

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## BEZDRÁTOVÝ PROTOKOL MIWI

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAKUB LEŠKA

BRNO 2015



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## **BEZDRÁTOVÝ PROTOKOL MIWI**

WIRELESS PROTOCOL MIWI

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**JAKUB LEŠKA**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. VLASTIMIL KOŠAŘ**

BRNO 2015

## **Abstrakt**

Přehled používaných směrovacích protokolů v bezdrátových sensorových sítích, popis bezdrátového protokolu mimac a jeho nádstavba miwi, návrh nového protokolu pro potřeby bezdrátové sensorové sítě v inteligentní domácnosti.

## **Abstract**

Overwiev of actual solutions in routing protocols in wireless sensors networks, description of wireless protocol mimac and its miwi, design of new protocol for sensor network used in intelligent housing

## **Klíčová slova**

MiWi, směrování, bezdrátový protokol, bezdrátové sensorové sítě

## **Keywords**

MiWi, routing, wireless protocol, wireless sensor network

## **Citace**

Jakub Leška: Bezdrátový protokol MiWi, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Bezdrátový protokol MiWi

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Vlastimila Košaře

.....  
Jakub Leška  
20. května 2015

## Poděkování

Děkuji za pomoc při tvorbě práce konzultantovi Josefovi Hájkovi

© Jakub Leška, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>4</b>
1.1 Bezdrátová komunikácia	4
<b>2 Bezdrátové senzorové siete</b>	<b>5</b>
2.1 Špeciálne charakteristiky WSN	7
2.2 Návrh smerovacích protokolov pre bezdrátové senzorové siete	7
2.3 Smerovacie protokoly	9
2.3.1 Spôsob klasifikácie	9
2.3.2 Flooding	10
2.3.3 Gossiping	11
2.3.4 SPIN – Sensor Protocols for Information via Negotiation	11
2.3.5 Directed Diffusion	12
2.3.6 Rumor routing	12
2.3.7 MCFA – Minimum Cost Forwarding Algorithm	13
2.3.8 EAR – Energy Aware Routing	13
2.3.9 LEACH – Low Energy Adaptive Clustering Hierarchy	13
2.3.10 HEED – Hybrid energy-efficient distributed clustering	14
2.3.11 TEEN & APTEEN – Threshold-sensitive Energy Efficient Protocols	14
2.3.12 SPAN – Coordination of Power Saving with Routing	14
2.3.13 Zoznam ďalších protokolov	15
<b>3 Riešenie od firmy Microchip</b>	<b>16</b>
3.1 MiMac	16
3.1.1 Format packetu	16
3.1.2 Nedostatky v Mimac	19
3.2 MiWi™	21
3.2.1 Bezdrátové protokoly v rámci MiWi™	22
3.2.2 Format paketu	23
3.2.3 Smerovanie	24
3.2.4 Prihlasovanie	26
3.2.5 Nedostatky v MiWi a Miwi PRO	26
3.2.6 MRF89	27
<b>4 Návrh a implementácia vlastného protokolu</b>	<b>29</b>
4.1 Požiadavky na návrh	29
4.2 Štruktúra	30
4.3 Smerovanie	30
4.4 Implementácia	31

4.4.1	hw	31
4.4.2	Fyzická vrstva	32
4.4.3	Linková vrstva	32
4.4.4	Sieťová vrstva	34
4.5	Zariadenia	35
4.6	Možné vylepšenia	35
<b>5</b>	<b>Testovanie vlastnej implementácie smerovacieho protokolu</b>	<b>37</b>
5.1	Prenosu paketov	37
5.2	Pripojovanie nových zariadení do siete	38
5.3	Návrh testov	39
<b>6</b>	<b>Záver</b>	<b>40</b>
<b>A</b>	<b>Obsah CD</b>	<b>42</b>

# Seznam obrázků

1.1	Diagram RF prenosu . . . . .	4
2.1	Znázornenie wsn . . . . .	5
2.2	Model senzoru v bezdrátových senzorových sieťach . . . . .	6
2.3	Hierarchické zobrazenie klasifikácie protokolov aj s ich zaradením . . . . .	10
2.4	Grafické znázornenie správ v Directed Diffusion protocale . . . . .	12
3.1	Mimac formát mimac rámca s fyzickou vrstvou . . . . .	16
3.2	MiMac hlavička media access control . . . . .	17
3.3	Štruktúra bytu frame control z hlavičky mac vrstvy . . . . .	18
3.4	MiWi <sup>TM</sup> štruktúra vývojového prostredia . . . . .	21
3.5	MiWi <sup>TM</sup> P2P ukážka topológie topológia v . . . . .	22
3.6	MiWi <sup>TM</sup> ukážka topológie topológia v . . . . .	22
3.7	MiWi <sup>TM</sup> PRO ukážka topológie topológia v . . . . .	23
3.8	MiWi <sup>TM</sup> PRO štruktúra Frame Control Bytu . . . . .	23
3.9	MiWi <sup>TM</sup> PRO hlavička . . . . .	24
3.10	Príklad topológie MiWi PRO siete . . . . .	25
3.11	Príklad tabuľky miwi tree routingu . . . . .	25
3.12	Vnútorne usporiadanie rádiového modulu z pohľadu komunikácie cez SPI . . . . .	27
4.1	Grafické znázornenie vrstiev protokolu . . . . .	31
5.1	Výsledok testu 1 . . . . .	38
5.2	Plošný plán miesta testovania . . . . .	38

# Kapitola 1

## Úvod

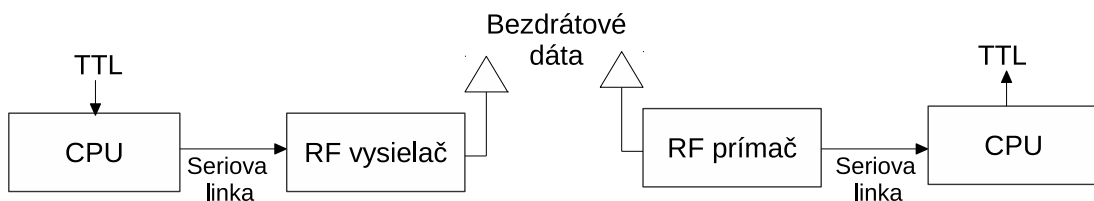
V poslednej dobe umožnil rozvoj v oblasti komunikačných technológií a microelektroniky vznik malých prenosných batériou napájaných zariadení, ktorých úlohou je zbierať informácie o fyzikálnych podmienkach v ich okolí. Tieto senzory sú často rozmiestnené v prostredí, kde nie je možné realizovať drátové pripojenie senzorov so zariadením určeným na zhromažďovanie týchto dát. Z tohto dôvodu vznikajú systémy využívajúce bezdrátovú komunikáciu.

Táto práca je rozdelená do 3 hlavných tématických celkov: prvý sa zaoberá popisom existujúcich smerovacích protokolov a ich kategorizáciou, druhá časť popisuje riešenie bezdrátovej siete navrhutej firmou microchip, v ktorého sekcii MiWi 3.2 bude popísaný hlavne protocol MiWi PRO ako aktuálne už existujúce riešenie a tretia sa zaoberá popisom návrhu a implementácie nového bezdrátového protokolu, ktorý vznikol ako podporná časť pre projekt inteligentnej domácnosti vyvíjanej na Fakulte informačných technológií Vysokého učenia technického v Brne.

### 1.1 Bezdrátová komunikácia

je prenos dát medzi dvomi alebo viacerými zariadeniami, ktoré nie sú spojené pomocou drátových spojov. Medzi najčastejšie technológie patrí prenos pomocou radiových vln. Použitie bezdrátových technológií prináša zvýšené nároky na zložitosť implementácie zariadení a na ich cenu. Aj napriek tomu sa v dnešnej dobe objavujú implementácie využívajúce túto technológiu a sú často nasadzované tam, kde je drátové spojenie nemožné alebo nežiadúce.

Hlavnou výhodou takéhoto riešenia je teda odstránenie potreby fyzického spojenia. Z tejto výhody ale plynie ďalšia požiadavka na výsledné zariadenia, ktoré musia obsahovať vlastný zdroj energie, čo sa vo väčšine prípadov rieši pridaním batérie.



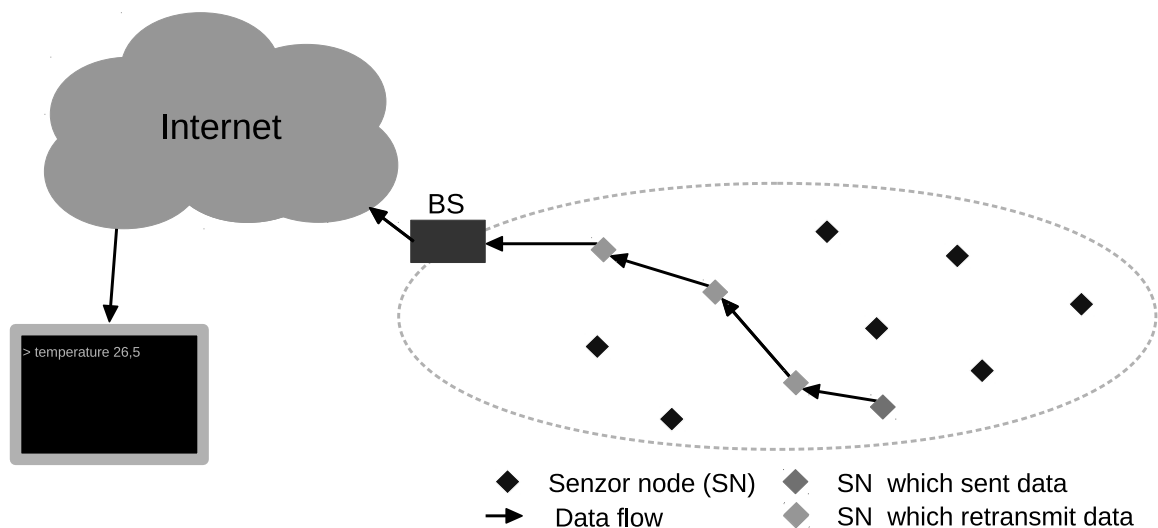
Obrázek 1.1: Diagram RF prenosu podľa[2]



## Kapitola 2

# Bezdrátové senzorové siete

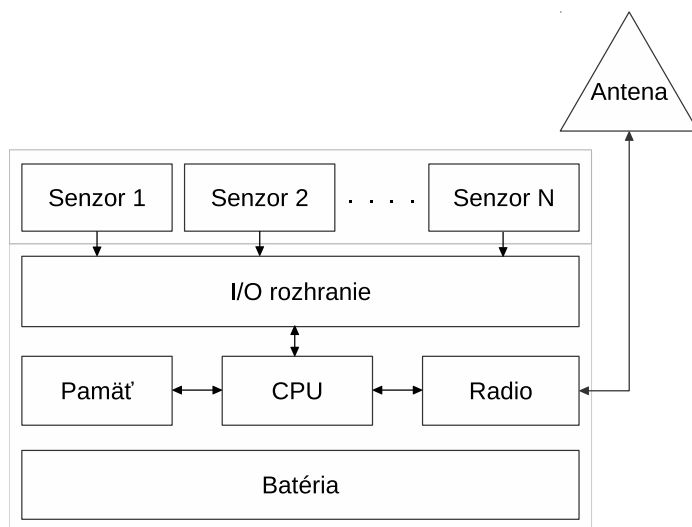
Bezdrátové senzorové siete ďalej len WSN(Wireless Sensors Network) alebo WSN(Wireless Sensor and Adapter Network) je priestorovo distribuovaný systém pozostávajúci z autonómnych senzorov, ktorých cieľom je monitorovať fyzikálne vlastnosti (napr. tlak, teplota, vlhkosť) prostredia, v ktorom sa nachádzajú a zároveň prenos týchto zozbieraných dát cez bezdrátovú sieť na jedno konkrétne miesto, kde je možné ich ďalej spracovávať alebo uchovávať. Toto miesto je zvyčajne označované ako centrálny uzol alebo base-station skrátené BS.



Obrázek 2.1: wsn

Vývoj WNS bol pôvodne započatý armádou pre aplikácie ako dohľad nad bojiskom. V dnešnej dobe sú WSN veľmi rozšírené v priemyselných a spotrebiteľských aplikáciách ako napríklad monitorovanie a kontrolovanie výrobných strojov.

WSN sú zariadenia (uzlov, nodes), kde je každé zariadenie pripojené na jedno alebo niekoľko ďalších zariadení. Každé toto zariadenie pozostáva z niekoľkých častí, ktoré je možné vidieť na schématickom znázornení na obrázku 2.



Obrázek 2.2: Senzor

Medzi hlavné časti zariadenia patrí microcontroller alebo iná výpočtová jednotka, rádiový modul s anténou, zdroj energie (prevažne používaná batéria) a senzoru/ov. Pri realizácii WSN je kladený dôraz na nízku obstarávaciu a prevádzkovú cenu, keďže takéto siete často obsahujú desiatky až stovky zariadení. Zariadenia sú preto osadzované procesormi s malým výpočtovým výkonom a rádiom s relatívne malým dosahom radovo v jednotkách až desiatkach metrov.

Pre fungovanie komplexných sietí s využitím takýchto zariadení je/bolo potrebné vyvinúť špecializované protokoly, ktoré sa budú snažiť riešiť špecifické ciele WSN.

