

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

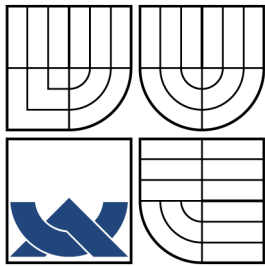
IMPLEMENTACE KOMUNIKAČNÍHO PROTOKOLU
WIRELESS M-BUS V SIMULAČNÍM PROSTŘEDÍ NS-3

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

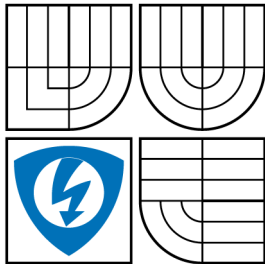
AUTOR PRÁCE
AUTHOR

Bc. KRYŠTOF ZEMAN

Brno 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

IMPLEMENTACE KOMUNIKAČNÍHO PROTOKOLU
WIRELESS M-BUS V SIMULAČNÍM PROSTŘEDÍ NS-3
IMPLEMENTATION OF WIRELESS M-BUS COMMUNICATION PROTOCOL TO
NS-3 SIMULATION ENVIROMENT

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. KRYŠTOF ZEMAN

VEDOUČÍ PRÁCE
SUPERVISOR

Ing. PAVEL MAŠEK

BRNO 2015



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Kryštof Zeman

ID: 134671

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Implementace komunikačního protokolu Wireless M-BUS v simulačním prostředí NS-3

POKYNY PRO VYPRACOVÁNÍ:

V rámci diplomové práce se bude nutné seznámit s problematikou M2M (Machine-To-Machine) komunikace v mobilních sítích. Pozornost bude věnována zejména popisu komunikačního protokolu Wireless M-BUS, který je v současné době využíván pro přenos dat od senzorů (MTCD) k sběrnému uzlu (MTCG). Teoretická část bude zaměřena na podrobný popis protokolu Wireless M-BUS (zejména popis fyzické a linkové vrstvy). Praktická část práce bude obsahovat návrh implementace protokolu Wireless M-BUS v simulačním prostředí NS-3 (Network Simulator 3).

DOPORUČENÁ LITERATURA:

[1] M2M communications: a systems approach. 1st ed. Editor David Boswarthick, Omar Elloumi, Olivier Hersent. Chichester: John Wiley, 2012, xxiii, 308 s. ISBN 978-1-119-99475-6.

[2] DONAHOO, Michael J a Kenneth L CALVERT. TCP/IP sockets in C: practical guide for programmers. 2nd ed. Boston: Morgan Kaufmann, c2009, xiii, 196 p. Morgan Kaufmann practical guides series. ISBN 01-237-4540-3.

Termín zadání: 9.2.2015

Termín odevzdání: 26.5.2015

Vedoucí práce: Ing. Pavel Mašek

Konzultanti diplomové práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Teoretická část práce se zaměřuje na historii a vývoj M2M komunikace, Wireless M-Bus protokolu a síťového simulátoru NS-3. Stručně popisuje charakteristické znaky M2M komunikace, protokolu Wireless M-bus a NS-3 simulátoru. Podrobněji se zaměřuje na fyzickou a linkovou vrstvu protokolu Wireless M-bus, čímž poskytuje čtenáři základy potřebné pro porozumění a orientaci v dané tématice. V praktické části je podrobně rozebrána struktura používaného Simple Wireless modulu, který je dále doplněn o metody pro simulaci přenosu dat reálných zařízení a je vytvořen simulační scénář, demonstrující možnosti upraveného modulu.

KLÍČOVÁ SLOVA

C++, M2M, NS-3, Síťový simulátor, Wireless M-Bus

ABSTRACT

The theoretical part of the thesis focuses on history and development of M2M communication, Wireless M-Bus protocol and NS-3 Network Simulator. It briefly describes characteristics of M2M communication, Wireless M-Bus and NS3 simulator. Physical and link layer of Wireless M-bus protocol are described in great detail to provide basics needed for understanding and orientation in the topic. Practical part of the thesis is focused on structure of used Simple Wireless module, which was supplemented by methods for simulating a data transfer between real devices and a simulation scenerio is built. The scenerio demonstrates basic functions of the upgraded module.

KEYWORDS

C++, M2M, NS-3, Network Simulator, Wireless M-Bus

ZEMAN, Kryštof *Implementace komunikačního protokolu Wireless M-BUS v simulačním prostředí NS-3*: diplomová práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2015. 68 s. Vedoucí práce byl Ing. Pavel Mašek

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Implementace komunikačního protokolu Wireless M-BUS v simulačním prostředí NS-3“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Pavlu Maškovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

PODĚKOVÁNÍ

Výzkum popsany v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....

(podpis autora)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	11
1 M2M Komunikace	12
1.1 Historie M2M	12
1.2 Charakteristické znaky M2M	13
1.3 Možnosti přenosu dat mezi M2M zařízeními	16
2 Wireless M-Bus	20
2.1 Historie Wireless M-Bus	20
2.2 Fyzická vrstva Wireless M-Bus	20
2.2.1 Režimy rádiového přenosu	21
2.2.2 Formát paketů Wireless M-Bus	25
2.2.3 Kódování a dekódování dat	26
2.2.4 Pořadí vysílání zakódovaných dat	27
2.3 Linková vrstva Wireless M-Bus	27
2.3.1 Formát rámců linkové vrstvy	28
2.4 Struktura dat u reálných zařízení	29
2.4.1 Bonega	29
2.4.2 WEPTECH	32
2.4.3 Pikkerton	35
2.4.4 ZPA	38
3 Network Simulator 3 (NS-3)	39
3.1 Historie	39
3.1.1 NS-1	39
3.1.2 NS-2	39
3.1.3 NS-3	39
3.2 Úvod do NS-3	40
3.2.1 Základní design simulačního modelu	41
3.2.2 Vytvoření a spuštění simulace	44
4 Rozšíření modelu SW	49
4.1 Přidání metod pro přenos dat určitého formátu	49
4.2 Simulační scénář	51
4.2.1 Popis jednotlivých částí simulace	52
5 Závěr	57

Literatura	58
Seznam symbolů, veličin a zkratk	61
Seznam příloh	63
A Tabulky s parametry jednotlivých režimů protokolu Wireless M-Bus	64
A.1 Detailní parametry režimu S	64
A.2 Detailní parametry režimu T	65
A.3 Detailní parametry režimu R	66
B Tabulka adresářové struktury síťového modulu NS-3	67
C Obsah přiložených DVD	68

SEZNAM OBRÁZKŮ

1.1	Základní schema M2M Komunikace[6]	12
1.2	Základní schéma mobilní a kapilární komunikace [6]	17
2.1	Základní topologie režimu S rozdělená dle jeho podrežimů [16]	22
2.2	Základní topologie režimu T rozdělená dle jeho podrežimů [16]	23
2.3	Základní topologie režimu R rozdělená dle jeho podrežimů [16]	24
2.4	Základní princip kódování Manchester [17]	26
2.5	Formát datové jednotky protokolu Wireless M-bus WEPTTECH	32
3.1	Softwarova organizace [22]	41
3.2	Základní diagram bloků NS-3 [23]	43
3.3	Diagram postupu nastavení simulace za použití funkcí pomocníků topologie [23]	45
3.4	Diagram znázorňující hierarchii dědičnosti v modelu šíření[23]	46
3.5	Základní adresářová struktura sw modelu	47
4.1	Diagram znázorňující topologii vytvořené simulace	52
B.1	Adresářová struktura síťového modulu	67

SEZNAM TABULEK

1.1	Kvalitativní porovnání bezdrátových technologií krátkého dosahu [6]	19
2.1	Základní rozdělení režimů [15]	21
2.2	Režim S - Hlavní parametry [14]	22
2.3	Režim T - Hlavní parametry [14]	23
2.4	Režim R - Hlavní parametry [14]	24
2.5	Formát datové jednotky [8]	25
2.6	Formát datové jednotky protokolu Wireless M-bus [8]	25
2.7	Zkrácený formát datové jednotky [8]	25
2.8	Převodní tabulka kódování 3 ze 6 [14]	27
2.9	Formát datové jednotky protokolu Wireless M-bus Bonega[8]	29
2.10	Zachycená datová jednotka výrobce Bonega[8]	31
2.11	Zachycená datová jednotka výrobce Weptech[8]	34
2.12	Formát datové jednotky protokolu Wireless M-bus Pikkerton[8]	35
2.13	Formát datové jednotky protokolu Wireless M-bus Pikkerton[8]	36
2.14	Formát datové jednotky protokolu Wireless M-bus ZPA[8]	38
A.1	Režim S - Detailní parametry vysílače [14]	64
A.2	Režim S - Detailní parametry přijímače [14]	64
A.3	Režim T - Detailní parametry vysílače [14]	65
A.4	Režim T - Detailní parametry přijímače [14]	65
A.5	Režim R - Detailní parametry vysílače [14]	66
A.6	Režim R - Detailní parametry přijímače [14]	66

ÚVOD

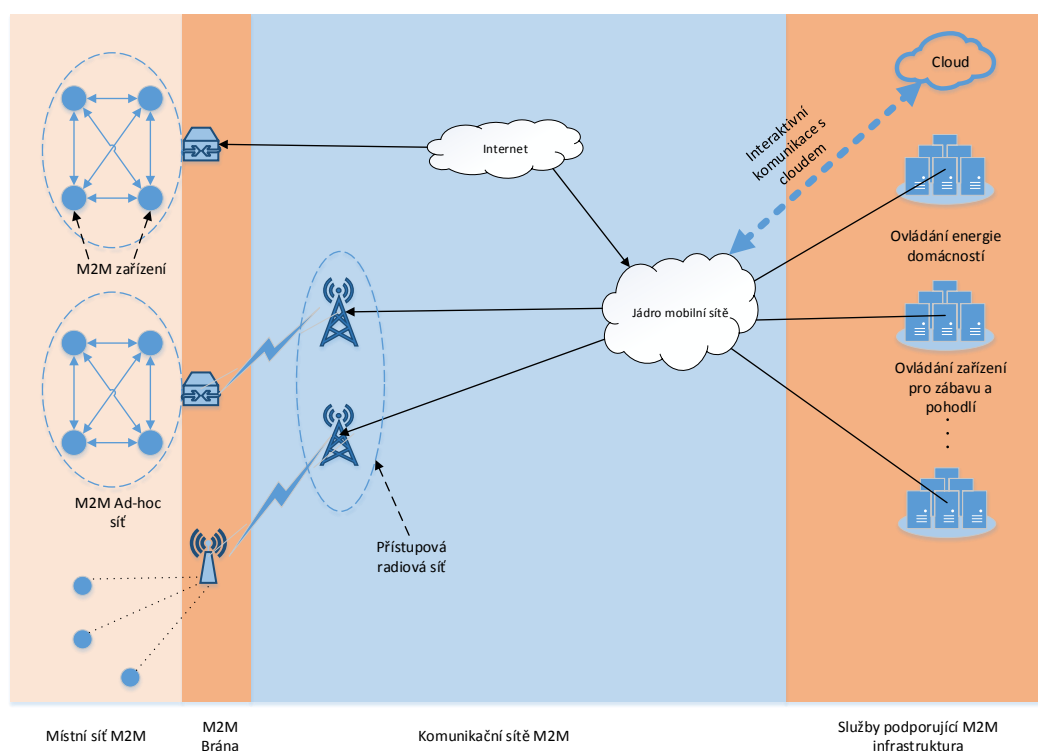
V dnešním světě existují dvě skupiny provozu zařízení, jsou to H2H (Human-to-Human) a M2M (Machine-to-Machine). Tato práce se zaměřuje na M2M komunikaci a její simulaci v síťovém simulátoru NS-3 (Network Simulator 3), protože v dnešní době, kdy vše spěje k automatizaci a senzory již dokáží pracovat bez nutnosti zásahu člověka, se právě M2M jeví jako typický představitel IoT (Internet of Things). Nově tak vzniká velké množství zařízení a odvětví, ve kterých se tento specifický způsob komunikace využívá a z důvodu různých nasazení je vhodné odsimulovat chování M2M zařízení v simulačních nástrojích před samotným nasazením.

V této práci se k tomuto účelu zvolil simulátor NS-3, zejména kvůli jeho přednostem oproti konkurenci (například OPNET Modeleru). NS-3 simulátor je zdarma jak pro soukromé tak i veřejné použití, jeho aktualizace vycházejí v krátkých intervalech a díky GPU licenci je možné do něj implementovat vlastní moduly. Pro M2M komunikaci byl zvolen protokol Wireless M-Bus, protože je jedním z nejrozšířenějších a navíc je založen na protokolu M-Bus, který je osvědčený a velmi rozšířený.

V teoretické části práce je rozebrána historie a hlavní vlastnosti protokolu Wireless M-bus a NS-3 simulátoru spolu s popisem M2M komunikace. V praktické části se práce zaměřuje na rozšíření Simple Wireless modulu, který již byl vytvořen Junseokem Kimem a dále rošířen Zenonem Kuderem. Tento model je nejdříve rozebrán, popsán a následně doplněn o metody potřebné k simulaci přenosu reálných dat protokolu Wireless M-Bus. Dále je tento model implementován v aplikaci zobrazující jeho možnosti na příkladu přenosu dat mezi koncentrátorem a uzlem.

1 M2M KOMUNIKACE

Machine to Machine komunikace (M2M komunikace) je termín, kterým označujeme skupinu přístrojů, služeb a řetězců hodnot ¹ potřebných k propojení elektronických přístrojů. Toto propojení je dnes obvykle bezdrátové a umožňuje primárně automatizovanou komunikaci mezi vzdálenými stroji (senzory) a jednou nebo více vrstvami aplikace centrálního řízení 1.1. M2M komunikace poskytuje možnosti měření a kontroly v reálném čase bez potřeby lidského řízení [1].



Obr. 1.1: Základní schema M2M Komunikace[6]

1.1 Historie M2M

M2M komunikace existuje už od počátků automatizace měření. První zmínky o tomto typu komunikace jsou v podobě telemetrie, průmyslových čidel a měřicích zařízení. Většina M2M komunikace však byla používána ve vědeckých oblastech a

¹Hodnotový řetězec slouží jako primární nástroj k identifikaci možností, jak vytvořit větší hodnotu pro zákazníka. Jde například o fyzický návrh výrobku, jeho prodej apod.

probíhala jednosměrně v podobě sběru dat (první byla NASA (National Aeronautics and Space Administration), dále následovaly předpovědi počasí apod.). Důvodem pomalejšího rozvoje M2M komunikace byl fakt, že zatímco rozvoj mobilních sítí probíhal velmi rychle, většina M2M zařízení komunikovala přes soukromé pozemní linky (POTS (Plain Old Telephone Service), DSL (Digital Subscriber Line)) a kabelové spoje [2].

První větší rozšíření odstartovalo roku 1995, kdy firma Siemens představila GSM modul M1, díky kterému M2M komunikace mohla začít probíhat bezdrátově [2]. Tento modul byl nejdříve nasazován v oblastech vzdáleného monitorování a sledování, PoS (Point of Sale) terminálech, telematické apod.². Průkopníky v tomto odvětví byly společnosti General Motors s OnStar technologií a Hughes Electronics Corporation [3]. Další rozšiřování se konalo s nástupem sítí GSM (Global System for Mobile Communications, původně Groupe Spécial Mobile), GPS (Global Positioning System), následně WiMAX (Worldwide Interoperability for Microwave Access) a HSPA (High Speed Packet Acces) [2].

K velkému rozšiřování M2M dochází v posledních letech díky nástupu nových telekomunikačních technologií, hlavně přenosu na krátké vzdálenosti jako jsou RFID (Radio Frequency Identification), Wi-Fi, Bluetooth, NFC (Near Field Communication) apod. V současné době se začínají využívat mobilní sítě 4. generace, označované jako LTE (Long Term Evolution). Tyto technologie se rozšiřují hlavně díky nástupu chytrých zařízení (telefony, tablety a nově i nositelnosti (wearables))³.

1.2 Charakteristické znaky M2M

V případě M2M komunikace se lze setkat s odlišnou charakteristikou datového provozu v porovnání s tradiční komunikací, často označovanou jako H2H (Human-to-Human). Nejpodstatnější rozdíly jsou rozebrány níže, zejména pak dopad na požadavky aplikací a sítí, které dosud nebyly brány v potaz [4].

