

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

KOMUNIKAČNÍ SCHÉMA WSN V PROSTŘEDÍ OMNET

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

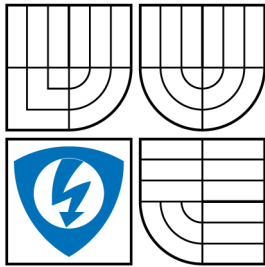
AUTOR PRÁCE
AUTHOR

JIŘÍ MIKAUŠ

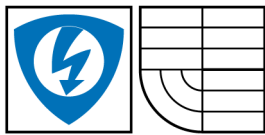
BRNO 2010



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

KOMUNIKAČNÍ SCHÉMA WSN V PROSTŘEDÍ OMNET COMMUNICATION SCHEME OF WSN IN OMNET SIMULATOR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ MIKAUŠ

VEDOUcí PRÁCE
SUPERVISOR

ING. PAVEL HOLEŠINSKÝ

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Jiří Mikauš

ID: 109698

Ročník: 3

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Komunikační schéma WSN v prostředí Omnet

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit a popsat vybrané komunikační schémata bezdrátové sensorové sítě v simulačním nástroji Omnet. Nejdříve je nutné nastudovat teorii sensorových sítí. Dále pak bude následovat sestavení komunikačních modelů a jejich simulace.

DOPORUČENÁ LITERATURA:

[1] BULUSU N., JHA S., Wireless sensor networks. Boston: Artech House, 2005, 326 stran. ISBN: 978-1580538671.

[2] KUORILEHTO M., KOHVAKKA M., SUHONEN J. Ultra-Low Energy Wireless Sensor Networks in Practice

Chichester: Wiley. 368 stran. ISBN: 978-0-470-05786-5.

Termín zadání: 29.1.2010

Termín odevzdání: 2.6.2010

Vedoucí práce: Ing. Pavel Holešinský

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské 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.

ZDE VLOŽIT PRVNÍ LIST LICENČNÍ
SMOUVY

Z důvodu správného číslování stránek

ZDE VLOŽIT DRUHÝ LIST LICENČNÍ
SMOUVY

Z důvodu správného číslování stránek

ZDE VLOŽIT PROHLÁŠENÍ O SHODĚ
LISTINNÉ A ELEKTRONICKÉ FORMY
PRÁCE

Z důvodu správného číslování stránek

ABSTRAKT

V úvodu této semestrální práce je uvedena teorie bezdrátových sensorových sítí, které nacházejí stále širší spektrum uplatnění. Bezdrátové sensorové sítě obsahují až několik tisíc uzlů, které spolu komunikují pomocí bezdrátového kanálu a dohromady tak tvoří strukturu sítě. Uzly jsou nízkoenergetické a předávají informace o naměřeném médiu základové stanici, která je zpracovává a poskytuje uživateli. V práci jsou popsány protokoly pro kontrolu přístupu k médiu a směrování. Ty kladou důraz na co největší možnou redukci spotřeby elektrické energie a tím napomáhají k prodloužení životnosti celé sítě. Dále byly provedeny simulace v simulačním nástroji Castalia, kde jsou porovnány energetické nároky pro jednotlivé protokoly kontroly přístupu k médiu a porovnány dvě reálně používané komunikační jednotky.

KLÍČOVÁ SLOVA

bezdrátová sensorová síť, MAC , sensorový uzel, směrování, WSN

ABSTRACT

In the introduction of this semestral thesis there is an theory about wireless sensor networks which are still more used. Wireless sensor networks contains thousands nodes which communicate with each other due to wireless channel and so they create together a structure of the network. The nodes are low-energy and transfer information about measured medium to the base station which processes them and further provides to user. In the work are described protocols for medium access control and routing described. They place emphasis on as high reduction of consumption of energy as possible and thus they help to extend a network lifetime. Further the simulations in simulation program Castalia were performed where the energetic requirement for individual protocols of medium access are compared. As next there were also compared two communication units used in real application.

KEYWORDS

MAC, routing, sensor node, WSN, wireless sensor network

MIKAUŠ J. *Komunikační schéma WSN v prostředí Omnet*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2010. 62 s. Vedoucí semestrální práce Ing. Pavel Holešinský.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Komunikační schéma WSN v prostředí Omnet“ jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce Ing. Pavlu Holešinskému za pomoc při zpracování bakalářské práce a za poskytnuté konzultace.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	14
1 Vlastnosti WSN	15
1.1 Limitované zdroje	15
2 Kontrola přístupu k médiu	17
2.1 S-MAC	18
2.2 T-MAC	19
3 Směrování ve WSN	21
3.1 Problematika návrhu směrování ve WSN	21
4 Směrovací protokoly ve WSN	23
4.1 Distribuované směrování	23
4.1.1 SPIN	23
4.1.2 Řízené šíření	24
4.1.3 Rumor routing	25
4.2 Hierarchické směrování	26
4.2.1 LEACH	26
4.2.2 PEGASIS	27
4.2.3 TEEN a APTEEN	28
4.3 Geografické směrování	28
4.3.1 GAF	29
4.3.2 GEAR	29
5 Simulační nástroj Castalia	31
5.1 Úvod	31
5.2 Simulace v Castalii	33
5.2.1 Bezdrátový kanál	35
5.2.2 Rádiová jednotka	36
5.2.3 MAC	37
5.2.4 Směrování	39
5.2.5 Fyzický proces	40
5.2.6 Modul řízení zdrojů	40
6 Návrhy simulací	41
6.1 Simulace MAC protokolů	41
6.2 Konektivita sítě	46

7	Vyhodnocení výsledků	48
8	Závěr	52
	Literatura	53
	Seznam symbolů, veličin a zkratk	55
	Seznam příloh	57
A	Komunikační jednotky	58
	A.1 Mica2	58
	A.2 Imote2	60
B	Obsah přiloženého DVD	62

SEZNAM OBRÁZKŮ

2.1	Časové sloty u a) S-MAC protokolu, b) T-MAC protokolu	19
4.1	Vyžádání dat u SPIN protokolu	24
4.2	Řízené šíření: a) vysílání zájmů, b) nastavení gradientů, c) vyztužení trasy	25
4.3	GAF: výpočet délky strany buňky	30
5.1	Struktura Castalie	31
5.2	Vnitřní struktura uzlu v Castalii	32
5.3	Průběh simulace	34
6.1	Rozmístění uzlů v síti	46
7.1	Průměrná spotřeba energie pro jednotlivá kola simulace	48
7.2	Průměrná spotřeba energie pro MAC protokoly	50
A.1	Mica2	58
A.2	Imote2	60

SEZNAM TABULEK

7.1	Průměrná spotřeba komunikačních jednotek pro jednotlivá kola simulace	48
7.2	Průměrná spotřeba komunikačních jednotek pro MAC protokoly . . .	49
7.3	Výsledky třetího experimentu	51
A.1	Vlastnosti Uzlu Mica2 [6]	59
A.2	Vlastnosti Uzlu Imote2 [5]	61

ÚVOD

Bezdrátové sensorové sítě mají širokou škálu uplatnění ve zdravotnictví, armádě nebo průmyslu. Nejčastěji je jejich úkolem získat data, tyto lokálně zpracovat a poskytnout skrze základnovou stanici uživateli zprávu o měření. Liší se od jiných bezdrátových sítí zejména počtem uzlů a jejich omezenými vlastnostmi jako je výpočetní kapacita, šířka přenosové rychlosti a kapacita napájecího zdroje. Proto pro tyto sítě byla navrženo a vyvinuto mnoho speciálních protokolů. Výběr správného protokolu je primárním úkolem při návrhu konkrétní aplikace. Neexistuje jeden univerzální protokol, který by byl vhodný pro jakoukoliv aplikaci, protože některé aplikace potřebují získat měřená data s co nejnižší odezvou bez důrazu na přesnost a jiné vyžadují přesná data byť i s časovým zpožděním. Protože sensorové uzly jsou v drtivé většině napájeny z baterií, je cílem co největší snížení spotřebované energie při přenosu dat i při samotném měření. Největší zásluhy na poli snižování energie nese především správná volba protokolu kontroly přístupu k médiu a směrovacího protokolu.

