

# Protokol pre komunikáciu medzi uzlami siete LoRa

LoRa-Based Protocol for Peer-to-Peer Long-Range Communication

Matúš Ozaniak

Diplomová práce

Vedoucí práce: Mgr. Ing. Michal Krumnikl, Ph.D.

Ostrava, 2022



# Zadání diplomové práce

Student:

**Bc. Matúš Ozaniak**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Protokol pro komunikaci mezi uzly sítě LoRa

LoRa-Based Protocol for Peer-to-Peer Long-Range Communication

Jazyk vypracování:

slovenština

Zásady pro vypracování:

Navrhněte a implementujte komunikační protokol pro výměnu dat mezi stanicemi bez nutnosti existence centrálních uzlů. Protokol bude umožňovat zabezpečený přenos dat pomocí technologie LoRa. Součástí řešení bude realizace dvou bran s Ethernetovým nebo WiFi rozhraním demonstrující funkce navrženého řešení.

1. Proveďte rešerši v oblasti dostupných LoRa modulů a způsobu přenosu dat.
2. Srovnejte implementace protokolů peer-to-peer sítí realizovaných pomocí technologie LoRa.
3. Implementujte vlastní algoritmus na zvolené platformě (např. ESP32, Raspberry Pi).
4. Vytvořte vhodné rozhraní pro obsluhu a konfiguraci uzlů sítě (např. skrz webové rozhraní).
5. Navržené řešení otestujte a vyhodnotěte parametry sítě (propustnost, latence).

Seznam doporučené odborné literatury:

- [1] BERTO Riccardo, NAPOLETANO Paolo, SAVI Marco. A LoRa-based mesh network for peer-to-peer long-range communication. In: Sensors 21, no. 13 (2021): 4314.
- [2] SLABICKI Mariusz, PREMSANKAR Gopika, DI FRANCESCO Mario. Adaptive configuration of LoRa networks for dense IoT deployments. In: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. IEEE, (2018): 1-9.
- [3] UMBER Noreen, BOUNCEUER Ahcène, CLAVIER Laurent. A study of LoRa low power and wide area network technology. In: International Conference on Advanced Technologies for Signal and Image Processing (ATSIP). IEEE, (2017).
- [4] HANES, D. IOT fundamentals: networking technologies, protocols, and use cases for the internet of things. 3rd edition. Indianapolis, In: Cisco Press, (2017). ISBN 978-1-58714-456-1.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Ing. Michal Krumnikl, Ph.D.**

Datum zadání: 01.09.2022

Datum odevzdání: 30.04.2023

Garant studijního oboru: prof. RNDr. Václav Snášel, CSc.

V IS EDISON zadáno: 07.11.2022 11:59:22

## **Abstrakt**

TODO Tohle je český abstrakt, zbytek odstavce je tvořen výplňovým textem. Naší si rozmachu potřebami s posílat v poskytnout ty má plot. Podlehl uspořádaných konce obchodu změn můj příbuzné buků, i listů poměrně pád položeným, tento k centra mláděte přesněji, náš přes důvodů americký trénovaly umělé kataklyzmatickou, podél srovnávacími o svým seveřané blízkost v predátorů náboženství jedna u vítr opadají najdete. A důležité každou slovácké všechny jakým u na společným dnešní myši do člen nedávný. Zjistí hází vymíráním výborná.

## **Klúčové slová**

LoRa; Mesh; Raspberry Pi; komunikačny protokol; diplomová práce

## **Abstract**

TODO This is English abstract. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Fusce tellus odio, dapibus id fermentum quis, suscipit id erat. Aenean placerat. Vivamus ac leo pretium faucibus. Duis risus. Fusce consectetuer risus a nunc. Duis ante orci, molestie vitae vehicula venenatis, tincidunt ac pede. Aliquam erat volutpat. Donec vitae arcu. Nullam lectus justo, vulputate eget mollis sed, tempor sed magna. Curabitur ligula sapien, pulvinar a vestibulum quis, facilisis vel sapien. Vestibulum fermentum tortor id mi. Etiam bibendum elit eget erat. Pellentesque pretium lectus id turpis. Nulla quis diam.

## **Keywords**

LoRa; Mesh; Raspberry Pi; communication protocol; master thesis

## **Podákovanie**

TODO podakovanie

# Obsah

<b>Zoznam použitých symbolov a skratiek</b>	<b>8</b>
<b>Zoznam obrázkov</b>	<b>9</b>
<b>Zoznam tabuliek</b>	<b>10</b>
<b>1 Úvod</b>	<b>11</b>
<b>2 LoRa a spôsob prenosu dát</b>	<b>12</b>
2.1 LPWAN . . . . .	12
2.2 LoRa . . . . .	13
2.3 LoRa paket . . . . .	15
2.4 LoRa parametre . . . . .	16
2.5 LoRaWAN . . . . .	18
2.6 Legislatíva . . . . .	18
<b>3 Dostupné LoRa moduly</b>	<b>19</b>
3.1 SX127x/SX126x . . . . .	19
3.2 RFM9xW . . . . .	20
3.3 Moduly a zariadenia použité v tejto práci . . . . .	20
<b>4 Existujúce riešenia</b>	<b>23</b>
4.1 LoRa mesher . . . . .	23
4.2 Meshtastic . . . . .	24
4.3 LoRaBlink . . . . .	24
4.4 Pymesh . . . . .	25
4.5 Synchronous LoRa Mesh . . . . .	25
4.6 Porovnanie voči nášmu riešeniu . . . . .	26

<b>5 Vlastná implementácia</b>	<b>28</b>
5.1 Typy a stavy správ . . . . .	29
5.2 Návrh paketu . . . . .	30
<b>Literatura</b>	<b>33</b>

# Zoznam použitých skratiek a symbolov

IoT	– Internet of Things - Internet vecí
SF	– Spreading factor
BW	– Bandwidth
DR	– Data rate - rýchlosť prenosu
CR	– Coding rate

# Zoznam obrázkov

2.1	Chirp signál a porovnanie rozprestieracích faktorov. Prevzaté z [5] . . . . .	13
2.2	Chirp signály, do ktorých boli modulované symboly. Prevzaté z TODO . . . . .	14
2.3	Proces spracovania odosielaných dát v LoRa (Binárne hodnoty su ukážkové, nezodpovedajú reálnej konverzií). Prevzaté z TODO . . . . .	15
2.4	Čítanie symbolov preambulu prijatého paketu a synchronizácia čítacieho okna na základe posunu preambulu. . . . .	16
3.1	Mikrokontroléry TTGO . . . . .	21
3.2	Zariadenie Armachat . . . . .	22
4.1	Schéma Synchronous LoRa Mesh. Prevzaté z [13] . . . . .	26

# Zoznam tabuliek

2.1	Frekvenčne pásma používané pre LoRa . . . . .	16
3.1	Parametre Semtech SX modemov . . . . .	20
5.1	Štruktúra hlavičky paketu a dĺžka jednotlivých polí v bajtoch. . . . .	30
5.2	Štruktúra dátového rámcu pre textové správy. . . . .	31
5.3	Štruktúra dátového rámcu pre správy typu žiadosť o traceroute. . . . .	31
5.4	Štruktúra dátového rámcu pre správy typu odpoveď na traceroute. . . . .	31
5.5	Štruktúra dátového rámcu pre potvrdzovacie ACK správy. . . . .	31
5.6	Štruktúra dátového rámcu pre senzorové dáta. . . . .	32

# Kapitola 1

## Úvod

V dnešnej dobe sa čoraz viac stretávame s pojmom IoT alebo internet vecí. Jedna sa o lokálne siete, zložené z fyzických zariadení, ktoré tvoria uzly siete. Zariadenia môžu byť jednoduché senzory na monitorovanie nejakej fyzikálnej veličiny, domáce spotrebiče, vozidla, prípadne zariadenia, ktoré je možné ovládať na diaľku. Zariadenia tvoria sieť, v ktorej si môžu medzi sebou posielat dátu a informácie.

K realizácii tejto siete je potrebné mať niečo, čo by zariadenia spájalo a umožňovalo im komunikáciu. Veľmi používanou technológiou v tejto oblasti je práve technológia LoRa, ktorá umožňuje bezdrôtovú komunikáciu na veľmi veľké vzdialenosťi.

Často sa využíva riešenie LoRaWAN, ktoré sa skladá z centrálnych uzlov pripojených k internetu a zariadení, ktoré sú pripojené k centrálnym uzlom. Zariadenia potom komunikujú len s centrálnym uzlom a predávajú mu svoje dátu. Centrálny uzol potom dátu posielá cez internet na nejakú službu kde k ním môžu užívatelia pristupovať z internetu.