- **Množství** - Je nejdiskutovanější změnou, kterou M2M přináší. Všeobecně se věří, že množství zařízení spojených pomocí M2M brzy výrazně přesáhne počet těch, které přímo ovládají lidé (jako jsou například mobilní telefony, PC, tablety atd.). Podle údajů společnosti Cisco bude do roku 2018 přes dvě miliardy M2M zařízení, což činí 43% ze složené roční míry růstu (CAGR - Compound Annual Growth Rate). Do roku 2018 bude 51% M2M zařízení

²Telematika je odvětví zabývající se technologiemi ve vozidlech - například senzory pro zvýšení bezpečnosti, navigace, multimediální zařízení, bezdrátová komunikace apod.

³Nositelnosti (wearables) jsou chytrá zařízení, které člověk může „obléci“. Například chytré hodinky, náramky apod.

připojeno pomocí 3G technologií (nárůst o 23% oproti roku 2013) a 14% pomocí 4G technologií (nárůst o 13,5% oproti roku 2013)[5]. To vede k větším nárokům na architekturu aplikací a propustnost sítí což vytváří problém ve škálovatelnosti systémů, původně určených pro obsluhu malého počtu zařízení s mnohem většími úrovněmi zatížení a množstvím druhů provozu. Jedním z prvních případů, kde se objevil tento problém jsou mobilní sítě, které nebyly pro tento způsob komunikace navrženy a v současné době probíhá intenzivní výzkum v oblasti optimalizace mobilních sítí pro M2M komunikaci při zachování definované kvality služeb (QoS - Quality of Service) pro H2H komunikaci [4].

- **Rozmanitost** - Již nyní existuje velké množství zdokumentovaných možností jak používat M2M zařízení. První implementace M2M aplikací vedla ke vzniku velkého množství zařízení s velmi odlišnými požadavky na rychlost přenosu, faktor tvarování M2M (Machine to Machine Form Factor), výpočetní výkon a schopnosti komunikace ⁴. To způsobilo vznik mnoha druhů zařízení, což představuje velký problém z hlediska kompatibility jednotlivých zařízení a pravděpodobně povede k problémům se sjednocením M2M zařízení [4].
- **Neviditelnost** - Je jedním z nejdůležitějších požadavků v M2M aplikacích. M2M zařízení musejí poskytovat služby jen s velmi malým, ideálně žádným lidským zásahem. Tento druh fungování však zamezuje lidem moci opravovat chyby (a také je vytvářet). Důsledkem toho se stává správa zařízení klíčovou částí služeb sítě více než kdy dříve a musí být bezchybná [4].
- **Kritičnost** - Některé zařízení, jako ty na poli eHealth zachraňují životy ⁵. Jiné jsou klíčovými prvky v inženýrských infrastrukturách, například jako detektory fáze a napětí, pojistky apod. v inteligentních sítích Smart Grid. Jejich provozní požadavky pak kladou obrovské nároky na odezvu (hlavně u eHealth zařízení) či spolehlivost, čímž mohou převyšovat možnosti dnešních sítí [4].
- **Zneužitelnost** - Mnoho M2M zařízení je navrhováno pro „lepší řízení“ systémů zabývajících se komfortem, zdravím apod. koncových uživatelů. Příkladem jsou výše zmíněné eHealth zařízení, chytré měřiče pro sledování a ovládnání elektrické spotřeby domácností apod. To bohužel vede k problémům v otázce soukromí. V podstatě to není nic nového z hlediska informačních technologií, nicméně je pravděpodobné, že soukromí bude překážkou při zavádění M2M systémů. Příkladem tohoto problému může být například situace, kdy

⁴Faktor tvarování určuje velikost, konfiguraci a fyzické uspořádání součástí - zejména SIM karty

⁵eHealth označuje elektronické zdravotnictví, které využívá informačních a komunikačních technologií pro diagnostiku, správu a podporu léčebně preventivní péče.

se distributor elektrické energie dostane do sporu s koncovými zákazníky kvůli implementaci chytrých měřičů do domácností, které potřebuje k lepší regulaci špiček v síti a které zároveň mohou omezovat soukromí zákazníků [4].

Krom výše zmíněných charakteristik a jejich dopadu na architekturu M2M systémů je důležité myslet též na další vlastnosti M2M zařízení, což může znamenat nové požadavky na způsob jejich komunikace v síti. Je tedy možné, že bude potřeba najít nové způsoby jak jednotlivé zařízení spojovat do skupin. Krom jiného, zařízení mohou být:

- **S omezenou funkčností** - Většina M2M zařízení má výpočetní výkon řádově nižší než moderní přenosné počítače nebo chytré telefony. Často též postrádají možnost aktualizací. Důvodem pro návrh takových zařízení je často cenová politika firem (firmy potřebují být konkurenceschopné). Dobře je tento trend patrný u chytrých měřičů. Dalším důvodem bývá logické rozhodnutí založené na povaze vyměňovaných informací - velká část senzorů odesílá naměřená data v přesných časových intervalech a tak není nutná implementace pokročilých funkcí [4].
- **S nízkou spotřebou** - I přes fakt, že velká část senzorů je napojena na elektrickou síť jsou zde i zařízení, které je z různých důvodů potřeba napájet jinak (zejména bateriemi). Například část z nich je (nebo bude) umístěna ve venkovních prostorech a nepůjde napojit na rozvodnou síť. Toto je problém například průmyslových senzorů, vodoměrů apod. Z toho vyplývá potřeba zařízení komunikovat co nejméně a na vhodných frekvencích, čímž se prodlouží jejich životnost (v některých případech i na 10let) [4].
- **Zabudovaná** - Velká část zařízení je a bude nasazena ve specifických operačních podmínkách, které způsobí, že bude těžké je změnit bez výraznějších zásahů do celého systému. Dobrým příkladem jsou systémy zabudované v domech či automobilech, které se často špatně mění (například když jsou připojené k motoru auta apod.) [4].
- **S dlouhou životností** - Mnoho M2M zařízení bude nasazeno i mimo oblast informačních komunikačních technologií, kde se počítá s naprosto odlišnou životností. Jedná se zejména o odvětví, kde M2M zařízení budou zabudované či jinak začleněné do systému, který se modernizuje mnohem pomaleji než tomu je v ICT (Information and Communication Technologies) [4].

1.3 Možnosti přenosu dat mezi M2M zařízeními

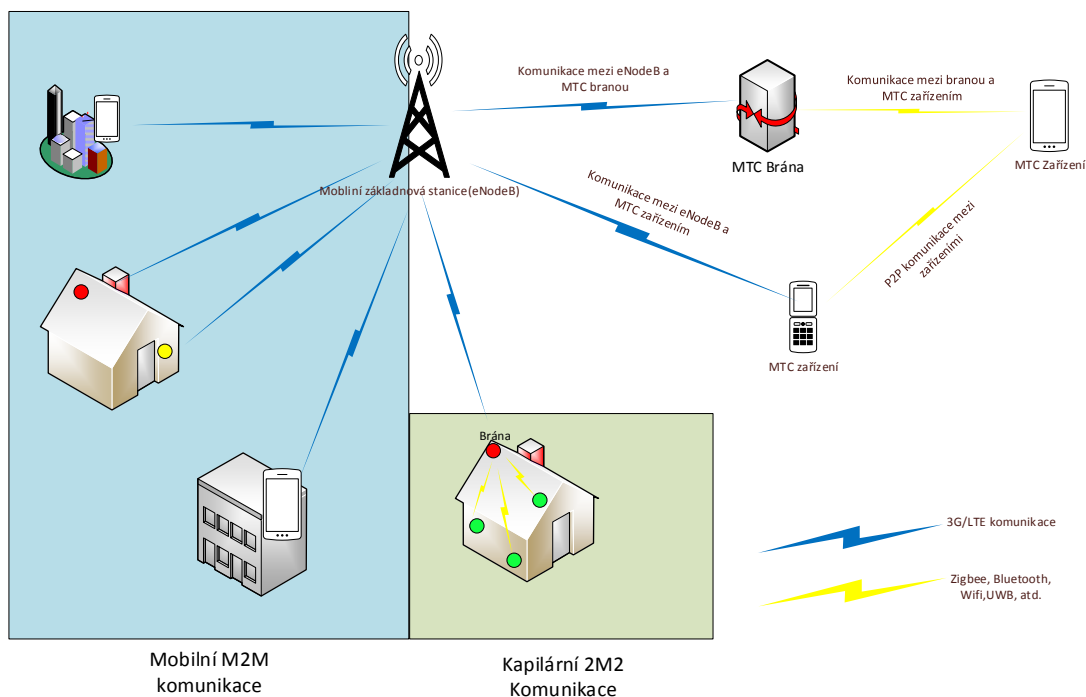
M2M má velmi specifické požadavky na transportní média. Tyto požadavky se odvíjí vždy od druhu aplikace senzoru. V dnešní době je velká část M2M komunikace prováděna bezdrátově a lze tedy zařízení rozdělit podle způsobu přístupu do sítě:

1. **Mobilní M2M komunikace** - V mobilní M2M komunikaci se jednotlivé terminály připojují přímo do 3G nebo 4G sítě a mohou tak využít prakticky všudypřítomného pokrytí, spolehlivého doručování a pokročilých zabezpečení. Bohužel tyto sítě nejsou stavěny pro obrovské množství vysílacích zařízení vytěžujících síť odesíláním dat (zejména je problém nadměrné vytížení přístupové části mobilní sítě). Díky tomuto faktu není vhodné používat tyto sítě přímo pro aplikace, kde je velké množství senzorů, ale spíše senzory spojit do kapilárních sítí a následně propojit s mobilní sítí [6].
2. **Kapilární M2M komunikace** - Tvoří chytrá zařízení organizovaná do podsítí, které jsou pomocí brány připojeny do mobilní sítě. Tím se v případě velkých počtů senzorů zajistí mnohem vyšší efektivita (z pohledu přenosu dat) a také snížení vytížení základnové stanice v LTE síti (označované jak eNodeB). Společnost Cisco uvádí, že do roku 2018 bude na zemi přes 10 miliard mobilních zařízení (včetně M2M) a více jak polovinu z nich budou tvořit „chytrá zařízení“, produkující více jak 88% celkového datového provozu [5].

Další možné rozdělení M2M zařízení je dle použité radiové transportní technologie. V následující části budou M2M zařízení rozdělena do dvou skupin podle energetické náročnosti a v rámci každé skupiny budou popsány nejčastěji používané technologie:

1. **Technologie s velmi-nízkou spotřebou a krátkým dosahem** - Do této skupiny se řadí hlavně zařízení používaná v kapilárních sítích, která jsou typická malou četností přenosů, minimální mobilitou a minimalizovanou spotřebou. Nejpoužívanějšími zástupci jsou ZigBee (pro velké kapilární sítě) a Bluetooth (pro malé sítě)[6].
 - **Bluetooth** - Byl navržen jako standard pro bezdrátovou komunikaci na krátké vzdálenosti a v současnosti je využíván zejména pro spojení řady osobních zařízení podporujících datové a hlasové služby. Jako poskytovatel WPAN (Wireless Personal Area Network) je Bluetooth schopen propojit 2 až 8 zařízení do pikosítě⁶, ve které jsou zařízení synchronizována pomocí společné časovací a skokové frekvence na stejném fyzickém kanálu. Řídící zařízení pikosítě je označeno jako master a všechny ostatní

⁶Pikosít je síť vytvořená pomocí Bluetooth zařízení, ve které může být maximálně 8 aktivních zařízení(důvodem je tříbitová MAC adresa)



Obr. 1.2: Základní schéma mobilní a kapilární komunikace [6]

zařízení jsou formovány do topologie hvězdy jako slave. Bluetooth pracuje na frekvenci 2.4 GHz užívané pro průmysl, vědu a medicínu. Využívá přeskokování frekvencí mezi 79 kanály s šířkou pásma 1 MHz rychlostí 1 600 skoku za vteřinu, čímž se snižuje rušení. I přesto, že má technologie nízkou energetickou náročnost, lze v případě Bluetooth pozorovat problémy s odezvou pokud je v používán pro velký počet zařízení a tudíž není vhodný pro implementaci do velkých sítí[6].

- **IEEE 802.15.X a ZigBee** - Existuje několik verzí standardu IEEE 802.15, ale pro M2M komunikaci se využívá verze IEEE 802.15.4, která určuje standardy fyzické (PHY) a linkové (MAC) vrstvy technologie ZigBee a podporuje velmi nízkou spotřebu, což je velmi důležité z hlediska konkurenceschopnosti celé technologie. Standard IEEE 802.15.4 podporuje dva operační módy - se zapnutým a vypnutým zasíláním řídicího rámce beacon. Dle některých studií tento standard může být rušen WLAN vysíláním[7]. Největší problém standardu IEEE 802.15.4 však zatím zůstává maximální přenosová rychlost, která je pouze 250 kb/s, což je dostatečné pouze pro zasílání dat ze senzorů, ale může být nedostatečné pro služby s větším datovým tokem. Na zvýšení přenosové rychlosti pracuje

výzkumná skupina pod záštitou IEEE 802.15.6. [6].

2. **Širokopásmové rádiové technologie** - Ve srovnání s technologiemi s velmi nízkou spotřebou a nízkou přenosovou rychlostí poskytují širokopásmové rádiové technologie mnohem vyšší propustnost ale také vyšší spotřebu. Do této skupiny patří hlavně HomeRF (Home Radio Frequency), UWB (ultra-wideband) a WiFi.
- **HomeRF** - Využívá FHSS (Frequency Hopping Spread Spectrum) v pásmu 2.4 GHz a může dosahovat maximálně 10 Mbit/s. Dostupné HomeRF zařízení podporují 1.6 Mbit/s, což je velmi málo oproti WiFi. Například IEEE 802.11b má teoretickou maximální propustnost 11 Mbit/s a standard IEEE 802.11n může dosahovat až 600 Mbit/s [6].
 - **UWB** - Podle FCC (Federal Communications Commission = Federální komise Spojených států amerických pro komunikaci) může být jakákoliv radiová technologie přesahující šířku pásma 500 MHz nebo 20% aritmetické střední frekvence považována za UWB. Dle regulí této komise je možno používat volné pásmo v rozsahu 3.1-10.6 GHz poskytující relativně nízkou spektrální hustotu emisí. Díky tomu se UWB hodí pro aplikace přenosu na krátké vzdálenosti, v budovách a v prostředích citlivých na emise radiových frekvencí (nemocnice apod.). Dnes dostupné UWB zařízení podporují až 480 Mb/s což využívají hlavně bezdrátová multimediální zařízení jako jsou monitory a digitální audio a video přehrávače [6].

Pro upřesnění a lepší představu využití jednotlivých technologií jsou výše popsané informace shrnuty v Tab.1.1:

Standard	Podsítě	Četnost	Energeticky omezen	Typické aplikace	Typ dat	Druh terminálu
Zigbee	Chytré podsítě	Nízká	Ano	Senzory, sledování	Senzory, sledování	Chytré měřiče
Bluetooth	Podsítě domácí kanceláře a zábavy	Nízká	Ano	Sdílení hudby	Hlas, hudba, malý přenos dat	Chytrý telefon, PDA
UWB	Podsítě domácí kanceláře a zábavy	Vysoká	Ne	Video, sdílení souborů	Video, velký přenos dat,	Video projektory, videokamery
IEEE 802.15.6	Podsítě v oblasti lidského těla	Nízká	Ano	Zdravotnictví	Biomedicínská data	Chytrý měřič tlaku
WiFi	Podsítě domácí kanceláře a zábavy	Vysoká	Ne	Internet, sdílení dat	VoIP, data, video	Chytrý fotoaparát, notebook
Femtocell	Mobilní stanice s malým výkonem	Vysoká	Ne	Mobilní telefony	VoIP, data, video	Domácí Node B
HomeRF	Podsítě domácí zábavy a komfortu	Vysoká	Ne	Hlasové služby	Hlas, data	Videotelefon

Tab. 1.1: Kvalitativní porovnání bezdrátových technologií krátkého dosahu [6]

2 WIRELESS M-BUS

V předchozí kapitole (1) byl proveden stručný popis M2M komunikace, jejích typů a charakteristiky. V dnešní době se na poli M2M komunikace jeví jako velmi perspektivní protokol Wireless M-Bus. Tento protokol vychází z jeho předchůdce M-Bus protokolu, avšak na úrovni fyzické vrstvy přenáší data bezdrátově vzduchem a ne pomocí krouceného páru. V této kapitole bude protokol Wireless M-Bus popsán podrobněji. Pozornost bude věnována zejména popisu jednotlivých vrstev [8] 2.2 2.3.