V první části bakalářské práce jsou popsány základní vlastnosti kontroly přístupu k médiu. Jsou zde uvedeny obecné vlastnosti těchto protokolů a dále jsou uvedeny dva základní protokoly, které budou v poslední části podrobeny zkoumání v simulačním programu Castalia.

V druhé části je úvod do směrování v bezdrátových sensorových sítích a následně úskalí při návrhu těchto protokolů, protože bezdrátové sensorové sítě se v mnohých ohledech liší od klasických sítí zejména sníženými hardwarovými možnostmi. Dále je uvedeno několik směrovacích protokolů, které jsou využívány každý pro různé účely.

V poslední části je popis simulačního programu Castalia, kde je v prvním experimentu ukázka závislosti spotřeby sensorů na délce pracovního cyklu protokolu kontroly přístupu k médiu a v druhém experimentu byly porovnány dva reálně hojně používané protokoly. Třetí experiment se zaměřuje na porovnání obou protokolů z hlediska konektivity sítě. Při všech simulacích jsou také porovnány vlastnosti dvou komunikačních jednotek, jež jsou popsány v příloze A.

1 VLASTNOSTI WSN

Současná technologie a pokročilé výrobní procesy umožňují výrobu malých a technologicky vyspělých sensorových uzlů za nízkou cenu. Bezdrátové sensorové sítě (WSN - Wireless Sensor Network) jsou v mnoha ohledech podobné ad-hoc sítím a bezdrátovým MESH sítím, mohou obsahovat stovky až tisíce sensorových uzlů. Uzly jsou schopny komunikovat přes mezilehlé uzly (multihop) nebo přímo se základovou stanicí (BS - Base Station), která shromažďuje data, zpracovává je a poskytuje uživateli např. prostřednictvím internetu. Větší počet uzlů v síti umožňuje měřit rozsáhlejší plochy území. Při vyšší hustotě rozmístění uzlů na jednotku plochy lze dosáhnout vyšší přesnosti naměřených hodnot. Bezdrátové sensorové sítě nacházejí díky své přizpůsobitelnosti široké oblasti využití. Například nenápadný monitoring divočiny, domovní zabezpečení, vojenské sledování nepřátelského cíle, sledování kvality vody, odhalování katastrof (požáry, zemětřesení, povodně), sledování zdravotního stavu pacienta, snímání teploty, zvuku, světla, tlaku, atd.

Na síť jsou kladeny požadavky podle jejího budoucího využití. Několik požadavků je však společných pro většinu aplikací. Sensorové uzly, a tedy i celá síť, musí pracovat i po dobu několika let, jinak by byla celá aplikace ekonomicky nevýhodná. Tento požadavek klade velké nároky na úspory spotřebované energie. Uzly jsou nejčastěji napájeny z AA nebo AAA baterií, jejichž výměna může být značně problematická. Z důvodu cenové perspektivy musí celá sensorová síť pracovat samostatně bez vnějších zásahů lidské ruky.

Síť musí být odolná vůči chybám při měření i přenosu dat bezdrátovým kanálem. Uzly mohou zkolabovat z důvodu vybití baterií, nebo poškozením baterie a rádiová komunikace může být rušena okolními signály. Sensorové uzly jsou vyrobeny nízkonákladově, a tudíž jsou náchylnější na poruchu nebo snímání chybných údajů. Data generovaná periodicky nebo měřená na základě zvolených událostí musí být přeneseny do základové stanice pro pozdější zpracování a jejich poskytnutí uživateli.

1.1 Limitované zdroje

Rádiová část uzlu je energeticky nejnáročnější komponentou typického uzlu, velké úspory mohou být dosaženy na spojové vrstvě, kde MAC protokol kontroluje použití rádiové části. MAC (Medium Access Control) protokoly navržené pro WSN na rozdíl od parametrů u jiných sítí (propustnost, odezva, spravedlivost při doručení dat) jsou zaměřeny na co největší snížení spotřeby energie a tím dosažení co nejdélejší životnosti celé sítě.

Současným trendem je zvyšování výkonu všech uzlů, zejména základnových stanic. CPU se vyrábějí s vyšší taktovací frekvencí a obsahují větší vnitřní vyrovnávací

paměť a pomáhají tak k řízení přenosů s vyšším datovým tokem. Přičemž spotřeba celého systému je konstantní nebo nižší (v praxi je spotřeba cca 100 mW). Díky modernějším výrobním technologiím může síť zajistit ještě více úkolů při použití stejného napájecího zdroje. Vybití baterií záleží zejména na době po kterou uzel setrvává v režimu spánku, kdy je spotřeba minimální, téměř nulová. Energetické řízení uzlů má na starosti hlavně MAC vrstva. Dnešní vývoj směřuje k tvoření sensorových sítí skupinami nenáročných nízko energetických uzlů pro sběr dat, které obsluhuje páteřní síť tvořená výkonnějšími, ale také dražšími uzly. [10]

CPU: vládnu 8 bitové procesory, ale stále více se prosazují 16 bitové s hodinovým taktem v rozsahu 1–10 MHz, což postačuje pro běh protokolu, řízení rádiové části uzlu a aplikačního zpracování dat. Výkonnější jednotky jsou osazeny CPU s proměnnou taktovací frekvencí. Např. Imote2 od 13 MHz do 416 MHz viz A

Paměť: velikost paměti pro uchování kódu programu se pohybuje okolo 4–10 kB u nízko výkonných uzlů. Operační paměť uzlu je cca 10 kB, což nutí vývojáře softwaru minimalizovat jejich programy s ohledem na alokaci této paměti.

Rádio: v porovnání s dnešními standardy v bezdrátových LAN sítích IEEE 802.11x (teoreticky až 300 Mbps, při využití MIMO i více) je šířka pásma u sensorových sítí je pouze 10–250 kbps. Většina aplikací nepotřebuje velkou šířku pásma, ale záleží na dosahu uzlů (desítky metrů) a kvalitě spoje. Dosah uzlů je poněkud horší, neduh je způsoben jednoduchými modulačními technikami citlivými na šum a integrovanými anténami na desky plošných spojů. Nové komunikační jednotky mívají možnost připojení externí antény [5]. [10]

2 KONTROLA PŘÍSTUPU K MÉDIU

MAC (Medium Access Control – Kontrola přístupu k médiu) protokoly, které byly primárně navrženy pro typické ad-hoc sítě, byly zaměřeny na optimalizaci propustnosti a spravedlnosti při doručování dat bez větších ohledů na spotřebu energie. Protože typickým problémem ve WSN je nedostatek energie, musely být MAC protokoly navrženy s robustním a energeticky úsporným řešením. Ukázalo se, že i uzel v režimu klidu, může spotřebovávat velké množství energie stejně jako při režimu vysílání, obzvláště v sítích, kde je relativně malý provoz. Proto se většina protokolů pro kontrolu přístupu k médiu v sensorových sítích snaží snížit spotřebu energie přepínáním rádiových modulů uzlů do režimu spánku tak často, jak je to jen možné. Efektivně navržený protokol poskytuje největší možnosti při snižování spotřeby energie.