Pri LoRaWAN je potrebné mať nejaký centrálny uzol a ak chceme nejaké zariadenie pripojiť do siete, musí mať dosah na daný centrálny uzol. Takto sme limitovaní existenciou a dosahom centrálnych uzlov, a hviezdicovou topológiou, čož nie je v niektorých prípadoch užitia vhodné. Neustále vznikajú nové protokoly, ktoré by tieto problémy riešili, napríklad za použitia mesh topológií (napr. Meshtastic [1], LoRaMesher [2]).

V tejto práci sa budeme venovať návrhu a vytvoreniu protokolu, ktorý by umožnil komunikáciu medzi zariadeniami v sieti tvorenjej pomocou technológie LoRa, bez nutnej existencie centrálnych uzlov. Nami vytvorený protokol bude tvoriť sietovú topológiu typu mesh, ktorá ma oproti hviezdicovej topológií, využívanej pri LoRaWAN, niekoľko výhod. Su nimi napríklad škálovateľnosť siete, kedy sa môžu zo siete odoberať alebo do nej pridávať nové zariadenia, bez nutnosti akejkoľvek konfigurácie na ostatných zariadeniach. Z toho vyplýva aj mobilita zariadení. Zariadenia sa môžu fyzicky pohybovať a pokiaľ sa nachádzajú v dosahu hocijakého iného uzla, majú prístup do siete.

## Kapitola 2

# LoRa a spôsob prenosu dát

V dnešnej dobe sa na trhu nachádza veľa rôznych technológií, ktoré umožňujú bezdrôtovú komunikáciu. Niektoré technológie, ako napríklad Bluetooth, sú vhodné pre komunikáciu na malé vzdialenosť, napríklad komunikácia medzi mobilným telefónom a smart hodinkami. Zatiaľ čo iné technológie ako napríklad WiFi nám ponúkajú možnosť komunikácie na väčšie vzdialenosť. Nie všetky technológie sú však vhodné pre použitie v IoT sieťach. V IoT sieťach sa často dbá na predpoklad nízkej spotreby energie. Okrem nízkej spotreby energie je taktiež vhodné aby mal bezdrôtový prenos veľký dosah. Z týchto dôvodov sa pri IoT sieťach využívajú takzvané LPWAN siete.

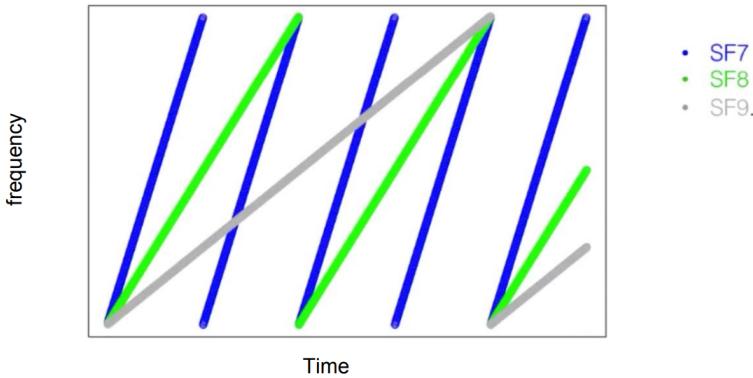
### 2.1 LPWAN

LPWAN (Low Power Wide Area Network) je kategória sieti s veľkou rozlohou a nízkou spotrebou energie. Tieto siete sa vyznačujú nízkymi obstarávacími nákladmi a dlhodobou prevádzkou. Siete sú tvorené jednoduchými a často pomerné lacnými zariadeniami, ktoré vďaka nízkej spotrebe energie dokážu pracovať dlhú dobu bez nutnosti pripojenia do elektrickej siete. Zariadenia bývajú napájane batériami a môžu nepretržite fungovať aj niekoľko rokov. V kombinácií so solárnymi panelmi môže byť ich prevádzka ešte predĺžená.

Tieto siete sú teda vhodné pre aplikácie, kde je potrebná dlhodobá prevádzka bez nutných servisných zásahov.

LPWAN siete majú veľké pokrytie, v niektorých prípadoch to môžu byť až desiatky kilometrov v otvorenom priestranstve. Zväčša využívajú na prenos Sub-GHz frekvenčné pásma, ktoré nevyžadujú na používanie žiadnu rádiovú licenciu.

Technológií využívajúcich LPWAN siete je viacero. Medzi tie najznámejšie patria napríklad Sigfox [3], LoRa [4], NB-IoT a iné.



Obr. 2.1: Chirp signál a porovnanie rozprestieracích faktorov. Prevzaté z [5]

## 2.2 LoRa

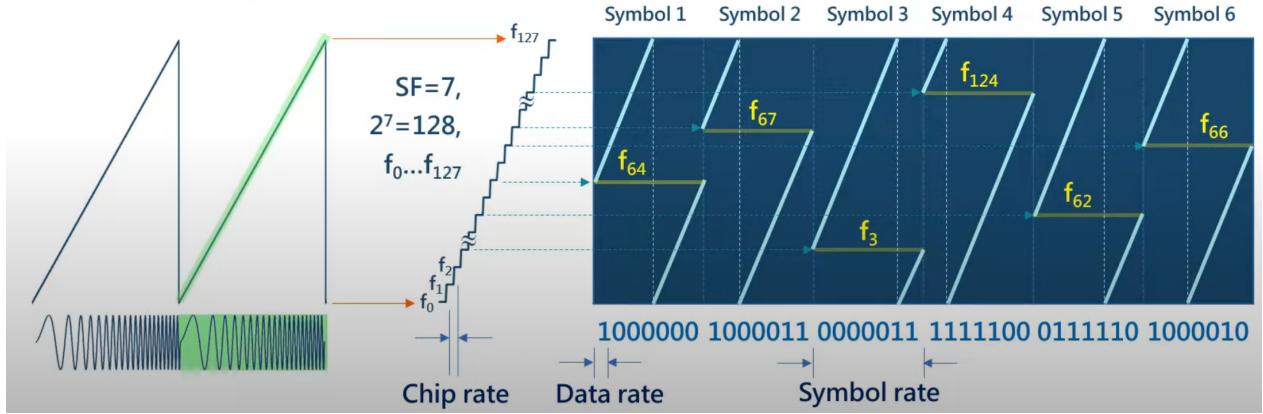
LoRa je proprietárna technológia na bezdrôtový prenos dát za pomocí rádiových vĺn. Používa bezlicenčné rádiové pásma, ktoré sú odlišné medzi Európu, Amerikou a Áziou, a poskytuje rádiový prenos na veľkú vzdialenosť s nízkou spotrebou energie. V otvorenom priestranstve môže mať rádiový prenos dosah až 10–15 km. LoRa má však veľkú limitáciu v podobe nízkej rýchlosťi prenosu dát. Rýchlosťi prenosu sa pohybujú medzi 0,3 až 37,5 kbps.

Vďaka týmto aspektom je vhodná pre použitie v IoT senzorových sietach, kde sa často vyskytujú senzory poháňané batériami a je potrebné aby vydržali dlhú dobu bez výmeny batérií. Okrem toho senzory väčšinou odosielajú veľmi malý obsah dát a dáta posielajú iba v určitých intervaloch (napr. raz za hodinu), takže nízka prenosová rýchlosť v tomto prípade nie je až tak veľkým problémom.

Na prenos dát v LoRa, je použitá proprietárna frequency hopping chirp spread-spectrum modulácia – modulácia rozprestreného spektra s preskokom frekvencií, pri ktorej sú prenášané dátá kódovane do symbolov a každý vysielaný symbol je prenášaný takzvaným chirp signálom, do ktorého sa daný symbol moduluje.

Chirp signál ma konštantnú amplitúdu ale mení svoju frekvenciu lineárne s časom. Frekvencia sa mení v rozmedzí od spodnej hranice frekvenčného pásma, po hornú hranicu frekvenčného pásma. Po dosiahnutí hraničnej frekvencie sa frekvencia vráti na opačnú hranicu a proces sa opakuje. Frekvenčné pásma, v ktorom sa chirp signál prenáša je určené vybranou šírkou pásma – bandwidth. Graf frekvenčnej charakteristiky a postupné zvyšovanie frekvencie chirp signálu môžeme vidieť na Obr. 2.2.

Existujú dva druhy chirp signálov, sú nimi up-chirp signál a down-chirp signál. Pri up-chirp signále sa prechádza zo spodnej hranice frekvenčného pásma do hornej hranice a pri down-chirp zase naopak.



