

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

PROTOKOL ZIGBEE PRO BEZDRÁTOVÉ SENZOROVÉ SÍTĚ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

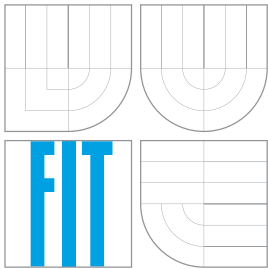
AUTHOR

LUKÁŠ BARTEK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

PROTOKOL ZIGBEE PRO BEZDRÁTOVÉ SENZOROVÉ SÍTĚ

ZIGBEE PROTOCOL FOR WIRELESS SENSOR NETWORKS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ BARTEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. FRANTIŠEK ZBOŘIL, Ph.D.

BRNO 2008

Zadání bakalářské práce

Řešitel: **Bartek Lukáš**
Obor: Informační technologie
Téma: **Protokol ZigBee pro bezdrátové senzorové sítě**
Kategorie: Počítačové sítě

Pokyny:

1. Nastudujte komunikační protokol 802.15.4 ZigBee a problematiku komunikace v bezdrátových senzorových sítích.
2. Vyhledejte nástroje pro simulaci tohoto protokolu a na jednoduchých modelech si vyzkoušejte jeho činnost.
3. Navrhněte modely pro demonstraci předávání senzorických dat a jejich propagaci v síti.
4. Proveďte experimenty s tímto modelem a vyhodnoďte vhodnost tohoto simulačního nástroje pro pedagogické a vývojové účely.

Literatura:

- Haenselmann, T.: Sensor Networks, GFDL Wireless Sensor Network textbook, 2006
- ZigBee Specification, Zigbee Standards Organization, 2006

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zbořil František, Ing., Ph.D., UITS FIT VUT**
Datum zadání: 1. listopadu 2007
Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Lukáš Bartek**
Id studenta: 79110
Bytem: Nová 177, 664 51 Jiříkovice
Narozen: 20. 05. 1984, Brno
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

**Článek 1
Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: Protokol ZigBee pro bezdrátové senzorové sítě
Vedoucí/školitel VŠKP: Zbořil František, Ing., Ph.D.
Ústav: Ústav inteligentních systémů
Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

tištěné formě	počet exemplářů: 1
elektronické formě	počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2 **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3 **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....

Autor

Abstrakt

Tato práce se zabývá průzkumem standardu ZigBee - jeho vlastnostmi, možnostmi použití a testováním jeho implementace v simulátorech. Práce je rozdělena do čtyřech částí.

První část rozebírá strukturu protokolového zásobníku standardu IEEE 802.15.4 a ZigBee, způsoby komunikace, zabezpečení a vývoj těchto standardů.

Druhá část popisuje vlastnosti jednotlivých dostupných simulátorů.

Ve třetí části je návod na vytvoření prostředí pro simulaci ve zvoleném simulátoru.

Poslední, čtvrtá část, předvádí praktické provedení simulace. Popisuje její přípravu, průběh a interpretaci výsledků. Výsledkem práce je zhodnocení testovaného simulátoru pro výukové a vývojové účely.

Klíčová slova

ZigBee, bezdrátové sensorové sítě, IEEE 802.15.4, OMNET++, Castalia, simulace, nod, sensor, propagace dat, agregace dat

Abstract

The aim of this work is to analyze ZigBee standard - its characteristic, potential use and testing its implementation in simulators. This work consists of four sections.

The first section analyses the structure of IEEE 802.15.4 and ZigBee protocol stack means of communications, security and development of this standard.

The second section describes features of available simulators.

The third section is the manual for creating simulation environment in the selected simulator.

The final section shows practical preparation, process and result interpretation.

The goal of this project is evaluation of tested simulator for education and development.

Keywords

ZigBee, wireless sensor networks, IEEE 802.15.4, OMNET++, Castalia, simulation, nod, sensor, data propagation, data aggregation

Citace

Lukáš Bartek: Protokol ZigBee pro bezdrátové sensorové sítě, bakalářská práce, Brno, FIT VUT v Brně, 2008

Protokol ZigBee pro bezdrátové senzorové sítě

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Františka Zbořila, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Lukáš Bartek
9. května 2008

Poděkování

Děkuji panu Ing. Františkovi Zbořilovi, Ph.D. za konzultace, připomínky a návrhy, které ochotně poskytoval v době vypracovávání této práce.

© Lukáš Bartek, 2008.

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

Obsah

1	Úvod	2
2	Pojem ”ZigBee”	3
2.1	Struktura ISO/OSI modelu protokolu Zigbee	4
3	Standard IEEE 802.15.4/ZigBee	5
3.1	Fyzická vrstva	5
3.2	MAC vrstva	6
3.3	Síťová vrstva protokolu ZigBee	10
3.4	Bezpečnostní vrstva protokolu ZigBee	12
3.5	Další vývoj standardu	12
4	Simulátory pro IEEE 802.15.4/ZigBee	13
4.1	Simulátor OPNET Modeler	13
4.2	ZigBee modul do simulátoru ns2	14
4.3	Implementace protokolového zásobníku v TinyOS	14
4.4	Simulátor OMNET/OMNET++	14
5	Simulátor OMNET++/Castalia	16
5.1	Struktura nodu v simulátoru Castalia	16
5.2	Moduly nodu a jejich parametry	17
6	Model agregace dat v simulátoru OMNET++/Castalia	21
6.1	Nastavení parametrů modulů	22
6.2	Spuštění a běh simulace	24
6.3	Výsledek simulace	26
6.4	Možnosti dalšího rozšíření	26
7	Závěr	28
A	Seznam použitých zkratk	30

Kapitola 1

Úvod

Bezdrátové sítě se stávají stále oblíbenější jak v oblasti spotřební elektroniky, tak i v oblasti průmyslu. Pojem bezdrátová radiová komunikace zahrnuje dnes již velké množství různých standardů s různou složitostí, dosahem, rychlostmi přenosu dat a zabezpečením.

Pro svou provozní nenáročnost a jednoduchou instalaci se v průmyslu nahrazují klasické drátové senzory bezdrátovými. Pro přenos dat z těchto senzorů však nejsou zcela vyhovující stávající technologie bezdrátového přenosu, jako je GPRS (General Packet Radio Service), Wi-Fi (Wireless Fidelity) nebo Bluetooth. Požadavky na přenos dat jsou odlišné – je třeba přenášet malé objemy dat v delších časových intervalech. Senzory navíc musí být schopny bezúdržbově odesílat data po dlouhou dobu. Často bývají napájeny bateriově a není neobvyklé, že výměna baterie po umístění senzoru již není možná. Hlavním aspektem těchto zařízení tedy musí být velmi nízká spotřeba energie. Tomuto požadavku se přizpůsobují i protokoly pro přenos dat a řídicích informací. Proto je čím dále populárnější protokol IEEE 802.15.4 s nadstavbou ZigBee, vyvinutý speciálně pro tyto účely.

Standart IEEE 802.15.4/ZigBee napomáhá rozšířit aplikace senzorových sítí do oblastí automatizace, sensoriky, spotřební elektroniky a zabezpečovacích zařízení.

Kapitola 2

Pojem "ZigBee"

Komunikační technologie Zigbee je nízkorychlostní standard bezdrátové komunikace, zaměřený především na oblasti automatizace a řídicí techniky. Patří do skupiny standardů pro bezdrátovou radiovou komunikaci, viz tabulka 2.1. Není však zaměřen jako konkurence stávajících, spíše jako doplněk pro oblasti, ve kterých dosavadní technologie nemohou být vhodně nasazeny. ZigBee byl primárně určen pro průmyslovou komunikaci. První verze protokolu byla vydána roku 2004 skupinou ZigBee Alliance [2], která sdružuje významné podniky v elektronickém průmyslu a vývoji. Od té doby byla již uvolněna verze ZigBee2006 a připravuje se vydání třetí verze. Mezi nejvýznamější nadnárodní firmy patřící do ZigBee Alliance patří Freescale Semiconductors, Honeywell, Mitsubishi Electric, Samsung, Philips a Motorola.

Obchodní jméno Standard	GPRS GSM	Wi-Fi 802.11	Bluetooth 802.15.1	ZigBee 802.15.4
Aplikační zaměření	Hlas a data	Web, Email, Video	Náhrada za kabel	Monitorování a řízení
Systémová paměť	16 MB	1 MB	250 kB	4–32 kB
Životnost baterií (dny)	1–7	0.5–5	1–7	100–1000
Max. velikost sítě (uzly)	1	32	7	65000
Přenosová rychlost (kb/s)	64–128	11000	720	20–250
Komunikační dosah (m)	1000	1–100	1–10	1–100
Výhody	dosažitelnost, kvalita	rychlost, flexibilita	cena, jednoduchost	spolehlivost, výkon/cena

Tabulka 2.1: Technologie pro bezdrátové sítě

Některé konkrétní příklady využití [11]:

- sběr dat ze senzorů, náhrada za RS-232 komunikační protokol
- pohyblivé senzory a zařízení
- zabezpečovací zařízení
- bezdrátová uživatelská rozhraní (displeje, klávesnice, myši)

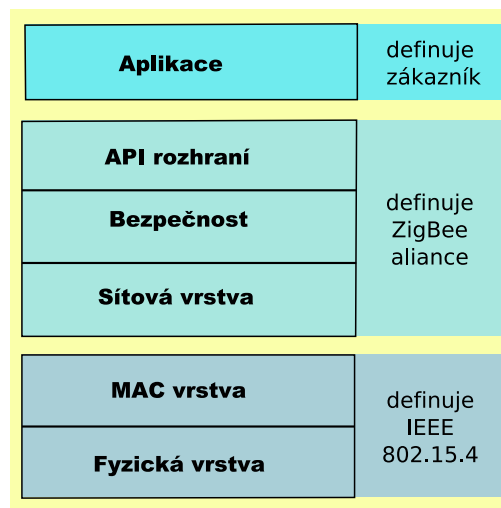
- ovládání topení, klimatizace a osvětlení
- zdravotnické pomůcky

2.1 Struktura ISO/OSI modelu protokolu Zigbee

Referenční model standardu Zigbee vychází ze sedmivrstvého modelu ISO/OSI (International Standards Organization / Open System Interconnection) (viz obrázek 2.1). Používá z něj však pouze ty vrstvy, které jsou významné pro dosažení požadované funkčnosti. Dvě nejnižší vrstvy, fyzická (PHY) a podvrstva spojové vrstvy (MAC), jsou definovány standardem IEEE 802.15.4. Tento standard je použit kvůli schopnosti nejnižší bitové chybovosti při přenosu v prostředí s vysokou úrovní šumu.

