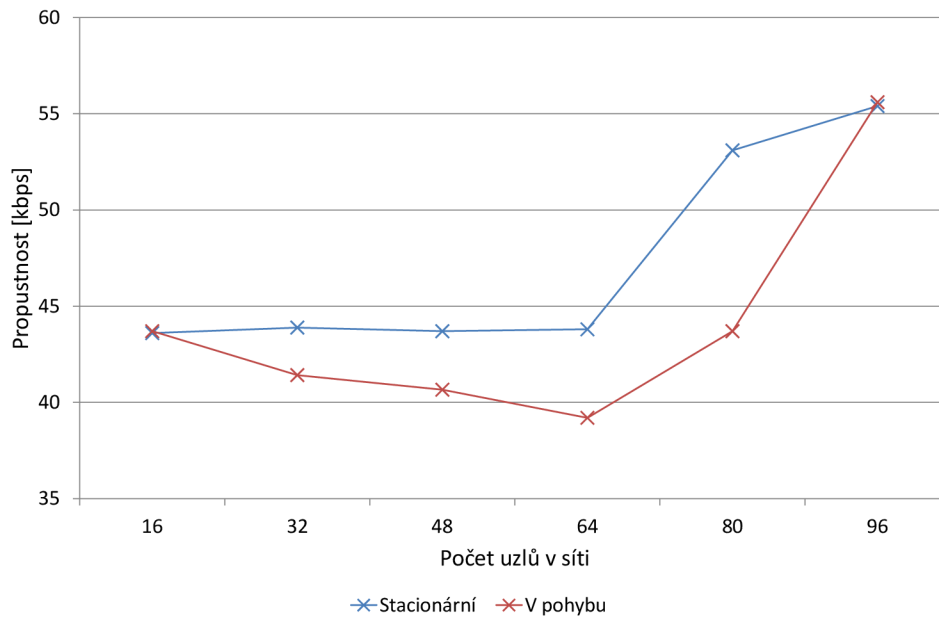
Při nastavení vysílání paketů hned po startu simulace (v čase 0s) dojde k určité ztrátovosti, prvních 11 paketů není doručeno. K doručení dojde až po sestavení směrovacích tabulek, vzhledem k intervalu zpráv HELLO až v čase 2s, viz Obr.6.3. V případě posunutí zasílání paketů na čas 2,5s dojde k doručení všech paketů, bez žádné ztrátovosti. K zobrazení statistik byl použit nástroj NS-3 FlowMonitor, který data uloží do xml souboru, ty jsou následně načíst v programu NetAnim. Porovnání statistik je zobrazeno na Obr.6.14, vlevo je statistika simulace vysílající pakety hned po startu simulace, vpravo statistika simulace kdy jsou pakety vysílány až v čase 2,5s.

UDP 10.1.1.12/49153---->10.1.1.1/80		UDP 10.1.1.12/49153---->10.1.1.1/80	
Tx bitrate:43.4044kbps		Tx bitrate:43.2842kbps	
Rx bitrate:46.89kbps		Rx bitrate:43.5804kbps	
Mean delay:40.916ms		Mean delay:39.4217ms	
Packet Loss ratio:52.6316%		Packet Loss ratio:0%	
timeFirstTxPacket= 2e+08ns		timeFirstTxPacket= 2.5e+09ns	
timeFirstRxPacket= 2.25741e+09ns		timeFirstRxPacket= 2.56174e+09ns	
timeLastTxPacket= 3.8e+09ns		timeLastTxPacket= 6.3e+09ns	
timeLastRxPacket= 3.83591e+09ns		timeLastRxPacket= 6.33591e+09ns	
delaySum= 3.68244e+08ns		delaySum= 7.88433e+08ns	
jitterSum= 4.7822e+07ns		jitterSum= 1.11998e+08ns	
lastDelay= 3.68244e+08ns		lastDelay= 7.88433e+08ns	
txBytes= 19532		txBytes= 20560	
rxBytes= 9252		rxBytes= 20560	
txPackets= 19		txPackets= 20	
rxPackets= 9		rxPackets= 20	
lostPackets= 10		lostPackets= 0	
timesForwarded= 27		timesForwarded= 60	

Obr. 6.14: Porovnání statistik v programu NetAnim

### 6.2.3 Statistiky protokolu

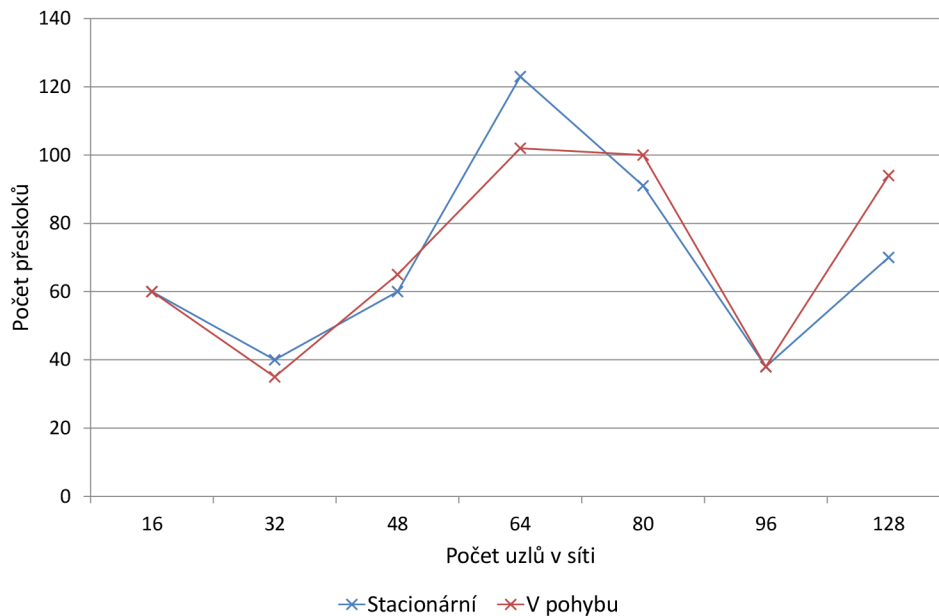
Po několika simulacích je možná vypočítat několik statistik. Danou síť je potřeba vždy vyváženě nastavit, simulace této práce vycházely z náhodného rozmístění uzlů, vždy se tedy upravovaly rozměry sítě vzhledem k zachování určité hustoty uzlů tak, aby jich byla dosahem pokryta většina. Na Obr.6.15 lze vidět, že s vyšší hustotou uzlů v síti dostáváme vyšší propustnost.