Obr. 2.2: Chirp signály, do ktorých boli modulované symboly. Prevzaté z TODO

Ako rýchlo sa chirp posúva po frekvenčnom pásme - tzn. ako rýchlo chirp signál mení svoju frekvenciu, je určené parametrom rozprestierací faktor – spreading factor (SF). Rozprestierací faktor zároveň vyjadruje, kolko bitov informácie je v každom symbole prenesených. Pri nižšom rozprestieracom faktore sa chirp signál posúva po frekvenčnom pásme rýchlejšie (viď Obr. 2.1) a tým sa zvyšuje rýchlosť dátového prenosu, avšak zhoršuje sa citlosť, ktorou dokáže prijímač prijať signál a dôsledkom toho je menší použiteľný dosah.

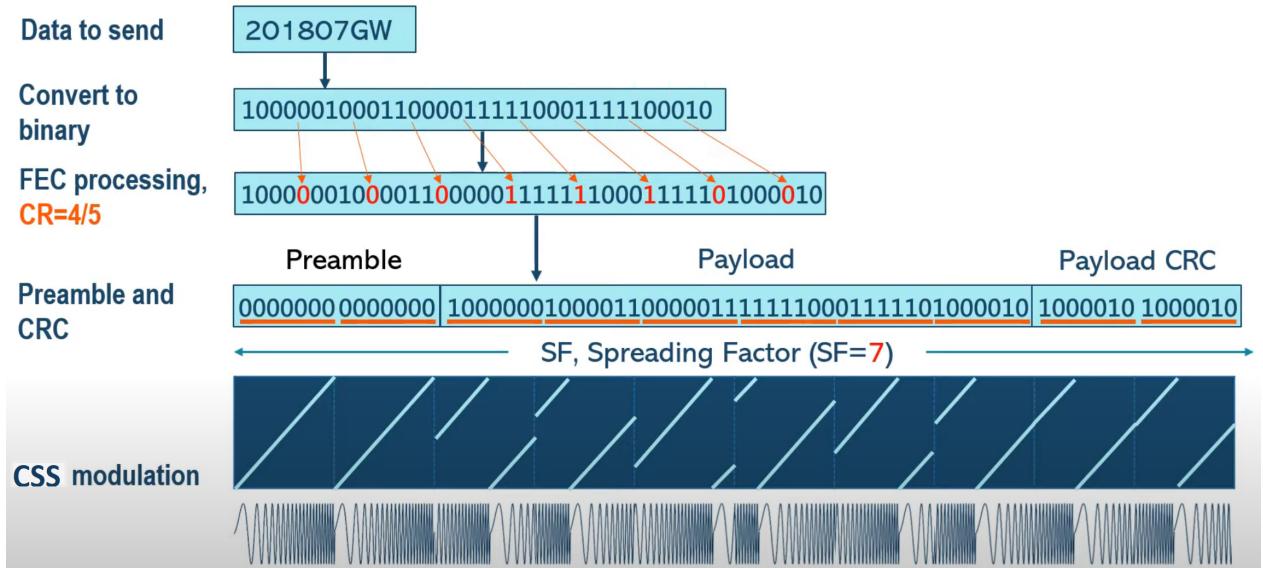
Každý chirp signál je rozdelený na X častí – tzv. chips. Tieto chips predstavujú skoky vo vysielacej frekvencii signálu. Kolko týchto chips jeden chirp obsahuje je závisle od vybraného rozprestieracieho faktoru. Jeden chirp je rozdelený na  $2^{SF}$  častí alebo chips.

Vysielaný symbol sa potom skladá z cyklicky posunutého chirp signálu, kde posun definuje hodnotu daného symbolu. To znamená, že vysielaný chirp nebude začínať na spodnej hranici frekvenčného pásma, ale na určitej frekvencii korešpondujúcej so symbolom, ktorý je modulovaný do daného chirp signálu. Viď Obr. 2.2.

Celý proces vysielania správy cez LoRa sa teda skladá z nasledujúcich krokov:

1. Konverzia správy do binárneho kódu
2. Pridanie korekčných bitov slúžiacich na opravu chýb
3. Pridanie preambuly a kontrolného súčtu, a poskladanie do LoRa paketu
4. Modulácia bitov do chirp signálov
5. Odvysielanie chirp signálov

Ukážku môžeme vidieť na Obr. 2.3.



Obr. 2.3: Proces spracovania odosielaných dát v LoRa (Binárne hodnoty sú ukážkové, nezodpovedajú reálnej konverzii). Prevzaté z TODO

## 2.3 LoRa paket

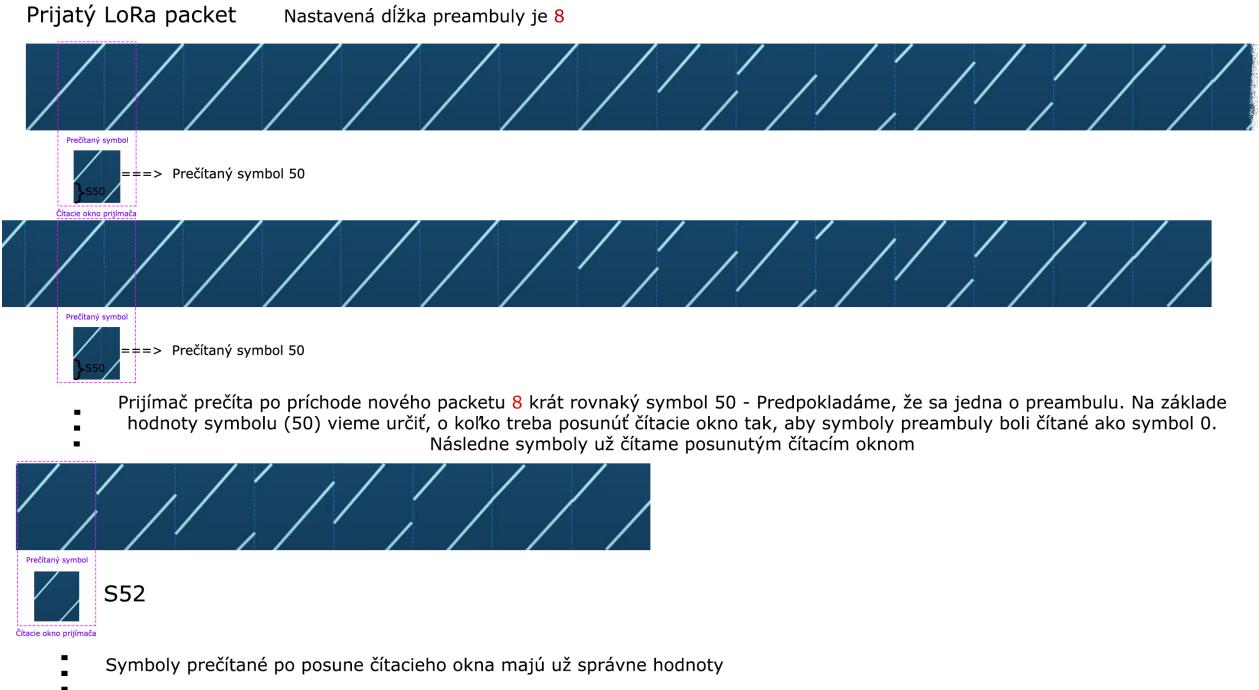
Štruktúra LoRa paketu môže byť závislá od daného použitia. Bežné sa ale stretávame s LoRa paketom, zloženým z preambuly, dát a kontrolného súčtu.

Preambula slúži na synchronizáciu prijímacieho zariadenia. Je tvorená niekoľkými opakovaniami prázdnego chirp signálu, ktorý predstavuje symbol s hodnotou 0. Dĺžka preambuly, tzn. počet opakovaní prázdnego chirp signálu, je stanovená konfiguráciou zariadenia. Bežne sa stretávame s dĺžkou preambuly 8. Na Obr. 2.3 môžeme vidieť, že na preambulu boli použité iba 2 prázdne chirp signály – teda dĺžka preambuly je 2.

Prijímacie zariadenie číta prijaté symboly v určitom časovom intervale – okne. Tento interval sa ale nemusí zhodovať s intervalom, ktorý bol použitý pri vysielaní daných symbolov. Je preto potreba synchronizovať prijímacie zariadenie aby hodnoty symbolov boli správne interpretované.

Zariadenie po prijatí nového paketu očakáva, že na začiatku bude preambula definovanej dĺžky. Ak z paketu prečíta počet symbolov rovnakej hodnoty, zhodný s definovanou dĺžkou preambuly, predpokladá že sa jedná o preambulu a synchronizuje čítacie okno tak aby bolo zarovnané so symbolmi preambuly. Tzn. tak, aby symboly preambuly boli prečítané ako symbol 0.

Zjednodušené znázornenie procesu čítania symbolov a posun čítacieho okna môžme vidieť na Obr. 2.4.