Nad vrstvou MAC (Media Access Control) je umístěna síťová vrstva. Vyskytuje se však pouze u plně funkčních zařízení (tato zařízení dokáží mimo získávání dat pracovat jako router; opakem jsou zařízení s omezenou funkcionalitou sloužící pouze pro získávání dat). Bezpečnostní vrstvu a vrstvu API rozhraní poskytuje tzv. aplikační vrstva. Tato část standardu je vytvořena ZigBee aliancí [2].

Nejvyšší vrstva (aplikace) již není definována žádným standardem a její návrh záleží na koncovém uživateli.



Obrázek 2.1: ISO/OSI model protokolu ZigBee

Kapitola 3

Standard IEEE 802.15.4/ZigBee

Protokol ZigBee je neoddělitelně spojen s protokolem IEEE 802.15.4, který mu poskytuje spodní vrstvy – fyzickou a MAC. Pro další popis můžeme předpokládat pod pojmem ZigBee protokol i tuto spodní část ISO/OSI modelu. Toto zjednodušení je používané i v odborné literatuře a vychází z toho, že pro ZigBee není definována jiná spodní část než IEEE 802.15.4. V této kapitole bude vysvětlena funkce všech vrstev protokolu podle obrázku 2.1 až po vrstvu API rozhraní. Také je zde objasněno jak dojde k inicializaci sítě a jakým způsobem probíhá předávání dat mezi prvky sítě (nody).

3.1 Fyzická vrstva

Fyzická vrstva je odpovědná za vysílání a příjem dat s využitím jistého radiového kanálu, modulace a dalších přenosových parametrů [10],[6]. IEEE 802.15.4 podporuje tři frekvenční pásma. Existují rozdílné definice radiových pásem v Americe a Evropě:

1. pásmo ISM 2,4 GHz, 16 kanálů, přenosová rychlost 250 kb/s, definováno globálně
2. pásmo 915 MHz, 10 kanálů, přenosová rychlost 40 kb/s, definováno pro Ameriku
3. pásmo 868 MHz, 1 kanál, přenosová rychlost 20 kb/s, definováno pro Evropu

Protokol umožňuje dynamický výběr kanálu, skenování dostupných podporovaných kanálů na přítomnost přenášených rámců, detekci výkonu vysílače a zjištění kvality linky. Fyzická vrstva plní tyto úkoly:

1. aktivace a deaktivace radiového transceiveru. Transceiver může pracovat ve třech režimech – vysílání, příjem a spánek. Vysílač je zapnut nebo vypnut na požadavek MAC vrstvy.
2. detekce energie (ED). Změření síly přijímaného signálu na určitém kanálu. Nejedná se o identifikaci ani o dekodování signálu. Toto měření je typicky použito na síťové vrstvě jako část algoritmu pro výběr kanálu k určení, zda kanál je volný nebo již používaný.
3. indikace kvality linky (LQI). Měření zjišťuje poměr síla/kvalita signálu.
4. výběr volného kanálu (CCA). Tato operace slouží pro zjištění stavu média (volné nebo obsazené). CCA má 3 operační módy:

detekce energie : stav busy je ohlášen při překročení určité energetické úrovně naměřeného signálu

detekce nosné : stav busy je ohlášen při detekci signálu s modulací podle IEEE 802.15.4

detekce nosné současně s detekcí energie : kombinace obou předchozích podmínek musí být splněna pro oznámení stavu busy

Pro přenos se datový signál moduluje metodou OQPKS a vzduchem se přenáší prostřednictvím techniky rozprostření spektra DSSS. Pro přístup na kanál vysílač používá metodu CSMA/CA. Dosah vysílače je přibližně 10–50 metrů v závislosti na lokálních podmínkách šíření signálu. Existují již komerčně dostupné nody splňující tyto specifikace, například výrobky platformy MICA s čipem CC2420 [3].

3.2 MAC vrstva

MAC podvrstva protokolu poskytuje rozhraní mezi fyzickou vrstvou a vyššími vrstvami protokolu. Je odvozena od standardu bezdrátových sítí IEEE 802.11 s některými úpravami, především pro snížení počtu kolizí při vysílání [8]. MAC protokol podporuje dva operační módy, které mohou být zvoleny koordinátorem sítě:

beacon-enabled mód : beacons jsou periodicky generovány k synchronizaci připojených zařízení a pro identifikaci PAN (Personal Area Network). Beacon rámeček je prvním rámečkem sady po sobě jdoucích rámečků (zvané superrámeček). Přenosy dat mezi nody jsou uskutečňovány v době trvání tohoto superrámečku. Pokud koordinátor zvolí beacon-enabled mód, začne používat superrámeček pro správu připojených zařízení.

non beacon-enabled mód : každé zařízení může kdykoliv vyslat svá data použitím neslotovaného CSMA/CA mechanismu. Není používána struktura superrámečku. Pokud koordinátor zvolí non beacon-enabled mód, nejsou použity beacons ani superrámeček.

Komunikaci mezi dvěma sousedními zařízeními (uzly sítě) probíhá prostřednictvím rámečků. Jsou definovány čtyři typy komunikačních rámečků využívané buď pro přenos užitečných informací, nebo k servisním účelům (sestavení spojení, řízení, správa sítě):

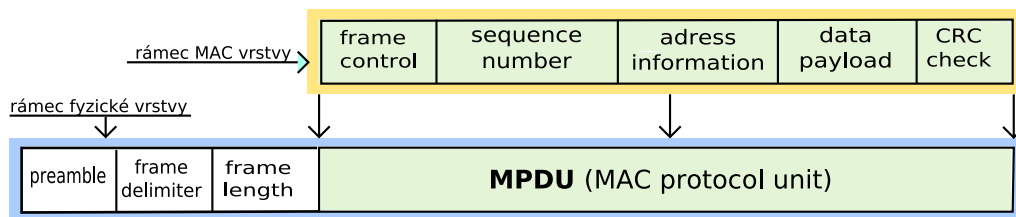
Data Frame : rámeček pro přenos užitečné informace

Acknowledgement Frame : rámeček využívaný pro potvrzovanou komunikaci.

MAC Command Frame : rámeček pro centralizovanou konfiguraci, nastavení a řízení klientských zařízení v síti.

Beacon Frame : rámeček k synchronizaci zařízení v síti, využíván hlavně při konfiguraci sítě v módu v němž umožňuje uvádění klientských zařízení do spánkového režimu s extrémně nízkou spotřebou.

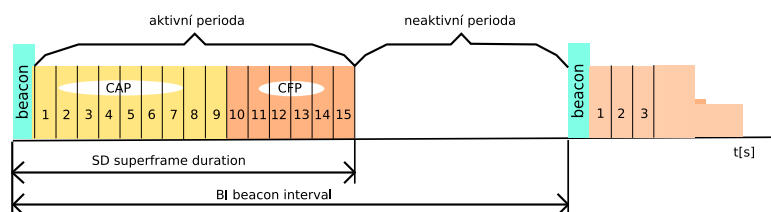
Na obrázku 3.1 je blokově zobrazen datový rámeček MAC vrstvy a jeho transformace do paketu MPDU (Message Protocol Data Unit) fyzické vrstvy. MPDU obsahuje mimo přenášených dat i informaci o adrese příjemce, pořadové číslo daného paketu a řídicí informace.



Obrázek 3.1: Datový rámeček MAC vrstvy IEEE 802.15.4

Struktura superrámce

Superrámec trvá po dobu ohraničenou příchodem dvou následujících beacon rámečků. První z těchto beacon rámečků je součástí aktuálního superrámce, další beacon patří již následujícímu superrámci. Doba trvání celého superrámce se značí BI (Beacon Interval). Superrámec je složen z aktivní části SD (Superframe Duration), jež je rozdělena do 16 stejných časových slotů. Za aktivní částí může následovat neaktivní část, která trvá až do začátku dalšího superrámce, viz obrázek 3.2.



Obrázek 3.2: Struktura superrámce

Aktivní část superrámce je složena ze tří částí:

Beacon : beacon je úvodní rámeček, přenášený bez použití techniky CSMA/CA na začátku slotu 0. Obsahuje informace jako jsou adresní pole, specifikace superrámce, přidělení GTS polí a další informace pro správu.

CAP : startuje bezprostředně ze beacon rámečkem a končí před začátkem CFP (pokud je používána – jinak CAP končí na konci aktivní periody). Všechny přenosy za trvání CAP periody jsou prováděny přes přístupovou metodu CSMA/CA. Zařízení, které nestihne vyslat svá data během trvání CAP, musí vysílání odložit na dobu dalšího vysílání superrámce.

CFP : startuje bezprostředně po skončení CAP a musí skončit před příchodem dalšího beacon rámečku. Všechny GTS sloty (může jich existovat 0–7), které jsou alokovány PAN koordinátorem, jsou umístěny v CFP bezprostředně za sebou. Každý GTS slot může obsadit několik základních časových slotů. CFP se tedy rozšiřuje a zužuje v závislosti na celkové délce všech GTS, nelze ji však rozšířit natolik, aby doba CAS periody klesla pod danou minimální mez ($aMinCAPLength$). CFP přenosy nesoupeří o přístup na médium.

Přístupová metoda CSMA/CA

IEEE 802.15.4 definuje 2 verze mechanismu CSMA/CA:

- slotovaný CSMA/CA používaný v beacon-enabled módu
- neslotovaný CSMA/CA používaný v non beacon-enabled módu

V obou případech je CSMA/CA založen na odloženém vysílání (back-off period) [6]. Back-off perioda je základní časová jednotka MAC protokolu a přístup na médium se může vyskytnout pouze na hranicích této periody. Ve slotované CSMA/CA jsou hranice back-off periody zarovnané s hranicemi slotů v superrámci, zatímco v neslotované CSMA/CA jsou hranice back-off periody každého zařízení nezávislé na ostatních v PAN. Přístup na médium je řízen podle stavu tří proměnných:

NB : udává kolikrát byl CSMA/CA algoritmus požádán o odložení vysílání při pokusu o přístup na aktuální kanál. Před každým pokusem o přenos datového rámce je vygenerován pseudonáhodně dlouhý časový interval, který se během nečinnosti na médium postupně snižuje. Vysílání začne až po dosažení nulové hodnoty. Pokud se během odpočtu změní stav na médium (jiné zařízení začalo vysílat), zastaví se i tento odpočet a navýší se počet odložení vysílání.