Obr. 6.15: Statistika propustnosti v závislosti na počtu uzlů v síti

V případě mobilních uzlů dochází k určité ztrátovosti a také zpoždění, celková propustnost je tedy nižší oproti stacionárním uzlům.

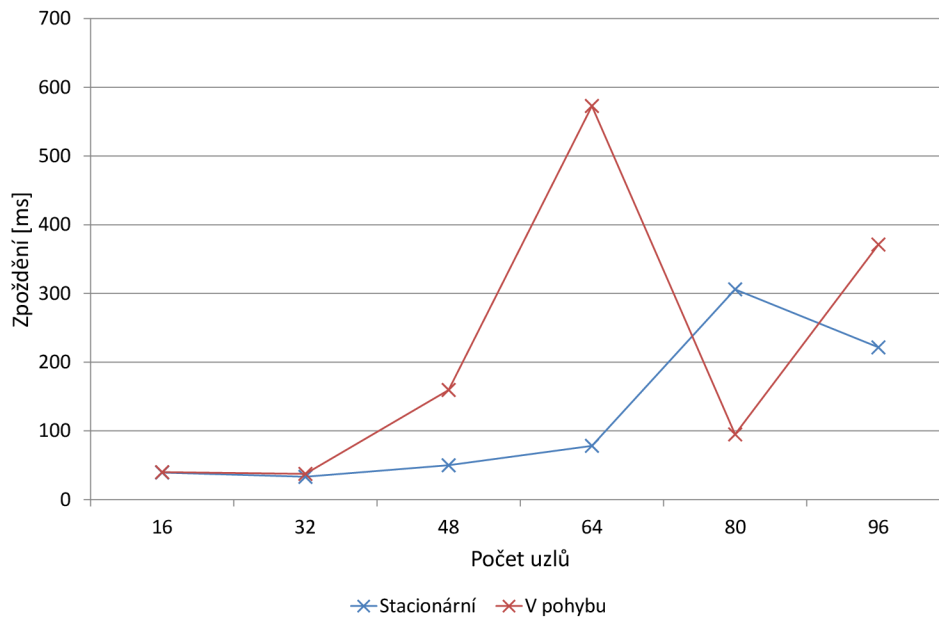
Další statistikou je počet přeskoků paketů mezi uzly během směrování. S mobilitou uzlů nastávají v určité změny v topologii, některé linky zaniknou, nové se utvoří a v ideálním případě by to mohlo znamenat navýšení počtu linek a tedy rychlejší směrování - snížení počtu přeskoků. Tato statistika je zobrazena na Obr.6.16.



Obr. 6.16: Statistika počtu přeskoků v závislosti na počtu uzlů v síti

Během směrování byla komunikující dvojice vždy zvolena ze dvou okrajových uzlů, aby trasa paketů byla napříč celou topologií. Pro stacionární uzly to mohlo znamenat vyšší počet přeskoků, který mohl být díky mobilitě uzlů snížen. Z Obr.6.16 lze vyčíst, že v některých případech nastalo snížení počtu skoků až o hodnotu 21, někdy byla hodnota přibližně stejná jako u stacionárních uzlů, případně byl počet skoků k cílové destinaci vyšší. Zde záleží pohybu uzlů, zda se krajní uzly pohybují směrem k centru topologie či od něj, pokud je pohyb příliš rychlý a nové linky nejsou sestaveny včas k zefektivnění komunikace.

Během směrování paketů sítě dochází vždy ke zpoždění, s každým přeskokem paketu směrem k cíli dojde kvůli směrování a přenosu dat k určitému zpoždění bez ohledu na efektivitu směrování. S vyšším počtem uzlů v síti tedy dochází k většímu zpoždění jak je zobrazeno na Obr.6.17.

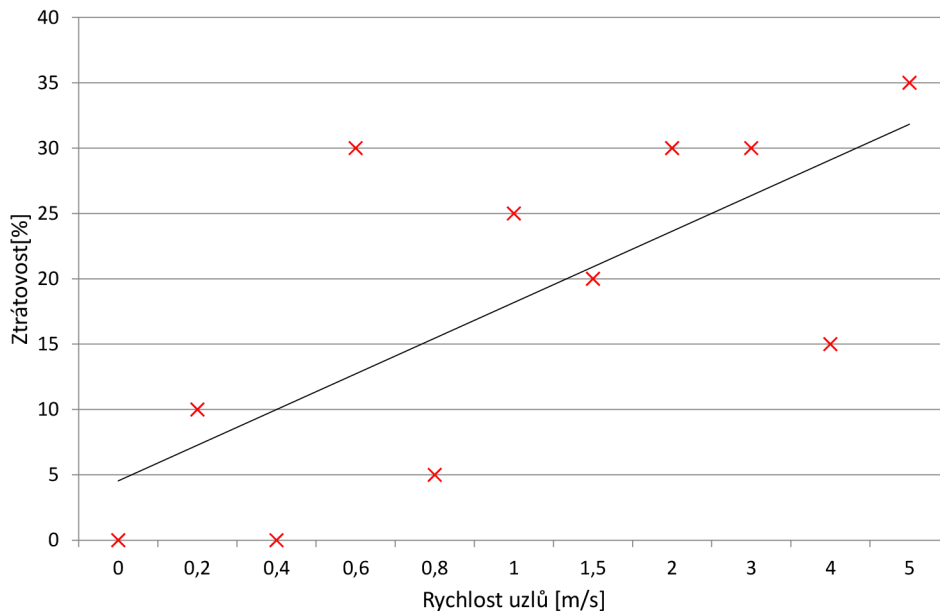


Obr. 6.17: Statistika zpoždění v závislosti na počtu uzlů

Zde je vidět, že ve většině případů je zpoždění u mobilních uzlů vyšší oproti stacionární topologii. Nižší je pouze tehdy, když se uzel díky pohybu dostane svým dosahem k novým uzlům, nastaví pro ně linku a následně je zpoždění nižší například kvůli menšímu počtu přeskoků od zdroje k cíli.



Vzhledem k mobilitě uzlů je důležité uvažovat ztrátovost. Rychlost uzlů v závislosti na intervalu OLSR zpráv může způsobit to, že nové linky nebudou sestaveny/uloženy včas a v případě ztráty linky může dojít ke ztrátovosti paketů. Tato statistika je zobrazena na Obr.6.18.