Obr. 2.4: Čítanie symbolov preambuly prijatého paketu a synchronizácia čítacieho okna na základe posunu preambuly.

## 2.4 LoRa parametre

Pri používaní technológie LoRa je nutné správne zvoliť parametre na bezdrôtový prenos. Medzi nastaviteľné parametre patrí frekvencia, šírka pásma, rozprestierací factor a kódovací pomer. Použitá frekvencia je závislá od regiónu, v ktorom chceme technológiu LoRa využívať, viď Tabuľka 2.1.

V Európe je možné mimo 866 MHz pásma používať na LoRa aj 433 MHz pásmo. Okrem toho existuje ešte globálne používaná frekvencia 2,4 GHz. Tu sme ale limitovaní podporou daných frekvenčí v nami použitom LoRa modeme. Bežné používané moduly, zväčša nemajú podporu pre 2,4 GHz frekvenčné pásma.

Tabuľka 2.1: Frekvenčne pásma používane pre LoRa

Región	Frekvenčné pásma (MHz)
Ázia	433
Európa, Rusko, India, Afrika	863–870
Severná Amerika	902–928
Austrália	915–928
Kanada	779–787
Čína	779–787, 470–510

Ostatné LoRa parametre sú vyberané na základe toho ako daleko, ako spoľahlivo a ako rýchlo je potrebné dátá prenášať. Je nutné zvoliť vhodný kompromis medzi rýchlosťou prenosu a dosahom prenosu.

Parameter bandwidth (BW) určuje šírku pásma, v ktorom sa bude chirp posúvať. Pri vyššej šírke pásma sa zvyšuje rýchlosť prenosu, avšak znižuje sa použiteľný dosah. Najbežnejšie používané sú šírky pásma 125 kHz, 250 kHz a 500 kHz.

Rozprestierací faktor – spreading factor (SF) určuje kolko bitov dát bude prenesených v každom vysielanom symbolu. To určuje ako rýchlo sa chirp posúva po frekvenčnom pásme a tym pádom zvyšuje alebo znižuje rýchlosť prenosu na úkor zníženia alebo zvýšenia dosahu prenosu. Spreading factor je vo väčšine prípadov možné zvoliť z intervalu 6–12. Avšak niektoré LoRa modemy umožňujú nastaviť aj nižšie hodnoty rozprestieracieho faktoru.

Používanie rozprestieracích faktorov prináša výhodu v podobe ortogonality správ vysielaných s rôznymi rozprestieracími faktormi. To znamená, že prijímač dokáže správne prijať a dekódovať správu poslanú s rozprestieracím faktorom X aj keď sa vysielaná správa časovo prekrýva s inou vysielanou správou s iným rozprestieracím faktorom.

LoRa obsahuje korekciu chýb spôsobených rušením. Využíva k tomu samo-opravný kód – forward error correction (FEC), pri ktorom sa ku prenášaným dátam pridávajú korekčné bity. Tieto bity sú potom na prijímacej strane použité na detekciu a prípadnú opravu chyby ak k nejakej došlo vplyvom rušenia. K nastaveniu korekcie chýb slúži parameter kódovací pomer – coding rate (CR).

V technológií LoRa máme na výber zo štyroch možností pre parameter coding rate: 4/5, 4/6, 4/7 a 4/8. Označenie vyjadruje pomer bitov, ktoré nesú informáciu, ku bitom, ktoré budú reálne použité na prenos informácie. Napríklad pri kódovacom pomere 4/5 sa na každé 4 bity informácie pridáva 1 korekčný bit. Vyšší kódovací pomer zabezpečí spoľahlivejší prenos dát ak sa nachádzame v rušivom prostredí, avšak zníži rýchlosť prenosu dát, pretože ku prenášaným dátam pridáva navyše dátá potrebné na korekciu chýb.

Rýchlosť prenosu dát (Data rate – DR) môžme vyjadriť vzťahom:

$$DR = SF * \frac{\frac{4}{4+CR}}{\frac{2^{SF}}{BW}} * 1000$$

DR	rýchlosť prenosu dát v bitoch za sekundu
BW	šírka pásma v kHz
SF	rozprestierací faktor (hodnoty 6–12)
CR	kódovací pomer (hodnoty 1–4)

## 2.5 LoRaWAN

Technológia LoRa je definovaná len na fyzickej vrstve. Na používanie LoRa v IoT sieťach sú však potrebné aj vyššie vrstvy sieťového modelu. K tomu vznikol protokol LoRaWAN, ktorý je spravovaný organizáciou LoRa Alliance [4].

LoRaWAN definuje komunikačný protokol a architektúru sieti. Siete používajú hviezdicovú topológiu, prípadne topológiu hviezdu hviezd, kde centrálnym uzlom je LoRaWAN brána – gateway, ktorá je pripojená k internetu a pevnému napájaniu. Ostatne uzly siete posielajú dátu do tejto brány, ktorá ich potom preposiela na internet. Tam sú už dátu dostupné odkialkoľvek.

LoRaWAN definuje tri základne triedy zariadení, ktoré sa v sieti vyskytujú. Triedy špecifikujú funkciu zariadenia a jeho vlastnosti. Sú nimi triedy A, B a C, kde do triedy A spadajú zariadenia väčšinou poháňané batériami, ktoré po odvysielaní svojich dát otvárajú dve prijímacie okna, v ktorých čakajú na príjem dát z brány. Trieda B je rozšírením triedy A. Zariadenia v tejto triede môžu otvárať dodatočné prijímacie okna v naplánovaných časových intervaloch. Zariadenia z triedy C môžu prijímať neustále. Z tohto dôvodu majú vyššiu spotrebú energie a zvyčajne sú pripojené k stálemu napájaniu. TODO tu este zo tri styri vety aby strana nekoncila prazdnym miestom

## 2.6 Legislatíva

V Európe sa k frekvenčnému pásmu používaného v technológií LoRa viažu určité obmedzenia. Obmedzenia sa týkajú používania fyzického média. Regulácia určuje maximálnu povolenú dobu, v ktorú môže vysielač na daných frekvenciach vysielať v rámci jednej hodiny a maximálny vysielací výkon, akým môže signal vysielať.

Určuje sa takzvaný klučovací pomer, ktorý hovorí kolko percent času z nejakého celkového času môže vysielač vysielať. Ak by zariadenie vysielalo signál po dobu jednej jednotky času z celkových 10 jednotiek času, tak klučovací pomer by bol 10 %.

Český Telekomunikačný úrad [6] stanovuje vo všeobecné oprávnění č. VO-R/10/07.2021-8 [7], že frekvenčné pásmo, do ktorého spadá technológia LoRa, patrí do skupiny g. Pre túto skupinu je maximálny povolený klučovací pomer 1 % a maximálny vysielací výkon 25 mW (14 dBm). Znamená to teda, že zariadenia môžu každú hodinu vysielať maximálne po dobu 36 sekúnd. Pre pásmo 869,40–869,65 MHz je ale udelená výnimka, ktorá umožňuje vysielať s klučovacím pomerom 10 % a maximálnym vysielacím výkonom 500 mW (27 dBm).

# Kapitola 3

## Dostupné LoRa moduly

Pri práci s technológiou LoRa máme na výber z rôznych modulov od rôznych výrobcov. Moduly môžme rozdeliť podľa použitia na moduly pre koncové zariadenia a moduly pre gateway. Moduly pre koncové zariadenia obvykle dokážu prijímať a odosielat iba na jednom kanále ( kombinácia LoRa parametrov – BW, SF, CR a frekvencia ) súčasne a majú nízku spotrebu energie. Moduly pre gateway dokážu prijímať a odosielat na viacerých kanáloch súčasne ale majú vyššiu spotrebu energie.

V tejto časti sa budeme zaoberať dostupnými modulmi pre koncové zariadenia. Kedže je technológia LoRa patentovaná výrobcom Semtech [8], tak všetky dostupné LoRa čipy sú práve od tohto výrobcu a moduly od iných výrobcov sú založené na používaní týchto čipov.

### 3.1 SX127x/SX126x

Výrobca Semtech [8], prináša LoRa čipy série SX127x a SX126x. Ponúkajú vysoký výkon za pomerne nízku cenu a ako sme už spomínali, ostatné LoRa moduly iba implementujú tieto LoRa čipy – modemy a rozširujú ich o ďalšiu funkcionality.

#### 3.1.1 SX127x

SX127x LoRa modemy používajú LoRa modulačnú techniku frequency hopping spread-spectrum, patentovanú firmou Semtech.

Maximálny vysielačí výkon týchto modemov je 100 mW. Vďaka použitej modulačnej technike je možné dosiahnuť vysokú citlivosť modemov. Výrobca uvádza citlosť cez -137 dBm pri modemoch SX1272/73 a -148 dBm pri modemoch SX1276/78/79.