CW : udává počet back-off period, po kterých je kanál označen za volný a může na něm být vysíláno.

BE : udává kolik back-off period má zařízení čekat při detekci cizího vysílání, než znovu ověří aktivitu na kanále.

Spuštění a provoz PAN

PAN může být vytvořena pouze po vykonání skenu, zjišťujícího aktivitu a úroveň energie na kanále. PAN koordinátor zvolí jednoznačné číslo identifikující PAN (PAN identifikátor) takový, že nekoliduje s dalšími PAN v dosahu. Jakmile je PAN spuštěna, musí její koordinátor generovat a rozesílat řídicí rámce, spravovat asociaci a deasociaci ostatních zařízení do PAN. Beacon rámce a superrámec může generovat pouze jediné zařízení v PAN a to koordinátor.

Pokud se zařízení chce připojit k existující síti, musí se asociovat do již vytvořené PAN. Proces připojování začíná aktivním nebo pasivním skenem kanálu. Pasivní sken pouze shromažďuje vysílané rámce od ostatních zařízení a na jejich základě si udělá obraz o struktuře sítě. Aktivní sken přímo vyšle dotaz na koordinátora s požadavkem o asociaci. Pasivní, stejně jako aktivní sken umožňuje zařízení lokalizovat koordinátora vysílajícího beacon rámce v dané pokryté oblasti POS (Personal Operating Space). Výsledky skenu jsou poté použity pro výběr vhodné PAN charakterizované fyzickým kanálem (phyCurrentChannel), identifikátorem sítě (macPANid), rozšířenou a krátkou adresou koordinátora (macCoordExtendedAddress, macCoordShortAddress). Všechny tyto atributy jsou vyžadovány vyšší vrstvou (síťovou, aplikační) k dokončení asociace. Koordinátor nemusí povolovat asociaci dalších zařízení do sítě, pokud však koordinátor vyhoví požadavku na asociaci, odešle zařízení zpět potvrzovací rámec s identikačními údaji o sobě. Po přijetí rámce se zařízení podle těchto parametrů nastaví a stává se součástí sítě.

Proces odpojení zařízení [8] může být inicializován buď od koordinátora nebo od zařízení:

disasociace inicializovaná koordinátorem : pokud koordinátor chce odpojit jedno ze svých zařízení, pošle mu příkaz „disassociation notification“. Jakmile zařízení přijme tento příkaz, mělo by poslat zpět potvrzení. I když potvrzení koordinátor neobdrží, považuje zařízení za odstraněné a smaže si všechny záznamy o něm.

disasociace inicializovaná zařízením : zařízení pošle koordinátorovi příkaz na svoje odpojení. Po přijetí koordinátor odstraní všechny záznamy na toto zařízení a pošle mu potvrzení o odebrání ze seznamu. I pokud zařízení neobdrží potvrzení, považuje sebe za odpojené a odstraní ze svých záznamů všechny údaje o koordinátorovi.

Synchronizace prvků v síti

Standard definuje mechanismy pro synchronizaci koordinátorů s jejich asociovanými zařízeními. Tato procedura je obzvláště důležitá v beacon-enabled módu, kde každé asociované zařízení musí synchronizovat své přenosy s přenosy beacon rámců od svého koordinátora. Tedy pro síť s podporou beacon rámců je synchronizace zajišťována pomocí dekódování obsahu přijatých beacon rámců. Zařízení získává synchronizaci pouze z beacon rámců obsahujících identifikační číslo jejich koordinátora (macPANid). Pokud identifikační číslo neodpovídá, není tento rámeček zpracováván. Pro PAN v non beacon-enabled módu je synchronizace poskytována dotazováním se koordinátora. Problém se synchronizací vznikne pokud zařízení není schopno přijmout beacon rámeček od svého koordinátora nebo pokud selže stanovený počet pokusů o získání rámce. Zařízení pak předpokládá, že se stalo tzv. sirotkem, resetuje MAC vrstvu a pokusí se vyhledat dostupné sítě ve svém okolí.

Vysílání a příjem dat

Způsob přenosu dat závisí na operačním módu PAN. V beacon-enabled PAN zařízení, jenž chce vysílat data, musí respektovat časový průběh superrámce s použitím CSMA/CA v CAP periodě nebo využít sloty GTS v CFP periodě. V non beacon-enabled módu zařízení jednoduše zahájí vysílání za použití neslotovaného CSMA/CA. Každé zařízení může přijmout vysílání ze všech zařízení kompatibilních se standardem a používajících stejný kanál, nacházejících se v dané oblasti (POS). MAC podvrstva přijímá pouze rámce, které splní následující požadavky:

- typ rámce musí být platným typem
- pokud je rámeček beacon rámečkem, musí obsahovat totožné macPANid, jaké má uložen nod
- pokud je uvedena cílová adresa, musí odpovídat adrese přijímajícího zařízení, nebo být rovna 0xFFFF (pak jde o broadcast)
- pokud je vyplněna pouze cílová adresa, rámeček smí být přijat pouze koordinátorem, který zprostředkuje jeho doručení

GTS alokace a řízení přenosu

GTS je část superrámce, která je vyhrazena exkluzivně pro dané zařízení. Umožňuje tomuto zařízení přístup na médium bez soupeření v době CFP. GTS může být alokován pouze PAN koordinátorem a musí být použit pouze pro komunikaci mezi PAN koordinátorem a

zařizováním. GTS musí být alokován před použitím a může být dealokován kdykoliv v závislosti na uvážení koordinátora nebo zařízení, které má slot přidělené.

Za přidělování GTS je zodpovědný koordinátor, který musí mít dostatek prostředků pro správu až 7 GTS [8]. Pro každý GTS si koordinátor uchovává startovací slot (starting slot), délku (length), směr přenosu dat (direction) a adresu asociovaného zařízení (associated device address). Všechny tyto údaje musí zařízení předat koordinátorovi při požadavku na alokaci GTS slotu. Každé zařízení může alokovat nejvíce jeden slot v každém směru současně. Při požadavku o alokaci nebo dealokaci GTS slotu vyšle zařízení svému koordinátorovi příkaz „GTS request“. Tento požadavek se skládá z pole parametrů, obsahujícího GTS Length (počet požadovaných slotů), GTS Direction (určení směru přenosu), Characteristic Type (rozlišuje požadavek na alokaci a dealokaci). Po přijetí GTS požadavku PAN koordinátor zkontroluje zda má dostupné zdroje a poté pošle potvrzovací rámeček. Zdroje jsou dostupné, pokud maximální počet GTS nebyl obsazen a alokace GTS neomezují minimální délku CAP periody. Přenos dat poté proběhne na základě požadavku vyšší vrstvy na odeslání dat přes GTS. Odeslání musí proběhnout během trvání přiděleného slotu.

3.3 Síťová vrstva protokolu ZigBee

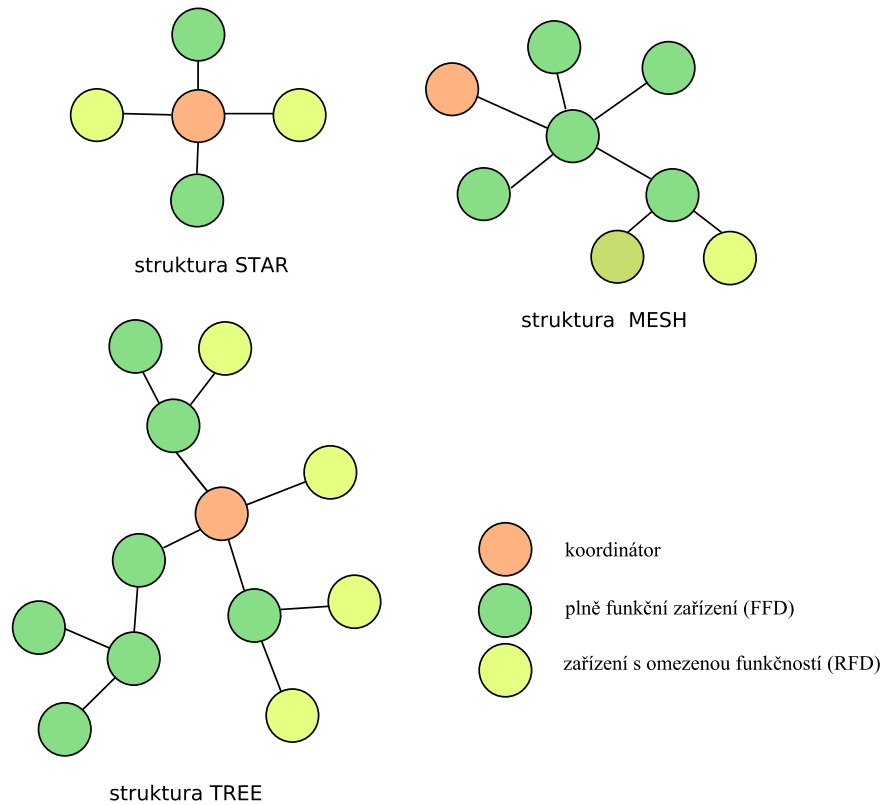
Standard IEEE 802.15.4 využívá pro adresaci jednotlivých zařízení adresovací kódy, které mohou být dlouhé (64 bit) nebo zkrácené (16 bit). Dlouhá adresa umožňuje jednoznačnou identifikaci zařízení ve všech sítích, zatímco zkrácená adresa umožňuje jednoznačně identifikovat zařízení pouze v jedné síti, což dává možnost umístit až 65536 zařízení do jedné sítě. Každá sestavená síť je dále identifikována 16bitovým identifikátorem PAN ID, který slouží pro rozlišení překrývajících se sítí, v případě že se v daném prostoru nachází více sítí standardu IEEE 802.15.4. Každou síť s jedinečným PAN ID zakládá a spravuje koordinátor (centrální stanice). Ostatní stanice pracují ve funkci směrovače nebo koncového zařízení.