## 2.1 Špeciálne charakteristiky WSN

- **vysoká hustota uzlov** výrazne vyššia hustota ako v mobilných ad-hoc sieťach
- **obvykle napájanie batériami** uzly sú často umiestnené na miestach, kde nie je možné batérie dobíjať alebo vymieňať
- **hardwarové obmedzenia** veľmi obmedzené napájanie, výpočtový výkon, pamäť
- **náhodne rozmiestňovanie** senzory sú často rozmiestňované náhodne a na miesta bez prístupu, preto je potrebné aby boli schopné sa sami konfigurovať a pripájať do siete
- **nespolahlivosť uzlov** uzly sú často umiestnené vo fyzicky náročnom prostredí, kde je vysoké riziko ich poškodenia alebo úplného zničenia
- **redundantnosť dát** uzly sú umiestnené v sledovanej oblasti veľmi husto, a preto môžu niektoré ich dáta korelovať prípadne byť redundantné
- **obvykle viazané na jedno použitie**
- **zmeny topológie** veľmi časté zmeny topológie vplyvom poškodzovania, vybíjania, presúvania, pridávania, odstraňovania uzlov alebo nestability média

## 2.2 Návrh smerovacích protokolov pre bezdrátové senzorové siete

Ako už bolo zmienené v 2.1 v bezdrátových senzorových sieťach je množstvo obmedzení, a preto návrh smerovacích protokolov musí riešiť viacere výzvy. Medzi najčastejšie patrí:

**Limitovaný zdroj energie** uzly senzorových sietí sú často napájané batériami, čo znamená, že zdroj energie uzla je limitovaný. Okrem toho batérie nie je často možné ani dobíjať prípadne vymieňať vzhľadom na možnosť umiestnenia senzoru v priestore, ktorý nie je dostupný. Príkladom takéhoto prostredia môže byť aktívna sopka alebo bojové pole. Vyčerpanie energetických zásob jedného uzla môže navyše spôsobiť v prípade používania tohto senzoru ako repeatra vo viacerých (multi-hop) sieťach výpadok celej podsiete, z ktorej boli dáta preposielané. Z uvedeného dôvodu sa snažia byť smerovacie protokoly veľmi šetrné na využívanie energie, ale zároveň aj na jej rovnomerné využitie na jednotlivých uzloch, aby sa dosiahlo predĺženie doby životnosti siete.

**Obmedzené výpočtové a pamäťové zdroje** používanie lacných a energeticky úsporných súčastí (procesor, pamäť) pri stavbe senzoru má vplyv na jeho výsledný výpočtový výkon kvôli čomu môže mať výsledný senzor len obmedzenú funkčnosť. Z uvedeného dôvodu je nutné pri návrhu protokolu dbať na to, aby vôbec bola realizácia na danom hardware možná.

**Rozmiestňovanie uzlov** senzory môžu byť do siete usporiadované dvoma spôsobmi, a to buď podľa vopred stanovenej topológie alebo náhodne. V prvom prípade je možné stanoviť cesty, po ktorých budú medzi sebou zariadenia komunikovať, vopred. Vďaka tomu môžu byť navrhnuté cesty veľmi efektívne, zároveň môžu byť použité protokoly

výrazne jednoduchšie, keďže nemusia plánovať usporiadanie cesty. Prevažná časť senzorových sietí v súčasnej dobe však vopred určenú topológiu nemá, preto je nutné aby mohli uzly prípadne sieť meniť svoju topológiu autonómne tak, aby bola dosiahnutá čo najväčšia pripojiteľnosť a zároveň vysoká miera úspory energie.

**Prostredie nasadenia, dynamickosť, prenosové médiá** prostredie, v ktorom sú zariadenia umiestnené, je často fyzicky veľmi náročné, čo môže spôsobovať dočasné prerušenie funkčnosti alebo až úplné zničenie zariadenia. Prostredie, v ktorom sa uzly nachádzajú, je navyše často dynamické, čo môže viesť k častým zmenám topológie v sieti. K častým zmenám topológie prispieva aj možnosť častého pridávania a odobrania zariadení, prípadne ich presúvanie v rámci siete. Ďalším faktorom je prenosové médium, kde môže vplyvom pôsobenia okolitého prostredia dochádzať k prerušovaniu spojenia, napríklad v dôsledku vysokej hladiny elektromagnetického šumu alebo pridaním fyzickej prekážky do monitorovaného priestoru. Z týchto dôvodov by bolo vhodné, aby sieť určovala smer ciest dynamicky.

**Homogenita zariadení v sieti** jednotlivé uzly môžu mať rôzne vlastnosti. Pri navrhovaní topológie by mal protokol posudzovať tieto odlišnosti a mal by sa snažiť vytvoriť sieť s čo najlepšími vlastnosťami. Dobrým príkladom je sieť, kde sa okrem batériami napájaných zariadení nachádzajú aj zariadenia napájané z rozvodnej siete. V takejto sieti majú niektoré zariadenia prakticky neobmedzený zdroj energie a z toho dôvodu by mal protokol primárne používať tieto zariadenia na energeticky náročnejšie úlohy ako je preposielanie dát naprieč sieťou.

**Skalovateľnosť** protokol by mal byť schopný adaptovať sa na veľkosť siete, keďže veľkosť siete nie je vopred známa a počet zariadení sa môže pohybovať medzi jednotkami až tisíckami kusov.

**QoS** do úvahy treba brať aj požiadavky aplikácie na kvalitu spojenia. V niektorých aplikáciách sa môžeme stretnúť s požiadavkami na doručenie správy do určitého času od jej vzniku, vzhľadom na to, že pri doručení dát po tomto čase už sú dáta bezcenné. Prípadne môže aplikácia vyžadovať garantovanú minimálnu rýchlosť prenosu. V takýchto sieťach je dobre definovať chovanie pri znížení energetických zásob pod určitú kritickú hodnotu, vzhľadom na to, že môže byť dôležitejšia doba funkčnosti siete pred kvalitou služby.

## 2.3 Smerovacie protokoly

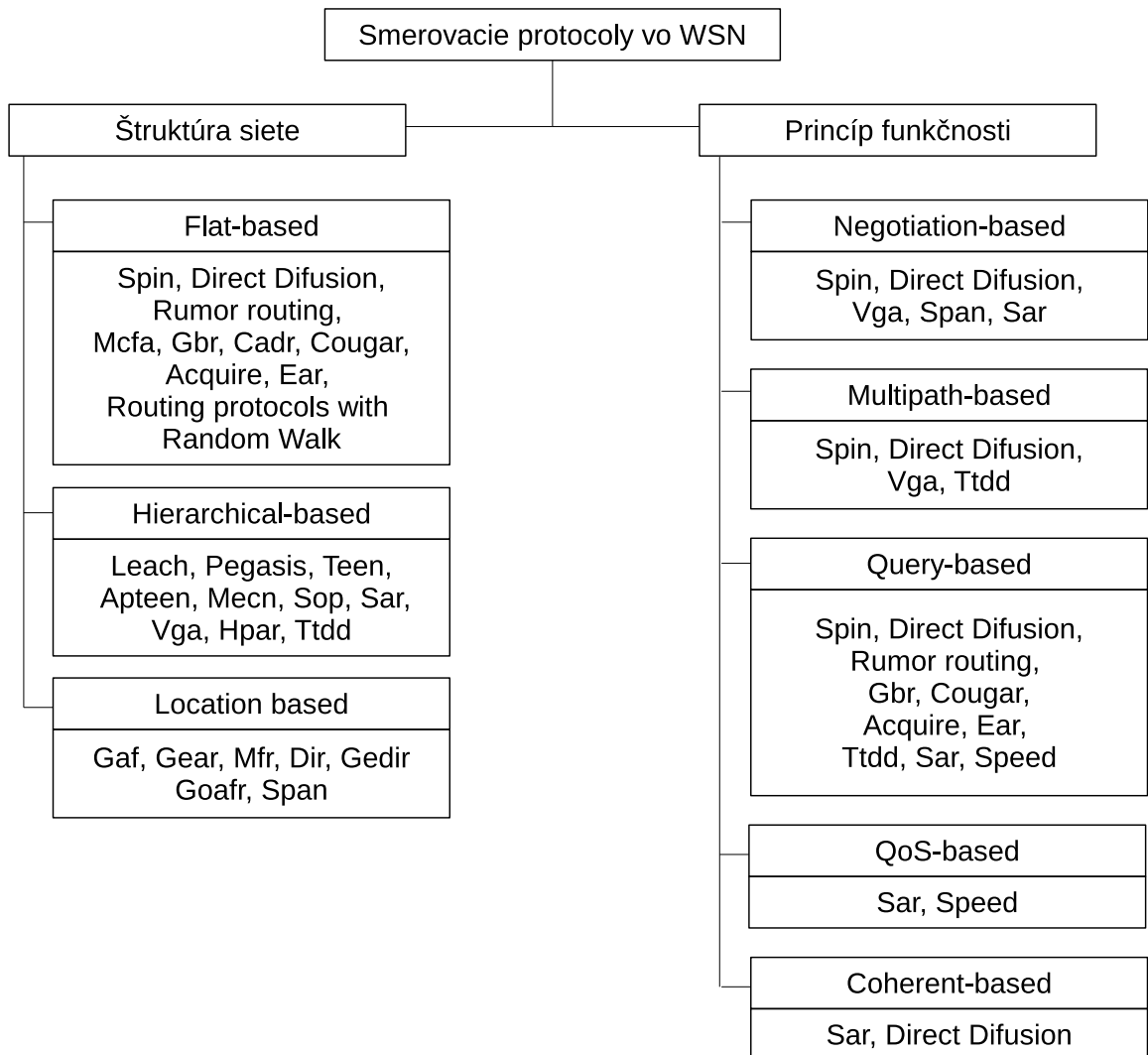
Klasifikovať smerovacie protokoly je možné z rôznych pohľadov. Táto kapitola popisuje základné spôsoby klasifikácie bezdrátových sensorových sietí podľa [1]. A ďalej obsahuje zoznam a popis vybraných protokolov z [1] [6].

### 2.3.1 Spôsob klasifikácie

Smerovacie protokoly môžeme posudzovať z rôznych pohľadov:

- **štruktúra**  
sieť má veľký význam vzhľadom na funkčnosť protokolu
  - **flat-based**  
uzly siete sú zvyčajne homogénne a v rámci siete zastávajú rovnakú úlohu. Zvyčajne tieto siete nepriradujú každému zariadeniu globálny identifikátor. BS zvyčajne posiela dotazy do určitej časti siete a čaká na dáta zo sensorov.
  - **hierarchical-based**  
tiež nazývané cluster-based, bol pôvodne využívaný už v pevných (drátových) sieťach. Existujú väčšinou dvojvrstvové protokoly, kde jedna vrstva je používaná na voľbu clustrov a druhá pre smerovanie, pričom väčšina protokolov v tejto skupine sa venuje prevažne vrstve, ktorá realizuje voľbu a nastavovanie clustrov.
  - **location-based** v týchto protokoloch bývajú zariadenia adresované ich polohou. Informácie o polohe sú potrebné pre sensorovú sieť, ktorá sa na ich základe snaží spočítať energetickú spotrebu medzi uzlami a určiť energeticky optimálnu cestu.
- **adaptívnosť** protokol môže byť označený za adaptívny, ak niektorý parameter smerovania je kontrolovaný na základe aktuálnych podmienok.
- **princípu funkčnosti**
  - **multipath**
  - **query based**
  - **negotiation based**
  - **QoS based**
  - **coherent based**
- **spôsobu hľadania cesty k cieľu**
  - **proactive** v proaktívnych protokoloch dochádza k výpočtu ciest ešte pred samotným smerovaním, táto metóda je ideálna pre siete, kde senzory sú rozmiestnené staticky a môžu ušetriť veľké množstvo zdrojov oproti reaktívnemu smerovaniu, ktoré by tieto zdroje spotrebovalo na preskúmanie a nastavovanie ciest.
  - **reactive** k výpočtu ciest v reaktívnych protokoloch dochádza až za behu v dobe, keď sú bezprostredne potrebné, to prináša veľkú výhodu oproti proaktívnemu smerovaniu v dobe, keď rozmiestnenie zariadení v sieti nie je statické a umožňuje udržiavať smerovanie funkčné aj po premiestnení zariadení.

- **hybrid** u hybridných protokolov sa využíva kombinácia proaktívneho a reaktívneho určovania cesty s cieľom optimalizovať náročnosť na zdroje ako pri proaktívnom smerovaní a zároveň zachovať možnosť smerovania v dynamicky sa meniacej sieti.



Obrázek 2.3: Hierarchické zobrazenie klasifikácie protokolov aj s ich zaradením

### 2.3.2 Flooding

Jedna z najjednoduchších techník smerovania packetov vo WNS. Nevyžaduje udržiavanie topológie ani algoritmy zisťovania cesty. Medzi hlavné nedostatky patria duplicitné správy, ktoré vyplývajú z princípu funkčnosti algoritmu, kedy zariadenie dostáva tú istú správu od všetkých susedov a zároveň ju všetkým pomocou broadcastu posiela.

**Popis funkčnosti** správa sa po prijatí posiela ďalej broadcastom na všetky okolité zariadenia, ak nie je určená pre aktuálne zariadenie a nemá vyčerpaný maximálny počet preposielaní.

### 2.3.3 Gossiping

je modifikácia floodingu 2.3.2 pričom správa sa po prijatí na zariadenie neposielala ďalej broadcastom, ale pošle sa náhodnému susedovi. Táto modifikácia odstraňuje veľké množstvo duplicitných broadcastových správ v sieti. Jej najväčšou nevýhodou je predĺženie času doručenia, ktorý sa navyše stáva silne nedeterministickým a paket môže cestovať sieťou veľmi dlhý čas, čo v prípade použitia kontroly maximálneho počtu preposlaní môže spôsobiť, že sa tento packet stratí bez toho, aby dosiahol svoj cieľ.