Obr. 6.18: Statistika ztrátovosti v závislosti na rychlosti pohybu uzlů

V průměru (spojnice trendu - černá čára) je logicky ztrátovost lineárně vyšší se zvyšující se rychlostí uzlů. Hlavní důvodem tohoto navýšení jsou nevhodné hodnoty intervalů OLSR zpráv. S vyšší rychlostí dochází častěji k tomu, že uzly najdou ve svém dosahu nové uzly, pro které je potřeba sestavit linky a zjištění zda nový uzel má být zvolen jako MPR anebo pro něj pouze změnit směrovací informace. Jak bylo zmíněno dříve v podsekcí 6.2.1, uzly si navzájem musí vyměnit několik zpráv HELLO k vzájemnému uvědomění se o trasách, které každý uzel nabízí a případnému sestavení linky o určitém typu (symetrická, asymetrická nebo MPR). Pokud je pohyb uzlů příliš rychlý a k sestavení linek nedojde, nastává ztrátovost.

Důležitým úkolem při sestavení topologie je tedy i volba intervalů OLSR zpráv. Ty mohou být krátké - desetiny vteřin, ale zároveň by se mělo myslet na energetickou náročnost výsledné sítě. Při sestavování topologie je tedy nutné myslet na princip sítě, možnou rychlost uzlů a k tomu vhodně nastavené intervaly zpráv směrovacího protokolu v závislosti na energetické náročnosti.

## 6.3 Úkoly laboratorní úlohy

K úkolům využijte příklady v adresáři NS-3 a nebo web <https://www.nsnam.org/>

### 1. Sestavte komunikační dvojici pomocí UdpSocketFactory

Funkce pro generování provozu je stanovena na začátku kódu, stačí přiřadit hodnoty proměnným "sourceNode" a "sinkNode" (dle pořadí uzlu).

V případě vyslání paketu před sestavením směrovacích tabulek dochází k ARP dotazům, ale bez odpovědi. V momentě sestavených směrovacích tabulek již probíhá doručování paketu a směrování, to u protokolu OLSR probíhá vždy lokálně. Uzel vyšle paket vždy všem okolním uzlům - broadcast i když má paket danou cílovou adresu. V programu NetAnim v záložce Stats vyberte možnost Flow-monitor a pomocí tlačítka FlowMon file načtete soubor "/home/student/ns-allinone-3.21/ns-3.21.xml" k zobrazení statistik simulace.



### 2. Změňte dobu odeslání prvního paketu

Ve výchozí podobě je první paket odeslán v čase 1 s, dle statistik je ztrátovost vysoká. Pomocí programu WireShark zjistíte po jak dlouhé době od začátku simulace dojde k úspěšnému odeslání všech paketů.

### 3. Upravte kód tak, aby se uzly pohybovaly a měňte interval OLSR zpráv

V různých situacích jsou potřeba odlišné intervaly pro sestavení směrovacích tabulek, v případě stacionárních uzlů mohou být výchozí (2 s), případně delší. V momentě kdy jsou zvoleny sady MPR uzlů dochází pouze k ověřování stávajících tras. Pokud se jedná o síť s mobilními uzly, intervaly zpráv mohou být kratší, aby se zamezilo případné ztrátovosti. Zkontrolujte statistiky v NetAnim.

### 4. Vytvořte vlastní topologii s nulovou ztrátovostí

Rozšiřte aktuální topologii, upravte tomu rozestavení uzlů - rozsah sítě, aby pokrytí jednotlivých uzlů nebylo příliš velké a nastavte intervaly OLSR zpráv anebo čas poslání prvního paketu tak, aby nedocházelo ke ztrátovosti.

## Kontrolní otázky

- Jak probíhá směrování OLSR, jakým způsobem jsou zvoleny sady MPR?
- Kolik přenosů (intervalů) zpráv HELLO je potřeba k získání kompletních směrovacích informací tabulek?

# VZOROVÉ ŘEŠENÍ

## 1. Sestavte komunikační dvojici pomocí UdpSocketFactory

Sestavena komunikační dvojice, pakety budou vyslány z uzlu 9 k uzlu 0. První paket je odeslán v čase 1 s, některé pakety tedy budou ztraceny dokud nebudou sestaveny směrovací tabulky. Statistiky simulace:

```
Tx bitrate:8.65684kbps
Rx bitrate:9.14782kbps
Mean delay:27.8283ms
Packet Loss ratio:50%

timeFirstTxPacket= 1e+09ns
timeFirstRxPacket= 1.10367e+10ns
timeLastTxPacket= 2e+10ns
timeLastRxPacket= 2.00268e+10ns
delaySum= 2.78283e+08ns
jitterSum= 9.8788e+06ns
lastDelay= 2.78283e+08ns
txBytes= 20560
rxBytes= 10280
txPackets= 20
rxPackets= 10
lostPackets= 10
timesForwarded= 20
```

## 2. Změňte dobu odeslání prvního paketu

Ze souboru OLSR.routes lze vyčíst, že kompletní směrovací tabulka pro uzel 9 je sestavena v čase 11 s, vyslání prvního paketu má tedy smysl až v čase 11 s.

Úprava kódu:

```
Simulator::Schedule(Seconds(11.0), &GenerateTraffic, source,
    packetSize, numPackets, interPacketInterval);
```

Statistiky simulace:

```
txBytes= 20560
rxBytes= 20560
txPackets= 20
rxPackets= 20
lostPackets= 0
timesForwarded= 40
```

### 3. Upravte kód tak, aby se uzly pohybovaly a měňte interval OLSR zpráv

K mobilitě uzlů stačí "odkomentovat" tyto řádky:

```
mobility.SetMobilityModel ("ns3::RandomWalk2dMobilityModel",
    "Mode", StringValue ("Time"),
    "Time", StringValue ("0.5s"),
    "Distance", StringValue ("1.0"),
    "Direction", StringValue (
        "ns3::UniformRandomVariable [Min=0.0 | Max=6.283184] "),
    "Speed", StringValue (
        "ns3::UniformRandomVariable [Min=1.0 | Max=3.0] "),
    "Bounds", StringValue ("0|350|0|350"));
```

Pro změnu limitu zpráv OLSR stačí přidat následující řádek, zde pro zprávu HELLO:

```
olsr.Set("HelloInterval", TimeValue (Seconds (1)));
```

Intervaly ostatních zpráv by se měnily podobně. Z původních 2s byl interval změněn na 1s, výchozí interval kontrolních zpráv (TC) je 5s, v čase 6s mají tedy všechny uzly dostupné informace pro veškeré destinace v síti. Statistika simulace:

```
timeFirstTxPacket= 6e+09ns
timeFirstRxPacket= 6.04464e+09ns
timeLastTxPacket= 2.5e+10ns
timeLastRxPacket= 2.50268e+10ns
delaySum=         5.56457e+08ns
jitterSum=        2.14993e+07ns
lastDelay=        5.56457e+08ns
txBytes=          20560
rxBytes=          20560
txPackets=        20
rxPackets=        20
```