MAC protokoly poskytují mírně odlišné funkce závislé na konkrétní síti, možnostech zařízení a požadavcích vyšších vrstev. Několik funkcí je ovšem společných pro většinu MAC protokolů. Obecně MAC protokoly zajišťují dle lit. [9]:

- *Tvar rámců* – definuje formát datových rámců a provádí zapouzdření a zpětné rozpouzdření dat pro komunikaci mezi zařízeními.
- *Přístup k médiu* – řídí, které zařízení se zúčastní komunikace v určitém čase. Kontrola přístupu se stává hlavní funkcí bezdrátových MAC protokolů, protože vysíláním dochází ke snadnému porušení dat vlivem kolizí.
- *Spolehlivost* – zajišťuje úspěšné přenosy mezi zařízeními. Nejčastěji realizované přes potvrzovací ACK (acknowledgement - potvrzení) zprávy a následné přeposlání při neobdržení ACK.
- *Kontrola toku* – předcházení ztráty rámců z důvodu přetečení příjemcovy vstupní vyrovnávací paměti.
- *Kontrola chyb* – užívá kódy pro kontrolu chyb resp. jejich opravování ke kontrole míry chyb obsažených v rámcích doručených pro vyšší vrstvy.

Klasické bezdrátové sítě IEEE 802.11 využívají na MAC vrstvě CSMA/CA (Carrier Sense Multiple Access With Collision Avoidance), kde uzel ověřuje přítomnost provozu na sdíleném přenosovém médiu (bezdrátovém kanálu), kde vysílá a přijímá více uzlů. Před vysláním vysílač naslouchá na nosné a pokud je volná, započne vysílat. Pokud ne, s vysláním počká a vysílá později. Protokol zabraňuje kolizím tak, že každý uzel musí před vysláním informovat o tomto úmyslu ostatní uzly a poté vysílá data. Tento způsob komunikace aplikovaný na WSN, kde je nízký provoz, by byl zbytečným plýtváním energie kvůli následujícím vlastnostem.

Nečinné naslouchání – MAC protokol nemůže sdělovat informace o tom, kdy bude jaká zpráva doručena. Proto musí rádiová jednotka zůstat stále zapnutá, jinak nebude možno uzlu doručit některé zprávy, které jsou pro něj určeny. Toto nečinné naslouchání je značným zdrojem velké spotřeby, když nejsou zrovna přijímána data.

Přeslechy – další nepřehlédnutelnou vlastností neustálého naslouchání je přijímání všech zpráv, i těch které patří pouze sousedním uzlům. Přeslechy jsou problematické v hustě osazených sítích, kde je v dosahu uzlu mnoho jiných uzlů. Husté osazení je velmi časté, protože oblast snímané hodnoty je vždy mnohonásobně menší než doslech uzlu.

Kolísání provozu – provoz generovaný WSN aplikacemi je proměnný jak v čase, tak v místě odkud data přichází. Výsledné špičky mohou zahlcovat celou síť. Tím spotřeba roste do nežádoucích mezí.

Dalšími vlastnostmi jsou kolize zpráv v síti a režie protokolu, kde MAC hlavičky a kontrolní zprávy jsou považovány za zbytečné, protože nepřenášejí žádná data, pouze spotřebovávají energii. Alternativou je použití rozvrhů, což se na první pohled jeví jako atraktivní. Nečinné naslouchání, přeslechy a kolize nevznikají, protože po obdržení rozvrhu je jasné, ve kterých (např. časových) slotech může uzel přijímat a vysílat. Problém je, že toto řešení má omezenou flexibilitu. Dynamicky měnit počet slotů je neproveditelné. [10]

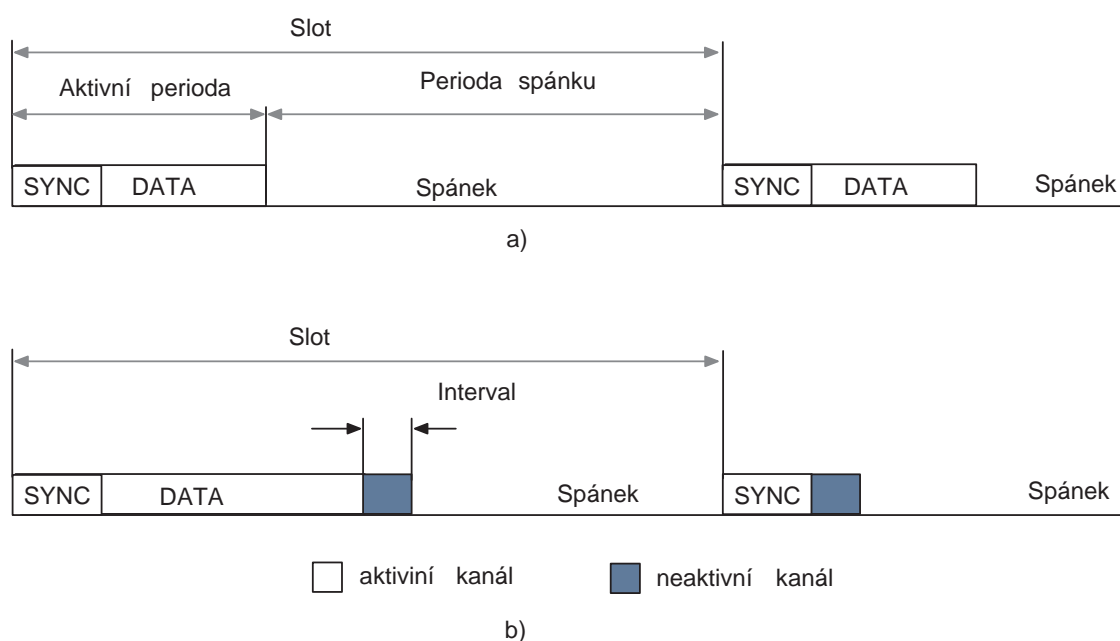
Ve zbytku kapitoly budou uvedeny dva MAC protokoly S-MAC a T-MAC, jež jsou součástí simulačního programu Castalia. Tyto protokoly organizují přístup k médiu pomocí slotů. Protokolů je ovšem celá řada. Nejnovějšími přírůstky v tomto odvětví jsou protokoly jako například X-MAC, který využívá náhodný přístup ke sdílenému rádiu nebo Crankshaft, využívající hybridní přístup. Jejich popis už přesahuje rámec tohoto textu.

2.1 S-MAC

S-MAC byl jeden z prvních navržených MAC protokolů pro WSN. Hlavní výhodou protokolu je jeho metoda fixního pracovního cyklu, která je jednoduchá a efektivně redukuje nečinné naslouchání. S-MAC synchronizuje mezi uzly rozvrhy spánku sousedních uzlů a vytvoří virtuální skupiny. Poté uzly pravidelně vysílají SYNC zprávy, umožňující ostatním uzlům naučit se rozvrhy svých sousedů, tak se mohou probouzet ve správný čas pro příjem zpráv. Nové uzly, které se chtějí připojit do sítě začnou naslouchat, zda-li nezachytí inicializační periodu. Po dobu několika časových slotů čekají na synchronizační zprávu, aby získaly informace o rozvrhu. Jeden slot se skládá z periody spánku a aktivní periody, která obsahuje SYNC a přenos dat. Pokud SYNC zpráva není odchycena, uzel usoudí že je první, kdo vytvoří tzv. virtuální

skupinu, a začne sám vysílat SYNC zprávy, ostatní se mohou později připojit. Pokud uzel zachytí více rozdílných rozvrhů, přijme je všechny a začne uskutečňovat komunikaci mezi dvěma skupinami. S-MAC také zahrnuje mechanismus posílání zpráv, kde jsou dlouhé pakety rozděleny na jednotlivé fragmenty. Jakmile se uzly připojí k rozvrhu začnou dle určeného pracovního cyklu zapínat a vypínat svoje rádiové obvody. U uzlů Mica2 je implementováno S-MAC s pevně nastavenou délkou aktivní periody na 300ms a nastavitelnou délkou slotu volitelně 1-3s.