### 2.3.4 SPIN – Sensor Protocols for Information via Negotiation

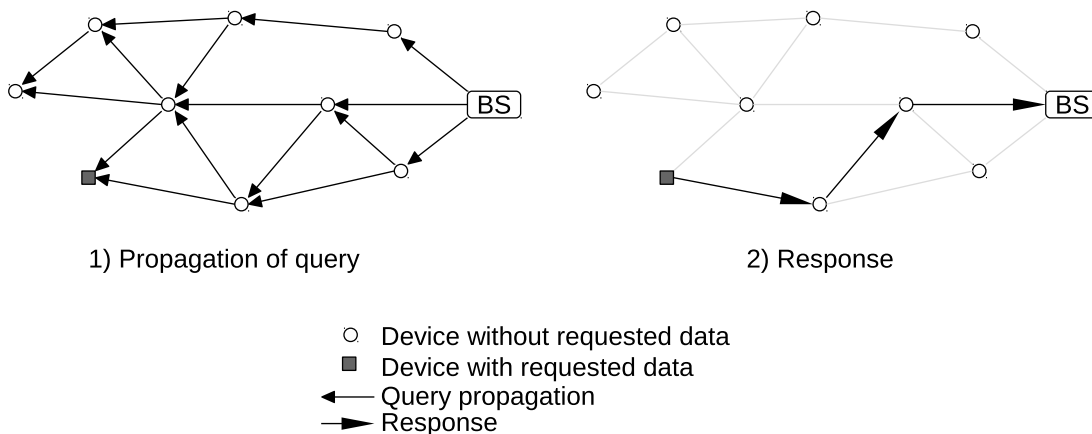
skupina protokolov je navrhnutá tak, aby sa zamerala na nedostatky klasického floodingu prostredníctvom negociácie(dohadovania sa). Všetky protokoly v tejto skupine berú do úvahy svoje zdroje a sú schopné sa adaptovať pre ich lepšie využívanie. Zariadenia s týmto protokolom robia pri komunikácii informované rozhodnutia na základe spotreby pre zvýšenie energetickej efektivity. Pri samotnej komunikácii sa často posielajú iba metadáta k skutočným dátam zo senzoru, ktoré by mali byť kratšie ako samotné dáta. Čo vedie k ďalšiemu znižovaniu energetických nárokov. Prenos dát v sieti prebieha pomocou 3 typov správ, a to ADV(advertise - ponuka), REQ(request - žiadosť) a DATA.

#### Popis funkčnosti:

1. ADV zaslanie metadát pomocou broadcastu všetkým okolitým zariadeniam
2. v prípade, že zariadenie prijme ADV a z metadát zistí, že má o dané dáta záujem, odpovie odosielateľovi REQ správou
3. po prijatí REQ dotazu na zariadení odosielaťcom dáta toto zariadenie odošle správu DATA

### 2.3.5 Directed Diffusion

podobne ako SPIN aj tento protokol sa snaží riešiť problémy jednoduchého floodingu. Na rozdiel od SPINu, kde sa posielajú dáta v smere od senzoru k BS, funguje tento protokol opačne, a to tak, že BS posiela dotaz do siete s cieľom získať údaje od špecifického senzoru. Jednou z výhod tohoto protokolu je, že trasa v smere od senzoru k BS vedie po optimálnej ceste.



Obrázek 2.4: Grafické znázornenie správ v Directed Diffusion protocoile

#### Popis funkčnosti:

1. centrálna jednotka(BS) zašle správu s popisom o aké dáta do siete má záujem jednotlivé uzly si postupne túto správu preposielajú a zároveň si ukladajú popis hľadaných dát a ukazovateľ na suseda, od ktorého žiadosť prišla
2. ak niektorý uzol má informácie zodpovedajúce popisu zašle tieto dáta späť do BS, pri prechode správy smerom späť sa využívajú ukazovatele na suseda, od ktorého žiadosť prišla a dáta sa posielajú len tomuto uzlu

### 2.3.6 Rumor routing

modifikácia Directed Diffusion 2.3.5 kde bol pridaný špeciálny typ paketu "agent", ktorý je vyslaný do siete po vzniku udalosti. Zo simulácií sa ukázalo ako energeticky úspornejšie riešenie, avšak je určené len pre siete s malým množstvom udalostí, inak nefunguje efektívne.



### 2.3.7 MCFA – Minimum Cost Forwarding Algorithm

počíta s pevnou pozíciou BS. Každý uzol uchováva svoju najmenšiu vzdialenosť k BS. Vzdialenosť k BS sa určuje tak, že každý uzol nastaví svoju hodnotu na nekonečno a BS pošle broadcastom správu s cenou 0. Zariadenia po prijatí správy zvýšia cenu o jedna a porovnajú túto hodnotu s aktuálnou cenou uzla. Ak je táto hodnota menšia, správa sa pošle ďalej s inkrementovanou cenou, ak nie, správa sa nepreposiela ďalej. Z dôvodu, že zariadenia vzdialené od BS, dostávajú veľký počet takýchto správ, pridalo sa do protokolu čakanie s dĺžkou závislou na cene spojenia aktuálneho uzla. Tým pádom je väčšina správ s nižšou cenou prijatá skôr ako uplynie toto čakanie a nevznikajú viacnásobné správy.

**Popis funkčnosti:** Správa sa posielajú broadcastom na všetky susedné uzly, ktoré po prijatí správy overujú, či je zariadenie s najkratšou cestou a ak áno, preposielajú správu ďalej do siete.

### 2.3.8 EAR – Energy Aware Routing

protokol navrhnutý hlavne na zvýšenie životnosti siete. Funguje podobne ako Directed Diffusion 2.3.5 prípadne MCFA 2.3.7 ale ceny uzlov sú určené podľa energetických rezerv zariadenia, vďaka čomu sa výrazne predlžuje doba života siete, keďže sa na preposielanie posielajú vždy zariadenia s najväčšou rezervou energie. Vedlajším efektom je, že v dobe, keď je energetická rezerva na niektorých zariadeniach takmer vyčerpaná a aj v celej sieti a v prípade, že by boli doplnené energetické zásoby len niektorých zariadení, tieto zariadenia by boli trvalo vyťažované až kým by sa ich energetická rezerva nezrovnala so zvyškom siete.

### 2.3.9 LEACH – Low Energy Adaptive Clustering Hierarchy

Protokol sa snaží zoskupiť zariadenia do hierarchie tak, aby minimalizoval spotrebu energie. V sieti sa nachádzajú zariadenia s úlohou clusterhead (správca skupiny) a k týmto zariadeniam sa následne pripájajú ostatné zariadenia a vytvárajú tak skupiny. Voľba clusterhead-ov je založená na náhode a výpočte podľa zbytkovej energie, čísla aktuálneho intervalu a podľa počtu clusterhead-ov, ktoré majú byť v sieti. Zariadenia označené ako clusterhead musia byť pripojené priamo na BS, čo je mierna nevýhoda tohto riešenia. Na druhej strane môže na clusterheadoch prebiehať agregácia dát a na BS sa posielajú len takto agregované dáta, čo šetrí energiu. Clusterhead navyše môže zariadeniam k nemu pripojeným zaslať časový harmonogram, podľa ktorého mu potom posielajú dáta, z čoho vyplýva veľká úspora energie vzhľadom na to, že zariadenie môže byť mimo času, kedy má byť aktívne podľa harmonogramu, uspať, a tým sa jeho energetická spotreba blíži nule.

LEACH dve fázy:

- **fáza nastavovania** Počas tejto fázy sa volia clusterheads, po zvolení začnú zariadenia vybrané ako clusterhead posielajú signál a ostatné stanice sa pripoja na základe najsilnejšieho signálu.
- fáza prenosu dát počas tejto fázy je štruktúra siete statická a prenášajú sa dáta. Táto fáza by mala byť výrazne dlhšia ako fáza nastavovania

### 2.3.10 HEED – Hybrid energy-efficient distributed clustering

Obdoba LEACH 2.3.9 protokolu, ktorý rieši jeho hlavnú nevýhodu, a to nutnosť pripojenia clusterhead zariadení priamo k BS. Na pripojenie clusterheadov k BS sa používa ľubovoľný iný protokol. Toto riešenie prináša ďalšiu energetickú úsporu vďaka možnosti znížiť vysielací výkon na clusterhead zariadeniach vďaka možnosti preposielať tieto dáta cez iné zariadenie.

### 2.3.11 TEEN & APTEEN – Threshold-sensitive Energy Efficient Protocols

Pri meraní fyzikálnych vlastností používa prahy, čím sa snaží znížiť počet prenosov, keďže senzor zasiela dáta len keď prekročí danú hodnotu a nie periodicky v čase. Využívajú sa dva prahy mäkký a pevný. Pevný prah určuje hodnotu, pri ktorej prekročení má byť poslaná správa a mäkký prah posiela správu, ak je zmena hodnoty od posledného merania väčšia ako tento prah. Hodnota mäkkého prahu sa môže meniť pri každej zmene clustra. Nevýhodou je, že ak sú hodnoty prahov zvolené nesprávne, nebudú dáta poslané na BS nikdy.

### 2.3.12 SPAN – Coordination of Power Saving with Routing

Protokol založený na určovaní špeciálnych smerovacích zariadení (coordinator). Zariadenie označí samo seba za koordinátora, ak medzi svojimi susedmi nájde dva vrcholy, ktoré vedia spolu komunikovať len cez neho samého a nie je v okolí žiadny koordinátor, ktorý by túto úlohu mohol zastať. Množina všetkých koordinátorov tvorí chrbticu tejto siete. Protokol obsahuje rutinu, ktorá v prípade, že v niektorom mieste siete existuje viac zariadení, ktoré sa v úlohe koordinátora môžu zastúpiť, bude sa úloha koordinátora pravidelne posúvať medzi týmito uzlami podľa energetických rezerv týchto zariadení. Koordinátor pravidelne komunikuje so svojimi susedmi a overuje, či uzly nevedia komunikovať medzi sebou priamo alebo v okolí nevznikol iný koordinátor, ktorý by túto úlohu mohol prevziať. Svojej úlohy sa môže teda vzdať len v prípade, že všetci jeho susedia vedia komunikovať medzi sebou navzájom bez jeho pomoci. Správova zložitosť tohto algoritmu je  $O(n)$ .

### 2.3.13 Zoznam ďalších protokolov

- **Gradient-Based Routing** upravená varianta Directed Diffusion [2.3.5](#)
- **COUGAR**
- **ACQUIRE**
- **Routing Protocols with Random Walks** balancovanie záťaže, vhodné pre veľké a statické siete, umožňuje vypínanie a zapínanie uzlov, využíva distribuovaný Bellman-Ford algoritmus
- **PEGASIS** Power-Efficient Gathering in Sensor Information Systems úprava LEACH [2.3.9](#) algoritmu so snahou znížiť energetickú náročnosť
- **MECN** Small Minimum Energy Communication Network, snaží sa nájsť energeticky efektívne podsiete, snaha preposielať dáta cez uzly tak, aby jednotlivé senzory mohli znížiť vysielačiaci výkon
- **SOP** Self Organizing Protocol, časť sensorov overuje vlastnosti prostredia a posiela dáta do špeciálne navrhnutých uzlov, ktoré slúžia ako smerovače. Smerovače slúžia ako chrbticová časť siete. Zozbierané dáta sú preposielané cez smerovače do BS. Každý senzor musí mať prístup aspoň k jednému smerovaču, aby mohol komunikovať so sieťou.
- **VGA** Virtual Grid Architecture routing
- **HPAR** Hierarchical Power-aware Routing
- **TTDD** Two-Tier Data Dissemination
- **GAF** Geographic Adaptive Fidelity
- **GEAR** Geographic and Energy Aware Routing
- **MFR** Most Forward within Radius
- **GEDIR** The Geographic Distance Routing
- **GOAFR** The Greedy Other Adaptive Face Routing

## Kapitola 3

# Riešenie od firmy Microchip

Firma Microchip je výrobcou microcontrollerov a analógových obvodov [7], ktorá v rámci svojich produktov vyrába aj moduly pre bezdrátovú komunikáciu [8]. V rámci podpory pre zakazníkov dodáva tieto moduly aj s knižnicami pre jednoduchú realizáciu bezdrátovej siete. V rámci tejto bakalárskej práce sa budem z ponúkaných produktov venovať výhradne protokolu MiWi 3.2 ktorý sa v práci používa spolu s bezdrátovým modulom MRF89 3.2.6.

### 3.1 MiMac

Microchip Wireless Media Access Control [9] je knižnica poskytovaná firmou Microchip ako nízkoúrovňová vrstva pre kontrolu rádia. MiMac zastáva funkciu fyzickej vrstvy v ISO/OSI modeli. Poskytuje jednotné rozhranie pre všetky podporované rádiá, to umožňuje jednoduchšiu implementáciu protokolu s vyššou mierou abstrakcie, navyše bude bez väčších úprav môcť byť použitý s viacerými rádiami. Vďaka jednoduchšiemu rozhraniu vyžaduje MiMac od programátora menšie znalosti o vnútornej implementácii rádia, čo uľahčuje a urýchľuje vývoj. MiMac obsahuje aj bezpečnostný modul, ktorý umožňuje šifrované spojenie.

#### 3.1.1 Format packetu

MiMac rámec sa podľa [9] skladá z dvoch hlavných častí, fyzického a mac rámca. Žiadna z hodnôt fyzického rámca nemôže byť priamo nastavovaná pomocou MiMac vrstvy. Grafické zobrazenie je na obrázku 3.1.1

LAYER	PHY			MAC		
NAME	Preamble	SFD	Packet Length	Header	Data	CRC
BYTE	Various	Various	0-1	2-21	Various	0-2

Obrázek 3.1: MiMac rámec

- **Fyzický rámeč**

Rámeč využívaný priamo rádiom na zaistenie synchronizácie a zistenie validity spojenia. Tento rámeč je väčšinou implementovaný priamo v hardveri rádia a jeho sú povolené, len ak to umožňuje rádio a musia sa prevádzať priamym nastavením registrov v rádiu.