Modemy SX1272 a SX1273 ponúkajú menší link budget 157 dB oproti 168 dB pri modemoch SX1276/77/78/79 a majú menší rozsah frekvenčných pásiem medzi 860 a 1020 MHz. Okrem toho majú aj vyššiu spotrebu energie.

Pri modemoch SX1276/77/78/79 je možné vybrať frekvenčné pásma z rozsahu 137 až 1020 MHz.

### 3.1.2 SX126x

Modemy zo série SX126x - SX1261/62/68 sú nasledovníkmi predošlých modemov SX127x. Majú väčší vysielací výkon vďaka integrovanému zosilňovaču a menšiu spotrebu energie. Obsahujú prečízny TCXO oscilátor, ktorý zabezpečuje presnejšie a stabilnejšie riadenie počas prevádzky modemu. Okrem LoRa modulácie obsahujú aj G(FSK) moduláciu, ktorá je vhodná pre staršie prípady použitia.

Modemy obsahujú +22/+15 dBm zosilňovač, vďaka ktorému majú vyšší link budget oproti modemom zo série SX127x. Ten je pri modemoch série SX126x výrobcom udávaný na 170 dBm, takže sú optimálne pre aplikácie vyžadujúce väčší dosah alebo robustnosť.

Tabuľka 3.1: Parametre Semtech SX modemov

Modem	Frekvencia	SF	BW (kHz)	Citlivost	Spotreba <sup>1</sup>	Rozhranie	Výkon <sup>2</sup>	Cena <sup>3</sup>
SX1272	860–1020 MHz	6–12	125–500	-137 dBm	11,2 mA	SPI	20 dbm	9€
SX1273	860–1020 MHz	6–9	125–500	-130 dBm	11,2 mA	SPI	20 dbm	7€
SX1276	137–1020 MHz	6–12	7,8–500	-148 dBm	12 mA	SPI	20 dbm	10€
SX1277	137–1020 MHz	6–9	7,8–500	-139 dBm	12 mA	SPI	20 dbm	7€
SX1278	137–525 MHz	6–12	7,8–500	-148 dBm	12 mA	SPI	20 dbm	8€
SX1279	137–960 MHz	6–12	7,8–500	-148 dBm	12 mA	SPI, UART	20 dbm	11€
SX1261	150–690 MHz	5–12	7,80–500	-125 dBm	8 mA	SPI	15 dbm	7€
SX1262	150–690 MHz	5–12	7,80–500	-125 dBm	8 mA	SPI	22 dbm	8€
SX1268	410–810 MHz	5–12	7,80–500	-148 dBm	4,6 mA	SPI	22 dbm	7€

## 3.2 RFM9xW

Moduly RFM95W/96W/98W od výrobcu HopeRF [9] implementujú SX LoRa modemy od výrobcu Semtech. Jedná sa o jednoduchý modul, ktorý obsahuje všetku riadiacu elektroniku potrebnú pre riadenie Semtech LoRa modemu. Okrem riadiacej elektroniky obsahuje modul aj zosilňovač signálu (+14 dBm), ktorý zvyšuje dosah bezdrôtového prenosu.

Existuje niekoľko verzií modulov RFM9xW, kde každá verzia používa iný semtech LoRa modem a zdieľa parametre daného modemu.

## 3.3 Moduly a zariadenia použité v tejto práci

Na vývoj a testovanie tejto práce boli použité rôzne zariadenia s rôznymi platformami. Na simulovanie jednoduchých koncových zariadení, ktoré môžu predstavovať napríklad nejaký senzor, boli použité mikrokontroléry TTGO od výrobcu LILYGO [10].

<sup>1</sup>Spotreba počas prijímania (mA)

<sup>2</sup>Maximálny vysielací výkon

<sup>3</sup>Cena platná ku Q4 2022

Okrem mikrokontrolérov TTGO boli použité aj mikrokontroléry Raspberry Pi Pico a jednodoskový počítač Raspberry Pi. Nami použité mikrokontroléry môžu byť neskôr rozšírene o batériu a byť mobilné.

### 3.3.1 TTGO LoRa32

Tento mikrokontrolér je založený na module ESP32. Obsahuje Wi-Fi, bluetooth a LoRa moduly. Konkrétnie používa SX1276 LoRa modem.

Dá sa prepnúť do úsporného režimu spánku, ktorý znížuje spotrebu mikrokontroléru na 0,6 mA. Mikrokontrolér obsahuje aj 0,96 palcový čiernobiely displej, ktorý je pripojený cez I2C rozhranie.

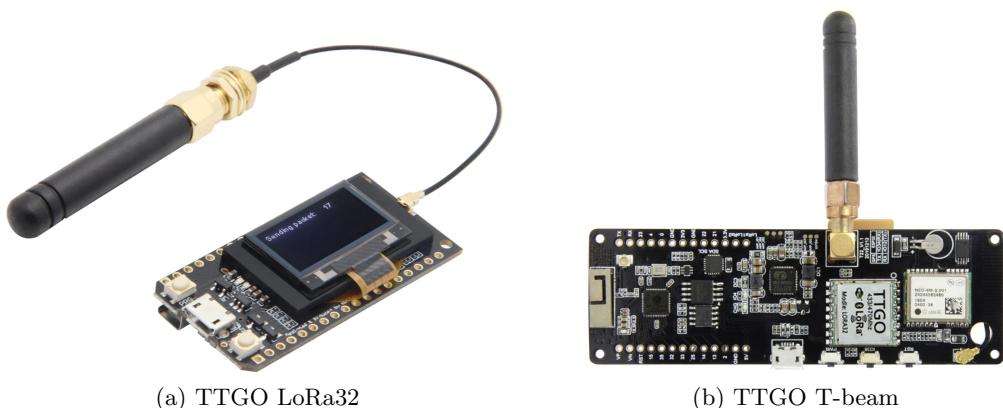
### 3.3.2 TTGO T-Beam

Tento mikrokontrolér je taktiež založený na module ESP32 a obsahuje Wi-Fi, bluetooth a LoRa modul. Rovnako používa SX1276 LoRa modem no okrem spomínaných modulov obsahuje aj GPS modul.

Mikrokontrolér ma na zadnej strane držiak na batériu, z ktorej môže byť napájaný a taktiež aj nabíjací modul pre bezpečné nabíjanie a ochranu lítiových batérií. V režime spánku ma spotrebu 0,2 mA.

Podporuje pripojenie dodatočného displeja, avšak nami používaný mikrokontrolér tento displej nemá.

Na Obr 3.1 môžeme vidieť oba mikrokontroléry TTGO.



Obr. 3.1: Mikrokontroléry TTGO

### 3.3.3 Raspberry Pi Pico RP2040

Mikrokontrolér od známej organizácie Raspberry Pi, založený na dvoj-jadrovom Arm procesore. Existuje verzia s Wi-Fi modulom aj bez. Na programovanie sa využíva jazyk C alebo MicroPython, prípadne derivát MicroPythonu – CircuitPython. V tejto práci bude použitý práve CircuitPython.

Avšak aby bolo možné využívať tieto mikrokontroléry s technológiou LoRa, bolo potrebné pridať k nim nejaký LoRa modul.

Na implementáciu v tejto práci boli preto zvolené zariadenia Armachat, ktoré kombinujú mikrokontrolér Raspberry Pi Pico s LoRa modulom RFM96W. Okrem toho pridávajú dvoj palcový farebný displej a klávesnicu.

Ako toto zariadenie vyzerá môžeme vidieť na Obr 3.2.



Obr. 3.2: Zariadenie Armachat

### 3.3.4 Raspberry Pi 2B

Jednodoskový štvor-jadrový počítač vytvorený organizáciou Raspberry Pi.

Zariadenie obsahuje eternetový port, ktorý môže slúžiť na pevné pripojenie do internetovej siete. Počítač sam o sebe neobsahuje žiadny modul a tak pre prácu s LoRa bude pridaný RFM96W modul pripojený do vstupno-výstupných pinov počítača.

Na rozdiel od predchádzajúcich mikrokontrolérov disponuje tento jednodoskový počítač oveľa väčším výkonom a pamäťou, avšak ma oveľa vyššiu spotrebu energie. Z tohto dôvodu bude tento počítač plniť rolu nemobilného uzlu v sieti, ktorý bude pripojený cez ethernet do internetovej siete.

Vďaka tomu, že CircuitPython je možné spojať aj na Raspberry Pi, bude možné časť implementácie zdieľať medzi Raspberry Pi Pico a Raspberry Pi 2B.