Jednoduchost protokolu využívajícího pevně nastavený pracovní cyklus má jednu nevýhodu. A to, že vývojář aplikace musí vybrat vhodnou dobu aktivních period ještě před samotným rozmístěním sensorových uzlů. [9],[10]



Obr. 2.1: Časové sloty u a) S-MAC protokolu, b) T-MAC protokolu

2.2 T-MAC

T-MAC vylepšuje S-MAC protokol využitím časovače pro indikaci skončení aktivní periody namísto trvání na pevném pracovním cyklu. T-MAC protokol zavádí proměnnou délku aktivní periody. Uzly naslouchají pouze po krátký časový interval na začátku slotu, a pokud nenastane komunikace (neaktivní kanál), přepne se uzel do režimu spánku. Interval trvá 15ms (u S-MAC je aktivní perioda dlouhá 300 ms). Pokud uzel naruší nebo přeslechne zprávu, počká na dokončení přenosu a v dalším

intervalu rozhodne, zda může přejít do režimu spánku či nikoliv. Pokud není uskutečněna komunikace, zůstává uzel v aktivním stavu pouze 15ms. Na obrázku 2.1 (převzatý z [9]) vidíme T-MAC časový slot, kde první aktivní perioda ukazuje uzel při přenosu zpráv a druhá aktivní perioda obsahuje pouze SYNC zprávu, poté není vyžadována komunikace a po vypršení intervalu přechází uzel do režimu spánku.

Simulace ukázaly, že T-MAC je schopno adaptovat na výkyvy provozu v čase i místě. V otázce spotřeby energie tento protokol předčí S-MAC díky zmenšování doby setrvávání ve stavu nečinnosti. Nevýhodou je nárůst odezvy a snížení propustnosti sítě vlivem agresivního vypínání rádiového modulu a čekání na další vysílací slot. [9],[10]

3 SMĚROVÁNÍ VE WSN

Směrování je ve WSN problematické vzhledem k vlastnostem, které odlišují tyto sítě od ostatních bezdrátových sítí jako jsou ad-hoc nebo buňkové sítě v několika aspektech. Za prvé vzhledem k relativně velkému počtu sensorových uzlů není možné uskutečnit celkové adresovací schéma pro rozmístění velkého počtu uzlů. Tudíž nepřichází v úvahu na WSN sítích aplikovat protokoly navržené pro IP síť. Za druhé ve WSN je někdy důležitější obdržet data, než znát identitu uzlů, od kterých jsou data doručena. Za třetí v porovnání s typickými komunikačními sítěmi většina aplikací sensorových sítí požaduje tok snímaných dat z mnoha zdrojů do jediné BS (Base Station – základová stanice). Za čtvrté uzly mají omezené zdroje energie, výpočetní kapacity a paměti. Za páté návrhové požadavky se liší pro každou aplikaci. Za šesté znalost pozice uzlů je důležitá pokud je sběr dat založen na lokalitě jejich původu. V současné době není vhodné používat GPS pro účely určení pozice uzlů. Raději se využívá metod triangulace, které umožňují uzlům určit jejich přibližnou pozici podle intenzity signálu přijatého z několika uzlů se známou pozicí. Vzhledem k těmto rozdílům bylo navrženo mnoho směrovacích řešení pouze pro bezdrátové sensorové sítě.

3.1 Problematika návrhu směrování ve WSN

WSN sítě mají spoustu omezení např. omezenou dodávku energie, limitovanou výpočetní kapacitu a šířku pásma pro bezdrátové linky spojující uzly. Jeden z hlavních požadavků při návrhu sensorových sítí je přenést data při co nejmenším snížení životnosti sítě a vyvarovat se zhoršení konektivity použitím směrovacích technik náročných na spotřebu energie. Nyní bude uvedeno několik úskalí při návrhu bezdrátových sensorových sítí, které je nutno vzít v úvahu pro dosažení co nejefektivnější komunikace [1].

Rozmístění uzlů: Rozmístění uzlů může být v bezdrátových sensorových sítích náhodné, anebo stanovené. Rozmístění záleží na použití sítě a určuje výkonnost směrovacího protokolu. Pokud není rozmístění uzlů rovnoměrné, je pro energeticky účinné síťové operace nevyhnutelné seskupování uzlů do skupinek. Dosah signálu je omezen výkonem vysílacích obvodů a integrovanými anténami. Proto je vhodné, aby trasa byla tvořena několika skoky. Při stanoveném rozmístění jsou uzly ručně umístěny a data jsou směrována předurčenými trasami.

Model doručení dat: Snímání a doručování dat je ve WSN závislé na aplikaci a časové náročnosti pro doručení dat. Poskytování dat může být rozděleno na časově řízené (kontinuální), řízené událostmi, doručení na požádání a hybridní. Časově řízené je vhodné pro aplikace které vyžadují periodické snímání. Uzly periodicky

přepínají mezi senzorovou a vysílací částí, měří prostředí a přenášejí data v opakujících se konstantních časových intervalech. U doručování řízeném událostmi nebo žádostmi uzly reagují na náhlé a kritické změny hodnot měřených vlastností. Tyto modely se mohou kombinovat a používají se v časově kritických aplikacích.

Energetická náročnost: Životnost uzlu je silně závislá na životnosti napájecího zdroje, většinou baterie. Při skokové komunikaci každý uzel hraje dvojí roli jako odesílatel dat a jako směrovač. Poruchy uzlů kvůli selhání napájení mohou znamenat významné změny topologie sítě, poté je nutné přesměrovat pakety jinými trasami a reorganizovat celou síť.

Tolerance chyb: Některé uzly mohou selhat, či být blokovány kvůli nedostatku energie, fyzickému poničení při rozmístování do terénu nebo rušením z okolí. Porucha uzlu by neměla ovlivnit úkol celé sítě. Pokud selže mnoho uzlů, kontrola přístupu k médiu a směrovací protokol musí uspořádat nové linky a trasy pro sběr dat do základové stanice. K tomu je zapotřebí aktivní nastavování vysílacího výkonu rádiové jednotky pro existující linky, což vede ke snížení spotřeby, nebo přesměrování paketů přes části sítě kde je dostatek energetických zdrojů.

Dynamika sítě: Většina síťových architektur uvažuje uzly jako stacionární. V některých aplikacích je zapotřebí mobilita jak základových stanic tak jednotlivých uzlů. Směrování zpráv od pohybujících se uzlů je náročnější na stabilitu tras, šířku pásma, spotřebu atd. Mimo to snímaná vlastnost může být také dynamická či statická, což záleží na použití. Např. dynamická při snímání pohybu cíle či statická při snímání kvality vody v řece.

Přenosové médium: V mnohaskokových (multi-hop) bezdrátových senzorových sítích jsou uzly spojeny bezdrátovým médiem. Funkčnost celé senzorové sítě proto mohou narušit tradiční problémy spojené s bezdrátovým kanálem jako jsou vysoká chybovost, ztrátovost, neúnosná odezva atd. Obecně je šířka pásma ve WSN nízká tj. cca 100 kb/s. Tuto problematiku řeší hlavně správný návrh MAC protokolu.