- **Preamble** preambula fyzickej vrstvy rádia, používa sa na synchronizáciu komunikácie, jej dĺžka a obsah je závislá na konkrétnom rádiovom module. Niektoré rádiové moduly umožňujú zmenu jej dĺžky a obsahu.
- **SFD** Start of Frame Delimiter oddeľovač preambule od dátovej časti fyzického rámca, je väčšinou používaný spolu s preambulou na overenie synchronizácie. Podobne ako preambula môže byť na niektorých rádiových moduloch nastaviteľný prípadne niektoré rádia umožňujú jeho úplné vypnutie.
- **Packet length** dĺžka packetu, parameter určuje množstvo dát, ktoré má byť odvysielané, na niektorých rádiách je možné nastaviť pevnú dĺžku packetu, v takom prípade môže byť táto hodnota vynechana.

- **MAC rámeč**

Popisuje dĺžky a semantiku jednotlivých častí rámca prenášaného fyzickou vrstvou. Rozdeľuje tento rámeč do troch častí:

- **Header** obsahuje informáciu pre príjemcu o obsahu packetu a tom, ako ma byť tento packet interpretovaný jeho formát je možné vidieť na obrázku [3.1.1](#)

NAME	Frame Control	Extra Control	Sequence Number	Destination Address	Source Address
BYTE	1	0-3	0-1	0-8	0-8

Obrázek 3.2: MiMac hlavička

Prvý byte tejto hlavičky obsahuje metadáta hlavičky, označenej na obrázku [3.1.1](#) ako **Frame control**. Tento byte sa používa pri interpretácii dát z mac hlavičky. Štruktúra frame control bytu je na obrázku [3.1.1](#). Význam jednotlivých bitov:

- \* **Packet type** dvojbítová hodnota určujúca o aký typ packetu sa jedná, Typ packetu môže byť :
  - Data, hodnota 0b01, správa tohto typu je priamo odovzdávaná vyššiemu protokolu na spracovanie, prípadne môže byť spracovávaná priamo aplikáciou, ak nad mac vrstvou nie je žiadny iný protokol
  - Command, hodnota 1, rovnako ako dáta je odpoveď tohto typu rovno posunutá na ďalšie spracovanie, ale pri odovzdávaní tejto správy je nastavený príznak, že sa jedná o command
  - Ack, hodnota 2, typ packetu používaný na potvrdzovanie doručenia správy priamo v mac vrstve, tento packet nie je nikdy odovzdaný na spracovanie. Neobsahuje ani adresu odosielateľa ani príjemcu len sekvencné číslo packetu, ktorý potvrdzuje.
  - Hodnota 3 je rezervovaná, ale nie je využitá.

- \* **Broadcast** príznak, či má byť správa poslaná všetkým zariadeniam v dosahu. V prípade, že je nastavený, nie je do packetu pridaná adresa cieľa. Ak je tento byt vynulovaný, jedná sa o poslanie správy unicastom na konkrétny cieľ.
- \* **Security** príznak, či bola správa pred odoslaním zašifrovaná. V prípade, že bol príznak nastavený, musí príjemca po prijatí packetu najskôr paket odšifrovať. V prípade, že je tento byt nastavený, pridáva sa za mac hlavičku hlavička bezpečnostného protokolu. Viac informácií o tejto hlavičke je možné nájsť v sekcii "MiMAC Security Module" v /citemimacAN.
- \* **Acknowledgement** príznak, či je od príjemcu potrebné potvrdenie. V prípade, že áno, musí príjemca odoslať paket s typom Ack a s rovnakým sekvencným číslom, ako má aktuálny paket.
- \* **Destination present** príznak, či je v mac hlavičke prítomná adresa cieľa. Tento príznak musí byť vynulovaný v prípade, že sa posielajú broadcastová správa alebo je posielaná správa typu Ack.
- \* **Source present** príznak, či sa nachádza v mac hlavičke adresa zdroja. Nie je nijak spracovávaná na úrovni mac vrstvy. Prítomnosť tejto hodnoty je závislá na potrebách konkrétnej aplikácie.

NAME	Packet Type	Broadcast	Security	Repeat	Ack	Destination present	Source present
BIT	2	1	1	1	1	1	1

Obrázek 3.3: Frame control

Byty 1–3 mac hlavičky obsahujú **Extra control** formát tohto packetu je popísaný v [9] a v rámci tejto práce nie je popisovaný, keďže ho rádio používané v tejto práci, hardverovo nepodporuje.

Byt 4 obsahuje text `Sequence number` (poradové číslo) aktuálneho packetu. Táto hodnota sa používa pre jednoznačné určenie a používa sa aj ako identifikačné číslo pre Ack packet. Jeho hodnota sa volí na každom zariadení náhodne a pri každom odoslanom pakete musí byť incrementovaná. V prípade, že sa jedná o správu typu broadcast, používa sa táto hodnota ako identifikátor tejto broadcastovej správy, vďaka čomu je možné rozlíšiť, či už bol daný broadcastovaný packet v histórii zariadením prijatý alebo nie a v prípade, že bol, tento paket sa už znova nepreposiela do siete a zabraňuje sa tak opakovanému odosielaniu tých istých broadcastových správ v sieti.

Byty 5–21 sú podľa nastavení v byte frame control použité na **Destination address** a **Source address** pričom samotná dĺžka adresy je závislá na nastavení knižnice a môže byť v rozmedzí 1–8. V prípade, že je dĺžka týchto dvoch adres kratšia, prípadne nie sú prítomné, je miesto použité ako Payload.

- **Payload** dáta prenášané pomocou mac vrstvy. Tieto dáta sú odovzdávané priamo pomocou programového rozhrania ďalším častiam aplikácie, či už sa jedná o vyšší protokol alebo priamo užívateľskú aplikáciu. V prípade, že boli dáta na mac

vrstve šifrované, sú ešte pred odovzdaním dešifrované. Maximálnu dĺžku tejto správy je možné spočítať ako maximálnu dĺžku správy prenášanú rádiom minus mac header a crc.

- **CRC** kontrolný súčet pre kontrolu integrity packetu. Na niektorých rádiách je počítaný hardwarovo. V prípade, že rádio podporu kontrolného súčtu nemá, je tento kontrolný súčet nahradený softwarovo v rámci mac vrstvy. V prípade, že má rádio hardwarovu implementáciu kontrolného súčtu, nie sú tieto dva byty potrebné.

### 3.1.2 Nedostatky v Mimac

Problémy a riešenia, ktoré na implementácii MiMac považujem za nevyhovujúce:

- **licencia a prenos na iné zariadenia** v licenčných podmienkach uvedených v hlavičkových súboroch je uvedené, že implementácia tejto vrstvy by mala byť používaná výhradne na procesoroch firmy Microchip a použitie na iných zariadeniach je porušením licenčných podmienok.
- **premenlivá dĺžka packetu** maximálna dĺžka packetu sa mení podľa typu správy. V prípade, že je nad touto vrstvou použitá nejaká ďalšia vrstva, užívateľ si nemôže byť istý maximálnou dĺžkou správy a z toho dôvodu musí používať najkratší možný variant. Navyše je dĺžka hlavičky pomerne veľká, pri implementácii na MRF89 a dĺžke mac adresy 4 Byty je veľkosť hlavičky až 10 Bytov v prípade, že sú prítomné obidve adresy.
- **strácanie a strnutie paketou** pri štúdiu implementácie pre rádio MRF89 bola nájdená v kóde chyba v spôsobe ukladania paketov do zásobníka. Prichádzajúce pakety boli ukladané do LIFO bufru, čo mohlo a pravdepodobne aj spôsobovalo veľké problémy pri stabilite spojenia, hlavne v dobe keď bolo na zariadenie posielané väčšie množstvo paketov. Paket sa mohol uložiť do fronty a vzhľadom na charakter fronty mohol byť poslaný na spracovanie po veľmi dlhej dobe. V nepriaznivých podmienkach by sa tento paket/pakety stratili natrvalo, keďže by nikdy nemohlo dôjsť k ich vyňatiu z fronty a navyše táto chyba by spôsobila obmedzenie veľkosti zásobníka na minimálnu hodnotu.
- **chýbajúca kvalitná detekcia chýb** v rámci tejto knižnice nie je možné nijakým spôsobom detekovať chybu prenosu dát medzi dvomi zariadeniami. Jediná možnosť ako zistiť, či bol paket prenesený na druhé zariadenie, je pomocou aktivovania žiadosti na acknowledge. Vzhľadom na to, ako toto potvrdzovanie funguje, je možné, že iné zariadenie v sieti pošle packet s rovnakým sekvenčným číslom a žiadosťou na potvrdenie a odpoveď pre toto zariadenie spôsobí potvrdenie obidvoch žiadostí.
- **potreba sekvenčného čísla** po celú dobu je potrebné v mac vrstve udržiavať sekvenčné číslo. Podľa definície má byť na začiatku nastavené toto číslo na náhodnú hodnotu. V implementácii sa ako zdroj takto nahodného čísla používa register z časovača, ktorý je podľa dokumentácie v dobe spustenia alebo reštartu microcontrolleru vynulovaný. Z toho dôvodu by táto hodnota nemala byť náhodná a ak je náhodná, jej rozptyl je len veľmi malý.
- **štruktúra** implementácia knižnice vyžaduje, aby boli v projekte prítomné špeciálne hlavičkové súbory, ktoré includeje a znemožňuje tak užívateľovi navrhnúť si vlastné

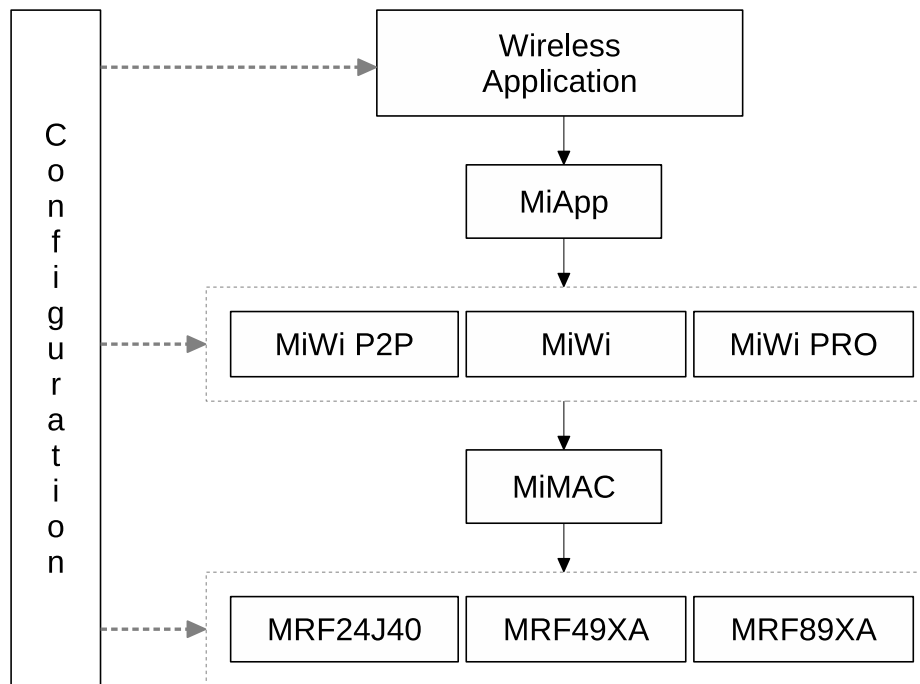
usporiadanie súborov v projekte. Možnou príčinou tohto problému je to, že implementácia počíta s projektami vytvorenými pomocou vývojového MPLABX keďže štruktúra týchto projektov zodpovedá súborom icludovaným v MiMac knižnici.

- **nadmerné používanie makier** veľká časť nastavení v MiMac je riešená pomocou deklarovania preprocesorového macra. Keďže sa na základe takto nastavených hodnôt generuje preprocesorom hlavičkový súbor, musia byť tieto nastavenia spravené na správnom mieste a v prípade, že prekladač zmení poradie expandovania makier alebo includovania hlavičkových súborov, môže ľahko dôjsť k zlému prekladu knižnice. Toto sa môže stať napríklad, keď prekladač preloží najskôr hackový súbor v súbore, kde nastavenia nie sú a až následne v súbore, kde boli definované. Keďže je v hlavičkovom súbore ochrana pomocou makier, aby sa prekladal hlavičkový súbor len raz, nedôjde k opätovnému dosadeniu hodnôt a užívateľské nastavenia nebudú použité.
- **nastavovanie kanálov a frekvencie** pri implementácii MiMac pre MRF89(ostatné zdrojové kódy neboli prezerané) bolo nastavovanie kanálov a frekvencie rádia riešené pomocou predpočítanej tabulky hodnôt pre 3 registre rádia. Výsledné tabulky zaberajú veľa miesta na mcu a zároveň neumožňujú meniť frekvenciu rádia za chodu zariadenia, ale je nutné celý kód opätovne preložiť.
- **implementácia** v implementácii tejto knižnice sú používané veľmi neprehľadné konštrukcie, ktoré zvyšujú náročnosť vykonávania dodatočných úprav, veľká časť kódu je veľmi neefektívna ako dátovo, tak výpočtovo, pričom úprava a zjednodušenie môžu byť veľmi jednoduché. Ako príklad môže byť switch, ktorý má niekoľko vetiev, ale v každej vetve sa vykonáva ten istý kód, čo vzhľadom na vypnuté optimalizácie vo free verzii prekladača môže výrazne zväčšiť veľkosť kódu.



## 3.2 MiWi™

Je skupina bezdrátových protokolov postavená ako nadstavba nad technológiou MiMac3.1. MiWi™ je skupina protokolov pre siete s rôznou topológiou a veľkosťou. Zároveň definuje jednotné aplikačné rozhranie spoločne pre všetky tieto protokoly. Podľa [10] [3] ide o protokoly určené pre siete s nízkou rýchlosťou prenosu, komunikujúce na krátku vzdialenosť a s nárokom na nízku cenu. Architektúra vysvetľujúca napojenie na MiMac, rádiá a užívateľskú aplikáciu je na obrázku.