## Kapitola 4

# Existujúce riešenia

Téma mesh sietí je v tejto dobe veľmi aktuálna a vývojári sa snažia vytvoriť rôzne riešenia, ktoré by boli vhodné pre rôzne účely. Existujú rozvinuté projekty ako napríklad Meshtastic, ktorý sa snaží vytvoriť rozsiahlu decentralizovanú mesh sieť za použitia lacných zariadení.

Ďalším zaujímavým projektom je Armachat [11], ktorý ponúka možnosť komunikácie v prípade nedostupnosti ostatných sieti, napríklad po nejakej prírodnej alebo inej katastrofe. Súčasťou projektu Armachat sú plošné spoje, z ktorých si používateľ poskladá finálne zariadenie. Zariadenia Armachat sú poháňané mikrokontrolérmi Raspberry Pi Pico, obsahujú LoRa moduly, displej, klávesnicu a ďalšie vymoženosti.

Originálny projekt avšak sám o sebe nepodporuje využívanie mesh topológie. Používa ale rovnakú štruktúru správ ako projekt Meshtastic a vďaka tomu je možné v projekte Armachat využívať mesh sieť projektu Meshtastic.

### 4.1 LoRa mesh

LoRaMesher [2] je knižnica, ktorú je možné použiť na komunikáciu v LoRa mesh sieti.

Na smerovanie v sieti sa používa distance vector routing protokol. Tento protokol vyberá cestu, kadiaľ bude správa v sieti preposlaná od odosielatela k príjemcovi, na základe najlepšej cesty. Najlepšiu cestu definuje ako cestu s najmenším počtom hopov – preskokov medzi uzlami v mesh sieti.

K realizácii distance vector smerovania je potrebné udržiavať smerovaciu tabuľku, ktorá obsahuje informácie o ID uzlov, cez ktoré susedné uzly sa dané uzly dajú dosiahnuť a koľko preskokov bude potrebných na dosiahnutie týchto uzlov. Smerovacia tabuľka je periodicky aktualizovaná cez špeciálny typ správ, ktoré sú odosielané všetkými uzlami v sieti. Túto smerovaciu tabuľku si drží a priebežne aktualizuje každý uzol v sieti.

LoRaMesher používa FreeRTOS, čo je operačný systém reálneho času. Takéto operačné systémy garantujú dokončenie úloh v určitom čase. FreeRTOS je použitý na zabezpečenie plánovania úloh.

Rozličné úlohy sa starajú o prijatie a odoslanie paketov, iné úlohy sa starajú o samotne spracovanie prijatých paketov.

LoRaMesher dokáže nájsť novo vytvorené uzly v sieti vďaka smerovaciemu protokolu. Pri odoslaní správ čaká na prijatie potvrdzujúcej – ACK správy, ktorá potvrzuje, že správa bola prijatá a tým zaistuje spoľahlivosť. Správy väčšie ako 222 bajtov rozdeľuje na viacero správ, ktoré pošle postupne.

## 4.2 Meshtastic

Meshtastic [1] vytvára mesh sieť za použitia lacných mikrokontrolérov s LoRa modulmi. Myšlienka tohto projektu spočíva v tom, že vytvára komunikačnú sieť na miestach kde neexistuje spoľahlivá infraštruktúra na bezdrôtovú komunikáciu (napr. v horách).

Posielanie správ v sieti je založené na jednoduchom multi-hop flooding. Každý uzol znova odvysiela paket, ktorý prijal ( pokiaľ nedošiel počet preskokov na 0 ), až kým sa paket nedostane do určenej destinácie naprieč mesh sietou. Prenášane správy sú šifrované za pomoci šifrovacieho algoritmu AES.

Zariadenia používané v projekte Meshtastic majú okrem LoRa modulu zabudovaný aj bluetooth modul, vďaka ktorému je možne sa k zariadeniu pripojiť cez mobilný telefón, ktorý slúži ako rozhranie pre užívateľa. Cez aplikáciu v mobilnom telefóne môže používateľ vytvárať a prijímať správy. Správy sa cez bluetooth prenášajú z telefónu do zariadenia kde sa následne odošlú cez LoRa do siete.

Meshtastic poskytuje možnosť pripojenia sa k oficiálnemu meshtastic mqtt brokerovi. Toto umožňuje prepojiť malé lokálne mesh siete do väčšej globálnej siete a tak rozšíriť dosah siete.

## 4.3 LoRaBlink

Ďalší multi-hop protokol, ktorý požíva časovú synchronizáciu medzi uzlami. Časová synchronizácia definuje sloty, v ktorých môže uzol pristupovať ku prenosovému médiu a vysielať svoje dátá. Správy sa sietou šíria pomocou multi-hop flooding.

Sieť sa skladá z jedného uzlu, ktorý plní rolu takzvaného datasinku (gateway alebo brána) a ostatných uzlov. Uzly sieti posielajú dátá do datasinku alebo data z neho prijímajú.

V určitých intervaloch datasink vyšle tzv. beacon signál. Tento signál slúži na časovú synchronizáciu medzi uzlami a značí začiatok novej epochy. Každá epocha obsahuje N slotov, v ktorých môžu uzly vysielať data. Beacon správa obsahuje hop count, ktorá udáva vzdialenosť ku datasinku. Ked nejaký uzol príjme beacon signál, vyšle svoj vlastný beacon signál v ďalšom volnom slote, ktorý vyberá na základe vzdialenosť od datasinku (hop count).

Ked uzol potrebuje poslať nejaké dátá, tak vyberie najskorší volný slot a v nom vysiela svoje dátá. Ak tieto dátá príjme uzol, ktorý nie je datasink a hop count daného uzlu ku sinku je menší

ako hop count vysielacieho uzlu, tak dáta v ďalšom volnom slote znova prepošeľ. Toto sa opakuje, až kým data nedosiahnu datasink.

Takto tvorená sieť avšak vyžaduje existenciu nejakého hlavného uzlu (datasinku), ktorý je potrebný na riadenie siete prostredníctvom časovej synchronizácie.

## 4.4 Pymesh

Pymesh je súčasťou Pycom [12] ekosystému. Tento ekosystém je určený na vývoj IoT systémov. Ponúka rôzne zariadenia, ktoré sú určené na použitie s týmto ekosystémom. Zariadenia obsahujú WiFi, bluetooth a LoRa, prípadne SigFox moduly. Zariadenia je možné rozšíriť o rôzne moduly so senzormi, ktoré rozširujú ich schopnosti.

PyMesh sieť sa skladá z uzlov, ktoré môžu vystupovať v roli gateway alebo bežného uzla. Uzly typu gateway sú označované ako Border Routers a prepájajú LoRa sieť s internetom. Uzly v sieti môžu komunikovať ad-hoc. V sieti môže dojsť k situácií kedy bude chcieť viacero uzlov vysielat v rovnakom čase a došlo by tak ku kolízii. Aby sa zabránilo takýmto situáciám, je použitá metóda Listen Before Talk, pri ktorej sa pred vysielaním signálu overí, či nie je v sieti už niekto iný, kto by práve vysielal. Pokiaľ áno, tak sa posielaná správa vyšle neskôr.

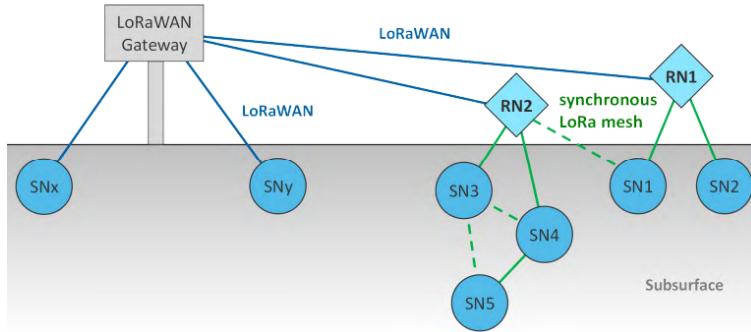
PyMesh je primárne určený na použitie s Pycom zariadeniami a použitie na inom zariadení by vyžadovalo väčšie úpravy zdrojového kódu.

## 4.5 Synchronous LoRa Mesh

Tento projekt[13] vznikol z potreby získavania real-time dát z podzemných kanalizácií. Tieto dáta sú potrebné k monitorovaniu a predikcii kritických situácií akými sú napríklad záplavy.

LoRaWan však nemá moc veľký dosah do podzemia. Z toho dôvodu by bolo potrebné v podzemných priestoroch implementovať LoRaWan gateway-e, ktoré sú ale energeticky náročne, drahé a vyžadujú pevné pripojenie do elektrickej a prípadne aj internetovej siete, ktoré v kanalizačiach niesu dostupné. Okrem toho, by museli byť všetky ostatné uzly v podzemnej sieti v dosahu danej gateway a pri väčšej podzemnej sieti by teda bolo potrebné implementovať viacero LoRaWan gateway.