2.1 Historie Wireless M-Bus

Protokol Wireless M-Bus vychází z jeho drátové varianty. Díky potřebě vzniku specializované směrnice pro vzdálené měření a odečítání stavu měřičů byl založen standard M-Bus s označením EN13757 „Communication system for meters and remote reading of meters“, v překladu komunikační systém pro měřiče a vzdálené čtení měřičů [10]. V posledních letech však v mnoha aplikacích začal drátovou komunikaci vytlačovat bezdrátový radiový přenos dat díky možnosti jednoduchého rozšíření a snadnější instalace. Díky tomuto trendu se v roce 2005 standard M-Bus rozšířil o protokol EN13757-4:2005 Wireless meter readout, dnes označovaný jako Wireless M-Bus, který obsahuje specifikace fyzické a linkové vrstvy [12]. Na něj pak přímo navazuje aplikační vrstva EN13757-3:2004 Dedicated application layer, která je shodná s původním M-Bus protokolem [8].

2.2 Fyzická vrstva Wireless M-Bus

Fyzická vrstva (PHY) protokolu Wireless M-Bus je definována v ČSN EN 13757-4 [11]. Definuje jak mají být bity kódovány a vysílány, radio-frekvenční charakteristiky (čipovou rychlost, úvodní sekvenci a synchronizační slovo) a radio-frekvenční parametry (modulaci, střední frekvenci a frekvenční odchylku). Fyzická vrstva je realizována pomocí kombinace hardwaru a firmwaru [9].

2.2.1 Režimy rádiového přenosu

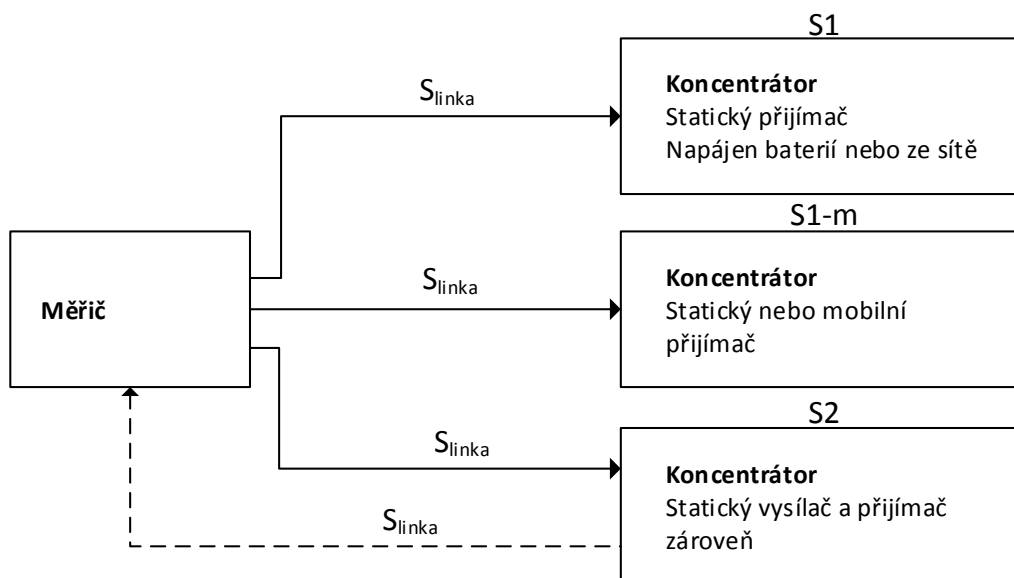
Fyzická vrstva dle normy ČSN EN 13757-4 definuje několik režimů označených jako S (S1, S1-m a S2), T (T1 a T2) a R2. Základní popis těchto režimů je uveden v Tab.2.1 [15].

Operační režim	Typ komunikace	Popis
S1	Jednosměrný	S označuje stacionární mód. Měřicí zařízení odesílá data několikrát denně. V tomto režimu lze šetřit energii koncentrátoru pomocí přepnutí do úsporného režimu, ze kterého se přepíná pouze po obdržení „wake up“ signálu od měřicích zařízení. Po „probuzení“ proběhne výměna dat a poté znovu-uspání koncentrátoru.
S1-m	Jednosměrný	Stejný jako režim S1, ale koncentrátor dat je mobilní a nelze přepnout do režimu snížené spotřeby.
S2	Obousměrný	Obousměrná verze operačního režimu S1.
T1	Jednosměrný	T označuje režim častého vysílání. V tomto režimu měřicí zařízení periodicky zasílá data do koncentrátoru dat. Interval zasílání dat lze nastavit dle potřeby (sekundy, minuty).
T2	Obousměrný	Obousměrná verze operačního režimu T1. Přidává koncentrátoru možnost vyžádání si dat mimo stanovený interval.
R2	Obousměrný	R označuje režim častého přijímání dat. Tento režim dovoluje příjem dat od více měřicích zařízení díky použití frekvenčního multiplexu.

Tab. 2.1: Základní rozdělení režimů [15]

Režim S

Stacionární režim, neboli režim S, byl definován pro jednosměrnou nebo obousměrnou komunikaci mezi stacionárními nebo mobilními zařízeními. Základní topologie režimu S je zobrazena na Obr.2.1.



Obr. 2.1: Základní topologie režimu S rozdělená dle jeho podrežimů [16]

Hlavní parametry režimu S jsou zobrazeny v Tab.2.2 [14].

Charakteristika	Minimum	Typicky	Maximum	Jednotka
Pásmo	868,0	868,3	868,6	MHz
Střída vysílače S2			1	%
Střída vysílače S1 a S1-m			0,02	%

Tab. 2.2: Režim S - Hlavní parametry [14]

Speciální pod-režim S1 je určený pro komunikaci mezi statickými měřiči a statickým koncentrátorem a je schopný pouze vysílat a může být optimalizován pro zařízení napájená z baterie pomocí dlouhé hlavičky paketu 2.3.1.

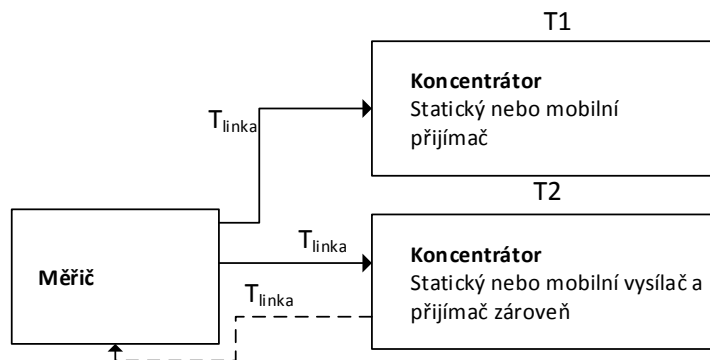
Pro komunikaci mezi statickými měřiči a mobilním koncentrátorem je vytvořen speciální pod-režim S1-m, používající krátkou hlavičku.

Pro měřiče s vždy zapnutým nebo synchronizovaným přijímačem bez nutnosti použití „budicí zprávy“ je určený pod-režim S2. Tento režim může obsahovat dlouhou hlavičku.

Detailní parametry definované pro vysílače v režimu S jsou zobrazeny v Tab.A.1 [14] a detailní parametry definované pro přijímače v režimu S jsou zobrazeny v Tab.A.2. Zkratky H_R (Highest performance = Nejvyšší výkon), M_R (Medium performance = Střední výkon), L_R (Lowest performance = Nejnižší výkon) jsou výkonové třídy přijímače stanovené ČSN EN 13757-4 [11].

Režim T

V režimu častého vysílání, neboli režimu T, měřič vysílá velmi krátký telegram (typicky 2-5ms) v určitém intervalu (typicky každých pár sekund), čímž umožňuje vyčítání dat při průjezdu okolo (Drive-by readout) ¹. Základní topologie režimu T je zobrazena na Obr.2.2.



Obr. 2.2: Základní topologie režimu T rozdělená dle jeho podrežimů [16]

Hlavní parametry režimu T jsou zobrazeny v Tab.2.3 [14] [8].

Charakteristika	Režim	Minimum	Typicky	Maximum	Jednotka
Pásmo: od měřiče k ostatním	T1,T2	868,7	868,95	869,2	MHz
Pásmo: od ostatních k měřiči	T2	868,0	868,3	868,6	MHz
Střída: od měřiče k ostatním	T1,T2			0,1	%
Střída: od ostatních k měřiči	T2			1	%

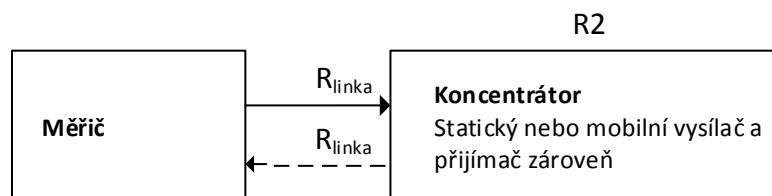
Tab. 2.3: Režim T - Hlavní parametry [14]

¹Vyčítáním při průjezdu okolo se označuje vyčítání dat pomocí zařízení zachytávajícího naměřené hodnoty při dostatečném přiblížení se k vysílajícím měřičům

Pod-režim T1 umožňuje pouze vysílat data. Tato data jsou redukována na minimum a obsahují pouze identifikační číslo měřiče a naměřené hodnoty. Jsou vysílána periodicky nebo náhodně. Tento režim lze použít pro přenos informací od měřičů tepla a vodoměrů. Pod-režim T2 umožňuje obousměrnou komunikaci. Zařízení periodicky zasílá krátký telegram obsahující minimálně jeho identifikační číslo, následně po velmi krátkou dobu čeká na příjem potvrzovacího paketu (ACK). Pokud je ACK přijat, naváže se obousměrný komunikační kanál. Detailní parametry definované pro vysílače v režimu T jsou zobrazeny v Tab.A.3 [14] [8] a detailní parametry definované pro přijímače v režimu T jsou zobrazeny v Tab.A.4 [14].

Režim R

V režimu častého přijímání dat, neboli R2, měřič v určitém intervalu (typicky každých pár vteřin) naslouchá pro příjem budící zprávy od mobilní radiostanice. Po obdržení budící zprávy se zařízení přepne do režimu pro komunikaci s radiostanicí. V tomto režimu je aktivován vícekanálový přijímací režim, který umožňuje vyčítání z více měřičů najednou. Každý měřič pak přenáší na určitém kanálu (lze volit v rozsahu 1-10). Režim R2 je vhodný spíše pro speciální případy, kde je nutné přenášet jen velmi malé množství dat a informací na velkou vzdálenost, protože komunikační rozhraní vykazují nejvyšší citlivost. Základní topologie režimu R je zobrazena na Obr.2.3.



Obr. 2.3: Základní topologie režimu R rozdělená dle jeho podrežimů [16]

Hlavní parametry režimu R jsou zobrazeny v Tab.2.4 [14] [8].

Charakteristika	Minimum	Typicky	Maximum	Jednotka
Pásmo	868,0	868,33	868,6	MHz
Rozestup kanálů		60		kHz
Střída vysílače			1	%

Tab. 2.4: Režim R - Hlavní parametry [14]

Detailní parametry definované pro vysílače v režimu R jsou zobrazeny v Tab.A.5. [14] a detailní parametry definované pro přijímače v režimu R jsou zobrazeny v Tab.A.6 [14] .

2.2.2 Formát paketů Wireless M-Bus

Moduly splňující standard Wireless M-Bus pracují následovně: Nadřízené aplikace (například měřící jednotka nebo koncentrátor) vyšlou data do RF (Radio Frequency) modemu ve formátu datové jednotky, který je zobrazen v Tab.2.5 [8]:

1 byte	1 byte	n bytes
Length	Cl	APP_Layer

Tab. 2.5: Formát datové jednotky [8]

Komunikační modul následně dle požadavků standardu Wireless M-Bus přidá následující pole[8]:

- Řídící pole C (Command Field).
- Označení výrobce ManID (Manufacturer ID).
- Unikátní komunikační adresy založené na parametrech uložených v paměti modulu (Address).
- Informace o síle přijímaného signálu RSSI (Received Signal Strength Indicator) - toto pole je nepovinné [8].

1 byte	1 byte	2 bytes	6 bytes	1 byte	n bytes	1 byte
Length	C	ManID	Address	Cl	APP_Layer	RSSI

Tab. 2.6: Formát datové jednotky protokolu Wireless M-bus [8]

Tento paket je dále šifrován (nejčastěji pomocí AES-128) a následně vyslán. V případě, že se realizuje pouze bezdrátové tunelování přenosu mezi dvěma Wireless M-Bus modemy, je povolen i zjednodušený režim bez zasílání adresy a jí přidružených informací o měřící jednotce. Rámec je pak výrazně kratší a jeho struktura je zobrazena v Tab.2.7 [8]:

1 byte	1 byte	n bytes	1 byte
Length	Cl	APP_Layer	RSSI

Tab. 2.7: Zkrácený formát datové jednotky [8]

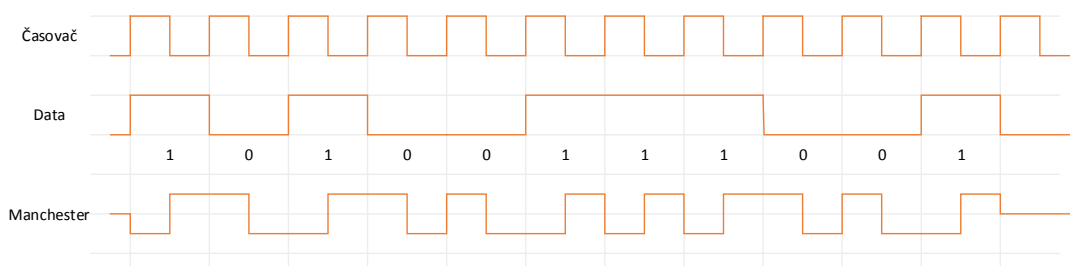
Obsah pole APP_LAYER je dán aplikační vrstvou definovanou ve standardu M-Bus, je tedy shodný s obsahem klasické drátové verze M-Bus. Komunikace mezi měřicí jednotkou a RF modemem či mezi koncentrátorem a RF modemem obvykle probíhá prostřednictvím sériového přenosu UART (Universal asynchronous receiver/transmitter) [8].

2.2.3 Kódování a dekódování dat

Wireless M-Bus protokol používá dva druhy kódování. Pro režimy S, R a pro režim T (od ostatních zařízení k měřičům) používá kódování Manchester. Režim T (od měřičů k ostatním zařízením) používá kódování 3 ze 6, které na rozdíl od kódování Manchester poskytuje vyšší efektivitu [9].

Kódování Manchester

Kódování Manchester dovoluje jednoduché kódování a dekódování při použití úzkého pásma. Každý bit je kódován jako „nula“, reprezentovaná pomocí čipové sekvence „10“ nebo jako „jedna“ reprezentovaná čipovou sekvencí „01“ [9]. Základní princip kódování je zobrazen na Obr.2.4.



Obr. 2.4: Základní princip kódování Manchester [17]

Kódování 3 ze 6

Při použití kódování 3 ze 6 je každé části (například úvodní sekvenci, začátku zprávy, atd.) přiřazen unikátní kód. Každý 4-bitový nibble (kousek) dat je kódován jako jedno 6-bitové slovo. Tyto slova byla vybraná ze 64 kombinací několika parametrů - musejí mít stejný počet jedniček a nul a musejí obsahovat minimálně dva přechody [9]. Převodní tabulka (Tab.2.8) kódování 3 ze 6 je zobrazena níže.

NRZ kód	Desítkově	6-bitový kód	Desítkově	Počet přechodů
0000	0	010110	22	4
0001	1	001101	13	3
0010	2	001110	14	2
0011	3	001011	11	3
0100	4	011100	28	2
0101	5	011001	25	3
0110	6	011010	26	4
0111	7	010011	19	3
1000	8	101100	44	3
1001	9	100101	37	4
1010	10	100110	38	3
1011	11	100011	35	2
1100	12	110100	52	3
1101	13	110001	49	2
1110	14	110010	50	3
1111	15	101001	41	4

Tab. 2.8: Převodní tabulka kódování 3 ze 6 [14]

2.2.4 Pořadí vysílání zakódovaných dat

V režimech S, T a R je každý byte vyslán s nejvíce důležitým bitem (MSB - Most Significant Bit) na prvním místě. Bytová sekvence CRC (Cyclic Redundancy Check) je: vysoký bit první (High Bite First) [12].

2.3 Linková vrstva Wireless M-Bus

Linková vrstva poskytuje rozhraní mezi fyzickou vrstvou (PHY) a aplikační vrstvou (AL). Její hlavní funkce jsou [9]:

- Poskytování služeb převádějících data mezi PHY a AL.
- Generování CRC pro odchozí zprávy.
- Detekování CRC chyb v příchozích zprávách.
- Poskytování adresování fyzické vrstvy.
- Kontrola ACK u obousměrných přenosů.
- Vytváření rámců.
- Kontrola chyb rámců v příchozích zprávách.