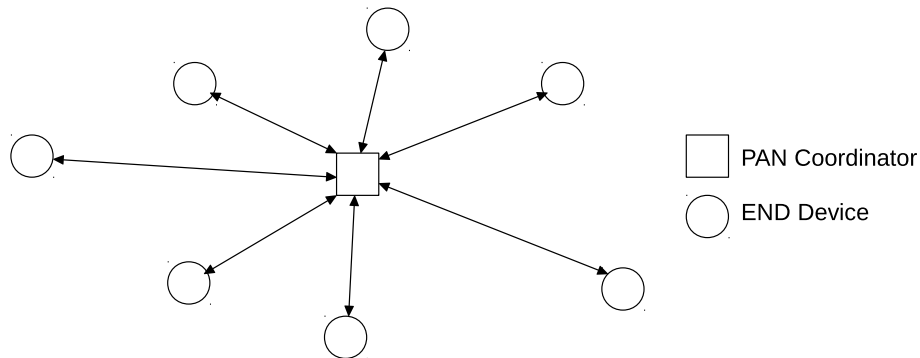
Obrázek 3.4: MiWi™ vývojové prostredie

V rámci ďalších podkapitol sa budem venovať prevažne protokolu MiWi PRO, ktorého úpravou som sa mal v tejto práci zaoberať. Protokoly MiWi P2P a MiWi nie sú pre navrhované univerzálne riešenie projektu inteligentnej domácnosti vhodné.

### 3.2.1 Bezdrátové protokoly v rámci MiWi<sup>TM</sup>

- **MiWi P2P**

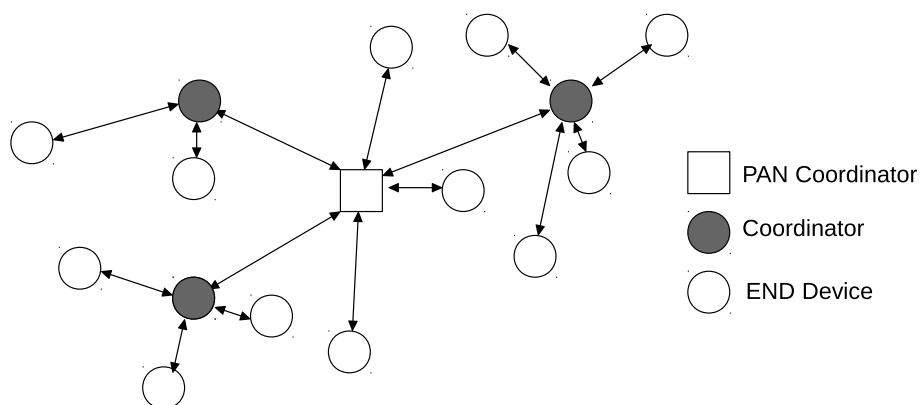
je najjednoduchšia varianta protokolu. Podporuje iba jednoduchú hviezdicovú alebo peer-to-peer topológiu. Nepodporuje žiadne smerovacie protokoly, preto môžu byť správy preposielané len medzi zdrojom a cieľom. Jedna z najväčších výhod tohto protokolu je jeho jednoduchosť. Kvôli neprítomnosti smerovacieho protokolu však nie je možné rozširovať dosah zariadenia pomocou preposielania cez iné uzly a kvôli tomu je dosah tohto protokolu relatívne malý.



Obrázek 3.5: MiWi<sup>TM</sup>P2P ukážka topológie

- **MiWi**

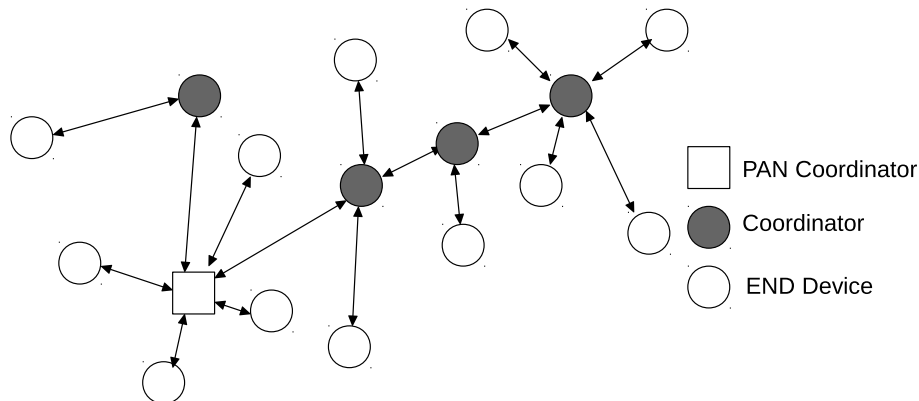
je rozšírením MiWi P2P o základné routovacie mechanizmy. Podporuje až 8 koordinátorov, zariadenie na smerovanie packetov, vrátane PAN koordinátora. Je povolené preposielanie packetov, pričom maximálny počet preposlaní je 4 pre komunikáciu medzi dvomi koncovými bodmi a 2 medzi pan koordinátorom a koncovým bodom. Zariadenie nastavené ako koordinátor sa môže pripájať len priamo k pan koordinátorovi. Zariadenie, ktoré má možnosť byť koordinátorom ale zároveň nemá prístup k PAN koordinátorovi alebo už je v sieti všetkých 8 koordinátorov sa prepne do stavu, kedy sa javí ako koncové zariadenie a pripojí sa do siete ku koordinátorovi v dosahu.



Obrázek 3.6: MiWi<sup>TM</sup>ukážka topológie

- **MiWi PRO**

je rozšírením MiWi o možnosť pripájať zariadenia nastavené ako koordinátor aj na iného koordinátora, nielen na PAN coordinator. Zároveň bol zvýšený maximálny počet koordinátorov v sieti z 8 na 64. V prípade, že by sieť mala lineárnu topológiu, celkový počet skokov v sieti by bol 65 od koncového zariadenia ku koncovému zariadeniu a 64 skokov medzi koncovým bodom a PAN koordinátorom. Chovanie MiWi PRO je rovnaké ako MiWi s rozdielom, že podporuje oveľa väčšie siete.



Obrázek 3.7: MiWi<sup>TM</sup>PRO ukážka topológie

### 3.2.2 Format paketu

Formát paketu v MiWi PRO definuje sieťovú vrstvu bezdrátového protokolu. Táto vrstva je posta, podporuje smerovanie s opakovaným preposielaním správy. Na obrázku 3.2.2 je formát hlavičky tejto vrstvy, kde jednotlivé časti majú tento význam:

- **Hops** počet možných preposielaní tohto paketu, hodnota 0x00 znamená nepreposielať tento paket.
- **Frame Control** skupina bytov ovplyvňujúca význam tohto paketu
  - **Ack Request** žiadosť o zaslanie potvrdenia o prijatí packetu
  - **Intrctlst** intra cluete byt, v implementácii miwi pro je rezervovaný a má vždy hodnotu 1.

NAME	Reserved	Ack Request	Intrctlst	Reserved
BIT	7-3	1	1	1

Obrázek 3.8: MiWi<sup>TM</sup>PRO štruktúra Frame Control Bytu

- **Destination PANID** unikátny identifikátor PAN coordinatora, do ktorej patrí cieľový uzol.
- **Destination Short Address** adresa cieľa packetu, formát podľa 3.2.2

- **Source PANID** unikátny identifikátor PAN coordinatora, do ktorej patrí cieľový uzol.
- **Source Short Address** Adresa zdroja paketu, formát podľa 3.2.2
- **Sequence Number** poradové číslo paketu, môže byť použité pre zisťovanie stavu paketu počas prechodu sieťou.
- **Report Type**
- **Report ID**

NAME	Hops	Frame Control	Destination PANID	Destination Short Address	Source PANID	Source Short Address	Sequence Number	Report Type	Report ID
BIT	1	1	2	2	2	2	1	1	1

Obrázek 3.9: MiWi<sup>TM</sup>PRO hlavička

**Formát adresy** Na adresáciu zdroja alebo cieľa paketu sa v MiWi používa 2Bytova adresa. Táto adresa sa skladá z:

- Adresy coordinatora, 6 bitova adresa koordinátora, ku ktorému je daný End Device pripojený.
- RxOnWhenIdle, 1bitovy priznak ci maku byt spravy dorucovane ked je zariadenie uspane
- Adresa End Devicu, 7 bitova hodnota, ktorú získava zariadenie po pripojení kukcoordinátorovi. V prípade, že je hodnota tejto adresy 0x00, je táto správa určená pre koordinátora.

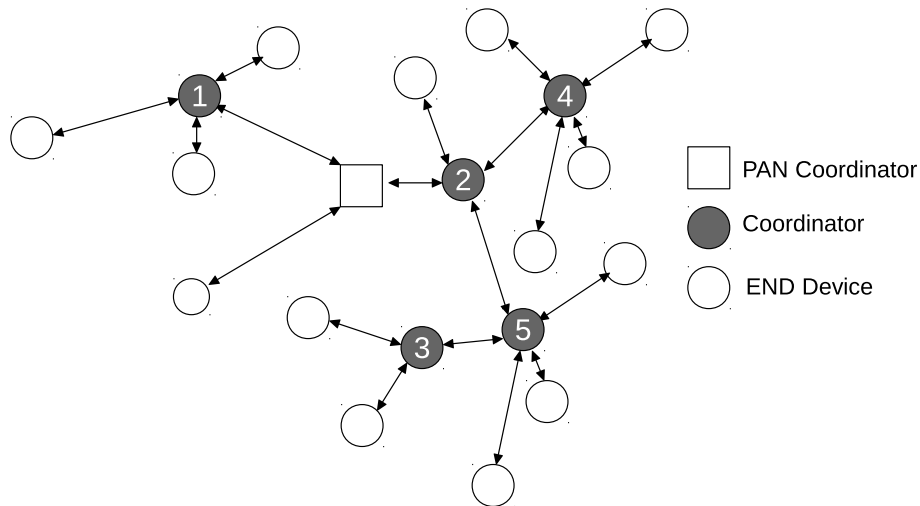
V rámci podpory boli v protokole rezervované adresy 0xFFFF pre broadcastové správy a adresy 0xFFFFE a 0xFFFFD pre multicastové správy na koncové zariadenia a koordinátorov a protokol aj broadcast a dve multicasovne skupiny. Pri poslaní správy na týchto adresách je automaticky správa broadcastovaná/multicastovaná cez celú sieť. To pomáha znižovať počet správ, ktoré by bolo nutné zasielať samostatne každému zariadeniu pri ulohách ako je napríklad distribuovanie smerovacích tabuliek.

### 3.2.3 Smerovanie

MiWi PRO podporuje dva rozdielne prístupy pri smerovaní dát.

- **Tree routing** je smerovací mechanizmus, kde je správa preposielaná medzi dvomi zariadeniami, ktoré majú medzi sebou vzťah rodič–potomok. Pre používanie tohto mechanizmu je potrebné, aby každý smerovací prvok, v prípade MiWi PRO sa jedná o koordinátora, mal dostupné informácie o topológii siete. V rámci obmedzenia za adresu smerovacieho prvku v MiWi PRO musí pridelovať PAN koordinátor, má PAN všetky informácie o topológii siete. Vďaka tomu je relatívne jednoduché zabezpečiť, aby všetky smerovacie zariadenia mali informácie o topológii, pričom stačí túto informáciu preposlať broadcastom na všetky smerovacie uzly pri každej zmene topológie. Informácie o topológii sú uchovávané v tabuľke, kde je pre každé zariadenie uložený

jeho rodič. Takáto tabuľka je veľmi efektívna na úschovu a vyhľadávanie dát. Takáto tabuľka môže byť (a je) realizovaná jednoduchým poľom adries, pričom index v tomto poli je index vyhľadávania v tejto tabuľke. Vyhľadanie rodiča k určitej adrese je výpočtovo veľmi jednoduché, keďže sa stačí pozrieť do tohto poľa s indexom rovným tejto adrese a hodnota uložená na tomto mieste bude adresa rodiča. Príklad takej tabuľky vidno na obrázku 3.2.3.



Obrázek 3.10: Príklad topológie MiWi PRO siete

Adresa koordinátora	Adresa rodičovského koordinátora	Popis
0	0	Pan coordinator je rodič sám sebe
1	0	Coordinátor 1 má rodiča PAN
2	0	Coordinátor 2 má rodiča PAN
3	5	Coordinátor 3 má rodiča koordinátora 5
4	2	Coordinátor 4 má rodiča koordinátora 2
5	2	Coordinátor 5 má rodiča koordinátora 2

Obrázek 3.11: Príklad tabuľky MiWi PRO tree routingu

V dobe, keď sa do siete prihlasuje nové zariadenie ako koordinátor, musí sa zariadenie, ku ktorému sa pripája, postarať o oznámenie tejto udalosti PAN koordinátorovi, ktorý v odpovedi zašle novú pridelenú adresu. Zariadenie musí preposlať túto správu na nové zariadenie.

- **Mesh smerovanie** Toto smerovanie sa používa v MiWi PRO primárne. Až v prípade, že sa mu nepodarí správu doručiť, je použitý tree routing. Smerovanie nie je založené na štruktúre rodič-potomok. Z toho plynie výhoda, že mesh routing funguje aj v dobe, keď sa z nejakého dôvodu poruší topológia siete, napríklad výpadok

koordinátora. Aby mohlo toto smerovanie fungovať, je potrebné aby uzly vedeli o susedných uzloch. Z toho dôvodu je na zariadení uložené, ktorých susedov môže zariadenie kontaktovať priamo. Obnovovanie tejto tabuľky sa deje periodicky počas doby, keď protokol normálne funguje a zariadenie si vymieňa tabuľky so svojimi susedmi.