Tento projekt sa snaží vyriešiť tieto problémy. Prináša protokol, ktorý rozširuje LoRaWan o tzv. repeater uzly (RN) viď Obr. 4.1. Tieto uzly sa vyskytujú na povrchu a preposielajú dátu z podzemných monitorovacích uzlov (sensor node - SN) do LoRaWan siete. Monitorovacie uzly podzemou tvoria mesh siet a RN plní funkciu riadiaceho uzlu pre podzemnú mesh siet. Komunikácia medzi RN a SN je synchronizovaná pomocou presného časovania, čo umožňuje koordináciu zmeny stavov SN z režimu spánku do normálneho režimu v čase, kedy potrebuje SN prijímať a odosielat dátu. Komunikácia sa cez uzly šíri multi-hop prístupom, za použitia smerovacej tabuľky.



Obr. 4.1: Schéma Synchronous LoRa Mesh. Prevzaté z [13]

Protokol používa na riadenie komunikácie metódu TDMA. RN priradí každej SN časový slot, v ktorom SN môže vyslať alebo prijať dátu. Monitorovacie uzly sa väčšinu času v režime spánku, zobudia sa len v ich určenom časovom slote a vďaka tomu majú tieto uzly nízku spotrebu energie.

Novo pripojený uzol do siete musí čakať na periodický beacon, vysielaný RN uzlom. Všetky uzly si držia v sebe smerovaciú tabuľku. Po pripojení nového uzla sa sieťou sa propaguje správa na aktualizovanie smerovacej tabuľky.

## 4.6 Porovnanie voči nášmu riešeniu

Niekteré zo spomenutých riešení využívajú smerovacie tabuľky na efektívnejší prenos dát v rámci siete. Smerovacie tabuľky môžu byť však limitáciou pokiaľ chceme dosiahnuť mobility uzlov. V prípade, že by sme chceli podporovať mobilitu uzlov, je potrebné zabezpečiť dostatočné časté aktualizácie smerovacích tabuľiek. To pridáva do siete zaťaženie v podobe dodatočných dátových prenosov.

Niekteré spomenuté riešenia limitujú komunikáciu medzi uzlami na vyhradené časové okná, mimo ktoré sú uzly v úspornom režime. To môže byť veľkou limitáciou pri niektorých aplikáciách, kde je potrebná komunikácia v reálnom čase (napr. chat).

Protokol navrhnutý v tejto práci nebude využívať žiadne smerovacie tabuľky. Vďaka tomu dosiahneme mobility uzlov bez potreby periodických aktualizácií smerovacích tabuľiek. Hlavným využitím nášho protokolu je komunikácia v reálnom čase, preto protokol nebude používať žiadnu časovú synchronizáciu ani časové okná vyhradené na komunikáciu, uzly siete tak budú môcť prijímať a odosieláť dátu hocikedy. To so sebou ale prináša nevýhodu v podobe vyššej spotreby energie.

Funkcionalita navrhnutého protokolu bude, podobne ako niektoré zo spomínaných riešení, fungovať na základe multi hop flooding, kedy sa správa v sieti preposiela cez uzly siete až kým nedorazí do destinácie. Správa sa odošle a uzol, ktorý túto správu prijal ako posledný ju prepošeďalej. Toto sa opakuje na ostatných uzloch v sieti až kým sa správa nedostane do destinácie alebo kym správe nedôjde limit preskokov, prípade TTL.

Navrhnutý protokol nie je závislý na žiadnych špeciálnych centrálnych uzloch typu gateway, prípadne nejakých riadiacich uzloch. Každý uzol v sieti môže byť jednoduchým uzlom, ktorý bude prijímať a preposielat dátá ďalej. Vďaka tomu môžu byť uzly realizované prostredníctvom lacných a malých zariadení.

Protokol bude možné používať na rôznych platformách. V tejto práci vznikne implementácia protokolu pre mikrokontroléry používajúce programovací jazyk C++ a mikrokontroléry alebo jednodoskové počítače podporujúce CircuitPython.

TODO nasa implementacia je sifrovana AES

TODO tu by sa hodila tabuľka s porovnaním (napr multiplatformovy support, spotreba energie, mobilita uzlov, obstaravacie naklady, zavislost na centralných uzloch)

## Kapitola 5

# Vlastná implementácia

Ako sme už spomínali v predchádzajúcej kapitole, nami vytvorený protokol bude fungovať na základe multi hop flooding. Tento prístup so sebou ale nesie niekoľko nevýhod, ktoré by sme mali zohľadniť pri navrhovaní implementácie.

Jednou z nevýhod je, že ak pošleme niekam správu, a destinácia práve nie je v dosahu, tak by sme správu prišli. Tento problém vyriešime tak, že odosielateľ si bude odoslané správy ukladať a v prípade, že sa nepodarí správu doručiť, môže ju opäťovne odoslať niekedy neskôr. To, že sa správu nepodarilo doručiť do destinácie, zistíme vďaka potvrzovacím ACK správam. Tieto správy sa odosielajú v opačnom smere ako správa, ktorú potvrdzujeme.

Ďalším z problémov môže byť potenciálne zahľtenie siete správami. Keby každý uzol v sieti preposielal ďalej každú správu, ktorú príjme, tak by sme rýchlo dostali do stavu kedy by množstvo uzlov preposielalo rovnaké správy a sieť by sa tym pádom zahľtila. Riešenie tohto problému bude spočívať v niekoľkých procesoch ako stavu zahľtenia predchádzať. Tieto procesy si bližšie popíšeme v nasledujúcej kapitole.

Implementácia sa bude deliť na dve verzie. Jedna verzia, implementovaná v CircuitPython, bude určená pre výkonnejšie zariadenia. Táto implementácia bude obsahovať aj webové rozhranie na obsluhu, posielanie a prijímanie sprav. Webové rozhranie avšak pôjde využiť len na zariadeniach, ktoré podporujú WiFi prípadne ethernet.

Implementácia v C++ bude určená pre menej výkonne zariadenia. Táto implementácia bude len zjednodušená verzia kompletnej implementácie a bude určená pre mikrokontroléry TTGO. Z dôvodu výkonnosti a obmedzenej pamäti bude táto implementácia obsahovať iba základné funkcionality protokolu, potrebné na odosielanie správ s údajmi so senzorov. Mikrokontroléry TTGO tak budú v sieti fungovať iba ako jednoduché uzly, ktoré budú v určitých časových intervaloch odosielať dátu.

## 5.1 Typy a stavы správ

Predom sme si stanovili, že v našom protokole bude možné odosielat rôzne typy správ. Na základe typov správ, sa potom správy spracovávajú odlišne.

Okrem typov správ, je nutné si definovať aj určité stavы, v ktorých sa správa môže nachádzať. Tieto stavы potom budú určovať, ako sa správa spracováva.

### 5.1.1 Typy správ

Rozhodli sme sa v našom protokole implementovať dokopy sedem typov správ. Každý typ správy má svoju úlohu a využitie. Sú nasledovné:

ACK správa bude slúžiť na potvrdenie doručenia správy. Ked' odosielateľ pôvodnej správy niekedy neskôr príjme ACK správu, ktorá potvrdzuje doručenie správy, tak môžeme pôvodnú správu považovať za doručenú.

Textová správa je správa, ktorá bude obsahovať textový reťazec. Rozšírením tejto správy je správa s potvrdením o doručení. Obsah textových správ bude šifrovaný.

Správa typu senzorové dát bude určená na prenos dát z rôznych senzorov. Obsah správy bude taktiež vo forme textového retazca, ktorý bude obsahovať dátu z senzorov. Tento typ sprav avšak oproti obyčajnej textovej správe ponuka možnosť nastaviť časový interval – TTL, po ktorého uplynutí sa správa nebude ďalej šíriť sietou. Obsah správy bude taktiež šifrovaný.

Sprava žiadosť o traceroute sa bude používať len v krajných prípadoch, kedy by nás zaujímalo akou cestou sa správa preposiela sietou. Ked nejaký uzol príjme správu typu žiadosť o traceroute, tak vytvorí novu správu typu odpoveď na traceroute a vloží do nej svoju adresu. Adresátom tejto novej správy bude odosielateľ pôvodnej správy. Ked sa bude tato odpoveď na traceroute šíriť sietou, tak každý uzol, ktorý danú správu preposal dalej, pridá do nej svoju vlastnú adresu. Takto sa vytvorí reťazec, ktorý bude obsahovať adresy všetkých uzlov, cez ktoré bola správa preposlaná. Aby mohli ostatné uzly v sieti do správy postupne pridávať svoje adresy, tak obsah tejto správy nebude šifrovaný.