#### 4. Vytvořte vlastní topologii s co nejnižší ztrátovostí

Důležitým parametrem při snaze o nízkou ztrátovost je interval zpráv HELLO a TC, pokud dojde v topologii ke změně, je nutné ji zaregistrovat a napravit. K tomu slouží právě tyto zprávy, HELLO kontroluje konektivitu a nachází nové linky, TC šíří směrovací informace mezi MPR uzly. K sestavení nové linky potřebuje zpráva HELLO 4 intervaly, interval zprávy by tedy měl být menší než 1 s, aby bylo možné zavčas reagovat na změny v síti (dosahu uzlu). K tomu se váže také rychlost pohybu uzlů, pokud je pohyb příliš rychlý, musí tomu odpovídat interval OLSR zpráv, to ale zároveň zatěžuje síť a zároveň je takové chování v praxi energeticky náročné. Dále je potřeba myslet na interval odesílání paketů, jeho prodloužením vzniká více času pro aktualizaci směrovacích informací. Změněny byly tyto hodnoty:

```
uint32_t numNodes = 48;
double interval = 0.5; // seconds

mobility.SetMobilityModel (
"ns3::RandomWalk2dMobilityModel", "Speed",
StringValue ("ns3::UniformRandomVariable [Min=0.1|Max=0.5]"))

olsr.Set("HelloInterval", TimeValue (Seconds (0.5)));
olsr.Set("TcInterval", TimeValue (Seconds (2)));
```

Z 20 vyslaných paketů se ztratily 3, došlo tedy ke 15%, k přiblížení se 0 by síť musela mít téměř minimální rychlost, nízký interval zpráv HELLO a delší interval odesílání paketů, nejednalo by se již ale o v praxi reálnou síť. V případě mobility uzlů se s určitou ztrátovostí počítá, následně již záleží na účelu sítě a k tomu zvolených parametrů.

#### Kontrolní otázky

**Jak probíhá směrování OLSR, jakým způsobem jsou zvoleny sady MPR?**

Výběr MPR sady je prováděn ve dvou krocích, kdy se uzel pomocí svého jedno-skokového okolí -  $N1(U)$  snaží pokrýt své dvou-skokové sousedy -  $N2(U)$ :

1. Výběr uzlů z  $N1(U)$  pokrývajících "izolované" uzly z  $N2(U)$ .
2. Uzly z  $N1(U)$ , které ještě nejsou zvoleny jako MPR, mající linku k  $N2(U)$ , které nebyly pokryty v prvním kroku (je k nim více linek). Volí se ty uzly z  $N1(U)$ , které pokryjí více uzlů z  $N2(U)$ .

**Kolik přenosů (intervalů) zpráv HELLO je potřeba k získání kompletních směrovacích informací tabulek?**

4 intervaly zprávy HELLO, bez ohledu na velikost topologie.

## 7 ZÁVĚR

Cílem diplomové práce bylo seznámení s ad hoc sítěmi, směrovacím protokolem OLSR využívaným v těchto sítích a provedení simulace využívající tento protokol. Účelem simulace bylo pochopení funkčnosti protokolu k následnému návrhu laboratorní úlohy, která by následně studentům vysvětlovala princip a využití tohoto směrovacího protokolu.

V teoretické části jsou popsány ad hoc sítě z hlediska využití a směrování v nich používaného. Následně bylo popsáno směrování jako takové, algoritmy směrování a směrovací protokoly ad hoc sítí. Hlavním bodem teoretické části byl důkladný popis směrovacího protokolu OLSR, jeho využití, struktura a funkčnost.

Praktickou částí bylo provedení simulací v simulátoru NS-3 k pochopení směrovacího protokolu OLSR, ze kterých se bude vycházet při návrhu laboratorní úlohy.

Provedené simulace ukázaly funkčnost protokolu OLSR v sítích volných - náhodná poloha uzlů, kdy se jednotlivé uzly musí rozpoznat podobu topologie za pomoci směrovacího protokolu OLSR. Pro uzly této náhodné topologie byl sestaven komunikační kanál - Wi-Fi síť a topologie uvedena do ad hoc stavu k otestování základní funkcionality protokolu OLSR. Zároveň byla také definována mobilita uzlů, díky které se průběžně topologie měnila a tedy i rozložení spojení MPR pro jednotlivé uzly. Následně byl vytvořen návrh laboratorní úlohy, kde si studenti mohou otestovat jak probíhá sestavení směrovacích tabulek, výběr klíčové sady MPR jakožto hlavní definice směrovacího protokolu OLSR, co směrování předchází a jak takové směrování probíhá. K laboratorní úloze bylo přiloženo vzorové řešení, tedy souhrn znalostí o protokolu OLSR, které by si měl student odnést.