Podle funkčnosti se zařízení dělí na plně funkční zařízení FFD (Full Function Device), která mohou plnit funkci koordinátora, směrovače nebo koncového zařízení a na redukováná zařízení RFD (Reduced Function Device), která mohou fungovat pouze jako koncová zařízení. Standard Zigbee předpokládá možnost vytvoření 3 typů síťové topologie. Základní je topologie typu hvězda (star topology), v níž je vždy definováno jedno zařízení, které přebírá funkci koordinátora sítě a ostatní zařízení působí ve funkci koncových zařízení. V topologii typu strom (tree topology) slouží jedno zařízení jako koordinátor a ostatní jako plně funkční nebo koncová zařízení. Na rozdíl od topologie hvězda však nemusí všechna zařízení komunikovat přímo s koordinátorem, ale mohou využít jiné plně funkční zařízení ve funkci směrovače. Díky tomu je možné zvětšit vzdálenost mezi koncovým zařízením a koordinátorem. Poslední používanou topologií je síť (mesh topology), jenž kombinuje vlastnosti topologií strom a hvězda. Umožňuje složit síť libovolným způsobem a obejít tím omezení hierarchické struktury. Schematicky jsou jednotlivé topologie znázorněny na obrázku 3.3.

Routování na síťové vrstvě

Směrování (routování) je jedna z nejdůležitějších činností síťové vrstvy. ZigBee používá algoritmus tree-routing, neighbour-routing a mesh-routing. V algoritmu tree-routing je umístění zařízení ve stromu jednoznačně definováno jeho adresou. Na základě cílové adresy zdrojové zařízení určí, kterým směrem pošle data. Omezujícími faktory tohoto protokolu jsou:

- maximum potomků na jeden router



Obrázek 3.3: Typy síťové topologie

- maximum dalších routerů na jeden router
- hloubka zanoření (počet úrovní) celé sítě

Neighbour-routing předpokládá použití tabulky sousedů. Pokud je cílové zařízení fyzicky dostupné vysílajícímu (což znamená, že v jeho tabulce sousedů je záznam o zdrojovém zařízení), může mu být zaslána zpráva. Tabulka sousedů obsahuje:

- PAN_id (16 bitová krátká adresa PAN)
- Network_Address (16 bitová krátká adresa zařízení)
- Device_Type (typ zařízení a vztah – rodič, potomek, sirotek, ostatní)
- Depth (hloubka zanoření zařízení)
- Permit_Joining (informace, zda zařízení akceptuje asociační požadavky)
- Logical_Channell (logický kanál, na němž zařízení vysílá)

Mesh-routing pracuje s routovacími tabulkami. Pokud je cílové zařízení uvedeno ve směrovací tabulce, cesta uložená u této položky je použita pro posílání dat na další router v cestě. Pokud však v routovací tabulce není záznam nalezen, uplatní se „route discovery“ proces pro doplnění záznamu. Při tomto procesu síťová zařízení spolupracují na vyhledání a navázání cesty k cíli skrz síť. Nejdříve vysílající zařízení rozešle broadcastem zprávu RREQ.

Zařízení, která zprávu přijmou, zkontrolují zda mají ve své směrovací tabulce záznam pro hledané zařízení. Pokud ne, vytvoří nový záznam a zařízení opět vysílá požadavek RREQ na další sousedy. Jakmile se zpráva dostane až k zařízení, ve kterém je hledaná položka uložena, ukončí se vysílání zprávy RREQ. Opačným směrem se nyní vysílá odpověď RREP až na úroveň dotazujícího se zařízení. To si přidá obdrženou cestu do své směrovací tabulky na příští použití.

3.4 Bezpečnostní vrstva protokolu ZigBee

Během přenosu přes fyzické médium může dojít k chybě. Pro odhalení takovýchto chyb se používá cyklického kódu (CRC, FCS), kdy každý rámec je doplněn o zbytek po dělení polynomem. Zabezpečován je celý rámec včetně záhlaví. Pro zvýšení spolehlivosti je vysílání provedeno technologií DSSS (Direct Sequence Spread Spectrum). Jednotlivé bity jsou zde nahrazeny početnější sekvencí bitů, která se pak vysílá. Signál je tak rozprostřen do větší části spektra a je tak více odolný proti rušení.

Přenášená data lze také zabezpečit proti odposlechu [6]. Standard definuje 3 režimy a to nezabezpečený přístup, přístup na základě práv a zabezpečený přístup. Je-li použit přístup na základě práv, pak síť odmítá rámce od neznámých zařízení. V zabezpečeném režimu mohou zařízení sítě využívat další služby:

- přístup na základě práv
- šifrování dat pomocí algoritmu AES 128b
- použití MIC
- odmítnutí opakujících se rámců

Šifrovací standard AES (Advanced Encryption Standard) bude nahrazen v nových verzích protokolu ZigBee standardem DES (Data Encryption Standard). Výhodou nového šifrování je, že nehrozí útok hrubou silou. MIC je kryptografický kontrolní součet, který je zahrnut do vysílaného rámce. Na přijímací straně se provádí stejná operace a hodnota součtu se porovnává s přijatou. Pokud se zpráva při přenosu změnila, budou se hodnoty lišit a rámec je odmítnut.

3.5 Další vývoj standardu

Praktické používání verze 1.0 ukázalo, že standard obsahuje nedostatky, bránící jeho implementaci především ve větších sítích obsahujících množství zařízení. Vzhledem k tomu, že tyto nedostatky nelze systematicky řešit v rámci původního standardu, jsou zařízení vystavěná na verzi 1.1 zpětně nekompatibilní. Standard 1.1 však ještě neobsahuje všechny požadované změny.

Aktuálně připravovanou verzí je ZigBee Pro. Nová verze již bude zpětně kompatibilní se ZigBee 1.1 a toto bude platit i pro následující verze. Znamená to tedy, že změny v síťové vrstvě jsou již dokončeny a další změny se budou týkat vyšších vrstev.

Na fyzické vrstvě je pro pásmo 2,4 GHz připravovaný standard IEEE 802.15.4a, který umožňuje v pásmu 2,4 GHz definovat kanál s komunikační rychlostí 32 kb/s až 2 Mb/s.

Kapitola 4

Simulátory pro IEEE 802.15.4/ZigBee

Při vytváření softwaru podle specifikace protokolu je vhodné mít možnost sledovat chování programu ještě před jeho přenesením do konkrétního hardwarového zařízení. Stejně tak při výuce potřebujeme sledovat pochody v programu na základě vnějších událostí. Pro splnění těchto potřeb je nutné při vývoji používat simulátory, které se svými vlastnostmi co nejvíce přibližují cílovému zařízení. Pro běžně používané síťové protokoly existuje množství takovýchto simulátorů a to jak komerčních, tak volně šířitelných. K dostupným simulátorům, zaměřeným přímo na bezdrátové senzorové sítě, resp. ZigBee protokol patří:

1. OPNET s přídatným modulem
2. implementace protokolového zásobníku do TinyOS
3. ns2 s přídatným modulem
4. OMNET++ s přídatným modulem

4.1 Simulátor OPNET Modeler

OPNET Modeler (Optimized Network Engineering Tools Modeler) je profesionální graficky orientovaný simulační nástroj vyvíjený firmou OPNET Technologies [9]. Umožňuje tvorbu modelů, vlastní simulaci a vyhodnocení získaných dat. OPNET je použitelný pro simulaci většiny síťových protokolů. Celý systém je složen z několika programových částí. Modeler Wireless Suite je část pro protokoly bezdrátových sítí. Do tohoto nástroje jsou dostupné moduly pro jednotlivé protokoly. Pro simulaci ZigBee protokolu (resp. jeho spodní vrstvy popsané IEEE 802.15.4) je v rámci projektu IPP HURRAY! (Polytechnic Institute of Porto, Portugal) vyvíjen modul „IEEE 802.15.4 OPNET Simulation Model“ [7].

Tento simulační model implementuje:

- fyzickou vrstvu IEEE 802.15.4 pro běh na pásmu 2.4 GHz s rychlostí přenosu 250 kb/s.
- MAC vrstvu podporující beacon-enabled mód a slotovanou CSMA/CA přístupovou metodu na sdílené médium
- modul baterie, počítající spotřebovanou a zbývající energii

- aplikační vrstvu, vytvářející datové rámce pomocí generátoru datového toku
- simulace šumu na pozadí, zpoždění při přenosu, interferencí

Jedná se o kompletní prostředí přesně vyhovující normě IEEE 802.15.4. Modul neza-
hrnuje síťovou vrstvu ZigBee, bylo by tedy nutné ji doimplementovat na aplikační vrstvě.
Vlastní práce s programem je velmi intuitivní, je k dispozici přehledné grafické prostředí
pro všechny činnosti spojené se simulací a zpracováním výsledků. Přestože se jedná o ko-
merční software, existuje i licence pro studijní účely. Podmínky ale nemohly být pro tuto
práci splněny.

4.2 ZigBee modul do simulátoru ns2

ns2 (network simulator) je objektově orientovaný simulátor diskretních událostí. Je na-
psaný v jazyce C++ a uživatelské rozhraní poskytuje přes skriptovací jazyk OTcl. Pro gra-
fické znázornění průběhu simulace používá nástroj Network Animator, napsaný v Tcl/Tk.
S použitím tohoto nástroje byl na univerzitě City Collegue of New York vytvořen modul
„802.15.4 and ZigBee routing Simulation“ [1]. Modul simuluje jak spodní část protokolu
ZigBee (IEEE 802.15.4), tak jeho horní vrstvy. Mimo jiné implementuje následující vlast-
nosti:

- neslotovaná i slotovaná CSMA/CA
- hvězdicová a cluster-tree síťová topologie
- beacon-enabled a non beacon-enabled přenosy
- asociace a deasociace zařízení

4.3 Implementace protokolového zásobníku v TinyOS

TinyOS je jedním z prvních operačních systémů, speciálně navržených pro vestavěná zařízení
[3]. Je napsán v jazyce nesC, což je rozšíření standardního jazyka C. Hlavním znakem
TinyOS je modularita komponent, které spolu navzájem komunikují přes přesně defino-
vaná rozhraní. Implementace komponent je skryta za rozhraním, což umožňuje dosažení
modularity a znovupoužitelnosti. Skupina IPP Hurray! vytvořila implementaci protokolu
IEEE 802.15.4 a ZigBee pro tento operační systém. Je možné ji použít pro testovací účely,
ale i pro reálné nasazení. Pro softwarovou simulaci aplikace běžící pod TinyOS lze použít
simulátor TOSSIM. Pro ladění aplikace běžící již na hardwaru je dostupná vývojová deska
CC2420DBK pro platformu MICA, která umožňuje získat mnohem lepší (reálnější) výsledky
než simulátor (přístup do všech registrů, do paměti, změna stavu komponent, zobrazení
struktury a obsahu všech zachycených rámců).