2.3.1 Formát rámců linkové vrstvy

Formát rámců protokolu Wireless M-Bus je odvozen od formátu FT3 (Frame Type 3) z normy IEC60870-5-2 [13]. Rámec se skládá z jednoho nebo více bloků dat. Každý blok dat obsahuje 16-bitové CRC pole. První blok má pevnou délku 12 bytů a obsahuje L-pole (L-field), C-pole (C-field), M-pole (M-field) a A-pole (A-field) [9].

L-pole

L-pole určuje velikost dat přenášených v rámci. Do této velikosti není zahrnuta velikost L-pole samotného ani CRC bytů. Protože maximální velikost kódovaných bytů je 255, maximální podporovaná velikost pro M-pole je 110 bytů při kódování Manchester a 148 při kódování 3 ze 6. Linková vrstva počítá délku L-pole při vysílání i přijímání [9].

C-pole

C-pole je kontrolní pole rámců. Identifikuje typ rámce (SEND, CONFIRM, REQUEST, RESPONSE) a používá se pro zasílání základních příkazů (Link Service Primitives). V případě SEND nebo REQUEST rámce C-pole určí, zda je očekáván CONFIRM nebo RESPONSE rámec. V základním TX režimu lze použít jakoukoliv hodnotu C-pole. V režimu se zasíláním základních příkazů je C-pole vyplněno automaticky dle normy ČSN EN 13757-4 [11].

M-pole

M-pole obsahuje třípísmenný kód výrobce, o který mohou výrobci zažádat na webové adrese <http://www.dlms.com/flag/INDEX.HTM>. Každé z písmen je kódováno jako 5 bitů, které se získávají z ASCII kódu písmena po odečtení 0x40 („A“). Nejvýznamnější bit (MSB) je nula [9].

A-pole

Adresní A-pole obsahuje jedinečnou 6-bytovou adresu pro každé zařízení. Tyto adresy přidělují jednotliví výrobci, kteří jsou zodpovědní za zachování jedinečnosti adres. Při přenosu rámců SEND a REQUEST je v A-poli uvedena adresa zařízení, které rámec vyslalo. Při přenosu rámců CONFIRM a RESPONSE je v A-poli uvedena adresa zařízení, kterému jsou pakety určeny [9].

Cl-pole

Cl-pole je hlavičkou aplikace a určuje typ přenášených dat aplikační vrstvy. Norma ČSN EN-13757-4:2005 povoluje pouze omezený počet hodnot, ale základní příkazy umožňují použít jakékoliv hodnoty [9].

CRC

CRC pole obsahuje kontrolní součet, dle kterého se následně kontroluje, zda při přenosu nedošlo k chybě. Polynom užívaný pro Wireless M-bus CRC je : $x^{16} + x^{13} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^2 + 1$ [9].

2.4 Struktura dat u reálných zařízení

V této části bude podrobněji popsána struktura dat u reálných zařízení od několika výrobců. Pro tuto práci byly vybráni výrobci Bonega, WEPTTECH, Pikkerton a ZPA. Tito čtyři výrobci byli zvoleni z důvodu dostupnosti dokumentace a popisu jejich zařízení.

2.4.1 Bonega

Výrobce Bonega zvolil pro strukturu přenášených dat formát zobrazený v Tab.2.9.

1 byte	1 byte	2 bytes	6 bytes	1 byte	20 bytes
Length	C	ManID	Address	Cl	Data

Tab. 2.9: Formát datové jednotky protokolu Wireless M-bus Bonega[8]

Z tabulky je patrné, že výrobce zachoval strukturu požadovanou normou. V V části označené data však už uplatňuje svojí strukturu, která je popsána dále.

Délka celé datové jednotky je 30 bytů, z čehož 20 B zabírají právě data. Tato část se dále dělí na:

- 1 byte pro přístupové číslo (Access number), které se při každém vysílání nových dat zvětší o jedna.
- 1 byte pro stavový byte (Status byte), určující stav, ve kterém se zařízení nachází. Konkrétně jde o stavy : vybitá baterie, fatální chyba, únik vody (více jak 4l/h po dobu více jak 48h), únik vody (více jak 900l, tedy 3x 300l/h), porušení kabelů.
- 2 byty pro podpisové pole (Signature field), určující mód přenosu a typ kódování.

- 16 bytů pro šifrovaná data (Encrypted data). V této části se přenášejí údaje o aktuální hodnotě na měřiči společně s datem a časem měření.

V Tab.2.10 je zobrazena struktura zachycené datové jednotky přenášené měřičem od výrobce Bonega.

Length	C	ManID	Address	CI	Data
1E	44	EE 09	21 01 00 00 01 06	7A	4F 00 10 05 1A B9 4C 4F DA 69 43 09 E3 47 E8 6F A4 37 79 0C

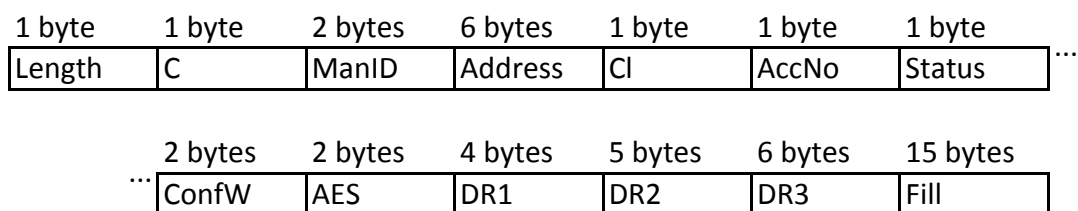
Tab. 2.10: Zachycená datová jednotka výrobce Bonega[8]

Byty jsou rozdělené do polí dle formátu udávaného výrobcem (viz. 2.9) a jejich význam je:

- Length - $1E_h$ odpovídá délce 30 bytů.
- C - určuje typ paketu. Zde 44_h značí paket SND-NR, neboli paket se odešle a nečeká odpověď.
- ManID - $09EE_h$ odpovídá BON, což je zkratka výrobce Bonega.
- Address - 210100000106_h První 4 byty udávají identifikační číslo, čtou se od MSB. Identifikační číslo je tedy 00 00 01 21. Další byte udává verzi, zde 01. Poslední byte udává typ měřiče - 06 je měřič teplé vody.
- CI - 4 bytová hlavička, zde $7A_h$ neboli 01111010_b.
- Data - $4F_h$ udává číslo přístupu, zde 79. 00_h značí stav bez chyby. 1005_h určuje typ šifrování a mód přenosu - zde 01_h = T1, 05_h = 128 AES, mód 5.
- 1AB94C4FDA694309E347E86FA437790C_h jsou zašifrovaná data. K jejich rozšifrování je potřeba IV složený z M Pole, A Pole a 8 bytů přístupového čísla. IV tedy odpovídá EE092101000001064F4F4F4F4F4F_h a AES klíč 2B7E151628AED2A6AB – F7158809CF4F3C_h. Pomocí AES dešifrování pak získáme: 2f2f04131a220000046d0328c4162f2f_h. Byty 1a220000_h odpovídají reálné hodnotě měřiče, čtou se zase od LSB, tedy 221a_h, což odpovídá hodnotě 8730. Byte 04_h označuje DIF, v našem případě datum a čas v F formátu. Tyto hodnoty získáme z následujících 4 bytů: 0328c416_h. První dva určují čas a další dva datum. Po převedení do binární soustavy získáme tedy 0000001100101000_b, ze kterého určíme čas 08:03 a 11000100000010110_b, ze kterého získáme datum 06.04.2014.

2.4.2 WEPTECH

Formát datové struktury používaný výrobcem WEPTECH je zobrazen na Obr.2.5



Obr. 2.5: Formát datové jednotky protokolu Wireless M-bus WEPTECH

I tento výrobce dodržel normu, a první část struktury je tedy stejná. Na rozdíl od firmy Bonega však zbylé byty neřadí hromadně do dat, ale nechává je rozdělené. Význam jednotlivých bytů je:

- Status byte je nastaven na 0. V případě rozpoznání sabotáže se nastaví na hodnotu 1 a pokud není sabotáž vyřešena do následujících dvou vysílání, je nastaven bit „permanentní“ chyby. Tento bit je nastaven také v případě slabého napájení či vybité baterie.
- Konfigurační slovo je složeno ze dvou bytů. První byte obsahuje NNNNCCHHb (NNNN - počet kódovaných bloků, CC - obsah telegramu, HH - počítací skoků). Pokud je šifrování vypnuto pak je byte 00h, jinak odpovídá počtu šifrovaných bloků. Druhý byte obsahuje BAS0MMMMb (B - obousměrný, A - dostupnost, S0 - synchronizace, MMMM - šifrování). Pokud je šifrování vypnuto je byte 00h, jinak je nastaven mód šifrování 5 (05h). Pokud se počítá se synchronním vysíláním tak se nastavuje také bit S.
- AES ověření se provádí pomocí dvou bytů nastavených na 2Fh.
- DR1 je složen ze čtyř bytů. První byte obsahuje DIF (Data Information Field) v podobě 4 číselného binárně kódovaného dekadického čísla. Druhý obsahuje VIF (Value Information Field), udávající informace o ukládané hodnotě (zde konkrétně 66h, odpovídající teplotě ve formátu $t * 10^{(-1)}$ ve stupních). Poslední dva byty DR1 obsahují uloženou konkrétní hodnotu měřené veličiny.
- DR2 je složen z pěti bytů. První je stejný jako u DR1. Druhý byte obsahuje VIF, nesoucí informaci o tom, že se jedná a první rozšířenou tabulku. Třetí byte obsahuje informace o měřené veličině (zde konkrétně 1Ah, odpovídající relativní vlhkosti $*10^{(-1)}$ v procentech). Poslední dva byty DR2 pole obsahují přesnou hodnotu měřené veličiny.
- DR3 je složen ze 6 bytů. První obsahuje DIF uložený jako 16-bitový integer. Druhý byte obsahuje VIF, nesoucí informaci o tom, že se jedná o druhou

rozšířenou tabulku. Třetí byte je rozšířený VIF nesoucí chybové vlajky (Error Flags). Čtvrtý byte je druhým rozšířením VIF bytu a nese údaje o normě (Standard Conform). Poslední dva byty DR3 jsou nastaveny na 0000h. Pokud dojde k sabotáži, nastaví se tzv. „falšovací“ bit. Pokud dojde k vybití baterie, nastaví se bit „prázdné baterie“.

- Zbývající byty jsou výplňové - nenesou žádnou informaci.

V Tab.2.11 je zobrazena struktura zachycené datové jednotky přenášené měřičem od výrobce Weptech.

L	C	MID	Address	CI	AN	S	CW	AES	DR1	DR2	DR3	Fill
2E	44	B05C	11 00 00 00 02 1B	7A	92	08	00 00	2F 2F	0A 66 67 02	0A FB 1A 56 04	02 FD 97 1D 01 00	2F * 15

Tab. 2.11: Zachycená datová jednotka výrobce Weptech[8]

Byty jsou rozdělené do polí dle formátu udávaného výrobcem (viz. 2.5) a jejich význam je:

- Length $2E_h$ odpovídá délce 46 bytů.
- C určuje typ paketu. Zde 44_h značí paket SND-NR, neboli paket se odešle a nečeká odpověď.
- ManID $B05C_h$ odpovídá WEP, což je zkratka výrobce Weptech.
- Address $1100000021B_h$ První 4 byty udávají identifikační číslo, čtou se od MSB. Identifikační číslo je tedy 00 00 00 11. Další byte udává verzi, zde 02. Poslední byte udává typ měřiče - 1B je měřič umístěný v místnosti.
- CI - 4 bytová hlavička, zde $7A_h$, neboli 01111010_b, značí odpověď od senzoru.
- AccNo 92_h udává o kolikáté vysílání se jedná - zde 146.
- Status udává v jakém stavu se zařízení nachází (porucha atd...) - 00_h odpovídá normálnímu stavu.
- Conf Word udává zda je aktivováno šifrování. 0000_h značí vypnuté šifrování.
- AES ověření $2F2F_h$.
- DR1 je pole pro ukládání dat prvního měření. $0A_h$ značí, že je šifrováno pomocí 4 číselního BCD (Binary Coded Decimal). 66_h značí, že je zaznamenávána teplota. 6702_h odpovídá teplotě 26, 7° C.
- DR2 je pole pro ukládání dat druhého měření. $0A_h$ značí šifrování pomocí 4 číselného BCD. FB_h je VIF značící první rozšiřující tabulku. $1A_h$ značí, že uložená hodnota odpovídá relativní vlhkosti. 5604_h odpovídá relativní vlhkosti 45,65%.
- DR3 je pole pro ukládání dat třetího měření. 02_h značí 16bitový integer. FD_h je VIF značící druhou rozšiřující tabulku. 97_h je pole obsahující vlajky chyb (Error Flags). $1D_h$ značí normu. 0100_h ukazuje, že zařízení má vybitou baterii.
- $15 * 2F_h$ jsou výplňové byty bez informace.

2.4.3 Pikkerton

Formát datové struktury používaný výrobcem Pikkerton je zobrazen v Tab.2.12

1 byte	1 byte	2 bytes	6 bytes	1 byte	1 byte	1 byte	2 bytes	x bytes
Length	C	ManID	Address	Cl	Counter	Status	Config	Data

Tab. 2.12: Formát datové jednotky protokolu Wireless M-bus Pikkerton[8]

Tento výrobce zvolil po normou dané části struktury použít volitelný blok označený data. Tento blok se liší dle typu měřiče a přenášených veličin. Celá struktura je tedy univerzální a mění se pouze poslední část, přenášející data měřičů.

V Tab.2.13 je zobrazena struktura zachycené datové jednotky přenášené měřičem od výrobce Pikkerton.

L	C	MID	Address	Cl	Ctr	Status	Conf	Data
28	44	2B 41	44 52 12 70 02 02	7A	35	00	00 00	04 FB 2C CD C3 00 00 01 FD 49 E0 02 FD 59 1A 01 02 2B 20 00 04 03 5C 00 00 00

Tab. 2.13: Formát datové jednotky protokolu Wireless M-bus Pikkerton[8]

Byty jsou rozdělené do polí dle formátu udávaného výrobcem (viz 2.12) a jejich význam je:

- Length 28_h odpovídá délce 40 bytů.
- C určuje typ paketu. Zde 44_h značí paket SND-NR, neboli paket se odešle a nečeká odpověď.
- ManID $2B41_h$ odpovídá PIK, což je zkratka výrobce Pikkerton.
- Address 445212700202_h První 4 byty udávají identifikační číslo, čtou se od MSB. Identifikační číslo je tedy 70 12 52 44. Další byte udává verzi, zde 02. Poslední byte udává typ měřiče - 02 elektroměr.
- Cl - 4 bytová hlavička, zde $7A_h$, neboli 01111010_b, značící odpověď od senzoru.
- AccNo 35_h udává o kolikáté vysílání se jedná - zde 53.
- Status udává v jakém stavu se zařízení nachází (porucha atd...) - 00_h odpovídá normálnímu stavu.
- Conf Word udává, zda je aktivováno šifrování či zařízení komunikuje obousměrně. 0000_h značí vypnuté šifrování a jednosměrnou komunikaci.
- Data se dají rozdělit na 5 částí. Každá z nich obsahuje naměřená data a je popsána níže:
 1. První část se dělí do následujících částí. 04_h udává, že se jedná o 32bitový integer. FB_h značí 1. rozšíření VIF kódu. $2C_h$ značí, že uložená hodnota je frekvence. $CDC30000_h$ odpovídá frekvenci 50,125 Hz.
 2. Druhá část obsahuje informace o napětí. 01_h značí 8 bitový integer. FD_h značí, že se jedná 2. rozšíření VIF kódu. 49_h značí, že uložená hodnota je napětí. E0 odpovídá napětí 224 V.

3. Třetí část obsahuje data o proudu. 02_h udává, že se jedná o 16 bitový integer. FD_h značí, že se jedná 2. rozšíření VIF kódu. 59_h značí, že uložená hodnota je proud. $1A01_h$ odpovídá hodnotě 282 mA.
4. Čtvrtá část obsahuje informace o výkonu. 02_h udává, že se jedná o 16bitový integer. $2B_h$ značí, že uložená hodnota je výkon. 2000_h odpovídá hodnotě výkonu 32 W.
5. V poslední části jsou uložené údaje o práci. 04_h udává, že se jedná o 32bitový integer. 03_h značí, že uložená hodnota je práce. $5c000000_h$ odpovídá hodnotě 92 Wh.