# LITERATURA

- [1] RFC 2501 *Mobile Ad hoc Networking (MANET)* [online]. Naval Research Laboratory: University of Maryland, 1999 [cit. 10 12 2017]. Dostupné z URL: <<https://tools.ietf.org/html/rfc2501.html>>.
- [2] ILYAS, Mohammad. *THE HANDBOOK OF AD HOC WIRELESS NETWORKS*. Florida: Florida Atlantic University, 2003 [cit. 10 12 2017]. ISBN 0-8493-1332-5.
- [3] ROY, Radhika Ranjan. *Handbook of Mobile Ad Hoc Networks for Mobility Models*. Fort Monmouth, New Jersey: United States Army Research, 2011 [cit. 10 12 2017]. ISBN 978-1-4419-6048-1.
- [4] ODOM, Wendell. *Cisco CCNA Routing and Switching ICND2 200-101 Official Cert Guide*. Indianapolis: Cisco Press, 2013 [cit. 10 12 2017]. ISBN-13: 978-1-58714-373-1, ISBN-10: 1-58714-373-9.
- [5] JEŘÁBEK, Jan. *Komunikační technologie* [online]. Brno: Vysoké učení technické v Brně, 2013 [cit. 10 12 2017]. ISBN 978-80-214-4713-4. Dostupné z URL: <[https://www.vutbr.cz/www\\_base/priloha.php?dpid=71518](https://www.vutbr.cz/www_base/priloha.php?dpid=71518)>.
- [6] JEŘÁBEK, Jan. *Pokročilé komunikační techniky*. [online]. Brno: Vysoké učení technické v Brně, 2012 [cit. 10 12 2017]. Dostupné z URL: <[https://www.vutbr.cz/www\\_base/priloha.php?dpid=67088](https://www.vutbr.cz/www_base/priloha.php?dpid=67088)>.
- [7] ROYER, Elizabeth M., CHAI-KEONG, Toh *A Review of Current Routing Protocols Ad Hoc Mobile Wireless Networks*. 1999 [cit. 10 12 2017].
- [8] RFC 3626 *Optimized Link State Routing Protocol (OLSR)* [online]. 2003 [cit. 10 12 2017]. Dostupné z URL: <<https://tools.ietf.org/html/rfc3626>>.
- [9] *ns-3* [online]. [cit. 10 12 2017]. Dostupné z URL: <<https://www.nsnam.org/>>.
- [10] RFC 3626 *The Optimized Link State Routing Protocol Version 2* [online]. 2014 [cit. 4 2014]. Dostupné z URL: <<http://www.rfc-editor.org/info/rfc7181>>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ACK	informační zpráva telekomunikačních protokolů značící potvrzení - Acknowledgement
AODV	reaktivní vektorový směrovací protokol sítí MANET - Ad hoc On-Demand Distance Vector Routing
ARP	komunikační protokol pro zjištění linkové adresy přidružené k IP adrese - Address Resolution Protocol
bajt	jednotka množství dat v informatičních technologiích (anglicky byte)
BAN	Body Area Network
DSDV	proaktivní distančně-sekvenční vektorový směrovací protokol sítí MANET - Destination-Sequenced Distance-Vector Routing
ETSI	Evropský ústav pro telekomunikační normy - European Telecommunications Standards Institute
FCS	detekční kód chyb komunikačních protokolů - Frame Check Sequence
IANA	organizace dohlížející celosvětově na přidělování IP adres, správu kořenových zón DNS a další náležitosti internetových protokolů - Internet Assigned Numbers Authority
ID	identifikátor
IGRP	vektorový směrovací protokol firmy Cisco pro užití uvnitř autonomních systémů - Interior Gateway Routing Protocol
IP	Internet Protocol
ISM	pásma pro rádiové vysílání, bez licenčních poplatků, avšak bez garance proti rušení - industrial, scientific and medical)
LAN	Local Area Network
MAC	unikátní fyzická adresa zařízení linkové vrstvy modelu OSI - Media Access Control
MANET	Mobile Ad hoc Network
MPR	vícebodové spojení/přenos protokolu OLSR - Multipoint Relay
NPDU	datová jednotka síťového protokolu - Network Protocol Data Unit
NS	Network Simulator
OLSR	stavový směrovací protokol sítí MANET - Optimized Link State Routing Protocol
OSI	referenční model vypracovaný organizací Open Systems Interconnection stanovující řešení komunikace v počítačových a telekomunikačních sítích pomocí vrstevnatého modelu
OSPF	stavový směrovací protokol určený pro užití uvnitř autonomního systému - Open Shortest Path First
PAN	Personal Area Network



PDA	Personal digital assistant
RIP	vektorový směrovací protokol - Routing Information Protocol
RREQ	požadavek o nalezení trasy protokolu AODV - Route Request
$TC_{REDUNDANCY}$	Kontrolní zpráva protokolu OLSR odesílaná zvoleným MPR rozšířená i o sousední uzly nezvolené jako MPR
TTL	Time to live
UDP	jednoduchý transportní protokol umožňující nespojovaný a nespolehlivý přenos dat - User Datagram Protocol
WAN	Wide Area Network
Wi-Fi	Wireless Fidelity
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WRP	proaktivní vektorový směrovací protokol sítí MANET - Wireless Routing Protocol

# SEZNAM PŘÍLOH

<b>A Přílohy</b>	<b>62</b>
A.1 Přehled metod pro rozmístění uzlů . . . . .	62
A.2 Přehled metod pro pohyb uzlů . . . . .	63
A.3 Přehled parametrů třídy YansWifiPhyHelper . . . . .	64
A.4 Laboratorní úloha . . . . .	64

# A PŘÍLOHY

## A.1 Přehled metod pro rozmístění uzlů

Tab. A.1: Metody rozmístění uzlů [9]

Metoda	Popis
GridPositionAllocator	Rozmístění uzlů do obdelníkové (dvourozměrné) mřížky
ListPositionAlocator	Rozmístění do specifické řady, dále specifikované uživatelem
PositionAllocator	Přímé určení pozice - [x,y]
RandomBoxPositionAllocator	Zvolení náhodné pozice v trojrozměrném objektu (krychle/kvádr)
RandomDiscPositionAllocator	Zvolení náhodné pozice v kruhu
RandomRectanglePositionAllocator	Zvolení náhodné pozice v obdelníku (2D objektu)
UniformDiscPositionAllocator	Náhodné rozmístění uzlů v kruhu s rovnoměrnou (konstantní) hustotou

## A.2 Přehled metod pro pohyb uzlů

Tab. A.2: Metody pohybu uzlů [9]

Metoda	Popis
ConstantAccelerationMobilityModel	Akcelerace uzlů je konstatní dokud se nenastaví nová hodnota
ConstantPositionMobilityModel	Pozice uzlů je změněna přímo, bez pohybu, $[x1,y1] \rightarrow [x2,y2]$
ConstantVelocityMobilityModel	Rychlost uzlů je konstatní dokud se nenastaví nová hodnota
GaussMarkovMobilityModel	3D pohyb uzlů, oproti ostatním modelům má tento paměť a je proměnlivý, pohyb každého uzlu je zcela náhodný
HierarchicalMobilityModel	Tento model povoluje specifikovat odlišnou pozici nového uzlu (potomku - child) oproti vzoru (parent)
RandomDirection2dMobilityModel	Pohyb objektů je založen na náhodném směru, každý objekt se po určitou dobu nepohybuje, následně zvolí náhodný směr a rychlost, které definují pohyb dokud objekt nedojde k hranicím modelu
RandomWalk2dMobilityModel	Každý uzel se pohybuje náhodnou rychlostí a směrem (určeným minimem a maximem tohoto parametru) dokud nedosáhne stanovené vzdálenosti nebo času simulace. Pokud dosáhne hranice modelu, odrazí se dle úhlu. Model často označován jako Brownův pohyb.
RandomWaypointMobilityModel	Každý objekt začne stopnutím času na náhodnou dobu proměnné Pause. Po každé pauze je zvolena nová náhodná rychlost směrem k umístěnému bodu.
SteadyStateRandomWaypoint-MobilityModel	Založen na předchozím modelu, rozdíl v nerovnoměrné distribuci parametrů rychlosti, pauzy. Dvojměrný model s nenulovou rychlostí uzlů.
WaypointMobilityModel	Objekty mají stanovený bod, ke kterému se pohybují. Tyto body se po jejich dosažení mění, po dosažení poslední bodu (Waypoint) se uzel zastaví.