### 3.2.4 Prihlasovanie

Spôsob prihlasovania zariadenia do MiWi PRO siete je závislý od druhu zariadenia a jeho umiestnenia. Pokiaľ sa jedná o nové zariadenie typu end-device, zariadenie pošle broadcastom žiadosť o prihlásenie (beacon request), na ktorú odpovedajú všetky koordinátory v dosahu. Zariadenie si vyberie podľa sily signálu zariadenie s navyššou intenzitou a pošle mu association request. V prípade, že je možné, aby sa toto zariadenie pripojilo, odošle sa zo zariadenia správa association response. Od tejto chvíle je end-device pripojený do siete ku koordinátorovi, s ktorým prebehlo prihlasovanie. V prípade, že sa jedná o zariadenie typu koordinátor, môžu nastať dva scenáre. Prihlasovanie začína rovnako ako pri end-device. Pri poslaní association request záleží na tom, komu bola táto správa poslaná. Buď bola poslaná priamo PAN koordinátorovi, v tomto prípade sa hneď po prijatí tejto správy PAN koordinátorom zisťuje, či môže byť toto zariadenie prijaté ako koordinátor alebo nie a v prípade, že áno, je mu rovno pridelená adresa. V prípade, že bol association request prijatý iným koordinátorom, preberá tento koordinátor zodpovednosť za jeho prihlásenie u PAN, kde zasiela coordinator report. Po jeho prijatí PAN koordinátorom dochádza opäť k určovaniu, či je možné zariadenie prijať ako koordinátora a je mu priradená adresa a následne je odoslaná správa coordinator response pre zariadenie, ktoré pripojenie sprostredkovalo a toto zariadenie sa stáva rodičovským koordinátorom. Pokiaľ sa bude zariadenie pripájať ako koordinátor, dochádza k zmene v routovacej tabuľke, kde sa tento koordinátor zapíše, a preto je potrebné, aby sa upravená tabuľka rozpropagovala na všetky koordinátory. V momente, keď je propagácia hotová, dostáva zariadenie správu association response od rodičovského koordinátora alebo od PAN a zariadenie je pripojené do siete. Po dokončení pripájania v rámci tree raoutingu, je potrebné, aby zariadenie nastavilo ešte tabuľky pre mesh routing. To sa deje už počas normálneho chodu siete, keď zariadenie odosiela pakety pomocou broadcastu na všetky okolité koordinátory so správou routing table report a tieto uzly mu odpovedajú routing table report.

### 3.2.5 Nedostatky v MiWi a Miwi PRO

V MiWi pro je nedostatkov výrazne menej ako pri MiMac 3.1 vrstve. Medzi hlavné nedostatky patrí opäť licencia, ktorá limituje používanie jej kódu len na microprocesory firmy Microchip a samozrejme slabá prehľadnosť programu. Najväčšiu časť kódu tvorí funkcia MiWiPROTasks, ktorá implementuje veľkú časť funkcionality knižnice. Obsahuje prijímanie, spracovávanie, ukladanie správ do buffru, odosielanie a aj prihlasovanie. Kvôli tomu, že táto funkcia nie je rozčlenená do menších na sebe nezávislých častí, stáva sa s jej dĺžkou vyše 2400 riadkov kódu veľmi neprehľadno a je veľmi náročné pochopiť ako funguje, prípadne ju upraviť.

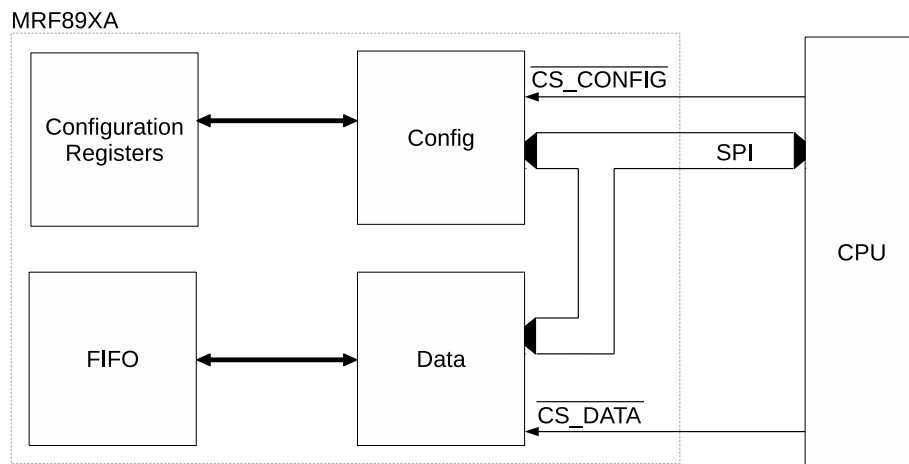
MiWi pro má jednu ďalšiu veľkú nevýhodu, ktorá je tiež spojená s funkciou MiWi-PROTasks, a to konkrétne nutnosť túto funkciu volať periodicky z užívateľského programu. Táto požiadavka výrazným spôsobom ovplyvňuje zložitost a čitateľnosť výslednej užívateľskej aplikácie. Jej príkladom môže byť aj ukážka z Aplikation Note [3] na strane 16 príklad 1, kde sa táto požiadavka prejavuje ako nutnosť implementovať celú aplikáciu ako

skupinu funkcií alebo blokov kódu a vytvorením stavového automatu, ktorý postupne tieto funkcie/bloky vykonáva.

### 3.2.6 MRF89

V rámci celej tejto práce a aj projektu inteligentnej domácnosti sa používa rádio od firmy Microchip s označením MRF89XA. V tejto sekcii je popísaná základná štruktúra a funkčnosť rádia a popis chýb, ktoré sa počas práce s rádiom vyskytli a je potrebné sa im vyvarovať. Manuály, z ktorých je čerpané a obsahujú plný popis rádia sú [4] [5].

Komunikácia medzi procesorom a rádiom prebieha pomocou sériovej linky SPI. V rámci rádia sa nachádzajú dva moduly pripojené k tejto zbernici, ako to je možné vidieť na obrázku 3.2.6. Medzi týmito dvomi koncovými zariadeniami je možné prepínať pomocou chip selectov CS\_DATA a CS\_CONFIG. Pri aktivovaní CS\_CONFIG je k SPI pripojená riadiaca časť rádia, ktorá umožňuje prístup k stavovým a konfiguračným registrom. Pri aktívnom CS\_DATA je na SPI dátový modul, pomocou ktorého je realizované samotné prijímanie a odosielanie dát.



Obrázek 3.12: Vnútorne usporiadanie rádiového modulu z pohľadu komunikácie cez SPI

Na čo je potrebné si dať pri práci s rádiom pozor:

- **Komunikačná rýchlosť SPI** konfiguračná a dátová časť rádia nemajú maximálnu komunikačnú rýchlosť. Zatiaľ čo konfiguračná časť je schopná komunikovať na frekvencii až 6MHz, dátová ma maximum 1MHz. Táto vlastnosť rádia spôsobuje, že pri nastavení vyššej komunikačnej rýchlosti ako je 1MHz, je možné nastavovať a čítať registre rádia, ale nie je možné odosielať alebo prijímať pakety. Táto chyba sa veľmi ťažko hľadá, nakoľko komunikácia s rádiom sa javí ako funkčná.
- **dĺžka paketu** rádio podporuje paketový režim komunikácie, pri jeho použití je potrebné, aby na prvom Byte tohoto paketu bola hodnota dĺžky paketu. Táto hodnota sa do rádia posielajú spolu s dátami a zároveň je s dátami vyčítaná pri prijatí paketu. Rádio má maximálnu dĺžku paketu 64 vrátane tohto prvého Bytu a táto hodnota je uložená na spodných šiestich bitoch. Problém nastáva, ak sa do horných dvoch bytob uloží nejaká ďalšia informácia. Nielenže sa táto hodnota zahodí, ale paket sa vôbec neodosle.

- **vyber registra** komunikacii s controlnou castou radia je nutne davat pozor na format v akom sa posiela cislo registra, aby komunikacia fungovala spravne musia byt k 5tim bitom adresy registra spravne nastavene start, stop a bit urcujuci operaciu ktora sa bude vykonavat.



## Kapitola 4

# Návrh a implementácia vlastného protokolu

V tejto práci bolo potrebné implementovať protokol pre potreby projektu inteligentnej domácnosti, ktorý by umožňoval spoľahlivo prenášať informácie medzi uzlami a zároveň umožňoval uspávanie uzlov pre úsporu energie. Spoľahlivý prenos je veľmi dôležitý z dôvodu, že v sieti sa okrem senzorov nachádzajú aj aktory, zariadenia manipulujúce prostredím, ktoré sú ovládané cez túto bezdrátovú sieť. Táto vlastnosť je jedna z kľúčových vecí, prečo nový protokol vznikol.

Vzhľadom na nadmernú zložitosť a licenčné podmienky použité v riešení od spoločnosti Microchip, ktoré zvyšujú náročnosť opravy chýb v protocoloch a zabraňujú prenosu systému na zariadenia bez microcontolera od tohto výrobcu mi bolo počas konzultácií povolené namiesto upravovania existujúceho kódu protocol kompletne prerobiť.

### 4.1 Požiadavky na návrh

- **spoľahlivosť prenosu** z dôvodu prítomnosti aktorov v sieti je potrebné nielen, aby sa informácia dostala spoľahlivo na koncový uzol, ale aj aby sa tak udialo v dostatočne krátkom čase. Riešenie s opakovanými pokusmi o doručenie a čakaním na potvrdenie preto nie je veľmi vhodná. Ďalej by bolo vhodné aby, ak v prípade, že sa nepodarilo takúto správu doručiť, aby bola táto chyba detekovaná a prípadne bolo možné na ňu reagovať.
- **úspora energie** v sieti existujú dva typy zariadení: aktor a senzor. Pri aktoroch je napájanie spravidla pripojené do rozvodnej siete, takže majú neobmedzený zdroj energie, na druhej strane sú v sieti senzory, ktoré sú napájané z batérie, a tým pádom je ich energetická rezerva veľmi malá. Vzhľadom na používanie tejto sensorovej siete v domácnosti nároky na výdrž batérií ešte rastú, pretože by nákup batérií zvyšoval náklady na prevádzku a mohlo by dôjsť k nepokojnosti zo strany užívateľa a dokonca aj k úplnému odstavenia systému.
- **dynamickosť** v rámci nasadenia systému do domácnosti, kde s ním užívateľ bude chcieť manipulovať, je potrebné, aby boli jednotlivé uzly autonómne schopné obnoviť svoju funkčnosť po presunutí.
- **spätná kompatibilita s MiWi** je potrebné, aby aplikačné rozhranie bolo menené pokiaľ možno čo najmenej a prípadne, aby sa toto rozhranie menilo v prospech zjedno-

dušenia už existujúcej implementácie vzhľadom na to, že veľká časť aplikačnej logiky už bola vybudovaná na MiWi protokolom.

- **prenositelnosť** jednoduchá prenositeľnosť medzi rôznymi systémami ku ktorým bude možné pripojiť priamo rádiový modul tak aby boli tieto zariadenia schopné komunikovať.
- **implementácia bez periodicky volaných funkcií**

## 4.2 Štruktúra

Štruktúra nového protokolu bola rozdelená do viacerých častí, ktoré by viac menej mali odpovedať referenčnému modelu ISO/OSI. Jednotlivé časti majú stanovené rozhranie, vďaka ktorému je možné jednotlivé časti implementácie obmieňať bez potreby upravovať ostatné časti.

Protokol bol rozdelený na nasledujúce časti:

- **hw**, je vrstva hardwarovej abstrakcie, ktorá by mala pokrývať rozdiely medzi použitým procesorom a rôznym pripojením tohto rádia k mikroprocesoru podľa použitého plošného spoja.
- **fyzická vrstva**, fyzická vrstva je abstrakcia rádiového modulu, jej úlohou je zapuzdriť prácu s rádiovým modulom. V rámci tejto práce bola vytvorená len jedna implementácia tejto vrstvy vzhľadom na používanie len jedného rádia MRF893.2.6. V prípade potreby použitia iného modulu, je potrebné upraviť túto vrstvu. Avšak nie je potrebné meniť implementáciu žiadnej inej (v prípade, že ostáva hardwarové napojenie rovnaké ako u rádia, s ktorého sa vychádza).
- **linková vrstva** je náhradou za MiMac3.1. Zabezpečuje bezpečný prenos informácie medzi dvomi zariadeniami s možnosťou detekcie chyby. Ďalej umožňuje prijímať a posielať špeciálne typy správ určené na komunikáciu mimo protokol s cieľom pripojenia sa do siete.
- **sieťová vrstva** stará sa o pripájanie zariadení do siete, ich odstraňovanie, začlenenie presunutých zariadení späť do siete, o správu uspatých správ a o smerovanie paketov v sieti. Na PAN coordinator spravuje databázu pripojených zariadení.
- **fitp** jednoduchý obal nad net vrstvou, implementuje len základnú konfiguráciu celého protokolu, v budúcnosti by mala slúžiť na udržiavanie stabilného rozhrania aj v dobe, keď dôjde k väčším zmenám na sieťovej vrstve, aby sa znížila potreba úprav v aplikáciách využívajúcich tento protokol.

## 4.3 Smerovanie

Pri smerovaní sa rovnako ako pri MiWi PRO3.2 používa tree routing. Vzhľadom na spôsob akým zariadenia komunikujú v projekte nie je potrebné implementovať podporu mesh routingu. Komunikácia v sieti prebieha v prevažnej väčšine prípadov medzi PAN koordinátorom a koncovým zariadením, či už sa jedná o aktor alebo senzor. V prípade, že je potrebné doručiť správu medzi dvomi zariadeniami, prechádza táto správa postupne po jednotlivých rodičovských koordinátoroch, kde sa zisťuje, či je už hľadaný cieľ v podstrome tohto uzla.