4.4 Simulátor OMNEST/OMNET++

OMNEST je otevřený, mnohostranný simulátor diskretních událostí s modulární architekту-
rou založenou na komponentách. Pro nekomerční použití existuje jeho varianta OMNET++
[9] s licencí postavenou na GPL. Ačkoliv OMNET++ není sám o sobě síťový simulátor,

začíná se pro simulaci sítí používat nejen ve výzkumných projektech, ale i v průmyslu. OMNET++ byl vytvořen na univerzitě Technical University of Budapest, Department of Telecommunications (BME-HIT). Hlavní využití OMNETu++ spočívá v:

- modelování toku dat telekomunikačními sítěmi
- modelování protokolů
- modelování s využitím front
- modelování multiprocesorových hardwarových systémů
- ověřování zapojení hardwaru
- vyhodnocování výkonostních charakteristik kompletních systémů

Simulační model v OMNET se skládá z hierarchicky uspořádaných modulů. Hloubka zanoření není limitována, což umožňuje reflektovat logickou strukturu modelovaného systému. Moduly komunikují pomocí zasílání zpráv v libovolném formátu. Každý z modulů má určité parametry, které mu uživatel přidělí a které poté specifikují chování modulu. Modul může popisovat např. jednu vrstvu ISO/OSI modelu (fyzická, síťová, aplikační) a vzájemně se propojí s dalšími moduly pomocí bran a portů. OMNET++ též velmi dobře podporuje paralelní simulace.

Pro simulaci bezdrátových sítí máme na výběr z více modulů od různých vývojářů:

- Mobility Framework
Modul je vhodný pro simulaci pevných a mobilních bezdrátových sítí, sensorových sítí, bezdrátových sítí pracujících na více kanálech.
- NesCT
Modul pro podporu simulace provozu bezdrátových nodů pod TinyOS operačním systémem.
- Castalia
Platformově nezávislý simulátor sensorových bezdrátových sítí. Castalia poskytuje velké množství parametrů umožňující detailní přizpůsobení simulovanému standardu. Projekt je aktivní, poslední verze byla uvolněna v únoru 2008 a v každé verzi jsou implementovány nové funkce [5]. Pro další práci byl vybrán právě tento modul.

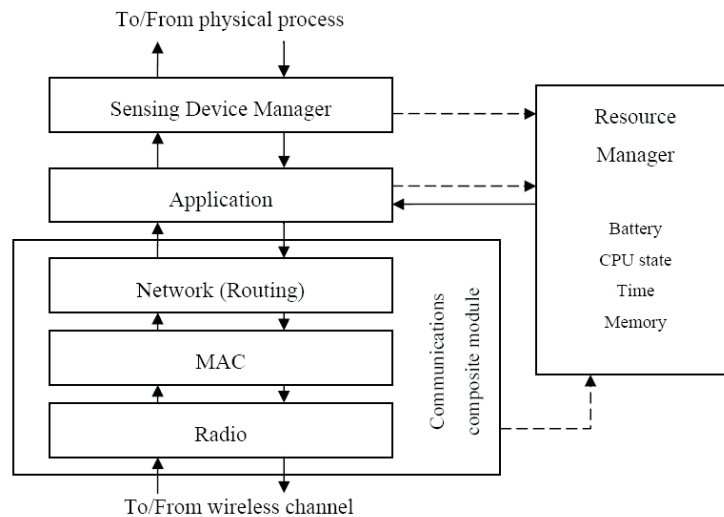
Kapitola 5

Simulátor OMNET++/Castalia

5.1 Struktura nodu v simulátoru Castalia

Obrázek 5.1 zobrazuje interní strukturu nodu, rozdělenou na moduly odpovídající programovým modulům v Castalia. Struktura každého z modulů je posaná pomocí jazyka NED simulátoru OMNET++. Tímto jazykem můžeme snadno definovat rozhraní modulů (vstupní/výstupní brány, umístění ve struktuře ostatních modulů). Kód jazyka NED je uložen v souborech s koncovkou .ned.

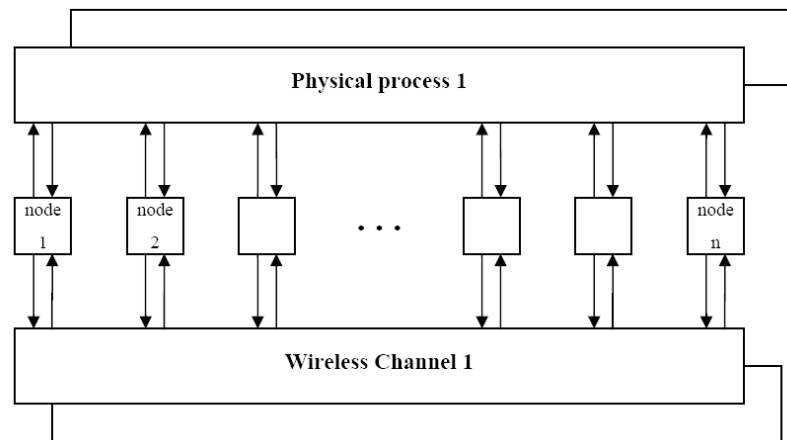
Každý modul v Castalia je reprezentován jedním adresářem, který vždy obsahuje příslušný .ned soubor definující rozhraní. Chování modulu je popsáno pomocí jazyka C++ (soubory s příponou .cc a .h). Celková struktura nodu je dána hierarchickou strukturou .ned souborů. Uživatel obvykle nemá důvod měnit tuto strukturu. Mění pouze parametry jednotlivých modulů a to jednak statickým nastavením před spuštěním simulace, jednak dynamicky při běhu simulace v reakci na změnu okolních podmínek [4].



Obrázek 5.1: Jednotlivý nod v prostředí Castalia

Obrázek 5.2 zobrazuje rozmístění více nodů v určitém prostředí, majícím zvolené fyzikální vlastnosti a umožňujícím nodům vzájemnou komunikaci. Nody nejsou propojeny

přímo, ale přes modul „Wireless Chanell“. Pomocí tohoto modulu lze simulovat fyzikální vlastnosti prostředí, kterým se šíří radiové vlny.



Obrázek 5.2: Nody a jejich propojení v prostředí Castalia

5.2 Moduly nodu a jejich parametry

Jednotlivé moduly nodu potřebují navzájem spolupracovat. Pro zajištění modularity je jejich komunikace zajišťována přes posílání zpráv. Každý modul ke zprávě přikládá datový rámec, který obsahuje parametry této zprávy.

Aplikační modul

Pro popis chování každého nodu musíme editovat soubor popisující aplikační model nodu (viz Application na obrázku 5.1). Tento soubor je souborem pro simulátor OMNET++ a je tedy vytvořen podle jeho pravidel. Obsahuje 4 části:

- sekce definic
- inicializace modulu
- sekce metod pro obsluhu přicházejících zpráv
- závěrečná sekce

Hlavní parametry aplikačního modulu jsou uvedeny v tabulce 5.1.

Komunikační modul (kompozitní)

Komunikační modul je nejpečlivěji modelovaným aspektem v simulátoru Castalia. Je složen ze čtyř submodulů: modulu bezdrátového kanálu (The Wireless Chanell), radio modulu (The Radio), MAC modulu (The MAC) a síťového modulu (The Network - Routing).

1. The Wireless Chanell

modeluje fyzikální prostředí v němž se šíří radiové vlny. Pro pokles síly signálu mezi dvěma nody (vysílačem a přijímačem) používá vztah

Parametr	Popis
node[x].nodeApplication.isSink	určuje, zda daný nod je sink
node[x].nodeApplication.maxAppPacketSize	maximální velikost paketu aplikační vrstvy [bytů]
node[x].nodeApplication.packetHeaderOverhead	maximální velikost hlavičky paketu aplikační vrstvy [bytů]
node[x].nodeApplication.constantDataPayload	celková velikost uživatelských přenášených dat [bytů]

Tabulka 5.1: Parametry aplikačního modulu

$$PL(d) = PL(d_0) + 10 \cdot \eta \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma \quad [dB]$$

$PL(d)$... pokles úrovně signálu na vzdálenosti d

$PL(d_0)$... známý pokles úrovně signálu na referenční vzdálenost d_0

η ... path loss exponent

X_σ ... náhodná proměnná daná Gaussovým rozložením se směrodatnou odchylkou σ

2. The Radio

modeluje vlastnosti typické pro nízkovýkonné transceivery v bezdrátových nodech. Podporuje tedy několik provozních stavů (vysílání, příjem, spánek) s rozdílnou spotřebou energie a s různým spožděním mezi přechody do jiného stavu. Vysílač může pracovat na několika stupních výkonu, je možná detekce vysílání na sledovaném pásmu. Modul se snaží simulovat chování skutečných zařízení (čipy CC1100 nebo CC2420 v TelosB a MICA platformách).

3. The MAC

Medium Access Control modul umožňuje široké nastavení vlastností MAC protokolu.

4. The Network - Routing

specifikuje obecný formát síťového rámce, zahrnující např. hlavičku rámce (zdroj, cíl, poslední skok, následující skok, doba života). Implementovány jsou dva jednoduché routovací protokoly. Prvním z nich je single-tree algoritmus, směřující data z nodů směrem k sinku (sink je kořenový nod stojící na vrcholu stromu). Druhý algoritmus, multipathRing, je robustnější než předchozí. Data z vyšší úrovně posílá také na nižší úroveň směrem k sinku, není však závislý na konkrétním rodiči. Data doručená z vyšší úrovně rozešle všem sousedům na své úrovni a ten soused, který má v dosahu nod na nižší úrovni mu data přepošle. Síťová vrstva je v Castalia na počátku vývoje, v dalších verzích bude rozšířena o protokoly používané ve WSN (Wireless Sensor Networks).

Hlavní parametry kompozitního modulu jsou uvedeny v tabulce 5.2.