Agregace dat: Uzly mohou poskytovat výrazně redundantní data. Podobná data od několika uzlů mohou být seskupeny (agregovány), a tím se sníží počet přenosů. Agregace dat je kombinace dat z různých zdrojů podle určitých agregačních funkcí, např. potlačení stejných dat, neodeslání minim a maxim atd. Tato technika je použita v mnoha směrovacích protokolech pro dosažení energetické efektivity a optimalizace datového přenosu.

QoS: (Quality of Service – kvalita služeb) Při určitých aplikacích WSN musí být naměřená data doručena s určitým časovým zpožděním od momentu, kdy jsou naměřena. Data doručená později jsou nevyužitelná. Proto maximální odezva pro doručená data je další podmínkou pro časově náročné aplikace.

4 SMĚROVACÍ PROTOKOLY VE WSN

Obecně může být směrování v bezdrátových sensorových sítích rozděleno dle struktury sítě na distribuované, hierarchické a geografické odvíjející se od struktury sítě. U distribuovaného směrování mají všechny uzly stejnou funkci nebo roli. Při hierarchickém směrování má každý uzel určenou jinou roli. A u geografického směrování je známá pozice uzlu využita ke směrování dat v síti. Protokoly také mohou být děleny dle funkčnosti protokolu na mnohacestné, založené na principu požadavků, vyjednávání, QoS principu nebo koherentní. Další možné dělení je na proaktivní, reaktivní a kombinované směrovací protokoly. U proaktivních protokolů jsou všechny trasy sestaveny ihned bez ohledu na jejich budoucí potřebu, zatímco u reaktivních se trasy sestavují až dle potřeby. Kombinované protokoly využívají myšlenku obou předchozích návrhů. Při statickém umístění uzlů nejsou reaktivní protokoly vhodné, protože značná část energie je spotřebována na vyhledání tras a jejich sestavení. Ve zbytku kapitoly budou popsány základní směrovací protokoly spadající do první skupiny tj. dělení dle struktury sítě.

4.1 Distribuované směrování

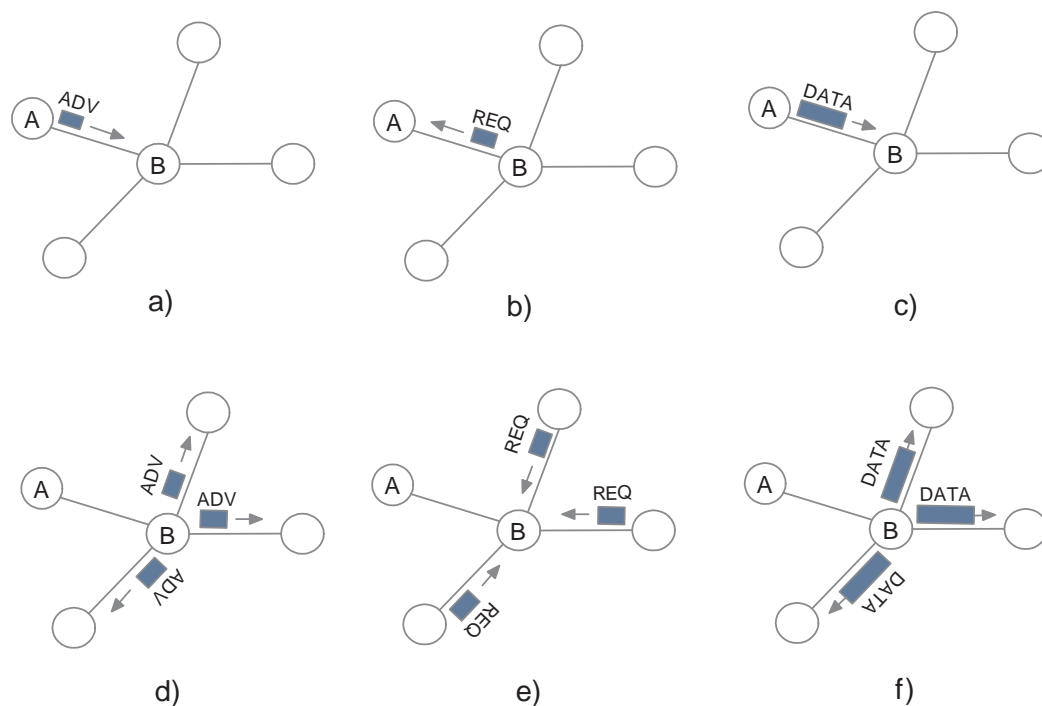
V distribuovaných (flat) sítích každý uzel hraje stejnou roli a všechny uzly společně spolupracují na splnění úkolu. Vzhledem k velkému množství uzlů v síti není vhodné, aby měl každý uzel globální identifikátor. Tato okolnost vedla k zavedení data-centrického směrování, kde základová stanice vysílá požadavky do určitých regionů, a poté čeká na data od uzlů ležících v těchto oblastech. Při přijímání dat na základě požadavků je nutné data označit atributy pro upřesnění jejich vlastností. Prvními data-centrickými protokoly byly SPIN a řízené šíření, které nastínily metody pro úsporu energie komunikací uskutečněnou pomocí vyjednávání. Tím redukuje nadbytečnost dat na minimum. Na základě těchto protokolů bylo navrženo i mnoho dalších. Nyní budou některé uvedeny.

4.1.1 SPIN

SPIN (Sensor Protocols for Information via Negotiation) protokol šíří všechny informace z uzlu na uzel, jako by všechny uzly v síti byly potencionální základovou stanicí. To umožňuje uživateli požádat kterýkoliv sensorový uzel a neprodleně obdržet data. Využívá se předpokladu, že uzly ležící v blízkosti mají podobná data, proto je potřeba poskytnout pouze data, která ostatní uzly nemají. Uzly u tohoto protokolu označují nasbíraná data, tato označení se nazývají meta-data. Meta-data mohou popisovat takové vlastnosti jako typ dat nebo pozici jejich vzniku. Jejich

specifikace není součástí protokolu SPIN, formát meta-dat je jedinečný pro každou aplikaci. Celá komunikace je zajištěna pomocí tří zpráv ADV, REQ a DATA. ADV (advertise - oznámení) oznamují uzly že obdržely nová data. REQ (request - žádost), tímto uzel oznamuje zájem o data a DATA jsou samotná data. Uzel A obdrží nová data, která bude přeposílat dále. Vyšle zprávu ADV obsahující meta-data svému bezprostřednímu sousedovi – uzlu B (obr. 4.1 a)), ta v případě, že má zájem o data, pošle zpět REQ zprávu, a poté obdrží data (obr. 4.1 b), c)). Tento sousední uzel B poté celý proces zopakuje se svými sousedy (obr. 4.1 d), e), f), převzatý z lit. [12]).