V dobe, keď to nastene, prestane sa správa posielat' na rodičovský koordinátor, ale pošle sa na najbližšieho koordinátora, ktrpý leží v ceste k danému zariadeniu. V najhorskšm prípade dôjde k nájdeniu podstromu na PAN koordinátorovi, pre ktorého je celá sieť podstrom.

Pri návrhu smerovania bol navrhnutý rozšírený spôsob tree routingu, ktorý bez pridania nároku na ďalšie pamäťové zdroje pridáva možnosť posielat' správy z koordinátora na uzly, na ktoré ma priamy dosah. Rozdiel v implementácii spočíva v pridaní k adrese uloženej v routovacej tabuľke príznak, či je možné na toto zariadenie odosielať správy priamo. Na pridanie tohto príznaku nie je potrebná ďalšia pamäť nakoľko veľkosť adresy v routovacej tabuľke je 8bitov avšak v aktuálnej implementácii je na adresu používaných len 6. Z dôvodu, že takéto posielanie nebolo potrebné, nebola táto možnosť zatiaľ implementovaná.

## 4.4 Implementácia

Celá implementácia protokolu je vzhľadom na hw nároky zo strany microcontrolerov napísaná v jazyku C, ktorý je možné jednoducho používať s vyššími programovacími jazykmi ako je C++ ??, prípadne python ??. V kóde sú dosť často používané externé funkcie, ktoré slúžia ako najjednoduchšia forma callbacku, keďže nie sú definované v module, ktorý túto funkciu volá. Toto riešenie má dve zásadné výhody oproti používaniu iných typov callback funkcií, ako napríklad ukazovateľ na funkciu. Tieto výhody spočívajú v rýchlosti výsledného riešenia, keďže takéto funkcia je vo výsledku preložená ako obyčajná funkcia a zároveň garantuje, že takéto funkcia bude niekde v programe definovaná, čo do veľkej miery zabraňuje možnosti zabudnúť inicializovať callback.



Obrázek 4.1: Grafické znázornenie vrstiev protokolu

### 4.4.1 hw

Hardwarova sa snaží abstrahovať:

- komunikáciu po SPI zbernici,
- výber častí rádia pomocou i/o pinou,
- spracovávanie prerušení prijatých od rádia
- povoľovanie a zakazovanie prijímania prerušení od rádia
- implementáciu časovača

V hardwarovej vrstve by mali byť implementované aj dve funkcie určené na logovanie dát prenesených rádiiom, keďže na väčšine zariadení môže byť logovanie a jeho forma veľmi odlišná. V prípade, že logovať nič nechceme, tieto funkcie ostanú prázdne.

#### 4.4.2 Fyzická vrstva

Na tejto vrstve je implementovaná všetka logika, ktorá pracuje s hardwarom rádia. Jej hlavnou úlohou je poskytovať jednotné rozhranie pre prácu so všetkými rádiami tak, aby nemuselo dochádzať k zmenám ostatných častí aplikácie pri výmene rádiového modulu za iný. Rozhranie tejto vrstvy je veľmi jednoduché a jeho hlavnú časť tvoria funkcie:

- **PHY\_init** úlohou funkcie je nastaviť rádiový modul do stavu, kedy je možné prijímať a posielať dáta. Požadované frekvenčné pásmo, kanál a rýchlosť prenosu sú odovzdávané tejto funkcii pomocou štruktúry.
- **PHY\_send\_with\_cca** je variantou funkcie **PHY\_send** ktorá sa tiež nachádza v rozhraní tejto vrstvy, avšak pridáva kontrolu média pred odosielaním. Táto varianta je preferovaná, aby nedochádzalo v sieti ku kolíziám jednotlivých paketov. Jej nevýhodou je čakanie na uvoľnenie média, ktoré môže pozastaviť vykonávanie programu na vopred nedefinovaný čas.
- **PHY\_process\_packet** funkcia definovaná ako callback pomocou externej funkcie, táto funkcia volá po úspešnom prijatí paketu.

Funkcie `PHY_send_with_cca`, `PHY_send` a `PHY_process_packet` zabezpečujú celú komunikáciu so sieťou a umožňujú posielať a prijímať dáta jednotným spôsobom, keďže všetky tieto funkcie majú rovnaké parametre, a to ukazovateľ na dáta, ktoré sa majú odoslať alebo boli prijaté a ich dĺžku.

Ostatné funkcie tejto vrstvy slúžia na zisťovanie stavu rádia alebo média, v ktorom sa toto rádio nachádza a na nastavovanie parametrov rádia ako sú rozsah frekvencie, kanál, rýchlosť odosielania.

Poslednou funkciou, ktorá zatiaľ nebola zahrnutá v žiadnej časti je funkcia `PHY_timer_interrupt`, ktorá je tiež definovaná ako callback pomocou externej funkcie a jej úlohou je len prevolať udalosť vzniknutú v hardwarovej vrstve prijatím prerušenia od časovača. Dôvod, prečo je táto funkcia zahrnutá vo fyzickej vrstve, je len odstránenie nutnosti pristupovať k funkciám z hw vrstvy [4.4.1](#)

#### 4.4.3 Linková vrstva

Je alternatívou k MiMac knižnici, avšak snaží sa riešiť problémy uvedené v sekcii [??](#). Najvýznamnejšou úlohou, ktorú zabezpečuje táto vrstva, je bezpečný prenos dát medzi zariadeniami, s možnosťou detekovania niekoľkých druhov chyby.

#### Realizácia prenosu

Pri navrhovaní spôsobu prenosu medzi dvomi zariadeniami bola snaha navrhnúť riešenie, ktoré zabráni ako strate paketu, tak aj vzniku duplicitných paketov. Aby bolo možné garantovať, že packet bude doručený a zároveň nebude doručený viackrát, musel byť rozšírený handshake medzi zariadeniami. Komunikácia medzi dvomi zariadeniami sa skladá minimálne zo štyroch správ. Každá správa má svoj vlastný typ a špecifickú úlohu.

Typy správ vymieňaných počas prenosu správy:

- **DATA** táto správa je nositeľom dát po jej odoslaní sa nastaví na príjemcovi nastaví timeout čakajúci na správu typu ACK. Pri prekročení tohoto času sa pokladá táto správa za stratenú. Po prijatí tejto správy druhým zariadením je správa v prípade, že je určená tomuto zariadeniu a sú dáta aj metadáta tejto správy uložené do zásobníku v prípade, že neexistuje v zásobníku varianta správy s rovnakými metadátami. Po spracovaní a uložení tejto správy dochádza k odoslaniu správy ACK a nastavovaniu timeoutu na primárny paket, ktorý musí byť pred vypršaním potvrdený správou COMMIT.
- **ACK** táto správa je odosielaná zo zariadenia, ktoré prijíma dáta ako potvrdenie prijatia dát zariadením. V prípade prijatia správy po úspešnej kontrole, či je táto správa určená aktuálnemu zariadeniu, sú skontrolované metadáta tohoto packetu a zisťuje sa, či sú zhodné s nejakými metadátami zo zásobníka odoslaných správ v prípade, že áno je zrušený timeout, ktorý bol nastavený pre túto správu pri odosielaní packetu DATA a je zmenený stav odosieleného packetu zo stavu odosielenia dát do stavu odosielenia potvrdenia COMMIT a táto správa je odoslaná
- **COMMIT** táto správa je poslaná zo zariadenia, ktoré odosiela dáta ako potvrdenie úspešného dokončenia prenosu. Po prijatí tejto správy príjemcom a úspešnej validácii adresy cieľa a metadát je príslušný paket predaný z bufra vyššej vrstve protokolu na spracovanie keďže v tejto chvíli obidva zariadenia majú informáciu o tom, že bol prenos úspešný. Po spracovaní packetu na vyššej vrstve je správa uvoľnená zo zásobníka, a tým je prenos ukončený. Bez ohľadu na to, či je správa odpovedajúca metadátam uložená v zásobníku, je na túto správu vždy odosielené potvrdenie COMMIT\_ACK.
- **COMMIT** sprava odosielená príjemcom potvrdzuje ukončenie prenosu packetu. Po prijatí tejto spravy zariadením ktoremu je určená je z fronty odosielených packetou zmazaný paket odpovedajúci metadátam keďže je tento paket úspešne doručený a totodoručenie je potvrdené.

V prípade, že zariadenie nie je schopné v ktorejkoľvek fáze pokračovať v prijímaní packetu, či už z dôvodu plných zásobníkov alebo z dôvodu dlhej obsluhy nejakého packetu vo vyšších vrstvách protokolu, je posielená späť správa rovnakého typu ako keby bola požiadavka úspešne spracovaná, ale je v hlavičke nastavený príznak zaneprázdnenia. V takomto prípade sa na zariadení, ktoré paket odosiela, prenasťava časovač, keďže nejde o chybu siete ale len o zaneprázdnené zariadenie.

Z dôvodu snahy o zníženie energetickej náročnosti bola snaha udržať veľkosť packetu čo najmenšiu, a preto neobsahujú správy ACK, COMMIT a COMMIT\_ACK žiadne ďalšie informácie okrem hlavičky.

### Format packetu

formát packetu má pevnú dĺžku aj napriek tomu, že jeho štruktúra nie je pevná. Jednotlivé varianty packetu je možné vidieť na obrázku ??.

Popis časti packetu:

- **Control** kontrolný Byte, ktorý obsahuje informácie o štruktúre hlavičky. Význam jednotlivých bitov:
- **Network id** adresa siet, v ktorej sú obidve zariadenia

- packet type dvojbitová hodnota určujúca typ paketu, možné hodnoty sú DATA, ACK, COMMIT a COMMIT\_ACK
- receiver adress type určuje typ zariadenia, ktorému je paket určený a podľa tohto bitu je závislá dĺžka jeho adresy, v prípade, že sa jedná o end-device, je dĺžka adresy 4Byte, v prípade koordinátora je to 1Byte,
- sender adress type typ adresy odosielateľa, rovnaká úloha ako receiver adress type,
- busy flag príznak, či je význam tejto správy, že zariadenie je zaneprázdnené
- move flag príznak, či sa jedná o žiadosť o presunutie zariadenia v sieti

typ adresy odosielateľa a príjemcu nesmie byť nikdy nastavený na typ end-device v oboch súčasne, takýto paket je zahodený ako nevalidný

- **Destination address** adresa cieľa, jej veľkosť je určená príznakom typu v kontrolnom Byte
- **Source address** adresa zdroja, jej veľkosť je určená príznakom typu v kontrolnom Byte
- **Seq** poradové číslo správy, táto položka je povolená len v dobe, keď sú obidva zariadenia typu koordinátor, nakoľko v takomto prípade je potrebná dĺžka hlavičky len 7Bytov. Poradové číslo v takomto prípade umožňuje poslať medzi dvomi koordinátormi viac správ paralelne, nakoľko je možné ich identifikovať týmto číslom.

### Podpora prihlasovania a presúvania prvkov v sieti

V rámci linkovej vrstvy sú detekované dva stavy, pri ktorých nie je formát paketu zhodný s týmto popisom. Prvou takouto situáciou sú join správy (správy, ktoré neobsahujú identifikátor siete), v takomto prípade sa jedná o broadcastovú správu, ktorú prijímajú všetky zariadenia v sieti. Táto správa obsahuje end-point adresu zariadenia, ktoré žiada o zaradenie do siete, a dáta, ktoré môžu byť do tejto vrstvy vložené vyšším protokolom.

Druhý prípad nastáva, keď je v kontrolnom Byte paketu nastavený príznak žiadosti o presunutie. V takomto prípade obsahuje zvyšok paketu rovnakú štruktúru ako žiadosť o prihlásenie.

Zatiaľ čo prvý typ žiada ľubovlnú sieť o možnosť pripojiť sa k nej, druhý žiada o zmenu svojho umiestnenia v rámci už nastavenej siete. Rozdiel medzi týmito správami je v spôsobe ich spracovaní, keďže správa od neznámeho zariadenia môže byť v prípade, že je vypnutý párovací režim siete zahadzovaná. Žiadosť o zmenu umiestnenia musí byť prijatá sieťou vždy, keďže sa tento typ správy používa ako prostriedok znovupripojenia do siete po jeho presnutí.

#### 4.4.4 Sieťová vrstva

sieťová vrstva nahrádza implementáciu MiWi PRO 3.2. Pri implementácii tejto vrstvy na microcontroleroch bol zistený problém s fungovaním linkovej vrstvy, ktorý nie je možné v linkovej vrstve vyriešiť priamo, a to zasekávanie odosielania viacerých paketov naraz. Zariadenia, na ktorých je tento algoritmus implementovaný, majú limitovanú pamäť, a preto používajú v linkovej vrstve zásobníky s limitovanou veľkosťou. Problém nastával pri odosielaní správ, ktoré sa žz do zásobníka linkovej vrstvy nevošli a takéto poslanie spôsobovalo zaseknutie zariadenia v aktívnom čakaní. Tento problém bol spôsobený snahou odstániť

nevhodné riešenie metódy používanej v MiWi, kde musela byť v užívateľskom kóde používaná funkcia, ktorá prevádzala úlohy Miwi vrstvy. Viac o tomto probléme je v sekcii ???. Riešením tohto problému bolo prerobenie sieťovej vrstvy do podoby automatu, ktorý odosiela vždy len jednu správu. Aby takéto riešenie mohlo fungovať, museli byť do linkovej vrstvy pridané špeciálne callback funkcie na notifikáciu, že posielanie predchádzajúcej správy skočilo, to znamená, že je možné poslať ďalšiu správu a funkcie na detekciu, či je zariadenie zaneprázdnené, v dobe keď neboli tieto úpravy realizované, považovalo sa zariadenie za zaneprázdnené, keď nemalo voľné prijímacie zásobníky, to ostalo nezmenené, alebo ak nemalo minimálne jedno voľné miesto v zásobníku na odosielanie.