## A.3 Přehled parametrů třídy YansWifiPhyHelper

Tab. A.3: Parametry třídy YansWifiPhyHelper [9]

Název atributu	Popis	Typ	Výchozí hodnota
Frequency	Provozní frekvence	uint16_t 0:65535	0 MHz
ChannelWidth	Šířka pásma	uint16_t 5:160	20 MHz
ChannelNumber	Volba frekvenčního kanálu	uint8_t 0:196	0
TxGain	Zisk vysílací antény	double -1.79769e+308: +1.79769e+308	0 dB
RxGain	Zisk přijímací antény	double -1.79769e+308: +1.79769e+308	0 dB
TxPowerLevels	Počet vysílaných výkonových úrovní mezi TxPowerStart a TxPowerEnd	uint8_t 0:255	1
TxPowerEnd	Maximální dostupná úroveň výkonu	double -1.79769e+308: +1.79769e+308 0:255	16.0206 dBm
TxPowerStart	Maximální možná úroveň výkonu	double -1.79769e+308: +1.79769e+308	16.0206 dBm
RxNoiseFigure	Výkonová ztráta v poměru SNR (Signal-to-Noise-Ratio)	double -1.79769e+308: +1.79769e+308	7 dB
State	Stav fyzické vrstvy (obsazení, příjem, vysílání, spánek atd.)	ns3::Ptr<ns3::Wifi-PhyStateHelper>	0
ChannelSwitch-Delay	Zpoždění mezi dvěma rámci vyslaných na jiných frekvencích	Time -922337203e+18: +922337203e+18	+250000.0 ns
Antennas	Počet antén na daném zařízení	uint8_t 1:8	1

## A.4 Laboratorní úloha

Na následujících stranách je kompletní návrh laboratorní úlohy. Vzorové řešení je v kapitole Výsledky simulace 6.3

# SMĚROVACÍ PROTOKOL OLSR

OLSR je optimalizovaný stavový směrovací protokol pro ad hoc sítě (Optimized Link State Routing). Protokol je optimalizací klasického stavového algoritmu přizpůsobenému požadavkům mobilních bezdrátových LAN sítí. Klíčový koncept protokolu spočívá ve vícebodovém spojení (MPR - multipoint relay). MPR jsou vybrané uzly, které přeposílají všesměrové zprávy během procesu zahlcení. Tato technika značně redukuje režii zpráv oproti klasickým mechanismům zahlcení, kde každý uzel přeposílá každou zprávu při obdržení její první kopie. U OLSR je stavová informace generována pouze uzly zvolenými jako MPR. Druhou optimalizací je dosažení minimalizace počtu kontrolních zpráv zahrnujících síť. Třetí optimalizací je to, že si MPR uzel může vybrat ohlašování pouze stavu linek mezi ním a uzly, které si jej jako MPR vybrali. Dále je oproti klasickému stavovému algoritmu distribuována pouze částečná stavová informace, která je později použita pro kalkulaci směrování. OLSR poskytuje optimální trasy ve smyslu nejnižšího počtu skoků. Protokol je vhodný i pro velké a husté sítě díky dobré funkcionalitě vícebodových přenosů (MPR).

OLSR je vyvinut pro mobilní ad hoc sítě. Funguje jako proaktivní (tabulkový) protokol - dochází k pravidelné výměně informací o topologii s ostatními uzly sítě. Každý uzel si vybere sadu sousedních uzlů jako MPR. U OLSR jsou pouze uzly vybrané jako MPR zodpovědné za předávání kontrolních zpráv, určených k šíření po celé síti. MPR uzly tak poskytují efektivní mechanismus pro zahlcování sítě kontrolními zprávami snížením počtu potřebných přenosů.

## Použitelnost

Protokol se dobře hodí k velkým a hustým mobilním sítím díky využití vícebodového spojení (MPR). Čím větší a hustější síť, tím lepší optimalizace lze dosáhnout oproti klasickým stavovým algoritmům. OLSR používá směrování "hop-by-hop" (krok po kroku). To znamená, že každý uzel využívá aktuální lokální informace ke směrování paketů.

OLSR je vhodný více pro sítě, kde přenos je náhodný a sporadický mezi spoustou uzlů než pro malou topologii specifických uzlů. Jako proaktivní protokol je zároveň vhodný také ve scénáři, kde se komunikující dvojice uzlů časem mění. Nepřibývá žádné kontrolní dopravy, jelikož trasy jsou udržovány pro všechny distance v jakýkoliv čas.

# MPR

Myšlenkou vícebodového spojení je minimalizovat režii zahlcení sítě zprávami zredukováním redundantního přeposílání (opětovného posílání) v dané části sítě. Každý uzel ( $U$ ) sítě si vybírá skupinu (sadu) uzlů v jeho symetrickém jedno-skokovém okolí, která může jeho zprávy přeposlat. Tato skupina vybraných sousedů je označena jako vícebodové spojení tohoto uzlu -  $MPR(U)$ . Sousedé tohoto uzlu, kteří nejsou v jeho skupině  $MPR$  obdrží a zpracovávají jeho broadcast zprávy, ale tyto zprávy nikam nepřeposílají.

Každý uzel si vybírá svoji sadu  $MPR$  z jeho jedno-skokového okolí. Tato sada je zvolena takovým způsobem, aby pokryla všechny symetrické dvou-skokové uzly (sousedy).  $MPR$  sada uzlu  $U$ , označena jako  $MPR(U)$ , je tedy náhodná skupina jedno-skokově vzdálených sousedů uzlu  $U$  a musí splňovat následující podmínku - každý uzel dvou-skokového okolí uzlu  $U$  musí mít linku k  $MPR(U)$ . Čím menší je sada  $MPR$ , tím menší je výsledná kontrolní režie směrovacího protokolu. Každý uzel si udržuje informace o skupině sousedů, které si jej zvolili jako  $MPR$ . Skupina se nazývá "vybraná sada vícebodového spojení" daného uzlu.

## Výběr $MPR$

Výběr  $MPR$  sady je prováděn ve dvou krocích, kdy se uzel pomocí svého jedno-skokového okolí -  $N1(U)$  snaží pokrýt své dvou-skokové sousedy -  $N2(U)$ :

1. Výběr uzlů z  $N1(U)$  pokrývajících "izolované" uzly z  $N2(U)$ .
2. Uzly z  $N1(U)$ , které ještě nejsou zvoleny jako  $MPR$ , mající linku k  $N2(U)$ , které nebyly pokryty v prvním kroku (je k nim více linek). Volí se ty uzly z  $N1(U)$ , které pokryjí více uzlů z  $N2(U)$ .