Výměna ADV zpráv s meta-daty před samotným přenosem dat zajišťuje, že sítě nejsou šířena nadbytečná data. Jednou z výhod SPIN protokolu je, že změny topologie sítě jsou vždy jen lokální, poněvadž uzel musí znát pouze bezprostřední sousedy. Nevýhodou je, že oznamovací způsob komunikace nezaručuje doručení dat. [1, 12]



Obr. 4.1: Vyžádání dat u SPIN protokolu

4.1.2 Řízené šíření

Řízené šíření (direct diffusion) je data-centrický protokol, jehož základní myšlenkou je agregovat data přicházející z odlišných zdrojů vyloučením nadbytečnosti a snížením počtu přenosů, a takto šetřit energii a prodloužit životnost sítě. Data jsou

značkována několika atributy. Řízené šíření se skládá z několika elementů: zájmů, datových zpráv a gradientů. Zájem popisuje úkol, který specifikuje požadavky uživatele. Tento popis je složen z několika vlastností resp. atributů. Jednotlivé atributy pro popis úlohy sledování vozidla mohou vypadat například takto:

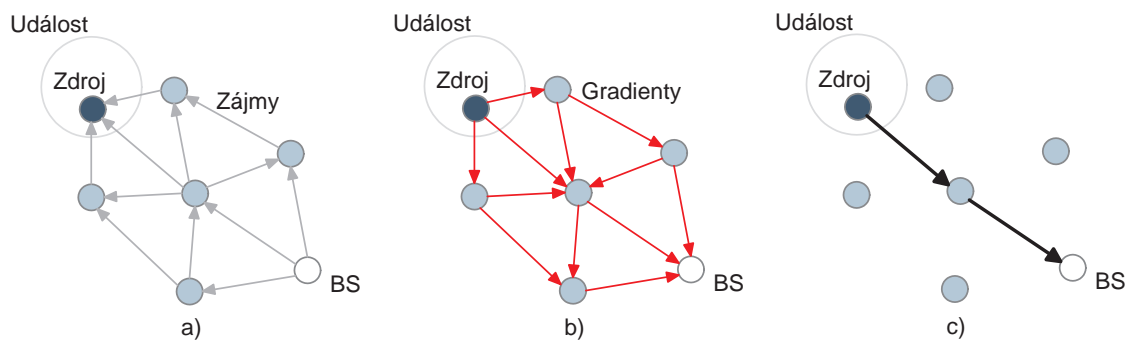
```

type = wheeled vehicle           // detekuj pozici vozidla
interval = 20ms                  // pošli každých 20 ms
duration = 10 seconds            // po dobu 10 sekund
rect = [-100, 100, 200, 400]    // souřadnice požadované oblasti

```

Základová stanice požádá o data vysláním zájmů. Zájmy se rozšiřují po síti skokově a jsou šířeny všemi uzly k jejich sousedům. Každý uzel, který obdrží zájmy nastaví gradient. Gradient je směr nastavený vůči sousednímu uzlu, od kterého byl přijat zájem. Takto komunikace pokračuje, dokud nejsou gradienty nastaveny od zdroje zpět do základové stanice. V této fázi nejsou detekovány smyčky, ale jsou odstraněny později. Nyní je trasa vytvořena z několika cest a nejlepší cesta se „vytuzí“. Data jsou agregovány uvnitř sítě. Fungování řízeného šíření je zobrazeno na obr. 4.2 z lit. [8].

Tento protokol není vhodné použít pro aplikace, kde je potřeba kontinuálního doručování dat do základové stanice např. monitorování prostředí. Dále je nepoužitelný pro jednorázové požadavky. Není výhodné nastavovat gradienty pro požadavky, které použijí vytyčenou trasu pouze jedenkrát. [1, 8]



Obr. 4.2: Řízené šíření: a) vysílání zájmů, b) nastavení gradientů, c) vytužení trasy

4.1.3 Rumor routing

Rumor routing volně přeloženo jako „zvěstovací směrování“ je určeno pro aplikace, kde se uskutečňují pouze jednorázová nebo krátkodobá měření. Na rozdíl od řízeného šíření, kde jsou cesty zřízeny pro vícenásobné požadavky. Klíčovou myšlenkou je směrovat požadavky k uzlům, kde byla zaznamenána událost raději než zaplavování

celé sítě pro obdržení informací o nastávajících událostech. Za účelem šíření událostí sítě využívá tento protokol dlouho žijící pakety nazývané agent. Když uzel detekuje událost, přidá ji do místní tabulky nazývané událostní tabulka a vygeneruje agenta. Ten putuje sítě propagující informaci o lokální události vzdáleným uzlům. Uzel, který má zájem o tuto událost vyše požadavek. Není zde potřeba zaplavovat celou síť, což šetří komunikační náklady. Na druhou stranu zřizuje pouze jednu trasu mezi zdrojem a cílem dat na rozdíl od řízeného zaplávání, kde mohou být data směrovány několika trasami. Rumor routing je vhodný, pouze pokud je počet událostí nízký. [1, 12]

4.2 Hierarchické směrování

Tyto techniky směrování mají výhodu ve škálovatelnosti a efektivní komunikaci. Byly původně navrženy pro sítě s metalickými linkami. Hierarchické směrování obsahuje dvě vrstvy. První je využita pro výběr vedoucího skupiny a druhá pro samotné směrování. V této architektuře mohou být uzly s vyšší energetickou kapacitou použity pro zpracování dat a následné odeslání informací. Zatímco uzly s nižší energií jsou použity pro měření v blízkosti cíle. Z toho plyne, že tvoření skupin a přiřazování speciálních úkolů pro tzv. skupinové vedoucí může vést ke zlepšení škálovatelnosti, energetické efektivitě a životnosti celé sítě. Data putují z nejnižší skupiny do vyšší až se dostanou do základové stanice. Uvnitř skupiny jsou data přenášeny skokově, ale mezi sousedními skupinami překonávají větší vzdálenosti.

4.2.1 LEACH

LEACH (Low Energy Adaptive Hierarchy), což v překladu znamená nízkoenergetická přizpůsobivá skupinová hierarchie. Raději než vynucovat, aby všechna data z uzlů byla doručena do základové stanice, která monitoruje prostředí, uzly v jedné oblasti mohou spolupracovat a posílat pouze jediný sumarizační paket pro celou skupinu. Uzly jsou rozděleny do skupin, kde každá obsahuje skupinového vedoucího. Role skupinového vedoucího se volí náhodně a je prohazována postupně tak, aby bylo dosaženo stejného objemu spotřebované energie pro všechny uzly. Pro redukci přeslechů vně skupiny a mezi sousedními skupinami je využito TDMA/CDMA kontroly přístupu k médiu.

Každá směna u LEACH je rozdělena do dvou fází – nastavovací fáze a fáze měření. V nastavovací fázi jsou zorganizovány skupinky a vybráni vedoucí skupin. Při fázi měření jsou samotná data doručována do základové stanice a její trvání je delší než nastavovací fáze. Během nastavovací fáze se uzly náhodně rozhodnou, zda

budou skupinovým vedoucím. Uzly které se takto rozhodnou poté rozešlou oznamovací zprávu, aby oznámili celé síti, že jsou skupinovými vedoucími. Na základě síly signálu těchto oznámení se všechny ostatní uzly v síti (nejsou-li skupinovými vedoucími) rozhodnou, do které skupiny chtějí patřit. Poté to oznámí příslušnému vedoucímu. Skupinový vedoucí po přijetí všech zpráv od uzlů, které chtějí patřit do jeho skupiny, a na základě počtu uzlů vytvoří rozvrh časového multiplexu (TDMA – Time Division Multiple Access) a přidělí každému uzlu časový slot, po jehož dobu může vysílat. Tento rozvrh rozešle všem uzlům ve skupině. Během fáze měření uzly snímají a odesílají data skupinovým vedoucímu. Skupinový vedoucí po obdržení všech dat provede jejich agregaci a pošle je základové stanici. Po určité době síť začne opět nastavovací fází a začíná nová směna. Doba směny je určena tak, že každý uzel může být v roli skupinového vedoucího jednou za dobu své životnosti. Aby byly eliminovány přeslechy uzlů ze sousedních skupin, tak každá skupina používá jiné kódování CDMA (Code Division Multiple Access).