Modul senzoru a fyzikálního děje

Modul fyzikálního děje slouží ke generování dat pro snímací zařízení (senzor), který je umístěn v každém nodu. Pro jednoduché simulace lze pouze přiřadit senzoru statickou hodnotu, kterou bude trvale vysílat. Takto se to řeší v obecných simulátorech. V Castalia je tato možnost také dostupná, nicméně je zde snaha o vyšší reálnost. V simulovaném

Parametr	Popis
wirelessChannel.pathLossExponent	určuje „ztrátovost“ prostředí. Hodnota: 2.0 (bezztrátové prostředí) – 6.0 (vysoce ztrátové prostředí)
wirelessChannel.PLd0	známý útlum signálu v referenční vzdálenosti d0 od vysílače [dBm]
wirelessChannel.d	referenční vzdálenost [metrů]
wirelessChannel.collisionModel	způsob modelování interferencí při přenosu signálu 0 žádné kolize 1 jednoduchý kolizní model 2 rozšířený kolizní model
wirelessChannel.sigma	směrodatná odchylka
wirelessChannel.allBidirectionalLinks	pokud true, tak kvalita linky v jednom směru bude stejná i ve směru opačném
node[x].networkInterface.Radio.dataRate	přenosová rychlost vysílače [kb/s]
node[x].networkInterface.Radio.modulationType	0 ideální modulace (žádné chyby přenosu) 1 FSK pro MICA2 motes 2 BPSK, QPSK, OQPSK pro CC2420
node[x].networkInterface.Radio.receiverSensitivity	citlivost přijímače [dBm] a) -98 pro MICA2 b) -95 pro TELOS
node[x].networkInterface.Radio.rxPower	spotřeba při vysílání [mW]
node[x].networkInterface.Radio.listenPower	spotřeba při příjmu [mW]
node[x].networkInterface.Radio.sleepPower	spotřeba při spánku [mW]
node[x].networkInterface.Radio.txPowerLevels	dostupné úrovně výkonu vysílače [dBm], první prvek je výchozí úroveň
node[x].networkInterface.Radio.bufferSize	počet MAC rámců uložitelných do paměti bufferu
node[x].networkInterface.Radio.maxPhyFrameSize	velikost fyzického rámce [bytů]
node[x].networkInterface.Radio.phyFrameOverhead	velikost hlaviček fyzického rámce [bytů]
node[x].networkInterface.Radio.delaySleep2Listen	doba potřebná na přechod mezi stavy spánek-naslouchání [ms]
node[x].networkInterface.Radio.delayListen2Sleep	doba potřebná na přechod mezi stavy naslouchání-spánek [ms]
node[x].networkInterface.Network.maxNetFrameSize	maximální velikost síťového rámce [bytů]
node[x].networkInterface.Network.netDataFrameOverhead	velikost hlaviček v každém síťovém paketu [bytů]
node[x].networkInterface.Network.maxNeighborsTableSize	maximální počet záznamů o sousedních nodech v tabulce sousedů
node[x].networkInterface.Network.maxNumberOfParents	maximální počet rodičů

Tabulka 5.2: Parametry kompozitního modulu

prostoru tak můžeme rozmístit několik zdrojů (např. teploty) na přesně dané souřadnice a přiřadit jim hodnotu (teplotu). Poté rozmístíme nody na jejich souřadnice a přesnou hodnotu (teploty) v místě umístění nodu zjistíme podle vzorce:

$$V(p, t) = \sum_{\text{all sources}} \frac{V_i(t)}{(Kd_i(t) + 1)^a}$$

$V(p, t)$... hodnota fyzikální veličiny v bodě p v čase t

$V_i(t)$... hodnota i -tého zdroje v čase t

$d_i(t)$... vzdálenost bodu p od i -tého zdroje v čase t

K, a ... multiplikativní a útlumová konstanta

Hlavní parametry modulu fyzikálního děje jsou uvedeny v tabulce 5.3.

Parametr	Popis
physicalProcess[0].directNodeValueAssignment	statické přiřazení hodnoty specifickému nodu
physicalProcess[0].multiplicative_k	multiplikativní parameter (k), udávající rozložení hodnoty zdroje v okolí
physicalProcess[0].attenuation_exp_a	attenuation exponent (a), udávající rozložení hodnoty zdroje v okolí
physicalProcess[0].sigma	směrodatná odchylka Gaussova šumu
physicalProcess[0].numSources	počet rozdílných zdrojů, maximálně 5
physicalProcess[0].source_0	řetězec „tuple“, udávající chování zdrojů v čase, místě a hodnotě [čas: Xsouř: Ysouř: hodnota]

Tabulka 5.3: Parametry modulu fyzikálního děje

Hlavní parametry modulu senzoru jsou uvedeny v tabulce 5.4.

Parametr	Popis
node[x].nodeSensorDevMgr.numSensingDevices	počet senzorů na nodu
node[x].nodeSensorDevMgr.pwrConsumptionPerDevice	spotřeba jednoho senzoru [mW]
node[x].nodeSensorDevMgr.maxSampleRates	počet vzorkování za sekundu pro každý senzor

Tabulka 5.4: Parametry modulu senzoru

Modul energetických a výpočetních zdrojů

Modul zavádí účtování pro fyzické zdroje nodu (použitá paměť, výkon CPU, spotřebovaný čas CPU, zbývající energie). Aktuální verze podporuje pouze správu stavu baterií. Hlavní parametry modulu energetických a výpočetních zdrojů jsou v tabulce 5.5.

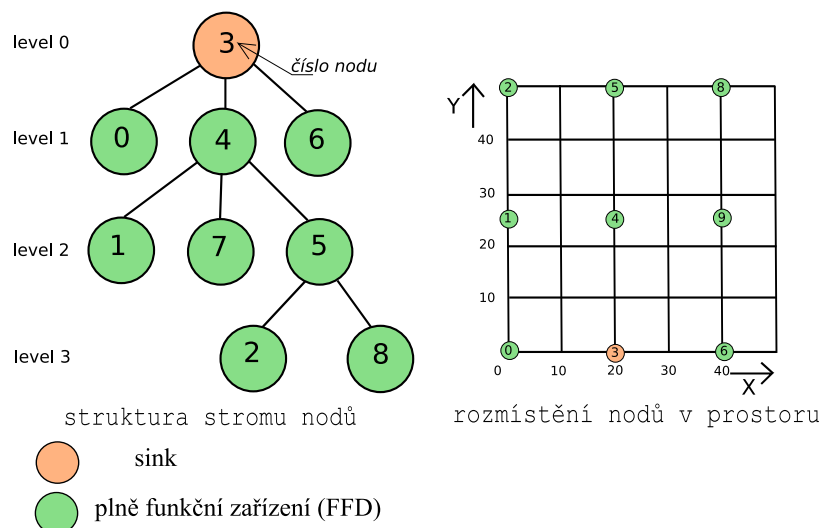
Parametr	Popis
node[x].nodeResourceMgr.initialEnergy	počáteční energie baterií [Joul]

Tabulka 5.5: Parametry energetického modulu a modulu výpočetních zdrojů

Kapitola 6

Model agregace dat v simulátoru OMNET++/Castalia

Ukázková simulace je založena na demonstrační simulaci simpleAggregation. Tato simulace využívá fyzickou, MAC, síťovou a aplikační vrstvu protokolu ZigBee. Agregace obecně znamená seskupení vybrané části určitých entit za účelem vytvoření nové entity. V této simulaci jsou entity tvořeny naměřenými teplotami. Každý rodič obdrží naměřené teploty svých potomků, provede na nich tzv. agregační funkci (v tomto případě určení nejvyšší teploty) a výsledek předá opět svému rodiči. Nody jsou uspořádány do struktury typu tree (obrázek 6.1 vlevo), každý nod má tedy svého rodiče (s výjimkou nodu na nejvyšší úrovni - sinku). Předpokládejme simulaci platformy MICA, používající transceiver CC2420 kompatibilní s fyzickou vrstvou IEEE 802.15.4. Napájení je zajištěno dvěma AA bateriemi. Dále určíme, že na ploše 50 x 50 metrů máme umístěn zdroj tepla (např. ohřívač, topení), který bude v čase měnit svoji polohu a teplotu. Na ploše také máme staticky umístěny nody se senzory (obrázek 6.1 vpravo). Budeme tedy sledovat nejvyšší teplotu v prostoru v daném čase simulace.



Obrázek 6.1: Uspořádání nodů do stromu a jejich rozložení v prostoru při vzorové simulaci

6.1 Nastavení parametrů modulů

Nejdříve zvolíme základní parametry simulace. Cesty k jednotlivým souborům s konfiguracemi jsou uváděny relativně vzhledem k adresáři *Castalia*.

V souboru */Castalia/Simulation/SimpleAgregation/omnetpp.ini* nastavíme parametry vztahující se k délce simulace, počtu a umístění nodů. Zvolíme si manuální rozmístění nodů.

```
sim-time-limit = 50
# délka běhu simulace bude 50 sekund
field_x = 50
# vzdálenost nodů v ose x bude 50 metrů
field_y = 50
# vzdálenost nodů v ose y bude 50 metrů
numNodes = 9
# celkový počet nodů pro simulaci
deploymentType = 4
# manuální umístění nodů na scénu
```

Souřadnice jednotlivých nodů pak určíme manuálně v souboru */Castalia/Simulation/SimpleAgregation/node.location.ini*

```
node[0].xCoor = 0
# souřadnice X nodu 0
node[0].yCoor = 0
# souřadnice Y nodu 0
node[1].xCoor = 0
# souřadnice X nodu 1
node[1].yCoor = 25
# souřadnice Y nodu 1
...
```

Následuje vložení (*include*) konfiguračních souborů pro jednotlivé moduly nodu. Soubor */Parameter_Include_Files/physicalProcess_0_one_source_at_2_38.ini* nastavuje fyzikální prostředí. V prostoru 50 x 50 metrů umístíme zdroj tepla. Tento zdroj se bude během simulace pohybovat v prostoru a přitom bude měnit svoji teplotu. Nody tak budou zachycovat pokaždé jinou teplotu. Vyjimku uděláme u nodu číslo 6, u kterého budeme simulovat poruchu – jeho senzor vždy naměří hodnotu 40°C.

```
physicalProcess[0].description = "Fire"
# popis prostředí – nody jsou umístěny mezi zdroji tepla
physicalProcess[0].inputType = 0
# způsob získání hodnoty teploty pro každý z nodů,
# v tomto případě přímo přidělena následujícím parametrem
physicalProcess[0].directNodeValueAssignment = "(0) 6:40"
# nod číslo 6 bude mít přiřazenu hodnotu 40°C
physicalProcess[0].numSources = 1
# počet zdrojů tepla
physicalProcess[0].source_0 = "0 2 38 10; 10 2 38 160; 20 2 38 60"
# změna zdroje v čase simulace, místě působení a teplotě
```