Sieťová vrstva má rôzne implementácie na základe typu zariadenia, a na ktorom sa používa. V koordinátoroch má táto implementácia podporovať aj prihlasovanie, presúvanie a smerovanie.

## Formát paketu

Formát paketu sa skladá z nasledujúcich častí”

- **type** určuje type správy, ktorá je prenášaná – 4b
- **dcid** určuje adresu koordinátora, ku ktorému je pripojený cieľ paketu – 6b
- **scid** určuje adresu koordinátora, ku ktorému je pripojený zdroj paketu – 6b
- **dedid** adresa cieľového end-point zariadenia – 32b
- **sedid** adresa zdrojového end-point zariadenia – 32b

## 4.5 Zariadenia

Vďaka kompletnému prepisu protokolu bolo možné počas vývoja začať používať aj zariadenia osadené čipmi od iných výrobcov, nakoľko už viac nie je potrebné dodržiavať originálnu licenciu firmy Microchip. V rámci práce bol protokol preportovaný na zariadenie s armovým procesorom, na ktorom beží linux. Táto úprava vyžadovala väčšie množstvo úprav, ako bolo možné zahrnúť do hardwarovej vrstvy vzhľadom na úplne odlišný princíp spracovania udalostí v operačnom systéme. Táto úprava bola potrebná len po fyzickú vrstvu a ostatné časti protokolu nebolo potrebné meniť. Ďalej bola v rámci práce vznesená požiadavka na vytvorenie spôsobu, ako používať tento protokol z programovacieho jazyka python, a preto bola vytvorená knižnica pre PAN koordinátora pomocou boost::python, ktorá práve toto umožňuje. Rozhranie protokolu ostalo viac menej nezmenené okrem odlišného spôsobu využívania externých callback funkcií, ktoré vzhľadom na charakteristiky programového jazyka python už nebolo možné realizovať.

## 4.6 Možné vylepšenia

Pri návrhu protokolu bolo brané do úvahy, aby bolo možné tento protokol jednoducho rozšíriť. Vďaka presunutiu PAN koordinátora na vykonávanie zariadenie, ktoré je možné pripojiť k internetu, je možné niektoré vlastnosti siete meniť jednoducho zmenou funkčnosti PAN koordinátora.

Medzi najväčšiu úpravu, ktorá by mohla byť realizovaná bez zmeny programu na jednotlivých zariadeniach v sieti, patrí možnosť rozšíriť počet aktívnych koordinátorov v sieti

na mnohonásobne vyšší počet. Podmienkou pre funkčnú realizáciu tohto riešenia je možnosť nájdania dvoch častí siete, ktoré by navzájom nemohli medzi sebou komunikovať. V prípade úspešného určenia takýchto častí je možné zariadeniam z oboch častí začať priradovať rovnaké adresy bez toho, aby začalo dochádzať ku konfliktom v procese smerovania. Najväčším problémom takéhoto riešenia by bolo zvýšenie nárokov na prenosovú rýchlosť rádia PAN koordinátora, obzvlášť v prípade, keď by sa tieto správy preposielali často z jednej časti stromu do druhej. Na obrázku je vidieť akým spôsobom by mohli byť adresy v takejto sieti pridelované.



## Kapitola 5

# Testovanie vlastnej implementácie smerovacieho protokolu

### 5.1 Prenosu paketov

**Ciel:** odtestovať, či sú pakety prenášané medzi dvomi zariadeniami prenášané bez strát na linkovej vrstve.

**Popis:** Rádiá sú označené ako odosielateľ a príjemca. Odosielateľ odosiela vopred stanovený počet paketov smerom k príjemcovi. Príjemca si ukladá celkový počet paketov zo siete. Počet prijatých paketov by sa mal zhodovať s počtom odoslaných.

Tento test opakujeme postupne so zvyšujúcou sa vzdialenosťou (zväčšovanie počtu prekážok) medzi zariadeniami až kým nedôjde k úplnému prerušeniu spojenia. Pre tento test by bolo vhodné, aby najväčší počet meraní prebehol v oblasti, kde je hranica dosahu, aby bolo správanie čo najnáhodnejšie.

Vzhľadom na implementáciu linkovej vrstvy, ktorá umožňuje detekciu chýb, je nutné zaznamenávať aj počet výskytu chýb, pričom podmienka úspešnosti testu sa zmení na počet prijatých paketov je zhodný počtu odoslaných mínus počet chýb detekovaných na oboch zariadeniach.

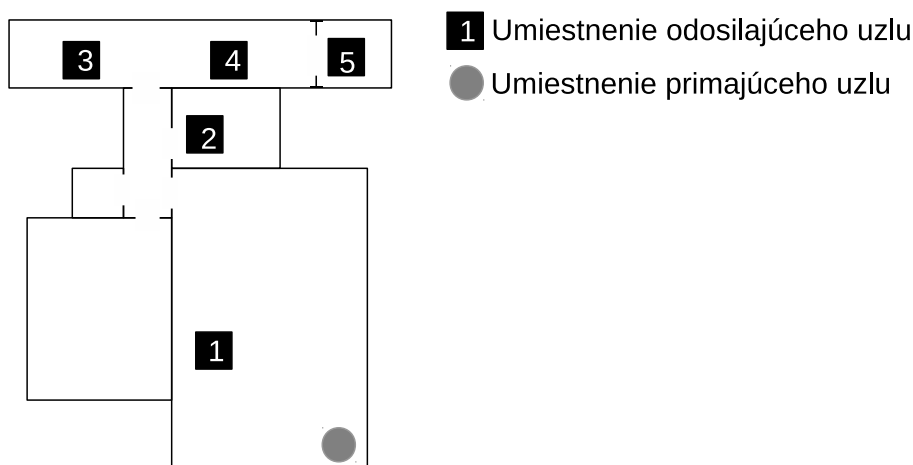
**Realizácia:** Test bol realizovaný v prostredí tehlového bytového domu, ktorého približný pôdorys je na obrázku 5.1, kde sú znázornené jednotlivé miesta merania, ktoré boli vopred vybrané vzhľadom na hranicu dosahu z väčšieho počtu meraní.

**Info:** Pre zvýšenie počtu chýb bol znížený počet povolených odosielaní na minimum a bola znížená aj doba čakania na prijatie paketu, vzhľadom na veľmi ťažké zbieranie dát z hranice dosahu, ktorá bola vplyvom týchto nastavení veľmi ostrá a vo väčšine dochádzalo buď k doručovaniu všetkých alebo žiadnych paketov.

**Merané výsledky:** Namerané hodnoty a výsledky testov podľa uvedeného pravidla sú v tabuľke na obrázku.

Umiestnenie	Počet odoslaných	Počet prijatých	Počet chýb pri odosielaní	Počet chýb pri prijímaní	Percent
1	300	300	0	0	100
2	300	274	7	19	100
3	300	50	235	47	107
4	300	35	255	34	108
5	300	0	300	0	100

Obrázek 5.1: Výsledok testu



Obrázek 5.2: Plošný plán miesta testovania

**Zhodnotenie testu:** Z jednotlivých meraní vyplynulo, že packety sa v rámci linkovej vrstvy doručujú spoľahlivo a vždy, keď sa paket stratil, bola oznamená chyba. V príadoch, keď bol počet odoslaných paketov nižší ako súčet počtu prijatých paketov a počtu chýb, dochádzalo k oznámeniu chyby na obidvoch zariadeniach, čo bolo kvôli podozreniu na možný vznik duplicity chyby a prijatého paketu preskúmané v logoch zo zariadenia.

## 5.2 Pripojovanie nových zariadení do siete

**Ciel:** otestovať schopnosť zariadenia pripojiť sa do bezdrátovej siete.

**Popis:** Test je rozdelený na štyri možné prípady, ktoré môžu v sieti nastať.

1. zariadenie sa pripája priamo k **PAN koordinátorovi** ktorý **má** povolený režim pridávania zariadení.
2. zariadenie sa pripája priamo k **PAN koordinátorovi** ktorý **nemá** povolený režim pridávania zariadení.
3. zariadenie sa pripája priamo k **koordinátorovi** ktorý **má** povolený režim pridávania zariadení.

4. zariadenie sa pripája priamo k **koordinátorovi** ktorý nemá povolený režim pridávania zariadení.

V prípade, že je táto požiadavka sieťou prijatá, je na PAN koordinátorovi oznámená žiadosť zariadenia o pridanie do siete. Po jej schválení je zariadenie pridané k uzlu, cez ktorý sa do siete prihlasovalo alebo v prípade existencie viacerých kandidátov je tento uzol vybraný PAN koordinátorom.

**Realizácia:** Po začiatku každého testu je vyprázdnená tabuľka zariadení koordinátora ak v nej nejake záznamy sú, následne sa nastaví testovací prípad podľa popisu a spustí sa prihlasovanie zariadenia.

**Info:** Bolo nutné nastavovať párovaí režim ručne z dôvodu, že pri nastavovaní na PAN koordinátore dochádza k odoslaniu príkazu na zmenu na všetky koordinátory v sieti.

**Zhodnotenie testu** Aj pri opakovaných testoch jednotlivých prípadov dochádza k pripojeniu respektíve oznámeniu nového zariadenia žiadajúceho o vstup do siete len v dobe, keď je aktívny párovací režim. Počas testovania bola objavená drobná chyba v prípade, že sa zariadenie snažilo pripojiť ku koordinátoru, občas došlo k zaseknutiu senzoru a bolo ho nutné reštartovať. Dôvod tejto chyby z dôvodu časovej tiesne nájdený a táto chyba bude opravená pri práci v projekte inteligentnej domácnosti.

### 5.3 Návrh testov

Okrem testov, ktoré boli realizované, bolo navrhnutých ešte niekoľko ďalších, ktoré preverujú iné chovanie nového protokolu a tieto testy môžu byť použité pri produkčnom testovaní, či nedošlo k znefunkčneniu protokolu.

- **propagácia routovacej tabuľky** tento test má odhaliť možné chyby pri šírení hromadnej správy sieťou. Jedna z vecí, na ktorú sa propagácia používa, je rozosielanie routovacej tabuľky do siete. Test spočíva v odoslaní takejto správy z PAN koordinátora a sleduje sa, či bola správa prijatá všetkými zariadeniami.
- **výpadok uzlu** tento test má za cieľ zistiť, či úu zariadenia schopné autoomne detekovať a reagovať na výpadok niektorého z nadradených uzlov a prípadne požiadať sieť o ich znovu začlenenenie. Test sa považuje za úspešný, ak sa komunikácia po opätovnom pripojení obnoví.
- **presun** implementácia protokolu by mala umožňovať na žiadosť PAN koordinátora meniť topológiu siete. Za týmto účelom sa používa nová správa. Test sa považuje za úspešný, ak po poslaní tohoto príkazu zariadeniu, začne toto zariadenie komunikovať pomocou novej cesty.

## Kapitola 6

### Záver

V rámci tejto práce som sa zoznámil s bezdrátovými senzorovými sieťami, s ich špecifickými vlastnosťami a požiadavkami. Naštudoval som problematiku smerovania v takýchto sieťach. Preskúmal som funkčnosť riešenia od firmy Microchip a po jeho vzore som na základe špecifických požiadaviek kladených na protokol v rámci projektu inteligentnej domácnosti vytvoril vlastnú implementáciu. Táto implementácia bola vďaka dobrému návrhu ľahko prenesiteľná na inú platformu čo umožnilo zjednodušenie hardwarovej realizácie zariadenia čo môže viesť k zníženiu výrobných cien. Nový protokol bol vďaka snahe o zachovanie spätnej kompatibility ľahko začlenený do stávajúcej implementácie kde sa už aktívne používa.

# Literatura

- [1] Al-karaki, J. N.; Kamal, A. E.: Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wireless Communications*, ročník 11, 2004: s. 6–28.
- [2] BENEŠ, V.: *Bezdrátová senzorová síť*. Diplomová práce, Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Brno, 2013.
- [3] Flowers, D.; Yang, Y.: MiWi™ Wireless Networking Protocol Stack.  
[http://ww1.microchip.com/downloads/en/AppNotes/MiWi%20Application%20Note\\_AN1066.pdf](http://ww1.microchip.com/downloads/en/AppNotes/MiWi%20Application%20Note_AN1066.pdf)  
2009, note: Application note 1066.
- [4] Microchip Technology Inc.: *MRF89XA Ultra Low-Power, Integrated ISM Band, Sub-GHz Transceiver*. 2010.
- [5] Microchip Technology Inc.: *MRF89XAM8A 868 MHz Ultra-Low Power Sub-GHz Transceiver Module*. 2010.
- [6] Shio Kumar Singh; M. P. Singh; Dharmendra K. Singh : Routing Protocols in Wireless Sensor Networks ?- A Survey. *International Journal of Computer Science & Engineering Survey (IJCES)*, ročník 1, č. 2, 2004.
- [7] WWW stránky: Microchip Corporate Overview.  
<http://www.microchip.com/pagehandler/en-us/aboutus/home.html>.
- [8] WWW stránky: Microchip Wireless Solutions.  
<http://www.microchip.com/pagehandler/en-us/products/wireless/home.html>.
- [9] Yang, Y.: MiWi™ Wireless Networking Protocol Stack.  
<http://ww1.microchip.com/downloads/en/AppNotes/01283a.pdf>, 2009, note: Application note 1283.
- [10] Yang, Y.: Microchip MiWi™ PRO Wireless Networking Protocol.  
<http://ww1.microchip.com/downloads/en/AppNotes/01371A.pdf>, 2011, note: Application note 1371.

# Příloha A

## Obsah CD

<code>/src</code>	zdrojové súbory protokolu
<code>/bp</code>	zdrojové súbory bakalárskej práce
<code>/doc</code>	dokumentácia
<code>/bachelors thesis.pdf</code>	elektronická verzia textu bakalárskej práce
<code>instal.txt</code>	popis inštalácie