# 1 POSTUP

Seznamte se s kódem simulace.

## 1.1 Model sítě

Model sítě se skládá z několika uzlů, ty jsou rozmístěny buď v mřížce (grid 4x4), případně náhodně rozmístěné v daném poli (mesh). K rozmístění uzlů do slouží třída `MobilityHelper` a její funkce `SetPositionAllocator`.

```
NodeContainer c;  
c.Create (16);
```

K rozmístění uzlů do slouží třída `MobilityHelper` a její funkce `SetPositionAllocator`. Pro tu se nastaví potřebné parametry pro metody rozmístění.

```
MobilityHelper mobility;  
mobility.SetPositionAllocator (  
    "ns3::GridPositionAllocator",  
    "MinX", DoubleValue (25.0),  
    "MinY", DoubleValue (25.0),  
    "DeltaX", DoubleValue (50.0),  
    "DeltaY", DoubleValue (50.0),  
    "GridWidth", UIntegerValue (4),  
    "LayoutType", StringValue ("RowFirst"));
```

Jak bylo zmíněno dříve, simulací bylo více, v předchozím výpisu kódu je ukázka přímého určení pozice uzlů do mřížky. Další simulace neměla pevně dané pozice uzlů, jedná se o náhodnou mesh topologii.

```
mobility.SetPositionAllocator (  
    "ns3::RandomDiscPositionAllocator",  
    "X", StringValue ("70.0"),  
    "Y", StringValue ("70.0"),  
    "Rho", StringValue (  
        "ns3::UniformRandomVariable [Min=30 | Max=70] "));
```

Zde se pomocí souřadnic X a Y určí střed kruhu a následně pomocí limit parametru Rho, která stanoví radius náhodných pozic od středu kruhu.



Po rozmístění uzlů je možnost definice, zda uzly budou stacionární

```
mobility.SetMobilityModel (  
    "ns3::ConstantPositionMobilityModel");  
mobility.InstallAll ();
```

a nebo pohyblivé. K jejich mobilitě se nabízí několik metod podtřídy Mobility-Model, ty jsou popsány v Tab.??.

```
mobility.SetMobilityModel (  
    "ns3::RandomWalk2dMobilityModel",  
    "Mode", StringValue ("Time"),  
    "Time", StringValue ("0.5s"),  
    "Distance", StringValue ("1.0"),  
    "Direction", StringValue (  
        "ns3::UniformRandomVariable [Min=0.0|Max=6.0]"), //gradient  
    "Speed", StringValue (  
        "ns3::UniformRandomVariable [Min=1.0|Max=2.0]"), //m/s  
    "Bounds", StringValue ("0|140|0|140"));
```

## 1.2 Nastavení komunikace

Mezi jednotlivými uzly je nyní potřeba vytvořit WiFi síť. K tomu se využilo několik tříd NS-3 - WifiHelper, YansWifiPhyHelper, YansWifiChannelHelper a WifiMacHelper.

```
WifiHelper wifi;  
YansWifiPhyHelper wifiPhy = YansWifiPhyHelper::Default();  
wifiPhy.Set("RxGain", DoubleValue(-32));  
wifiPhy.SetPcapDataLinkType(  
    ns3::YansWifiPhyHelper::DLT_IEEE802_11);  
YansWifiChannelHelper wifiChannel;  
wifiChannel.SetPropagationDelay(  
    "ns3::ConstantSpeedPropagationDelayModel");  
wifiChannel.AddPropagationLoss(  
    "ns3::FriisPropagationLossModel");  
wifiPhy.SetChannel(wifiChannel.Create());  
ns3::NqosWifiMacHelper wifiMac;  
wifi.SetStandard(WIFI_PHY_STANDARD_80211b);  
wifi.SetRemoteStationManager(  
    "ns3::ConstantRateWifiManager", "DataMode",  
    StringValue(phyMode), "ControlMode", StringValue(phyMode));
```

Třída `YansWifiPhyHelper` nám umožňuje nastavit fyzickou vrstvu modelu OSI, jednotlivé nastavitelné parametry najdete na webu <https://www.nsnam.org/>. Každý parametr má výchozí hodnotu, pro zisk vysílání/přijímání je to 0 dB. Z hlediska zjednodušení tohoto nastavování byla upravena pouze hodnota `RxGain` - zisk přijímací antény. WiFi kanál byl nastaven pomocí třídy `YansWifiChannelHelper`, která doplňuje nastavení fyzické vrstvy (`YansWifiPhyHelper`) o ztrátovost a zpoždění, obě tyto třídy se standardně využívají současně.

```
wifiMac.SetType("ns3::AdhocWifiMac");  
NetDeviceContainer devices = wifi.Install(wifiPhy, wifiMac, c);
```

Následně se pomocí třídy `WifiMacHelper` určí využití kvality služeb (`Qos/Nqos`), zvolí se typ podvrstvy MAC, v našem případě `AdhocWifiMac` a pomocí třídy `WifiHelper` se aplikuje toto nastavení na kontejner zařízení spolu s využitím některého z WiFi standardů (802.11b), tedy dojde k vytvoření komunikačního WiFi prostředí. Po nastavení komunikačního prostředí je nutné těmto uzlům nastavit IP adresy.

```
Ipv4AddressHelper ipv4;  
ipv4.SetBase("10.1.1.0", "255.255.255.0");  
Ipv4InterfaceContainer i = ipv4.Assign(devices);
```

Posledním bodem nastavení komunikace je určení směrování mezi uzly v topologii. K tomu slouží protokol OLSR.

```
OlsrHelper olsr;  
Ipv4StaticRoutingHelper staticRouting;  
Ipv4ListRoutingHelper list;  
list.Add(olsr, 10); // 10 - priorita  
InternetStackHelper internet;  
internet.SetRoutingHelper(list);  
internet.Install(c);
```

## 1.3 Generace datového provozu

Po nastavení komunikace už zbývá poslední část simulace a tou je generace datového provozu. Možností v NS-3 simulátoru je pro to více, v rámci jednoduché komunikace mezi dvěma uzly byla využita aplikace UdpEchoServer, stanovující zdroj komunikace a jeden nebo více cílových bodů - uzlů.

```
TypeId tid = TypeId::LookupByName("ns3::UdpSocketFactory");
Ptr<Socket> recvSink = Socket::CreateSocket(
c.Get(sinkNode), tid);
InetSocketAddress local = InetSocketAddress(
Ipv4Address::GetAny(), 80);
recvSink->Bind(local);
recvSink->SetRecvCallback(MakeCallback(&ReceivePacket));
Ptr<Socket> source = Socket::CreateSocket(
c.Get(sourceNode), tid);
InetSocketAddress remote = InetSocketAddress(
i.GetAddress(sinkNode, 0), 80);
source->Connect(remote);
```