Soubor */Parameter_Include_Files/resourceMgr_2AAbatteries.ini* nastavuje vlastnosti zdroje energie. Protože simulujeme platformu MICA, napájení budou zajišťovat dvě AA baterie.

```

node[*].nodeResourceManager.initialEnergy = 29160
# energie baterií v Joulech při
# použití 2 AA baterií, podle vztahu
# 1 AA baterie : 1.5V * 2700mAh = 4050 mWh
# 4050 mWh * 3600 sec/hr * 1/1000 = 14580 W/sec = 14580 J
# 2 AA baterie : 2 * 14580 = 29160 J

```

Soubor */Parameter_Include_Files/nodeSensorDevMgr_Temperature.ini* nastavuje vlastnosti senzorů – čidel. Každý senzor bude mít jeden teplotní snímač. Snímače mají také určitou vlastní spotřebu, předpokládejme že 0,02 mW.

```

node[*].nodeSensorDevMgr.numSensingDevices = 1
# počet senzorů na 1 nod
node[*].nodeSensorDevMgr.pwrConsumptionPerDevice = "0.02"
# spotřeba senzoru v mW
node[*].nodeSensorDevMgr.sensorTypes = "Temperature"
# typ=teplotní snímač

```

Soubor */Parameter_Include_Files/WChannel/No_Interference/WChannel_Ideal.ini* nastavuje vlastnosti bezdrátového kanálu. Pro simulaci nastavíme ideální prostředí pro šíření vln. V cestě nestojí žádné překážky, proto nevznikají nežádoucí interference.

```

wirelessChannel.pathLossExponent = 2.0
wirelessChannel.PLd0 = 55
# úroveň Path Loss
# při vzdálenosti d0 [dB]
wirelessChannel.d0 = 1
# referenční vzdálenost d0=1 metr
wirelessChannel.collisionModel = 0
# nenastanou žádné kolize při přenosu
wirelessChannel.allBidirectionalLinks = true
# stejné vlastnosti linky
# v obou směrech, ideální stav

```

Soubor */Parameter_Include_Files/Radio/TelosB_CC2420/radio_CC2420_IDEALmodulation.ini* nastavuje vlastnosti transceiveru (vysílače a přijímače). CC2420 v MICA zařízeních je čip zajišťující celou funkci transceiveru. Je to první čip podporující standard IEEE 802.15.4. Nastavené hodnoty vychází z technických parametrů zařízení MICA [3].

```

node[*].networkInterface.Radio.dataRate = 250
        # přenosová rychlost 250 kb/s
node[*].networkInterface.Radio.modulationType = 0
        # typ modulace bezkolizní
node[*].networkInterface.Radio.receiverSensitivity = -95
        # citlivost přijímače [dBm]
node[*].networkInterface.Radio.rxPower = 62
        # spotřeba při vysílání [mW]
node[*].networkInterface.Radio.listenPower = 62
        # spotřeba při příjmu [mW]
node[*].networkInterface.Radio.sleepPower = 1.4
        # spotřeba v režimu spánku [mW]
node[*].networkInterface.Radio.maxPhyFrameSize = 127
        # velikost datového rámce na fyzické vrstvě [bytů]
node[*].networkInterface.Radio.phyFrameOverhead = 6
        # velikost hlaviček datového rámce [bytů]

```

Soubor */Parameter_Include_Files/MAC_just_carrierSense.ini* nastavuje vlastnosti MAC vrstvy. V naší simulaci budou nody trvale zapnuty a nebudeme povolovat beacon rámce. Protože máme vytvořené ideální prostředí, můžeme zvolit pravděpodobnost přenosu rovnu 1, tedy vždy dojde k úspěšnému přenosu. Pro uplatnění určitých fyzikálních vlastností hardwaru spustíme vysílání až po určité, náhodně zvolené době, od příchodu požadavku k vysílání. Tím zabráníme současnému vysílání více nodů, které by při simulaci reálnějšího prostředí vyvolalo kolize rámců.

```

node[*].networkInterface.MAC.dutyCycle = 1
        # doba spánku / doba provozu = 1, nod tedy nespí
node[*].networkInterface.MAC.beaconIntervalFraction = 0
        # beacons nejsou vysílány
node[*].networkInterface.MAC.probTx = 1
        # pravděpodobnost úspěchu přenosu
node[*].networkInterface.MAC.randomTxOffset = 10
        # vysílání zpožděno o dobu z intervalu
        # [0...randomTxOffset] [ms]

```

Soubor */Parameter_Include_Files/Routing_simpleTree.ini* nastavuje parametry síťového modulu. Každý nod může zaznamenat údaje pouze o třech svých sousedech, z toho jeden bude rodič. V této simulaci je to dostatečné množství. Tak, jak je obvyklé v reálném použití, každý nod má pouze jednoho rodiče, kterému posílá svá data.

V souboru *omnetpp.ini* pak následují definice hodnot parametrů pro řídicí modul celého nodu, pro aplikační modul. Musíme určit, kterou implementaci v jazyce C++ použijeme pro řízení nodu. Dále určíme, že ve stromě bude pouze jeden sink (na úrovni 0) a ostatní nody mu budou podřízeny.

6.2 Spuštění a běh simulace

Jakmile máme nadefinovány tyto parametry, můžeme přikročit k vlastní simulaci. Tu spustíme skriptem */Castalia/Simulation/simpleAggregation/runSimpleAggregation*. Simulace je

```

node[*].networkInterface.Network.maxNetFrameSize = 60
                # velikost datového rámce na síťové
                # vrstvě [bytů]
node[*].networkInterface.Network.netDataFrameOverhead = 16
                # velikost hlaviček datového rámce
                # na síťové vrstvě [bytů]
node[*].networkInterface.Network.maxNeighborsTableSize = 3
                # maximální počet položek v tabulce sousedů
node[*].networkInterface.Network.maxNumberOfParents = 1
                # počet rodičů

```

```

node[*].appName = "simpleAggregation_ApplicationModule"
                # název souboru v C++ popisujícího chování nodu
node[*].nodeApplication.applicationID = "simpleAggregation"
                # jméno aplikačního modulu
node[*].nodeApplication.isSink = false
                # žádný nod není sink
node[3].nodeApplication.isSink = true
                # s výjimkou nodu číslo 3

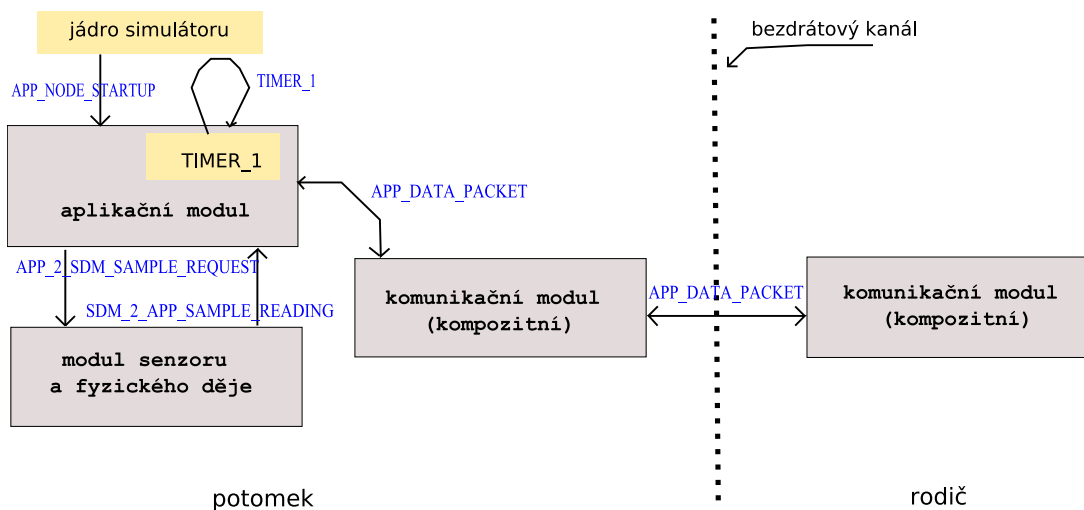
```

řízena kódem aplikačního modulu a její průběh se skládá z několika kroků:

1. Inicializují se proměnné na výchozí hodnotu, načtou se parametry ze souborů *.ini a aplikačnímu modulu se zašle zpráva APP_NODE_STARTUP.
2. Po přijetí zprávy APP_NODE_STARTUP aplikační modul spustí činnost nodu, tj. odblokuje jeho používání všem funkcím a naplánuje zaslání zprávy APP_REQUEST_SAMPLE pro zajištění další činnosti. Tím končí reakce na zprávu APP_NODE_STARTUP a od této chvíle je činnost nodu řízena pouze na základě příchozích zpráv od ostatních modulů a od vlastních časovačů.
3. Nod nyní očekává zprávu od síťové vrstvy, že byl zapojen do stromu na určitou úroveň pod sink. Sink je vždy inicializován na úrovni 0. Síťová vrstva vyhodnotí vhodnost připojení k nadřazenému uzlu podle metriky, kterou je v tomto případě vzdálenost od nodu. Jakmile je známa úroveň zanoření, změní se perioda časovače TIMER_1, který udává dobu mezi vysíláním naměřené hodnoty rodiči. Čím je nod hlouběji v hierarchii stromu, tím je tato perioda delší. Tím simulujeme zpoždění při přenosu dat k sinku.
4. Nod se stal členem stromu a periodicky mu chodí zpráva od jeho časovače TIMER_1 na požadavek o odeslání naměřené teploty. Dále mu periodicky chodí zpráva APP_SELF_REQUEST_SAMPLE požadující nové měření teploty. Aby mohl vyhovět musí kontaktovat modul senzoru zprávou APP_2_SDM_SAMPLE_REQUEST. Odpověď přichází opět v podobě zprávy. Jakmile senzor dokončí měření, pošle aplikačnímu modulu hodnotu teploty pomocí zprávy SDM_2_APP_SENZOR_READING.
5. Aplikační modul také dostává zprávy od svých potomků APP_DATA_PACKET s jimi naměřenou teplotou. Nad příchozími teplotami a svou teplotou provede agregační funkci (v tomto případě výběr maxima), výsledek uloží jako svou hodnotu a tu opět posílá svému rodiči zprávou APP_DATA_PACKET. Situace se opakuje až k sinku, který přijatou hodnotu zpracuje (vytiskne na výstup).