2.4.4 ZPA

Formát datové struktury používaný výrobcem ZPA je zobrazen v Tab.2.14

1 byte	1 byte	2 bytes	6 bytes	1 byte	1 byte	20 bytes
Length	C	ManID	Address	CRC	CI	Data

Tab. 2.14: Formát datové jednotky protokolu Wireless M-bus ZPA[8]

ZPA stejně jako výše zmínění výrobci používá normou danou strukturu, na kterou navazuje přesně daným formátem dat. Konkrétně odesílá údaje o dvou veličinách.

3 NETWORK SIMULATOR 3 (NS-3)

Simulátor NS-3 byl zvolen díky jeho specifickým vlastnostem, které nabízí. Jeho velkou výhodou je široká podpora nových standardů a technologií, široká skupina přispěvatelů a členů na fórech (viz : <https://groups.google.com/forum/#!forum/ns-3-users>). Navíc například oproti OPNET Modeleru (který je spíše pro průmysl) a NetSlimu je NS-3 zdarma a jeho aktualizace vycházejí častěji [18].

3.1 Historie

3.1.1 NS-1

První verzi Network Simulatoru byla NS-1, jejíž autory jsou Steve McCanne, Sally Floyd, Kevin Fall a další lidé ze společnosti GEEKLIME. Jádro aplikace je napsáno v jazyce C++, kde je pro vytváření simulací používán Tcl (Tool Command Language). Na dalším rozvoji simulátoru se podílely i firmy jako Sun Microsystems či universita Berkeley (projektem Daedelus)[19].

3.1.2 NS-2

V letech 1996-1997 byla Stevem McCannem představena verze NS-2, která byla založena na přepracované verzi NS-1. Tcl bylo nahrazeno OTcl (Object Tcl), což je objektově orientovaná verze Tcl. Jádro NS-2 je stále napsáno v C++, ale objekty užívané v simulacích jsou spojeny s objekty v OTcl a proměnné tak mohou být odkazovány mezi oběma jazyky. Celé simulační skripty jsou pak psány výhradně v OTcl. Aktuálně se NS-2 skládá z více než 300 000 řádků zdrojového kódu a minimálně stejného množství kódu od přispěvatelů (existuje velké množství verzí a modulů do NS-2), který není integrován do hlavní distribuce programu. NS-2 běží na GNU/Linuxu, FreeBSD, Solarisu, Mac OS X a verzích Windows podporujících Cygwin. Celá NS-2 je licencována pod všeobecnou veřejnou licenci GNU (GNU General Public License¹) verze 2 [19].

3.1.3 NS-3

Tým pod vedením Toma Hendersona, George Rileyho, Sally Floydové a Sumita Roye, sponzorovaný americkou NSF (National Science Foundation) pracuje ve spolupráci s projektem Planete pod záštitou INRIA (Institut national de recherche en informatique et en automatique) na náhradě za NS-2, nazvané NS-3. Celý tento projekt je distribuován jako open source [19].

¹GNU umožňuje veřejnosti bezplatně používat, studovat a sdílet daný software.

Při vývoji NS-3 se autoři rozhodli naprosto opustit zpětnou kompatibilitu s NS-2 a nový simulátor kompletně přepsat, za použití C++. Vývoj NS-3 začal v červenci 2006. Aplikační rámec (framework) pro generování vazeb Pythonu a kompilátor Waf byl vytvořen Gustavem Carneirem [19].

První verze NS-3.1 byla vydána v červnu 2008 a od té doby je nová verze vydávána ve čtvrtletních aktualizacích. Aktuální verzí NS je verze NS-3.21 [19].

3.2 Úvod do NS-3

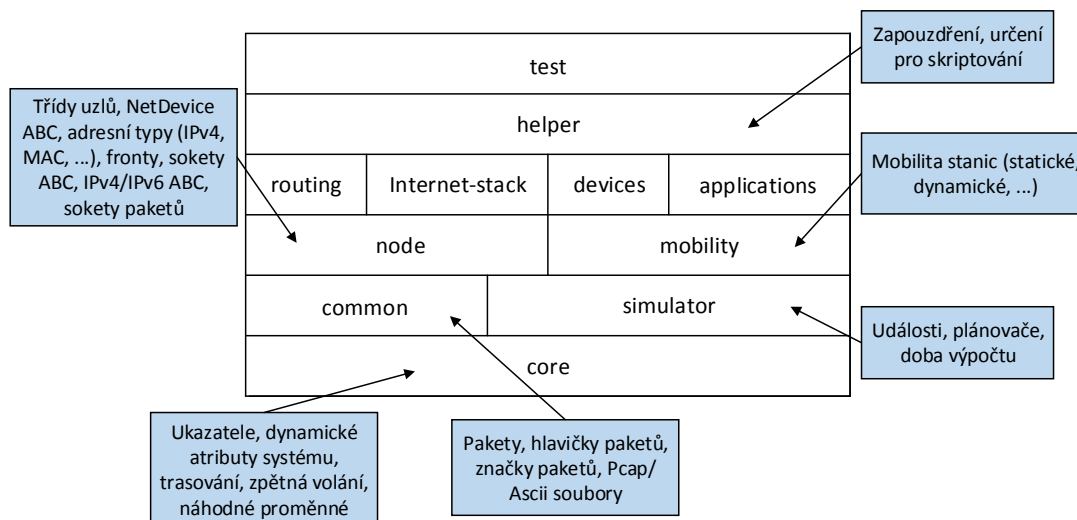
NS-3 je bezplatný software, licencovaný pod GNU GPLv2 licenci a je volně dostupný veřejnosti pro výzkum, vývoj a jiná použití [20]. NS-3 je „discrete-event“ síťový simulátor, určený hlavně pro výzkum a výukové účely. „Discrete-event“ znamená, že každá změna proměnné v systému je provedena v určité instantní události a mezi jednotlivými událostmi už k žádné další změně nedochází (respektive nedochází k žádné změně, která by ovlivnila stav simulátoru). Díky tomu může systém přímo přecházet z jedné události k další [25]. Dalším možným přístupem je „continuous“ simulace, která předpokládá neustálou změnu proměnných a tudíž nepřechází z jedné události do další, ale simulace probíhá nepřetržitě bez rozdělení na události či jednotlivé diskrétní úseky [25]. Poslední možností jsou „discrete“ simulace, které se vždy vztahují k počítatelnému problému a v každém čase přímo určují hodnotu. V dnešní době se proto díky své vhodnosti v 99% případů používají „discrete-event“ simulace sítí [25].

Celý projekt NS-3 byl vyvinut tak, aby zajistil pevné simulační jádro, které je dobře zdokumentované, jednoduše použitelné a laditelné a které dokáže pokrýt potřeby celé simulace (od jejího nastavení až po sběr a analýzu dat). Dále umožňuje vývoj simulačních modelů, které jsou dostatečně realistické, aby se daly používat jako simulátory sítí v reálném čase. Tyto modely se pak používají k simulaci reálných sítí za použití reálných protokolů. Ačkoliv NS-3 umožňuje výzkum jak v IP tak i v non-IP sítích, většina uživatelů jej využívá pro simulaci bezdrátových/IP sítí, které zahrnují modely pro Wi-Fi, WiMAX a LTE pro fyzickou a linkovou vrstvu ISO/OSI modelu a pro množství statických nebo dynamických směrovacích protokolů jako jsou OLSR (Optimized Link State Routing) a AODV (Ad hoc On-Demand Distance Vector) [20].

Další zajímavou funkcí NS-3 je podpora plánovače pracujícího v reálném čase umožňujícího používat několik „simulačních smyček“ (simulation-in-the-loop) schopných pracovat s reálnými systémy. Uživatelé tak mohou například vysílat a přijímat pakety generované NS-3 na reálných síťových prvcích a NS-3 může sloužit jako aplikační rámec (framework) simulující přenosovou síť mezi dvěma virtuálními stroji

[20] ².

Struktura NS-3 a softwarová organizace je zobrazená na Obr.3.1. Zdrojový kód aplikace je uložen v adresáři `src` a jádro celého simulátoru obsahující komponenty společné pro všechny protokoly, modely a hardware je implementováno v `src/core`. Základní objekty síťového simulátoru, pakety, jsou definovány v `src/network`[21].



Obr. 3.1: Softwarová organizace [22]

Simulační jádro i modely v NS-3 jsou naprogramovány v jazyce C++ a celý NS-3 simulátor se chová ve své podstatě jako knihovna, kterou je možné staticky nebo dynamicky spojit s hlavním programem, taktéž vytvořeném v jazyce C++. Hlavní program obsahuje topologii sítě a spouští simulaci. Celá knihovna NS-3 je pomocí knihovny pybindgen zapouzdřena do jazyka Python. Pybindgen pomocí parsování převádí C++ záhlaví NS-3 do gccxml, ze kterého Pygccxml následně automaticky vygeneruje odpovídající C++ soubory, které se sestaví do finálního NS-3 Python modulu umožňujícího pracovat s C++ NS-3 modely a jádrem pomocí skriptů v jazyce Python[21], [22].

3.2.1 Základní design simulačního modelu

Architektura NS-3 je založená na událostech, což si lze představit tak, že se skládá z funkcí, které spouštějí další funkce v přesně udaný čas. Simulační program zajišťuje spouštění funkcí ve správném pořadí a poskytuje obsáhlý framework pro ovládání

²Pro tento účel lze využít framework DCE viz <http://www.nsnam.org/overview/projects/direct-code-execution/>.

tohoto spouštění a definování nových simulačních scénářů za použití existujících modelů [21].

Modely v NS-3 jsou abstraktní reprezentací reálných objektů, protokolů, zařízení atd. Tyto reprezentace jsou implementovány jako třídy nebo skupiny tříd s daným rozhraním naprogramovaným k napodobování určitých vlastností reálných objektů. Výběr vlastností a míra, do které budou napodobeny je důležitou součástí návrhu a implementace modelu. Vždy je nutné vybrat správný poměr mezi požadovanou přesností, komplexností a rychlostí simulace [21].

Pro dosažení vysoké úrovně modularity a flexibility jsou v NS-3 implementována a často používána zpětná volání, agregace objektů a downcasting [23]:

- **Zpětná volání** umožňují předávat funkce jako parametry a jsou často používána pro výměnu dat mezi vrstvami. Tento přístup napomáhá modularitě, neboť jeden model nepotřebuje uchovávat odkaz na objekt určitého typu druhého modelu. Místo toho si pouze uchová odkaz zpětného volání.
- **Agregace** představuje volnou vazbu mezi celkem a součástí, kdy jeden objekt (celek) využívá služby dalších objektů (součástí) [24].
- **Downcasting** umožňuje získat ukazatel na podtřídu pomocí ukazatele na jeho rodičovskou (parent) třídu. Downcasting je stejně jako agregace v NS-3 řešen pomocí metody `GetObject()`.

NS-3 dále definuje objekty:

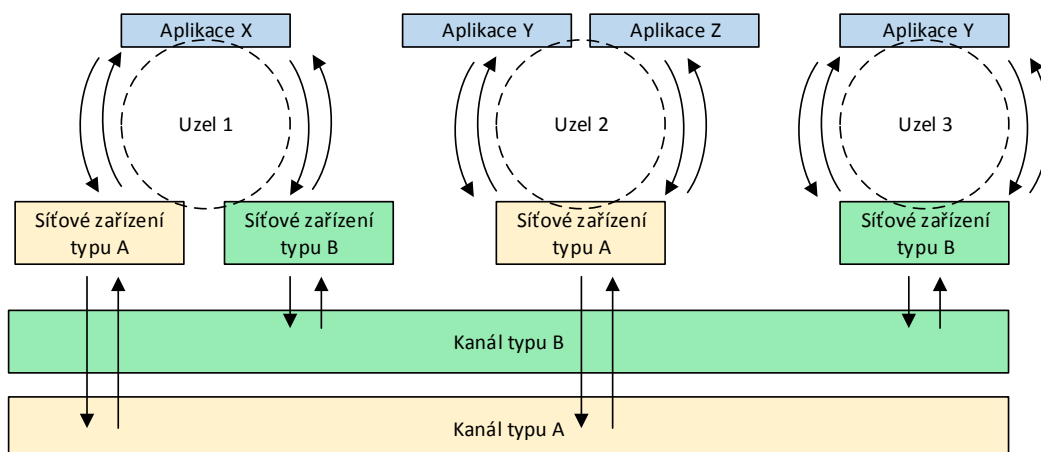
- **Uzel** (Node).
- **Aplikace** (Application).
- **Kanál** (Channel).
- **Síťové zařízení** (NetDevice).
- **Pomocníci topologie** (Topology Helpers).

Všechny tyto objekty mají za úkol ulehčit vývoj simulací a používání modelů. Diagram znázorňující jejich vztah je na Obr.3.2 [21], [23].

Diagram zobrazuje případ, kde jsou dva druhy kanálů (například bezdrátový a drátový), což znamená že musejí existovat i dva druhy síťových zařízení. Jeden uzel má síťové zařízení pro oba typy, zatímco druhé dva jsou připojeny pouze k jednomu typu. Aplikace může použít jakýkoliv kanál v závislosti na adrese a její funkci. Pomocníci topologie (topology helpers) nejsou v diagramu zobrazení kvůli zachování přehlednosti [23].

Všechny výše uvedené objekty jsou reprezentovány odpovídajícími třídami v jazyce C++ a jejich funkce jsou:

- **Uzel** v NS-3 představuje uzel v síťové topologii, přesněji se jedná o počítač či



Obr. 3.2: Základní diagram bloků NS-3 [23]

zařízení v síti. Uzlu lze přiřazovat zařízení a aplikace, na které si uzel následně uchovává ukazatele. Díky agregaci je možné spojit uzel s dalšími objekty jako například pohybovým modelem, který se používá pro určení pozice a pohybu daného uzlu. Agregace také činí celou simulaci flexibilnější, neboť modely jsou schopny ověřit agregaci jednotlivých objektů s uzly a podle toho s nimi dále pracovat. To v praxi znamená možnost vzájemné komunikace modelů, které obsahují agregované objekty s modely bez agregace [21], [23].

- **Aplikace** je abstraktním modelem pro uživatelský program generující určitou aktivitu. Aplikace tedy určuje funkce uzlu, kterými jsou hlavně generování provozu a zpracovávání příchozích dat. K tomu používá podtřídy, které buď vytvoří celý paket a následně jej přímo předávají modelům nižších vrstev, nebo postupně vytváří paket obsahující jejich hlavičku pomocí specifických modelů pro každou vrstvu. Výběr postupu pro tvorbu paketu se odvíjí od požadavků simulace [21], [23].
- **Sítové zařízení** zahrnuje softwarový ovladač i simulovaný hardware. Jeho hlavním úkolem je umožnit komunikaci uzlů přes určitý kanál. Sítové zařízení i kanál musejí být stejného typu aby spolu mohly komunikovat a stejně jako v reálném světě lze uzel připojit k více kanálům pomocí více síťových zařízení. Obecně se tato třída používá jako rozhraní mezi síťovou vrstvou a specifickými funkcemi zařízení, což umožňuje měnění a znovu-používání modelů vyšších vrstev pro různá zařízení [21], [23].
- **Kanál** poskytuje metody pro řízení objektů podsítí a jejich spojení s uzly. Kanál se dá dále specializovat v duchu objektově orientovaného programování. Specializace může být jak velmi jednoduchá jako například simulace přenos-

vého média (např. Ethernet) tak i velmi komplexní v podobě velkých ethernetových přepínačů nebo trojrozměrných prostorů plných překážek (v případě bezdrátových sítí). V případě modelů s více nezávislými, navzájem se nerušícími kanály je nutné použít čísla přenosových kanálů, střední frekvenci či jiné prostředky k jejich odlišení. To se dále dá využít například pro propojení uzlů využívajících pro komunikaci Frequency Division Multiplex (FDM). Tento přístup se však stává velice nepohodlným v simulacích, kde se počet použitých kanálů dynamicky mění [21], [23].

- **Pomocníci topologie** jsou třídy vytvořené za účelem zjednodušení a urychlení tvorby simulace. Obsahují metody pro konfiguraci parametrů, vytváření topologií a propojení výše uvedených tříd. Tyto metody obsahují velké množství příkazů a pomocníci topologie tedy výrazně zpřehledňují výsledný kód a není díky nim nutné tvořit všechny operace pro každý objekt znovu [21], [23].