Jedna z možností využití tohoto protokolu je pro konstantní monitorování prostředí, kde uživatel nepotřebuje data okamžitě. Naopak není vhodný pro rozložení uzlů na velkém prostranství, protože velké vzdálenosti mezi skupinovými vedoucími a základovou stanicí jsou na úkor velké spotřeby. [1],[12]

4.2.2 PEGASIS

PEGASIS (Power-Efficient Gathering in Sensor Information Systems) Protokol pracuje na principu řetězové komunikace. Ideou protokolu je, že za účelem prodloužit životnost sítě komunikují uzly pouze se svými nejbližšími sousedy a ti následovně se základovou stanicí. Když skončí kolo komunikace uzlů se základovými stanicemi, začíná nové. Na rozdíl od LEACH protokolu, PEGASIS nevyužívá rozdělení do skupin a ke komunikaci se základovou stanicí využívá pouze jednoho uzlu v řetězci. Nejbližší uzel se určí dle intenzity signálu a poté si uzel nastaví sílu signálu na svém rádiu tak, aby byl dosah pouze k tomuto nejbližšímu uzlu. Komunikační řetěz bude složen z těchto nejbližších uzlů a sestaví trasu do základové stanice. PEGASIS předpokládá, že každý uzel může komunikovat se základovou stanicí přímo.

V praxi se k dosažení základové stanice používá víceskokové komunikace. Výsledky simulací ukázaly, že tento protokol prodlužuje životnost sítě dvakrát než při použití LEACH protokolu. Předpokládá se, že všechny uzly jsou pevně umístěny a nebudou se pohybovat, pohybující se uzly naruší funkčnost protokolu. [1]

4.2.3 TEEN a APTEEN

Byly navrženy pro časově kritické doručení dat. U TEEN (Threshold-sensitive Energy Efficient sensor Network Protocol) uzly snímají kontinuálně, ale datové přenosy jsou méně časté. Uzel, který je vedoucí skupiny pošle svým členům vysoký práh, což je prahová hodnota měřené veličiny a nízký práh, což je malá změna měřené veličiny, při které se zapíná vysílač a uzel začne vysílat. Nízký práh snižuje nadbytečnost vysílání, která by mohla nastat při nízké nebo žádné změně snímané veličiny. Pokud je změna menší než velikost nízkého prahu, tak se nevysílá. Snížení hodnoty nízkého prahu nám poskytuje podrobnější přehled nad událostmi v síti, ovšem za cenu vyšší spotřeby. Vysoký práh umožňuje uzlům vysílat pouze hodnoty, které jsou v zajímavém rozsahu. Změřená hodnota (ZH) je uložena ve vnitřní paměti do stejné pojmenované proměnné. Uzel bude vysílat data pouze při splnění jedné z těchto podmínek. Za prvé aktuální hodnota změřené veličiny je větší než vysoký práh. Za druhé aktuální hodnota se liší od hodnoty uložené v proměnné ZH o číslo větší nebo rovné nízkému prahu. Hlavní nevýhodou protokolu je to, že pokud nejsou hodnoty prahů doručeny ke koncovým uzlům, tak ty nebudou nikdy komunikovat se zbytkem sítě.

Skupinový vedoucí u APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) šíří do sítě tyto parametry: *Vlastnosti* - fyzické parametry, o kterých chceme znát určité informace. *Prahy* - obsahující vysoký a nízký práh. *Rozvrh* - TDMA rozvrh přiřazující každému uzlu slot. *Časovač* - maximální velikost časové periody mezi jednotlivými zprávami vyslanými uzlem. Uzly stejně jako u TEEN měří prostředí nepřetržitě a vysílají pouze po splnění jedné z podmínek uvedených výše. Ale pokud uzel neodešle data po dobu rovnou časovači, je vynuceno snímání a odeslání dat.

Výhodou protokolu je nabídnutí velké flexibility umožněním uživateli nastavování časovače a prahů, čímž se může kontrolovat spotřeba sítě (vyšší hodnoty časovače = nižší spotřeba). Naopak složitá implementace těchto funkcí je jednou z nevýhod. [1],[7]

4.3 Geografické směrování

Při tomto způsobu směrování jsou uzly adresovány na základě jejich pozice. Pokud jsou uzly vybavené malým nízko energetickým GPS přijímačem, mohou svoji pozici určovat komunikací se satelitem. Vzdálenost mezi sebou uzly určují dle intenzity přijatého signálu. Některé systémy mohou za účelem úspory energie vyžadovat, aby se uzly bez aktivity přepnuly do režimu spánku. Největších úspor je dosaženo, pokud

je v režimu spánku tolik uzlů kolik je jen možné. Ve zbytku kapitoly bude uvedeno několik základních protokolů geografického směrování.