- Uvedený proces se opakuje až do doby vypršení času simulace. Činnost nodu může být ukončena předčasně před koncem simulace, pokud obdrží zprávu od modulu zdrojů `RESOURCE_MGR_OUT_OF_ENERGY`, signalizující vyčerpání baterií. Tuto situaci může způsobit například dlouhá doba simulace.

Pro názornost je systém zasílání zpráv schematicky zakreslen na obrázku 6.2.



Obrázek 6.2: Schéma zasílání zpráv mezi moduly nodu a nody v síti

6.3 Výsledek simulace

Všechny kontrolní výpisy jsou zapsány do souboru *Castalia-Debug.txt* a *Castalia-Primary-Output.txt*. Pro přehlednost jsou na standardní výstup za použití regulárních výrazů ve spouštěcím skriptu *runSimpleAggregation* vytištěny nejzajímavější údaje. Na ukázce výstupu (tabulka 6.1) jsou komentáře přímo v textu popisující řádky výstupu.

Na začátku simulace se nody uspořádají do stromové struktury podle obrázku 6.1. Při opakování simulace dojde vždy ke stejnému rozložení, což je dáno manuálním rozložením nodů. Inicializace proběhne do 1 sekundy od zahájení simulace. Jakmile se nody zařadí, začne výměna dat trvající až do vypršení času simulace. Sink průběžně vypisuje agregovanou hodnotu teploty, tak jak mu je doručována. Po skončení simulace je na výstup vypsána spotřebovaná energie z baterií jednotlivých nodů. Můžeme vidět, že při tomto neúspěšném režimu přenosu (transceiver neustále aktivní) by baterie vydržely napájet nod asi 5,5 dne. Proto je v praxi nutné využívat přenos v beacon-enable režimu s využitím uspávání nodů.

6.4 Možnosti dalšího rozšíření

Vhodným rozšířením algoritmu *simpleAggregation* by byla asociace nodů do sítě. Sink zde vystupuje jako PAN Coordinator, jednotlivá zařízení jako plně funkční zařízení schopná

číslo nodu	čas asociace	rodič
#[App_0]	t= 0.106397:	Parent 3#
#[App_3]	t= 0.106397:	Parent 3#
#[App_6]	t= 0.106397:	Parent 3#
#[App_1]	t= 0.156397:	Parent 4#
#[App_7]	t= 0.156397:	Parent 4#
#[App_5]	t= 0.159532:	Parent 4#
#[App_2]	t= 0.209532:	Parent 5#
#[App_8]	t= 0.209532:	Parent 5#
číslo sinku	simulační čas	agregovaná hodnota
#[App_3]	t= 1.01214:	Aggregated Value = 0
#[App_3]	t= 2.01216:	Aggregated Value = 8.12409
#[App_3]	t= 3.01218:	Aggregated Value = 19.8161
...		
#[App_3]	t= 46.0131:	Aggregated Value = 23.7314
#[App_3]	t= 47.0131:	Aggregated Value = 27.7415
#[App_3]	t= 48.0131:	Aggregated Value = 31.5887
#[App_3]	t= 49.0132:	Aggregated Value = 35.5211
číslo nodu	zbývající energie na konci simulace	
#[Energy_0]	t= 50: 3.10016	
#[Energy_1]	t= 50: 3.10066	
#[Energy_2]	t= 50: 3.10066	
#[Energy_3]	t= 50: 3.10099	
...		

Tabulka 6.1: Výstup simulace

pracovat jako router nebo jako koncové zařízení. Asociace spočívá v přidělování identifikátoru ID, sestávajícího se z identifikátoru PAN koordinátora a náhodně generovaného jednoznačného čísla, které identifikuje zařízení ve stromě. Aplikační vrstva každého nodu (mimo sinku) vytvoří datový paket s příznaky indikující požadavek na asociaci, resp. de-asociaci do sítě a ten se zašle sinku. Sink odpoví paketem obsahujícím přidělené ID. Pak může již probíhat přenos dat. Náročnější rozšíření by umožnilo využít beacon-enabled mód. V současné podobě simulace využívá komunikaci v non beacon-enabled módu, za použití přístupu na médium metodou CSMA/CA, ovšem v ideální podobě – bez kolizí.

Castalia je ideální nástroj pro testování detailů v nastavení spodních vrstev (fyzické a MAC) protokolu IEEE 802.15.4/ZigBee. Síťová vrstva je prozatím implementována pouze v podobě jednoduchých routovacích protokolů, i když v dalších verzích se očekává podstatné rozšíření. Je možné i naprogramovat vlastní routovací protokol, případně některý standardní. To se však již vymyká původnímu záměru práce nalézt vhodný nástroj pro výuku a vývoj. Proto při požadavku na simulaci se zřetelem na výzkum vyšších vrstev (síťová, bezpečnostní) bude vhodný jiný simulátor.

Kapitola 7

Závěr

Protože standard ZigBee je poměrně nový, očekával jsem problémy s dostupností simulátorů, které se ale nenaplnily. Po vyhledávání byly nalezeny čtyři uvedené programy. Jako nejvhodnější se ukázal simulátor OPNET, který uživateli poskytuje přehledné grafické rozhraní a neklade nároky na znalost dalšího programovacího jazyka. Jeho použití pro výuku by tak bylo ideální, poněvadž umožňuje zaměřit se pouze na problém komunikací v síti. Jedná se však o komerční produkt a podmínky pro získání licence byly závislé na vytvoření webové prezentace popisující vývoj v tomto simulátoru. Protože s tímto nebylo počítáno, bylo od simulátoru OPNET upuštěno. Jako další vhodná alternativa byl následně vybrán simulátor OMNET++/Castalia. Modul Castalia se ukázal jako velmi komplexní řešení pro simulaci bezdrátových sensorových sítí. Nepříjemnou vlastností je požadavek na znalost programovacího jazyka C++, kladený na uživatele. To společně s ovládáním a vyhodnocováním výsledků přes příkazový řádek snižuje intuitivnost ovládání. Simulátor OMNET++/Castalia umožňuje i tvorbu grafického uživatelského rozhraní. V modulu Castalia zatím GUI není implementováno, protože vývoj je teprve na počátku.

Jako pokračování této práce by tedy bylo vhodné doplnit OMNET++/Castalia o grafické rozhraní zobrazující průběh simulace a výsledky. Možné by také bylo rozšíření síťové vrstvy o implementaci dalších routovacích protokolů používaných ve standardu ZigBee, i když na tomto úkolu již pracují vývojáři simulátoru Castalia.

Literatura

- [1] 802.15.4 and ZigBee Routing Simulation at Samsung/CUNY. 2003, přednáška [online] [cit. 2008-04-10].
URL http://www.cuny.cuny.edu/cint/awnl/Software/WPAN_ZBR_pub.pdf
- [2] ZigBee Specification. Technická zpráva, ZigBee Alliance, 2008.
- [3] BASAVAN, C.: Research Integration: Platform Survey. Výzkumná zpráva EW-T21/D01-SICS-001-01, 2006.
- [4] BOULIS, A.: *Castalia. A simulator for Wireless Sensor Network version 1.3*. National ICT Austria, 2008.
- [5] Castalia - A simulator for WSNs. 2008, [online] [cit. 2008-04-10].
URL <http://castalia.npc.nicta.com.au/index.php>
- [6] HOLGER, K.; WILLIG, A.: *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005, ISBN 978-0-470-09510-2.
- [7] JURČÍK, P.; KOUBÁA, A.: The IEEE 802.15.4 OPNET Simulation Model: reference Guide v2.0. Technická zpráva, IPP Hurray!, Polytechnic Institute of Porto, 2007.
- [8] KOUBÁA, A.; ALVES, M.; TOVAR, E.: IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview. Technická zpráva, IPP Hurray!, Polytechnic Institute of Porto, 2005.
- [9] OMNeT++ Community Site. [online] [cit. 2008-04-10].
URL <http://www.omnetpp.org/>
- [10] SOHRABY, K.; MINOLI, D.; ZNATI, T.: *Wireless sensor networks - Technology, Protocols, and Applications*. John Wiley & Sons, 2007, ISBN 978-0-471-74300-2.
- [11] SZABADOS, L.; LÍŠKA, O.: Smart senzorové siete a príklady ich využitia. *Transfer inovácií*, 7 2004.

Dodatek A

Seznam použitých zkratek

AES	Advanced Encryption Standard
BE	Beacon Enabled
BI	Beacon Interval
CAP	Contension Access Period
CCA	Clear Channel Assessment
CDMA	Code Division Multiple Access
CFP	Contention Free Period
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CW	Contension Window length
DES	Data Encryption Standard
DSSS	Direct Sequence Spread Spectrum
FFD	Full Function Device
GPL	General Public License
GPRS	General Packet Radio Service
GSM	Global System for Mobile
GTS	Guaranted Time Slot
IEEE	Institute of Electrical and Electronics Engineers
ISM	Industrial, Scientific and Medical
ISO/OSI	International Standard Organization / Open System Interconnect
LQI	Link Quality Indicator
MAC	Media Access Control
MIC	Message Integrity Code
MPDU	Message Protocol Data Unit
OQPKS	Quadrature Phase-shift-keyed modulation
PAN	Personal Area Network
POS	Personal Operating Space
RFD	Reduced Function Device
SD	Superframe Duration
Wi-Fi	Wireless Fidelity
ZDO	ZigBee Device Object

Seznam příloh

CD-ROM obsahuje:

- kompletní stromovou strukturu souborů simulátoru Castalia s komentovaným zdrojovým kódem aplikačního modulu
(uloženo v adresáři Castalia-1.3)
- text této práce v souboru zigbee.pdf
(uloženo v kořenovém adresáři CD-ROM)