## 1.4 Výstupy simulace

Po spuštění simulace dojde ke komunikaci mezi jednotlivými uzly. OLSR protokol využívá pro vytvoření topologie směrování broadcast, to znamená, že každý jeden uzel začne do svého okolí, v rámci svého dosahu, vysílat zprávy typu HELLO k detekci sousedních uzlů a ověřování konektivity, viz Obr.1.1 - výstup pcap souboru v programu Wireshark.

```
▼Message: HELLO (1)
  Message Type: HELLO (1)
  Validity Time: 1,500 (in seconds)
  Message: 56
  Originator Address: 10.1.1.10 (10.1.1.10)
  TTL: 1
  Hop Count: 0
  Message Sequence Number: 3
  Hello Emission Interval: 0,500 (in seconds)
  Willingness to forward messages: Unknown (3)
  ▶Link Type: Symmetric Link (6)
  ▶Link Type: MPR Link (10)
  ▶Link Type: Symmetric Link (6)
  ▶Link Type: MPR Link (10)
  ▶Link Type: Symmetric Link (6)
```

Obr. 1.1: Detekce sousedních uzlů pomocí zpráv HELLO

Pro jednotlivé zprávy protokolu OLSR jsou nastaveny výchozí intervaly, pro zprávu HELLO jsou to 2s, pro ostatní 5s. Ačkoliv by se tento interval mohl zdát vysoký, již po několika zprávách zná každý uzel své sousedy a zvolí, které mu budou sloužit jako MPR jak lze vidět na Obr.1.1.

Dalším z výstupů simulace je textový soubor, který pro jednotlivé uzly vypisuje jejich směrovací tabulku se stanoveným intervalem - "OLSR.routes". Na začátku aktualizace zná každý uzel pouze svoji localhost adresu (127.0.0.0) a adresu sítě (10.1.1.0), po určité době zná každý uzel cestu ke každé destinaci v síti, viz Obr.1.4.

```

Node: 9 Time: 12s Ipv4ListRouting table
Priority: 10 Protocol: ns3::olsr::RoutingProtocol
Destination      NextHop          Interface        Distance
10.1.1.1         10.1.1.9        1                3
10.1.1.2         10.1.1.4        1                2
10.1.1.3         10.1.1.4        1                2
10.1.1.4         10.1.1.4        1                1
10.1.1.5         10.1.1.4        1                2
10.1.1.6         10.1.1.4        1                2
10.1.1.7         10.1.1.4        1                3
10.1.1.8         10.1.1.8        1                1
10.1.1.9         10.1.1.9        1                1
10.1.1.11        10.1.1.9        1                2
10.1.1.12        10.1.1.12       1                1
10.1.1.13        10.1.1.9        1                2
10.1.1.14        10.1.1.14       1                1
10.1.1.15        10.1.1.4        1                2
10.1.1.16        10.1.1.4        1                3
HNA Routing Table:
Priority: 0 Protocol: ns3::Ipv4StaticRouting
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
127.0.0.0        0.0.0.0         255.0.0.0       U      0     -    -    0
10.1.1.0         0.0.0.0         255.255.255.0  U      0     -    -    1

```

Obr. 1.2: Směrovací tabulka uzlu 9 v čase 12s

Dále je možno využít programu NetAnim, který zobrazuje zvolenou topologii a komunikaci. V záložce Stats je pak díky implementaci aplikace FlowMonitor možné kontrolovat statistiky sítě.

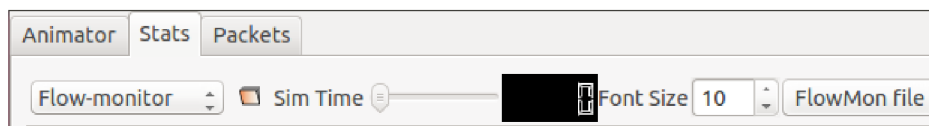
# Úkoly

Po sestavení směrovacích tabulek je možno generovat datový provoz. K úkolům využijte příklady v adresáři NS-3 a nebo web <https://www.nsnam.org/>

## 1. Sestavte komunikační dvojici pomocí UdpSocketFactory

Funkce pro generování provozu je stanovena na začátku kódu, stačí přiřadit hodnoty proměnným "sourceNode" a "sinkNode" (dle pořadí uzlu).

V případě vyslání paketu před sestavením směrovacích tabulek dochází k ARP dotazům, ale bez odpovědi. V momentě sestavených směrovacích tabulek již probíhá doručování paketu a směrování, to u protokolu OLSR probíhá vždy lokálně. Uzel vyšle paket vždy všem okolním uzlům - broadcast i když má paket danou cílovou adresu. V programu NetAnim v záložce Stats vyberte možnost Flow-monitor a pomocí tlačítka FlowMon file načtete soubor "/home/student/ns-allinone-3.21/ns-3.21.xml" k zobrazení statistik simulace.



## 2. Změňte dobu odeslání prvního paketu

Ve výchozí podobě je první paket odeslán v čase 1 s, dle statistik je ztrátovost vysoká. Pomocí program Wirekshark zjistíte po jak dlouhé době od začátku simulace dojde k úspěšnému odeslání všech paketů.

## 3. Upravte kód tak, aby se uzly pohybovaly a měňte interval OLSR zpráv

V různých situacích jsou potřeba odlišné intervaly pro sestavení směrovacích tabulek, v případě stacionárních uzlů mohou být výchozí (2 s), případně delší. V momentě kdy jsou zvoleny sady MPR uzlů dochází pouze k ověřování stávajících tras. Pokud se jedná o síť s mobilními uzly, intervaly zpráv mohou být kratší, aby se zamezilo případné ztrátovosti. Zkontrolujte statistiky v NetAnim.

## 4. Vytvořte vlastní topologii s co nejnižší ztrátovostí

Rozšiřte aktuální topologii, upravte tomu rozestavení uzlů - rozsah sítě, aby pokrytí jednotlivých uzlů nebylo příliš velké a nastavte intervaly a jiné potřebné parametry simulace tak, aby došlo k minimální ztrátovosti.

## Kontrolní otázky

- Jak probíhá směrování OLSR, jakým způsobem jsou zvoleny sady MPR?
- Kolik přenosů (intervalů) zpráv HELLO je potřeba k získání kompletních směrovacích informací tabulek?