Zatímco uzel je brán jako abstraktní objekt definovaný objekty s ním spojenými, nové modely aplikací, síťových zařízení a kanálu se vytváří pomocí podtříd. Naopak pomocníci topologie jsou vytvářeni jako samostatné třídy a obvykle nedědí z žádných NS-3 tříd. Účelem těchto tříd je vytvoření kostry, kterou mohou další modely používat, a rozhraní usnadňující další kompatibilitu[21], [23]. Některé z definovaných rozhraní jsou určeny pro IP (Internet Protocol) síť a tak nejsou vhodné pro Wireless Mbus protokol. Dobrým příkladem je metoda `send` definovaná NS-3 třídou `NetDevice`, používající cílovou adresu a číslo protokolu jako parametry. Číslo protokolu se používá pro rozlišení IP a ARP (Address Resolution Protocol) paketů, což je pro non-IP protokoly zbytečné. Wireless Mbus používá pro přenos dat broadcast a tudíž je i cílová adresa nepotřebná. Těmto problémům se dá buď vyhnout přepsáním příslušných metod, vytvořením vlastních metod nebo posíláním bezvýznamných dat těmto parametrům a s tím spojeným nepoužíváním[21], [23].

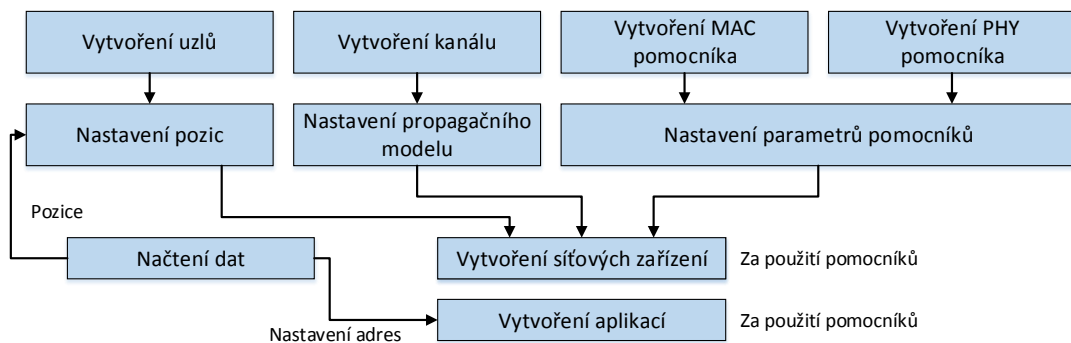
3.2.2 Vytvoření a spuštění simulace

Použití tříd pomocníků topologie a modelových API

Nejjednodušší cestou k vytvoření simulace je použití tříd pomocníků topologie. Příklad tohoto postupu najdete na Obr.3.3.

Jako první se vytvoří požadovaný počet uzlů, kanál určitého typu a pomocníci pro PHY a MAC vrstvy. Pomocníci se později použijí k vytvoření příslušných objektů a jejich atributů [21], [23].

Pomocník síťového zařízení používá kanál, kontejner uzlu obsahující uzel samotný a dva pomocníky pro vytvoření nového síťového zařízení pro každý uzel. Pomocník



Obr. 3.3: Diagram postupu nastavení simulace za použití funkcí pomocníků topologie [23]

síťového zařízení používá MAC a PHY pomocníky k vytvoření MAC a PHY modelu a jejich propojení s kanálem. Aplikační pomocník pak vytváří požadovanou aplikaci pro každý uzel. Načítání dat a jejich následné použití k nastavení adresy a pozice je specificky vytvořené pouze pro tento případ a není základní součástí NS-3 simulací[21], [23].

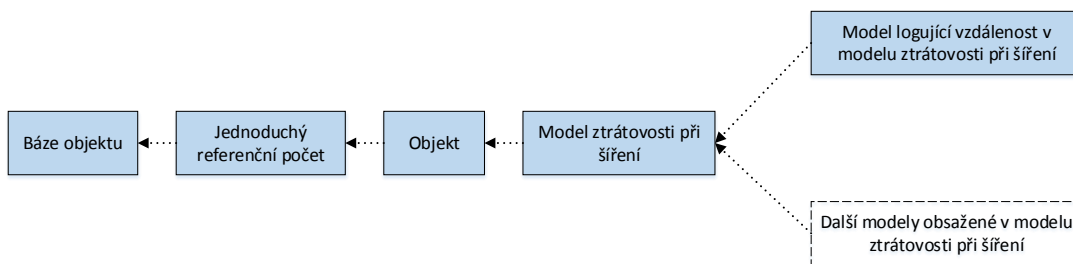
V některých případech pomocníci topologie nejsou dostačující či dostatečně flexibilní a je tedy nutné vytvořit všechny objekty a topologie ručně. K tomuto účelu slouží modelové API (Application Programming Interface), které umožňuje uživatelům implementovat všechny procedury jako vytváření kanálu a uzlů, spojování síťových zařízení s uzly a řetězení modelů pro každou vrstvu uzlu [21], [23].

Vytváření nového modelu

NS-3 obsahuje velký počet přednastavených modelů, ale pokud je potřeba vlastnosti nebo protokolu, který žádný z těchto modelů neobsahuje, nezbyvá než vytvořit model nový nebo některý z existujících upravit. NS-3 simulátor neobsahuje mnoho omezení pro definici struktury modelu nebo jeho implementaci. Jsou zde ale možnosti rozhraní, které se dají využít, a třídy, z kterých se dá dědit. Tyto možnosti výrazně šetří práci při vytváření modelu a navíc jej činí více kompatibilním a tudíž i více použitelným. Při vytváření modelu je důležité uvážit možnosti funkčnosti, znovupoužitelnosti a závislosti. V NS-3 simulátoru je velké množství tříd ze kterých může nový model dědit, v závislosti na jeho budoucím využití[21], [23]:

- Třída `Object` a její podtřídy umí využívat systém atributů NS-3, agregaci objektů a systém počítání chytrých ukazatelů (smart-pointers). Díky těmto vlastnostem je používána ve většině tříd v NS-3 hierarchii a je velmi dobrým stavebním kamenem pro většinu nových tříd.

- Třídy `NetDevice`, `Channel` a `Application` umí využívat rozhraní obsažené ve velkém množství existujících modelů, a tudíž je vhodné dědit i z nich.
- Třídy `Header` a `Trailer` umožňují definovat strukturu zapouzdření paketů pomocí přidávání hlavičky a konce paketu (traileru). Děděním z těchto tříd se tedy výrazně zjednoduší tvorba paketů, jelikož většina z nich se tvoří právě přidáním hlavičky a koncového pole paketu.
- Třída `Address` se používá k reprezentaci tříd. Nový typ adresy se však nevytváří jako podtřída třídy `Address`, ale jako nová základní třída. Tato třída pak implementuje metody `ConvertTo` a `ConvertFrom`, které převádějí specifickým typem adresy a objektem z původní `Address` třídy.
- Dalším vhodným způsobem jak tvořit nové modely je výše zmiňovaná úprava modelu stávajícího. Existují modely pro šíření, generování chyb, spotřebu energie a další jevy. Tyto modely jsou často definovány s obecnou rodičovskou třídou, což umožňuje vytvářet nové instance a jejich vložení do jiných modelů, bez jejich změny. Příklad takovéto implementace je znázorněn na Obr. 3.4.



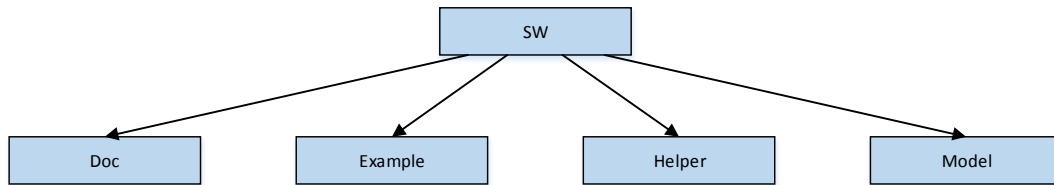
Obr. 3.4: Diagram znázorňující hierarchii dědičnosti v modelu šíření[23]

Nově vytvořený model musí být zapojen do simulace. Pokud je obecného typu, jako nový model šíření, lze jej použít jako náhradu za jakýkoliv jiný model tohoto typu. V ostatních případech musí být model zapojen do simulace například pomocí nastavení funkce zpětného volání, změnou kódu existujících modelů nebo nastavením spouštění metod nového modelu při začátku simulace. Pokud je model odvozen z `Application` třídy, měly by být jím definované funkce spouštěny v průběhu simulace[21], [23].

Jednoduchý síťový modul (Simple Wireless Module = sw)

NS-3 obsahuje velké množství komplexních modulů pro různé síťové protokoly. To je velkou výhodou při jejich používání, ale zároveň nevýhodou při jejich editaci. Pro simulaci Wireless M-bus protokolu tak v této práci bude popsán a modifikován jednoduchý síťový modul, napsaný Junseokem Kimem [23].

Adresářová struktura jednoduchého síťového modelu je velice přehledná a její zjednodušená podoba je zobrazena v Tab.3.5. Plná verze struktury je k nalezení v příloze B.1.



Obr. 3.5: Základní adresářová struktura sw modelu

Každý objekt v jednoduchém síťovém modulu je tvořen dvěma soubory. První má příponu `.h` a jedná se o tzv. hlavičkový soubor obsahující definici příkazů a struktury dané objektu. Druhým souborem je soubor s příponou `.cc` obsahující již samotnou implementaci jednotlivých metod a tříd. Jednoduchý síťový modul, vytvořený J. Kimem a následně upraven a doplněn Ing. Z. Kuderem, využívaný v této práci obsahuje tyto pomocníky topologie:

- `mbus-helper` - obsahuje podtřídy `MbusNodeHelper` a `MbusBaseHelper` umožňující vytvoření sad objektů stejného typu. Vstupními parametry těchto tříd jsou seznamy souřadnic a adres.
- `sw-helper` - obsahuje podtřídy `SwMac`, `SwPhy`, `SwChannel`, `SwMacHelper` a `SwPhyHelper`. Pomocí těchto tříd je schopný přiřadit síťovému zařízení MAC (Media Access Control) adresu, PHY adresu, definovat typ komunikačního kanálu a naslouchání na kanálu a dále přiřadit toto zařízení uzlu.
- `sw-mac-csma-helper` - usnadňuje nastavení CSMA (Carrier Sense Multiple Access) vlastností MAC adresy.
- `sw-phy-basic-helper` - usnadňuje přiřazení MAC adresy.

Jednoduchý síťový modul dále obsahuje tyto modely:

- `angles` model umožňuje převod mezi stupni a radiány. Obsahuje také strukturu definující sférické úhly azimutu a sklonu. Zápis těchto úhlů je stejný jako v [26]. Tento zápis odpovídá běžně používaným sférickým souřadnicím, kde se v rovině $x-y$ měří Φ proti směru hodinových ručiček od osy x . Theta (Θ) se měří v rovině $y-z$ od osy z rovněž proti směru hodinových ručiček. Dále `angles` model umožňuje tyto úhly přiřazovat různým strukturám.
- `antenna-model` poskytuje rozhraní pro definování rádiových vlastností antén. Definice těchto vlastností je postavena na použití sférických souřadnic.
- `ConcentratorMappingUtils` model poskytuje funkce pro převod mezi IP adresou, sériovým číslem a základním jménem. Tyto funkce jsou velice důležité,

neboť Wireless M-Bus nevyužívá klasické adresování a tudíž nelze využívat IP adresy.

- `cosine-antenna-model` umožňuje použití kosinového modelu popsaného v [27] a obsahuje pouze parametry v něm definované. Navíc obsahuje přidání konstantní zisk ve vertikální rovině, který má za úkol kompenzovat absenci úhlu elevace.
- `geography-functions` umožňuje pracovat s geografickými údaji. Mezi jeho funkce patří výpočet úhlu mezi základnou a uzlem a výpočet vzdálenosti dvou bodů ³.
- `isotropic-antenna-model` umožňuje využívat isotropickou anténu, což je nej-jednodušší model antény. Tato anténa má jednotný zisk 0 dB ve všech směrech.
- `mbus-app` model dědí z `Application` třídy NS-3 a obsahuje vše potřebné pro simulaci Wireless M-Bus protokolu. Obsahuje definici formátu rámců, typ režimu, broadcastový režim, typ rámců a typ trasování. Dále také obsahuje struktury definující typy uzlů a strukturu M-Bus. Součástí `mbus-app` modelu jsou také metody pro spuštění simulace, nastavení alarmů, nastavení naslouchání broadcastu.
- `node-loading-functions` je jednoduchá struktura pro ukládání adres a pozic ze souboru předtím, než se přidělí uzlům.
- `parabolic-antenna-model` definuje model pomocí parabolické aproximace hlavního laloku vyzařovacího diagramu. Parabolický model použitý v této metodě je stejný jako v [28]. Jediným rozdílem je absence elevační roviny.
- `sw-channel` obsahuje definici přenosových kanálů. Jeho součástí jsou metody umožňující identifikaci ID kanálu, určení počtu zařízení, odebírání nenaslouchajících zařízení, přidávání zařízení, posílání paketů, přijímání paketů atd.
- `sw-mac` obsahuje základní definici MAC adresy a jejích parametrů.
- `sw-mac-csma` dědí z třídy `sw-mac` a dále definuje CSMA parametry MAC adresy.
- `sw-mac-header` dědí z třídy `Header` a dále upravuje hlavičku paketů tak, aby byla vhodná pro bezdrátový přenos.
- `sw-net-device` obsahuje definici síťových zařízení. Jeho součástí jsou metody pro získání typu ID, nastavení naslouchání, všesměrového vysílání, uzlu, MAC adresy, PHY adresy, kanálu, adresy atd.
- `sw-phy` obsahuje základní definici PHY adresy a jejích parametrů.

³Pro tento výpočet používá Haversinův vzorec

4 ROZŠÍŘENÍ MODELU SW

Model sw je rozsáhlou aplikací pro simulaci přenosu dat mezi uzly a koncentrátory. Byl navržen však pro základní simulaci těchto přenosů, a z tohoto důvodu neobsahuje dostatek parametrů a informací pro úplnou simulaci protokolu Wireless M-Bus. Cílem práce je tyto nedostatky doplnit a vytvořit tak plně funkční model pro simulaci přenosu reálných dat. Pro správný chod celé simulace je potřeba vytvořit metody generující data ve formátu daného výrobce a metody, které data zpětně dekodují a zobrazí. Následně se tyto metody musí implementovat do stávajícího modelu a musí se zajistit přenos vygenerovaných dat mezi koncentrátory a uzly.

4.1 Přidání metod pro přenos dat určitého formátu

Prvním krokem práce bylo přidání metod pro tvorbu dat ve formátu odpovídajícím výrobcí daného měřiče. K tomu bylo potřeba přidat atributy M-Bus aplikaci. Konkrétně se jedná o parametry:

1. `producer` obsahující název výrobce.
2. `meterType` obsahující informace o typu měřiče.
3. `m_bonegaFrameData`, `m_pikkertonFrameData`, `m_weptechFrameData` a `m_zpaFrameData` obsahující základní data správného formátu jednotlivých výrobců.

Následuje ukázka kódu obsahující přidání parametru `producer`.

```
.AddAttribute ("producer",
               "Packet format of chosen producer",
               EnumValue (PIKKERTON),
               MakeEnumAccessor (&MbusApp::m_producer),
               MakeEnumChecker (MbusApp::PIKKERTON, "PIK", MbusApp
                               ::BONEGA, "BON",
                               MbusApp::WEPTech, "WEP", MbusApp::ZPA, "ZPA"))
```

Dále bylo potřebné vytvořit samotné metody starající se o plnění paketů aktuálními daty. Pro univerzálnost, lepší přehled a možnost změny pouze určitých parametrů byl zvolen postup plnění dat po jednotlivých částech, což znamená, že pro každou informaci obsaženou v paketu byla vytvořena jedna metoda, která ji nastavuje. Tyto metody plní data znak po znaku na přesné pozice do předem načteného paketu dle zvoleného výrobce 2.4. Z důvodu vyčítání dat od LSB ve většině případů tyto metody obsahují i část, ve které otáčejí pořadí jednotlivých dvojic bytů. Níže

je zobrazena metoda starající se o nastavení identifikačního čísla výrobce, která je společná pro všechny aktuálně podporované výrobce.

```
void MbusApp::setIDNumber(std::string idNum){
    std::string rightOrder = "00000000";
    rightOrder[0] = idNum[6];
    rightOrder[1] = idNum[7];
    rightOrder[2] = idNum[4];
    rightOrder[3] = idNum[5];
    rightOrder[4] = idNum[2];
    rightOrder[5] = idNum[3];
    rightOrder[6] = idNum[0];
    rightOrder[7] = idNum[1];

    for (unsigned int i = 0; i < 8; ++i) {
        m_dataToSend[8+i] = rightOrder[i];
    }
}
```

Stejně jako byly vytvořené metody pro plnění paketů jsou vytvořeny i metody pro vyčítání jednotlivých dat z paketů. Následující kód se stará o získání identifikačního čísla výrobce.

```
std::string MbusApp::getIDNumber(){
    std::string rightOrder = "00000000";
    std::string idNum = "00000000";

    for (unsigned int i = 0; i < 8; ++i) {
        idNum[i] = m_dataReceived[8+i];
    }

    rightOrder[0] = idNum[6];
    rightOrder[1] = idNum[7];
    rightOrder[2] = idNum[4];
    rightOrder[3] = idNum[5];
    rightOrder[4] = idNum[2];
    rightOrder[5] = idNum[3];
    rightOrder[6] = idNum[0];
    rightOrder[7] = idNum[1];
    return rightOrder;
}
```

Dalším krokem bylo doplnění přenosu dat mezi koncentrátorem a uzly. To bylo zajištěno vytvořením metod pro načtení základního paketu a dále úpravou metod `MbusApp::Send` a `MbusApp::Receive`. V metodě `MbusApp::Send` se konkrétně jednalo o volání metod pro načtení a následné nastavení jednotlivých částí paketu, v

závislosti na vybraném výrobci. Následuje krátká ukázka volání zmíněných metod.

```
else if ((m_producer == PIKKERTON)&& (m_meterType ==
    PikElectricity) ) {
    std::string actualAcc = intToHexString((m_acc + m_accIncrement)
    );
    setAccNumber(actualAcc);
    setIDNumber(id);
    setMeterType(type);
}
```

Naopak v metodě `MbusApp::Receive` se tyto nastavená data zpětně vyčítají a následně zobrazují do konzole. I zde se data vyčítají dle zvoleného výrobce.