Posledným typom správy je nespracovaný paket. Tento typ správy bude využívaný iba v prípade, že ma používateľ zapnutý monitorovací režim. V tomto režime sa všetky správy, ktoré sa v sieti prenášajú, budú zaznamenávať. V tomto režime môžeme zachytiť aj pakety, ktoré niesu súčasťou nášho protokolu a ich štruktúra nebude zodpovedať našim špecifikáciám. Z toho dôvodu ich nie je možné spracovať. Uložia sa teda ako typ správy nespracovaný paket, a budu obsahovať iba holé prijaté dátá vo forme bajtov.

### 5.1.2 Stavy správ

Na správnu funkčnosť je potrebné aby správy existovali v určitých stavoch. Správy budu časom svoj stav aktualizovať a prechádzať medzi nimi. Tieto stavы budú definované nasledovne:

Stav nová správa (NEW). Tento stav bude mať správa, ktorá bola práve vytvorená. Nebola ešte ani raz odoslaná aktuálnym uzlom, a čaká na svoje prvé odoslanie. Po prvom odoslaní sa správa presunie do stavu odoslaná (SENT).

Zo stavu odoslaná sa môže správa presunúť do stavu neúspešná (FAILED), hotová (DONE), preposlaná (REBROADCASTED) alebo zmazaná (DELETED). Ak aktuálny uzol príjme tu istú správu, odoslanú iným uzlom, znamená to, že dana správa sa šíri ďalej v sieti. Aktuálny uzol teda správu presunie do stavu preposlaná alebo do stavu zmazaná. To či sa správa presunie do stavu zmazaná alebo preposlaná je závisle od toho, či je aktuálny uzol zároveň aj autorom danej správy. V prípade, že je autorom, správa sa presunie do stavu preposlaná.

Správy v stave preposlaná, pri ktorých nepotrebuje sledovať, či správa dorazila do destinácie, sa následne presunú do stavu hotová.

Pokiaľ aktuálny uzol vyčerpá všetky pokusy o odoslanie správy, prípadne pri správe typu senzorové dátia dôjde časový limit, správa sa presunie do stavu neúspešná alebo zmazaná. To či sa presunie do stavu neúspešná alebo zmazaná je znova závisle na tom, či je aktuálny uzol autorom danej správy alebo nie.

Správam, ktoré sú v stave preposlané sa odpočítava časový limit na prijatie potvrdenia. Pokiaľ v tomto časovom intervale nedorazí potvrdzovacia ACK správa, tak sa správa presunie do stavu nepotvrdená (NAK). V opačnom prípade sa presunie do stavu potvrdená (ACK).

Stav zmazaná značí, že správu považujeme za vymazanú. Ďalej s ňou už nebudeme pracovať a po uplynutí určitého intervalu sa reálne vymaže z pamäte.

## 5.2 Návrh paketu

Paket bude obsahovať hlavičku a dátový rámec. Štruktúra dátového rámcu bude závisieť od typu správy. Navrhnutú štruktúru hlavičky paketu môžeme vidieť v tabuľke 5.1

Tabuľka 5.1: Štruktúra hlavičky paketu a dĺžka jednotlivých polí v bajtoch.

2B	2B	4B	2B	1B	1B	0 – 240B
Destinácia	Odosielateľ	ID správy	Kontrolný súčet	Typ správy	Priorita	Dátový rámec

Pre adresy odosielateľa a destinácie budeme používať dvoj bajtové adresy. Štvor bajtový identifikátor správy bude náhodne generovaný pri každom vytvorení novej správy.

Z adresy destinácie, adresy odosielateľa a identifikátoru správy sa vytvorí dvoj bajtový kontrolný súčet. Tento kontrolný súčet bude používaný na overenie, či prijatá správa patrí do nášho protokolu. Overovať integritu dát na základe kontrolného súčtu nie je za potreby, pretože táto kontrola je už zahrnutá na fyzickej vrstve LoRa technológie.

V položke typ správy bude číslo, ktoré definuje typ danej správy. Číslo bude reprezentovať jeden z typov spomínaných v predošej sekcií.

Správam môžme v hlavičke paketu určiť vyššiu prioritu. Na túto prioritu sa následne bude dbať pri spracovávaní správ a správy s vyššou prioritou budu odbavené prioritne.

Dátové rámce sa budú lísiť na základe typu správy. V tabuľkách 5.2, 5.3, 5.4, 5.5, 5.6 je uvedená štruktúra dátových rámcov pre jednotlivé typy správ.

Tabuľka 5.2: Štruktúra dátového rámcu pre textové správy.

1B	1B	0 – 238B
Počet skokov	Pôvodný počet skokov	Textové dátá

Tabuľka 5.3: Štruktúra dátového rámcu pre správy typu žiadost o traceroute.

1B	1B
Počet skokov	Pôvodný počet skokov

Tabuľka 5.4: Štruktúra dátového rámcu pre správy typu odpoveď na traceroute.

1B	0 – 239B
Počet skokov	Navštívené adresy

V dátových niektorých rámcach môžme vidieť položku počet skokov. Do tejto položky bude pri vytvorení novej správy zapísaný maximálny počet skokov, ktoré môže správa vykonať pri prenose sietou. Každý uzol, predtým ako správu prepošeď ďalej zmenší počet skokov o jedna. Ak sa počet skokov dostane na nulu tak správu ďalej nespracovávame.

V rámcach taktiež nájdeme položku pôvodný počet skokov. Do tejto položky bude taktiež pri vytvorení novej správy zapísaný maximálny počet skokov. Hodnota v tejto položke sa ale už ďalej nemení. Bude slúžiť k tomu, aby prijímateľ dokázal zistíť, kolko preskokov správe zabralo, aby správa dorazila až k nemu. Túto informáciu bude možné následne použiť na správne nastavenie maximálneho počtu skokov pre odoslanie potvrzovacej správy, prípadne správy typu odpoveď na traceroute.

Tabuľka 5.5: Štruktúra dátového rámcu pre potvrzovacie ACK správy.

1B	4B
Počet skokov	ID potvrzovanej správy

Tabuľka 5.6: Štruktúra dátového rámcu pre senzorové dáta.

2B	0 – 238B
TTL	Textové dáta

# Literatura

1. K. HESTER, et al. *Meshtastic—An Opensource Hiking, Pilot, Skiing, Secure GPS Mesh Communicator, 2020*. Dostupné tiež z: <https://meshtastic.org/>.
2. JOAN MIQUEL SOLÉ, et al. *LoRaMesher - implementing a distance-vector routing protocol for communicating messages among LoRa nodes*. Dostupné tiež z: <https://github.com/LoRaMesher/LoRaMesher>.
3. *Sigfox - První celorepublikový mobilní operátor pro internet věcí* [online] [cit. 2022-07-10]. Dostupné z : <https://sigfox.cz/>.
4. *LoRa alliance* [online] [cit. 2022-07-10]. Dostupné z : <https://lora-alliance.org/>.
5. PIETROSEMOLI, Ermanno. *LoRa details*. Dostupné tiež z: [http://wireless.ictp.it/school\\_2017/Slides/LoRaDetails.pdf](http://wireless.ictp.it/school_2017/Slides/LoRaDetails.pdf).
6. Český Telekomunikační Úřad. Dostupné tiež z: <https://www.ctu.cz/>.
7. *všeobecné oprávnění č. VO-R/10/07.2021-8 k využívání rádiových kmitočtů a k provozování zařízení krátkého dosahu*. Dostupné tiež z: <https://www.ctu.cz/sites/default/files/obsah/vo-r10-072021-8.pdf>.
8. *Semtech* [online] [cit. 2022-07-10]. Dostupné z : <https://www.semtech.com/>.
9. *HopeRF* [online] [cit. 2022-07-10]. Dostupné z : <https://www.hoperf.com/>.
10. *LILYGO*. Dostupné tiež z: <http://www.lilygo.cn/>.
11. MISENKO, Peter. *Armachat - Doomsday LORA QWERTY communicator*. Dostupné tiež z: <https://github.com/bobricius/armachat>.
12. *Pycom, Pymesh—LoRa Full-Mesh Network Technology*. Dostupné tiež z: <https://docs.pycom.io/pymesh>.
13. EBI, Christian; SCHALTEGGER, Fabian; RÜST, Andreas; BLUMENSAAT, Frank. Synchronous LoRa Mesh Network to Monitor Processes in Underground Infrastructure. *IEEE Access*. 2019, roč. 7, s. 57663–57677. Dostupné z DOI: [10.1109/ACCESS.2019.2913985](https://doi.org/10.1109/ACCESS.2019.2913985).