Po vytvoření všech metod potřebných pro načítání, ukládání a posílání dat bylo nutné upravit některé z helper metod stávajícího modelu. Konkrétně se jednalo o metody `MbusHelper::InstallCommon`, `MbusNodeHelper::Install` a `MbusBaseHelper::Install`. Do těchto metod bylo nutné implementovat nově přidané parametry pro určení typu měřiče a výrobce. Příklad implementace je zobrazen v následující části kódu.

```
ApplicationContainer MbusHelper::InstallCommon (Ptr<Node> node,
    MbusAddress address, MbusApp::Producer producer, MbusApp::
    MeterType meter, double maxTimeDelay, bool bNode)
{
    NS_LOG_FUNCTION("MbusNodeHelper installcommon node");
    ApplicationContainer applicationContainer;
    Ptr<MbusApp> mbusApp = CreateObject<MbusApp>();
    mbusApp->m_address = address;
    mbusApp->m_producer = producer;
    mbusApp->m_meterType = meter;

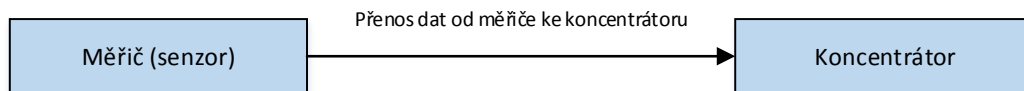
    node->AddApplication(mbusApp);
    mbusApp->SetNode(node);
    applicationContainer.Add(mbusApp);
}
```

Přidáním těchto metod bylo zajištěno vše potřebné pro vytvoření samotného simulačního scénáře, jehož stručný popis obsahuje následující kapitola 4.2.

4.2 Simulační scénář

V této kapitole bude popsána simulace vytvořená za účelem předvedení možností rozšířeného síťového modulu. Pro zachování přehlednosti celá simulace obsahuje

pouze dva uzly. První je koncentrátor a druhý měřič. Mezi nimi je vytvořen komunikační kanál, umožňující jejich vzájemnou komunikaci. Topologie simulace je zobrazena na Obr.4.1.



Obr. 4.1: Diagram znázorňující topologii vytvořené simulace

4.2.1 Popis jednotlivých částí simulace

V této části je uveden popis hlavních částí kódu simulace, rozdělený do jednotlivých bodů. Přestože se jedná pouze o důležité části kódu, jsou uspořádány ve stejném pořadí, v jakém se objevují v kódu.

- První věcí při tvorbě simulace bylo importování používaných modelů. To je v kódu provedeno pomocí příkazu `#include`. V uvedené aplikaci je importováno 23 modulů (například `ns3/sw-channel.h` atd. . .). Dále bylo nutné importovat jmenný prostor, ve kterém budeme pracovat (v našem případě `ns3`). Import je proveden pomocí příkazu `using namespace ns-3`.
- Dalším důležitým krokem bylo vytvoření samotné metody `main` ovládající průběh celé aplikace. Popis této metody byl rozdělen do následujících bodů:
 1. Vytvoření všech důležitých proměnných a aktivace možnosti volat a ovládat aplikaci z příkazové řádky pomocí parametrů. Parametry pro volání z příkazové řádky jsou nastaveny pomocí následujícího kódu.

```

CommandLine cmd;
cmd.AddValue("meterType", "Type of meter. Currently
available: PIKEL,WEPTMP", meterType);
  
```

2. Definice uzlů a vytvoření modelu pro simulaci ztrátovosti. K tomuto účelu jsou použity metody, které automaticky vytvoří předem zvolený počet uzlů a koncentrátorů. Dále se modelu ztrátovosti přiřadí jeho parametry a následně se model přiřadí kanálu.

```

MbusApp::NodeItemList nodeList = FakeNodeList();
MbusApp::NodeItemList baseList = FakeBaseList();
  
```

```

Ptr<PropagationLossModel> propagationLossModel;
if(useMeasuredPathloss){
    propagationLossModel = CreateObject<
        MatrixPropagationLossModel>();
}
else{
    propagationLossModel = CreateObject<
        LogDistancePropagationLossModel>();
    propagationLossModel->GetObject<
        LogDistancePropagationLossModel>()->
        SetReference(1, pathlossRef); // for 868 MHz
        20 * log((4 * pi * 1 * 868.95e6) / 3e8) =
        31.2216679
    propagationLossModel->GetObject<
        LogDistancePropagationLossModel>()->
        SetPathLossExponent(pathlossExponent);
}

Ptr<SwChannel> swChan = CreateObject<SwChannel> ();

swChan->SetAttribute("PropagationLossModel",
    PointerValue(propagationLossModel));
swChan->SetShadowVariableStd(8);

```

3. Definice pomocníků pro fyzickou a linkovou vrstvu. Po vytvoření jsou jim přiřazeny parametry. Fyzické vrstvě je přiřazena hodnota vysílacího výkonu, prahu detekce, poměru signálu k rušení a šumu. Linkové vrstvě zůstává základní nastavení. To zajišťuje kód:

```

SwMacCsmaHelper swMac = SwMacCsmaHelper::Default ();
SwPhyBasicHelper swPhy = SwPhyBasicHelper::Default
    ();

swPhy.Set("TxPower", DoubleValue(0));
swPhy.Set("CsPowerTh", DoubleValue(-200));
swPhy.Set("SinrTh", DoubleValue(pow(10, 8/10)));

```

4. Přiřazení kontejnerů s uzly do kontejnerů síťových zařízení. V tomto přiřazení už jednotlivé uzly obsahují parametry kanálu, fyzické a linkové vrstvy.

```

SwHelper sw;
NetDeviceContainer basesDevices = sw.Install (bases,
    swChan, swPhy, swMac);

```

```
NetDeviceContainer nodesDevices = sw.Install (nodes ,
      swChan , swPhy , swMac);
```

5. Vytvoření pomocníků aplikace, nastavení typu antén a přiřazení jednotlivých uzlů do aplikačních kontejnerů. Zde je nutné rozdělit aplikaci dle zvoleného výrobce a typu měřiče. O to se stará následující část kódu:

```
MbusBaseHelper mbusBaseHelper (MbusApp::C1 , MbusApp::
      B , mBusN);
MbusNodeHelper mbusNodeHelper (MbusApp::C1 , MbusApp::
      B , mBusN);

mbusBaseHelper.SetAntennaType (MbusHelper::ISOTROPIC)
      ;
mbusNodeHelper.SetAntennaType (MbusHelper::ISOTROPIC)
      ;

if (producer == "BONEGA") {
ApplicationContainer mbusNodeApplications =
      mbusNodeHelper.Install(nodes , nodeList , MbusApp::
      BONEGA , MbusApp::PikElectricity , 1); //nastavuje i
      mobility model
ApplicationContainer mbusBaseApplications =
      mbusBaseHelper.Install(bases , MbusApp::BONEGA ,
      MbusApp::PikElectricity , baseList);
```

6. Vytvoření rozhraní pro animaci a spuštění aplikací na uzlech a následně celé simulace. Simulace i aplikace se ukončí po 20 sekundách.

```
AnimationInterface anim (animFile); //netAnim

mbusBaseApplications.Start(Seconds(0.0));
mbusBaseApplications.Stop(Seconds(20));

mbusNodeApplications.Start(Seconds(0.0));
mbusNodeApplications.Stop(Seconds(20));

NS_LOG_INFO("Vypis komunikace ...");

Simulator::Run ();
Simulator::Stop(Seconds(20));
NS_LOG_INFO("Konec simulace.");
```

- Posledním krokem je kontrola samotného výstupu, ověřující funkčnost celého kódu. Výstup z konzole je zobrazen i s popisky níže a je z něj patrné, že dochází k nastavení výrobce, dat v paketu, přenosu rámců a následnému vyčítání

hodnot jednotlivých counterů.

- Výpis pro kontrolu nastavení parametrů, konkrétně pozice uzlu (x,y,z), typ uzlu (přenos / naslouchání), acc, počet rámců před přenosem plného rámce, Mbus N, a typ rámce

```
Vypis komunikace ...
1, 1, 1 #MbusAppXXX: listen?1 acc: 0
  fullFrameCountdownDefault:7 N:8 frame==B?1 mode==C1?:1
  producer==PIK?1. 0
2, 2, 2 #MbusAppXXX: listen?0 acc: 169
  fullFrameCountdownDefault:7 N:8 frame==B?1 mode==C1?:1
  producer==PIK?1. 0
```

- Výpis času, ve kterém proběhne další přenos

```
Scheduling next transmit to dt = +0.0ns. Acc
=169, inc=0, addr: 0
Scheduling next alarm to dt = +68066806408.0ns,
Acc=169, inc=0
```

- Výpis nastavených hodnot měřiče a jejich kontrola před odesláním

```
SIZE OF FREQ: 8, dataOfFreq = 00000001
SIZE OF VOLTAGE: 2, dataOfVoltage = 01
SIZE OF CUR: 4, dataOfCur = 0001
SIZE OF POWER: 4, dataOfPower = 0001
SIZE OF WORK: 8, dataOfWork = 00000001
Data Before send : 28442B414452127002027Aaa00000004FB
2C010000001FD490102FD590100022B0100040301000000
```

- Popis posílaného paketu, konkrétně čas odeslání, velikost, acc, pořadí a data

```
MbusApp::Sendtime: +0.0ns, Packet size:81, Acc=169, inc=1
,Data=28442
B414452127002027Aaa00000004FB2C010000001FD490102FD
590100022B0100040301000000
Sending Compact Frame
```

- Výpis plánovaného odeslání dalšího paketu

```
Scheduling next transmit to dt = +15828125000.0ns. Acc
=169, inc=1, addr: 0
```

- Výpis přijatých dat, síly signálu, velikosti paketu, času příchodu a adresy odesílatele


```
DATA RECEIVED: 28442B414452127002027Aaa00000004FB
2C0100000001FD490102FD590100022B010004030100000
MbusApp:Receive. Power: -34.7346Size: 81. Time: +7120004.0
nsBase address: 0. Sender address: 72783391635506
```

– Výpis vyčtených hodnot z dat přijatého paketu

```
AccReceived: 170
IDReceived: 70125244
MeterTypeReceived: 02
FrequencyReceived: 1
VoltageReceived: 1
CurrentReceived: 1
PowerReceived: 1
WorkReceived: 1
```

5 ZÁVĚR

V této diplomové práci byla popsána problematika M2M (Machine-to-Machine) komunikace pomocí protokolu Wireless M-bus a její simulace v simulačním programu NS-3 (Network Simulator 3). V první části 1 práce byla rozebrána M2M komunikace od její historie až po současné možnosti přenosu dat.

Následující část 2 se zaměřila na protokol Wireless M-Bus, konkrétně na jeho historii a jednotlivé vrstvy. Díky nutnosti znalosti fyzické a linkové vrstvy pro simulaci byly tyto vrstvy rozebrány podrobněji. Dále byla popsána struktura dat u výrobců Bonega, WEPTECH, Pikkerton a ZPA. Z popisu obsaženém v této části práce je vidět, jak široké je použití protokolu Wireless M-Bus. Největší uplatnění a s ním spojené rozšíření se dá očekávat hlavně v oblasti chytrých měřičů a a chytrých domů. Z tohoto důvodu bude stále nutnější vše nejdříve simulovat a následně implementovat.

Třetí část 3 se zabývá simulátorem NS-3, použitým následně k samotné simulaci v praktické části. Jsou zde popsány jeho předchozí verze, důvod jeho zvolení, design i kroky potřebné k vytvoření simulace.

Čtvrtá část 4 obsahuje popis vytvořené aplikace. Konkrétně jde o popis její topologie a následně o popis jednotlivých částí kódu. Účelem celé práce bylo vytvoření aplikace pro simulaci přenosu reálných dat mezi měřičem a koncentrátorem a následná demonstrace funkčnosti. Z výstupu aplikace (viditelném v konzoli) je patrné, že pakety obsahují příslušná data, komunikace mezi měřičem a koncentrátorem funguje a přenáší se data zvoleného výrobce, která se následně vyčítají a zobrazují. Pro zachování přehledného výpisu je v simulaci zvolen jeden koncentrátor a jeden uzel.

Nakonec bych rád zmínil další možnosti vývoje aplikace. V aktuální verzi je aplikace schopná simulovat přenos reálných dat mezi libovolným počtem uzlů a koncentrátorů. Je však schopná simulovat pouze data od výrobců Bonega, Pikkerton, WEPTECH a ZPA. V budoucnu by bylo dobré doplnit podporu více výrobců, AES kódování přenášených dat či přívětivé grafické uživatelské rozhraní (GUI).

LITERATURA

- [1] GENERAL INFORMATION. TELIT. *Telit Wireless Solutions* [online]. [cit. 2014-10-27]. Dostupné z: <http://www.telit.com/experience-telit/what-is-m2m/general-information/>
- [2] Machine to machine. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2006-204, 29.10.2014 [cit. 2014-10-30]. Dostupné z: http://en.wikipedia.org/wiki/Machine_to_machine
- [3] CRADLEPOINT TECHNOLOGY. *When Machines Talk to Machines: M2M deployment can make your business systems smarter* [PDF]. 2012, 13 s. [cit. 27.10.2014]. Dostupné z: http://www.streakwave.com/cradlepoint/CradlePoint_M2M_White_Paper.pdf
- [4] *M2M communications: a systems approach*. 1st ed. Editor David Boswarthick, Omar Elloumi, Olivier Hersent. Chichester: John Wiley, 2012, xxiii, 308 s. ISBN 978-1-119-99475-6.
- [5] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018. CISCO. *Cisco.com* [online]. 2014, 2014-02-05 [cit. 2014-11-01]. Dostupné z: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html
- [6] CHEN, Min, Jiafu WAN, Sergio GONZÁLEZ, Xiaofei LIAO a Victor C.M. LEUNG. A Survey of Recent Developments in Home M2M Networks. *Communications Surveys & Tutorials, IEEE* [online]. 2013, Vol. 16, 98 - 114, 2014-02-07 [cit. 2014-10-28]. Dostupné z: http://mmlab.snu.ac.kr/~mchen/min_paper/IEEE-COMST-M2M2014.pdf
- [7] HAUER, Jan-Hinrich, Vlado HANDZISKI a Adam WOLISZ. TELECOMMUNICATION NETWORKS GROUP. *Experimental Study of the Impact of WLAN Interference on IEEE 802.15.4 Body Area Networks*. Berlin, 2009. Dostupné z: http://www.tkn.tu-berlin.de/fileadmin/fg112/Papers/hauer_ewsn20091.pdf
- [8] Sběrnice Wireless M-BUS - jde to i bezdrátově... *Automatizace.hw.cz* [online]. 2010 [cit. 2014-10-28]. Dostupné z: <http://automatizace.hw.cz/sbernice-wireless-mbus-jde-i-bezdratove>

- [9] SILICON LABS. *WIRELESS M-BUS SOFTWARE IMPLEMENTATION*. 2010, 14 s. Dostupné z: <https://www.silabs.com/Support%20Documents/TechnicalDocs/AN451.pdf>
- [10] EN 13757-1. *Communication system for and remote reading of meters - Part 1: Data exchange*. Wien: Austrian Standards Institute, 2013-02-15. Dostupné z: <https://shop.austrian-standards.at/Preview.action;jsessionid=4B46107107AC62A5CB24E33F6A51A5E4?preview=&dokkey=467673&selectedLocale=en>
- [11] ČSN EN 13757-4. *Komunikační systémy pro měřidla a měřidla s dálkovým čtením – Část 4: Bezdrátová měřidla (radiometry pro provoz v pásmu SRD)*. Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2014. Dostupné z: http://www.technicke-normy-csn.cz/inc/nahled_normy.php?norma=258513-csn-en-13757-4&kat=94180
- [12] EN 13757-4. *Communication systems for meters and remote reading of meters - Part 4: Wireless meter readout (Radio Meter reading for operation in the 868-870 MHz SRD band)*. Brusel: EUROPEAN COMMITTEE FOR STANDARDIZATION, 2003. Dostupné z: <http://oldfjarrvarme.unc.se/download/1309/fj>
- [13] IEC 60870-5-2:1992. *TELECONTROL EQUIPMENT AND SYSTEMS Part 5: Transmission protocols Section 2: Link transmission procedures*. Geneva: INTERNATIONAL ELECTROTECHNICAL COMMISSION, 1992-04-30.
- [14] EUROPEAN COMMITTEE FOR STANDARDIZATION. *Communication systems for meters and remote reading of meters - Part 4: Wireless meter readout (Radio Meter reading for operation in the 868-870 MHz SRD band)*. 2003, 41 s. Dostupné z: <http://oldfjarrvarme.unc.se/download/1309/fj>
- [15] MAŠEK, Ing. Pavel. *Wireless M-Bus USB Adapter (AMB8425-M)*. 2013, 5 s. Dostupné z: <http://147.229.144.32/public.php?service=files&t=0a90a8682fc8e8d1ba17a624dd6a345a&path=/Literatura/Wireless%20M-BUS>
- [16] NXP SEMICONDUCTORS. *AN11017: Transceiver OL2381 using wireless M-BUS*. Rev. 2. 2011, 43 s. Dostupné z: http://www.nxp.com/documents/application_note/AN11017.pdf
- [17] Manchester code. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2014, 17-11-2014 [cit. 2014-11-18]. Dostupné z: http://en.wikipedia.org/wiki/Manchester_code

- [18] *NS-3* [online]. © 2011-14 [cit. 2014-12-15]. Dostupné z: <http://www.nsnam.org/>
- [19] Ns (simulator). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2014, 20 Listopad 2014 [cit. 2014-11-23]. Dostupné z: [http://en.wikipedia.org/wiki/Ns_\(simulator\)](http://en.wikipedia.org/wiki/Ns_(simulator))
- [20] NS-3: WHAT IS NS-3. NS-3 CONSORCIUM. *NS-3* [online]. 2014 [cit. 2014-11-23]. Dostupné z: <http://www.nsnam.org/overview/what-is-ns-3/>
- [21] NS-3 CONSORTIUM. *Ns-3 Tutorial*. Release ns-3.13. 2011. Dostupné z: <http://www.nsnam.org/docs/release/3.13/tutorial/ns-3-tutorial.pdf>
- [22] PAVEL, Mašek. *QOS MODEL PRO MOBILNÍ AD HOC SÍŤ*. Brno, 2013. Dostupné z: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=63715. Diplomová práce. VUT. Vedoucí práce Ing. PAVEL VAJSAR.
- [23] KUDER, Zenon. *ROUTING PROTOCOLS FOR LOSSY WIRELESS NETWORKS*. Brno, 2012. Dostupné z: http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=59199. Diplomová. VUT. Vedoucí práce doc. Ing. JIŘÍ ŠEBESTA, Ph.D.
- [24] Kapitola 1. Principy objektově orientovaného programování. *Katedra Informatiky VŠB* [online]. 2003 [cit. 2014-12-15]. Dostupné z: <http://www.cs.vsb.cz/benes/vyuka/pte/texty/objekty/index.html>
- [25] Discrete event simulation. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2014, 2014-03-28 [cit. 2014-12-05]. Dostupné z: http://en.wikipedia.org/wiki/Discrete_event_simulation
- [26] BALANIS, Constantine A. *Antenna theory: analysis and design*. 2nd ed. New York: Wiley, c1997, xvi, 941 s., příl. ISBN 04-715-9268-4.
- [27] CHUNJIAN, Li. *Efficient Antenna Patterns for Three-Sector WCDMA Systems*. Švédsko, 2003. Diplomová práce. Chalmers University of Technology.
- [28] CALCEV, George a Matt DILLON. Antenna tilt control in CDMA networks. *Proceedings of the 2nd annual international workshop on Wireless internet - WICON '06*. 2006, 2nd annual. DOI: 10.1145/1234161.1234186.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ACK	Acknowledgment
AL	Application Layer
AODV	Ad hoc On-Demand Distance Vector
API	Application Programming Interface
ARP	Address Resolution Protocol
BCD	Binary Coded Decimal
CAGR	Compound Annual Growth Rate
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
DSL	Digital subscriber line
FDM	Frequency Division Multiplex
FT3	Frame Type 3
GPS	Global Positioning System
GSM	Groupe Spécial Mobile
GUI	Graphical user interface
H _R	Highest performance
H2H	Human to Human
HomeRF	Home Radio Frequency
HSPA	High Speed Packet Access
IP	Internet Protocol
L _R	Lowest performance
LTE	Long Term Evolution
M _R	Medium performance
M2M	Machine to Machine

MAC	Media Access Control
NASA	National Aeronautics and Space Administration
NFC	Near Field Communication
NSF	National Science Foundation
OLSR	Optimized Link State Routing
PHY	Physical Layer
PoS	Point of Sale
POTS	Plain old telephone service
RF	Radio Frequency
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
SW	Simple Wireless
UART	Universal Asynchronous Receiver/Transmitter
WiMAX	Worldwide Interoperability for Microwave Access
WPAN	Wireless Personal Area Network

SEZNAM PŘÍLOH

A	Tabulky s parametry jednotlivých režimů protokolu Wireless M-Bus	64
A.1	Detailní parametry režimu S	64
A.2	Detailní parametry režimu T	65
A.3	Detailní parametry režimu R	66
B	Tabulka adresářové struktury síťového modulu NS-3	67
C	Obsah přiložených DVD	68

A TABULKY S PARAMETRY JEDNOTLIVÝCH REŽIMŮ PROTOKOLU WIRELESS M-BUS

A.1 Detailní parametry režimu S

Charakteristika	Režim	Sym	Min	Typ	Max	Jednotka
Střední frekvence (měříče pouze s vysílačem, pod-režim S1)			868,25	868,30	868,35	MHz
Střední frekvence (ostatní a režim S2)			868,278	868,300	868,322	MHz
Odchylka FSK			± 40	± 50	± 80	kHz
Rychlost přenosu čipů		f_{chip}		32,768		kcps
Tolerance rychlosti čipů			-1,5	0	1,5	%
Kolísání digitálních bitů (jitter)					± 3	us
Rychlost dat (Manchester)				$f_{chip}^{*1,2}$		bps
Délka úvodní sekvence včetně bit/byte synchronizace, v obou směrech	S2,S1-m		48			čipů
Délka úvodní sekvence včetně bit/byte synchronizace	S1		576			čipů
Délka závěrečné (trailer) sekvence			2		8	čipů
Zpoždění odezvy		t_{RO}	3		50	ms

Tab. A.1: Režim S - Detailní parametry vysílače [14]

Charakteristika	Režim	Sym	Min	Typ	Max	Jednotka
Citlivost (BER 1 z 10^2) nebo míra přijímání nad 80%	H_R	P_O	-100	-105		dBm
Blokovací výkon	L_R		3			Třída
Blokovací výkon	M_R		2			Třída
Blokovací výkon	H_R		2			Třída
Přijatelná tolerance rychlosti čipů		Df_{chip}	-2	0	2	%
Čípková rychlost (měříče)		f_{chip}		32,768		kcps

Tab. A.2: Režim S - Detailní parametry přijímače [14]

A.2 Detailní parametry režimu T

Charakteristika	Režim	Sym	Min	Typ	Max	Jednotka
Střední frekvence (od měřiče k ostatním)	T1,T2		868,90	868,95	869,00	MHz
Střední frekvence (od ostatních k měřiči)	T2		868,278	868,300	868,322	MHz
Odchylka FSK (od měřiče k ostatním)	T1,T		±40	±50	±80	kHz
Odchylka FSK (od ostatních k měřiči)	T2		±40	±50	±80	kHz
Rychlost přenosu čipů (od měřiče k ostatním)	T1,T2	f_{chip}	90	100	110	kcps
Rychlost změny v hlavičce a telegramu	T1,T2	Df_{chip}	-1	0	1	%
Rychlost dat (od měřiče k ostatním - kódování 3 z 6)	T1,T2			$f_{chip} * 2/3$		bps
Rychlost přenosu čipů (od ostatních k měřiči)	T2			32,768		kcps
Tolerance rychlosti přenosu čipů (od ostatních k měřiči)	T2		-1,5	0	1,5	%
Kolísání digitálních bitů	T2				3	us
Rychlost dat (od ostatních k měřiči - Manchester)	T2			$f_{chip} * 1/2$		bps
Délka úvodní sekvence včetně bit/byte synchronizace, v obou směrech	T1,T2		48			čipů
Délka závěrečné (trailer) sekvence	T1,T2		2		8	čipů
Zpoždění odezvy potvrzení	T2	t_{ACK}	2		3	ms

Tab. A.3: Režim T - Detailní parametry vysílače [14]

Charakteristika	Režim/ Třída	Sym	Min	Typ	Max	Jednotka
Citlivost (BER 1 z 10 ²) nebo míra přijímání nad 80%	H_R	P_O	-100	-105		dBm
Blokovací výkon	L_R		3			Třída
Blokovací výkon	M_R		2			Třída
Blokovací výkon	H_R		2			Třída
Přijatelná tolerance rychlosti čipů hlavičky ostatních	T1,T2	f_{chip}	88	100	112	kcps
Přijatelná tolerance rychlosti čipů hlavičky ostatních	T1,T2	Df_{chip}	-2	0	2	%
Čipová rychlost (měřiče)		f_{chip}		32,768		kcps
Přijatelná tolerance čipové rychlosti (měřiče)		Df_{chip}	-2	0	2	%

Tab. A.4: Režim T - Detailní parametry přijímače [14]

A.3 Detailní parametry režimu R

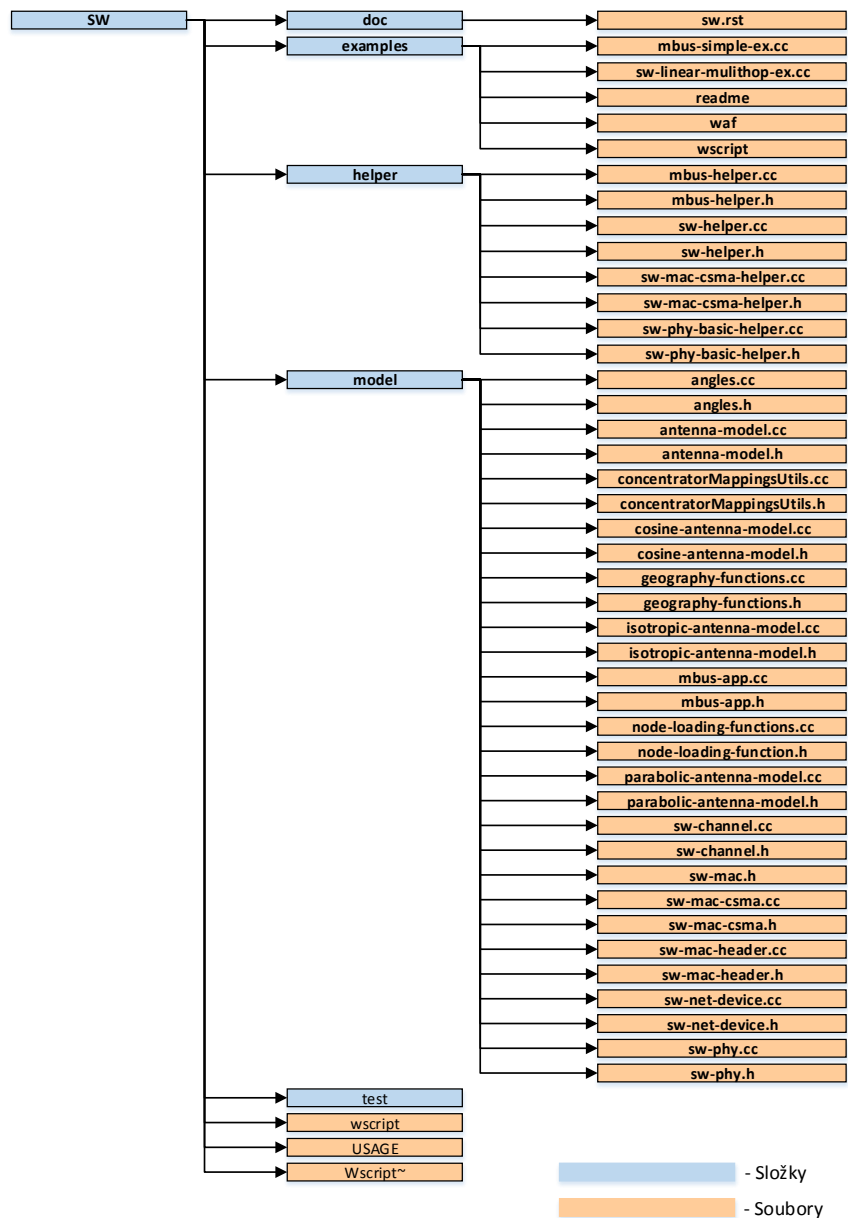
Charakteristika	Režim	Sym	Min	Typ	Max	Jednotka
Střední frekvence (ostatní)				868,330		MHz
Střední frekvence (měřič)				$868,030 + n \cdot 0,06$		MHz
Frekvenční tolerance			-17	0	17	kHz
Odchyška FSK (měřiče/ostatní)			± 8	± 6	± 2	kHz
Rychlost přenosu čipů buzení a komunikace				4,8		kcps
Tolerance rychlosti přenosu čipů buzení a komunikace			-1,5	0	1,8	%
Kolísání digitálních bitů					± 15	us
Rychlost dat (Manchester)				$f_{chip} \cdot 1/2$		bps
Délka úvodní sekvence včetně bit/byte synchronizace, v obou směrech			96			čipů
Délka závěrečné (trailer) sekvence			2		8	čipů
Zpoždění odezvy (ostatní)		t_{RO}	3		50	ms
Zpoždění odezvy (měřiče)		t_{RM}	10		10000	ms

Tab. A.5: Režim R - Detailní parametry vysílače [14]

Charakteristika	Třída	Sym	Min	Typ	Max	Jednotka
Citlivost ($BER 1 z 10^2$) nebo míra přijímání nad 80%	H_R	P_O	-105	-110		dBm
Blokovací výkon	L_R		3			Třída
Blokovací výkon	M_R		2			Třída
Blokovací výkon	H_R		2			Třída
Přijatelný rozsah rychlosti čipů		f_{chip}	4,7	4,8	4,9	kcps
Přijatelná tolerance rychlosti čipů hlavičky ostatních		Df_{chip}	-0,2	0	0,2	%

Tab. A.6: Režim R - Detailní parametry přijímače [14]

B TABULKA ADRESÁŘOVÉ STRUKTURY SÍŤOVÉHO MODULU NS-3



Obr. B.1: Adresářová struktura síťového modulu

C OBSAH PŘILOŽENÝCH DVD

K diplomové práci jsou přiloženy dvě DVD, obsahující VMware obraz operačního systému Ubuntu, ve kterém je nainstalováno vše potřebné ke spuštění aplikace. Dále je na prvním DVD uložena složka Files, obsahující upravené soubory aplikace, nutné nakopírovat do příslušných složek uvnitř systému Ubuntu.

Návod pro spuštění aplikace:

1. Z obou DVD rozbalte soubor: VMware - FEKT NS-3.13 WM-BUS.part0X.rar
2. Obraz následně otevřete v programu VMware a spusťte systém Ubuntu.
3. Ze složky Files na DVD 1 zkopírujte soubor `SimpleApplication.cc` do adresáře `\Home- \ns-allinone-3.13\ns-3.13\scratch`
4. Ze složky Files na DVD 1 zkopírujte složku `sw` do adresáře `\Home\ns-allinone-3.13\ns3.13\src`
5. Spusťte terminál a zadejte do něj:
 - (a) `cd ns-allinone-3.13/ns-3.13/`
 - (b) `./waf --run SimpleApplication`
6. Pro zobrazení informací o argumentech nastavujících parametry spuštění aplikace použijte příkaz `./waf --run "SimpleApplication --PrintHelp"`