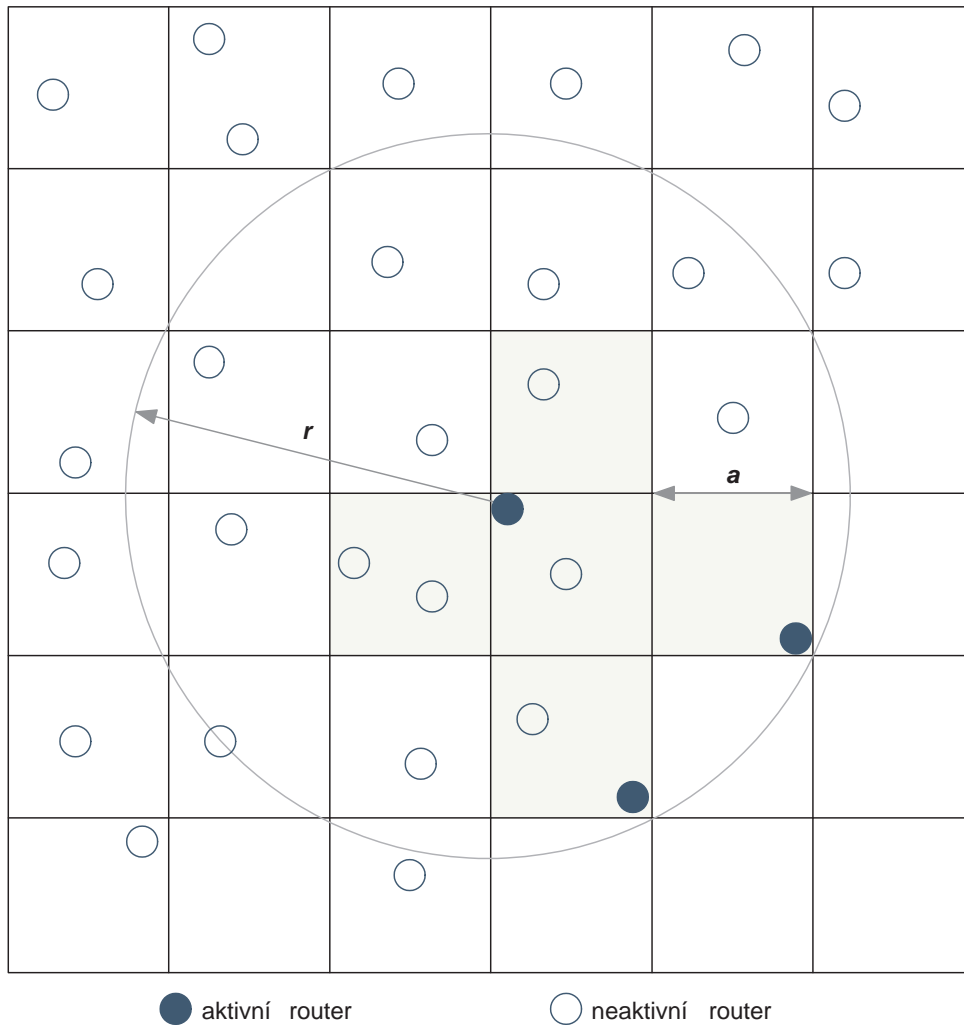
4.3.1 GAF

GAF (Geographic Adaptive Fidelity) je algoritmus navržený primárně pro mobilní ad-hoc sítě, ale dobře aplikovatelný i pro účely WSN. Síť je rozdělena do pevných buněk, které tvoří virtuální rastr. Uzly spolupracují v buňkách. Jeden uzel v buňce je vždy vybrán, aby zůstal aktivní po určitý časový interval a ostatní přejdou do režimu spánku. Tento uzel poté komunikuje se základovou stanicí, sám monitoruje a odesílá data za celou buňku. Volba velikosti buněk (čtverců) se odvíjí od velikosti vysílacího rádiusu. Strana čtverce se určí dle vzorce $a = \frac{r}{\sqrt{5}}$, kde r je maximální přenosový rádius. Jak je znázorněno na obrázku 4.3, nyní mohou uzly komunikovat v bezprostředně sousedních přiléhajících buňkách přímo a to ve čtyřech směrech. Buňky sousedící pouze rohem mohou být také v dosahu, ale vždy by to nebylo možné. GAF definuje tři fáze. První z nich je *objevování* - pro určení sousedů v buňce. Dále *aktivní fáze* - uzel se aktivně zúčastňuje směrování a poslední fáze je *spánek*, kdy je rádiová jednotka vypnuta. Každý uzel začíná stavem objevování, kdy je rádio zapnuto a uzel je připraven přepnout se do aktivního stavu. Uzly setrvávají v tomto stavu po pevně určený časový interval T_O . Pokud časovač vyprší, uzel se přepne do aktivního stavu, kde opět setrvává po čas T_A . Po vypršení se znovu přepíná do objevovacího stavu. Jakmile uzel změní stav z objevovacího do aktivního, pošle zprávu obsahující ID uzlu, ID buňky a hodnotu důležitosti měřené veličiny. Pokud uzel ve stavu objevování nebo aktivity obdrží zprávu od uzlu ze stejné buňky s vyšší hodnotou důležitosti, může se přepnout do režimu spánku na dobu T_S .

GAF šetří energii přepínáním uzlů do stavu spánku, ovšem bez snížení kvality směrování. Při zvyšování hustoty sítě počet aktivních uzlů v rastru zůstává stejný, ale počet uzlů v buňkách roste. Tím jsou uzly mnohdy zdvojené. Proto mohou některé uzly ve stejných částech buňky zůstat ve stavu spánku déle, což navyšuje životnost celé sítě. [1],[12],[13]

4.3.2 GEAR

Hlavním úkolem je snížit počet žádostí, jež využívá protokol řízeného šíření zaplavením pouze určitých regionů, než šíření zájmů do celé sítě. Tímto dochází ke značné úspoře energie oproti řízenému šíření. Každý uzel zná odhadované a naučené náklady pro dosažení cíle za pomoci svých sousedů. Odhadované náklady jsou spojení zbytkové energie a vzdálenosti k cíli. Naučené náklady vylepšují odhadnuté náklady, tím, že berou v úvahu směrování okolo výluk v síti. Pokud v síti nejsou



Obr. 4.3: GAF: výpočet délky strany buňky

výluky, náklady se rovnají. Výluky nastávají, pokud uzel nemá souseda blíže k cíli než sám sebe. Algoritmus sestává ze dvou částí.

1) Doručení paketů do cílové oblasti. Uzel poté co obdrží paket prověří okolní uzly, zda-li nemá některý blíž k cíli než on sám. Pokud je takových uzlů více, vybere se uzel nejbližší k cíli. Pokud na trase k cíli žádný sousední uzel není, znamená to výluky a musí se vybrat jiný sousední uzel pro směrování na základě naučených nákladů.

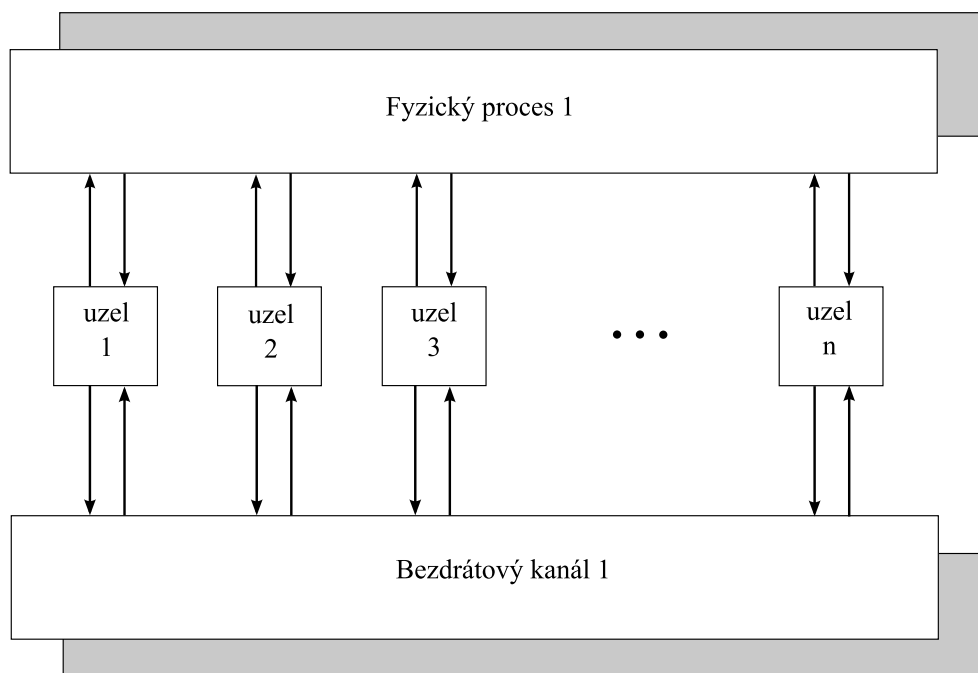
2) Doručení paketů uvnitř oblasti. Jakmile paket dorazí do oblasti, může být rozptýlen rekurzivním geografickým směrováním, nebo ohraničeným zaplavováním, které je vhodné řídicí osazené sítě. Ve vysoce zahuštěných sítích je nejvhodnější geografické rekurzivní směrování. V tomto případě je oblast rozdělena do čtyř podoblastí, do kterých jsou rozeslány nově vytvořené čtyři kopie přijatého paketu. Tento proces pokračuje dokud nezbudou regiony obsahující pouze jeden uzel. [1]

5 SIMULAČNÍ NÁSTROJ CASTALIA

5.1 Úvod

Castalia je simulační program určený pro bezdrátové senzorové sítě. Program je určen pro vývojáře a výzkumníky, kteří chtějí testovat nové algoritmy a protokoly s reálným bezdrátovým kanálem a chováním uzlů. Za pomoci Castalie můžeme také posuzovat různé vlastnosti specifických aplikací, poněvadž jsou zde pro mnoho platform velike možnosti nastavení parametrů a simulací. Castalia pro svůj běh využívá program OMNeT, který je doporučené instalovat na operační systém Linux. V našem případě byl použit operační systém Linux Ubuntu 9.04 kódové označení Jaunty Jackalope. OMNeT může pracovat jak v grafickém rozhraní, tak v příkazovém řádku. Castalia využívá pouze rozhraní příkazového řádku, protože je to jednodušší pro spouštění dávkových úkolů. Grafické rozhraní může být užitečné pro konečné ladění algoritmů, ale pro Castalii se nevyužívá. Popis struktury Castalie a simulace v Castalii jsou čerpány z uživatelského návodu [2].

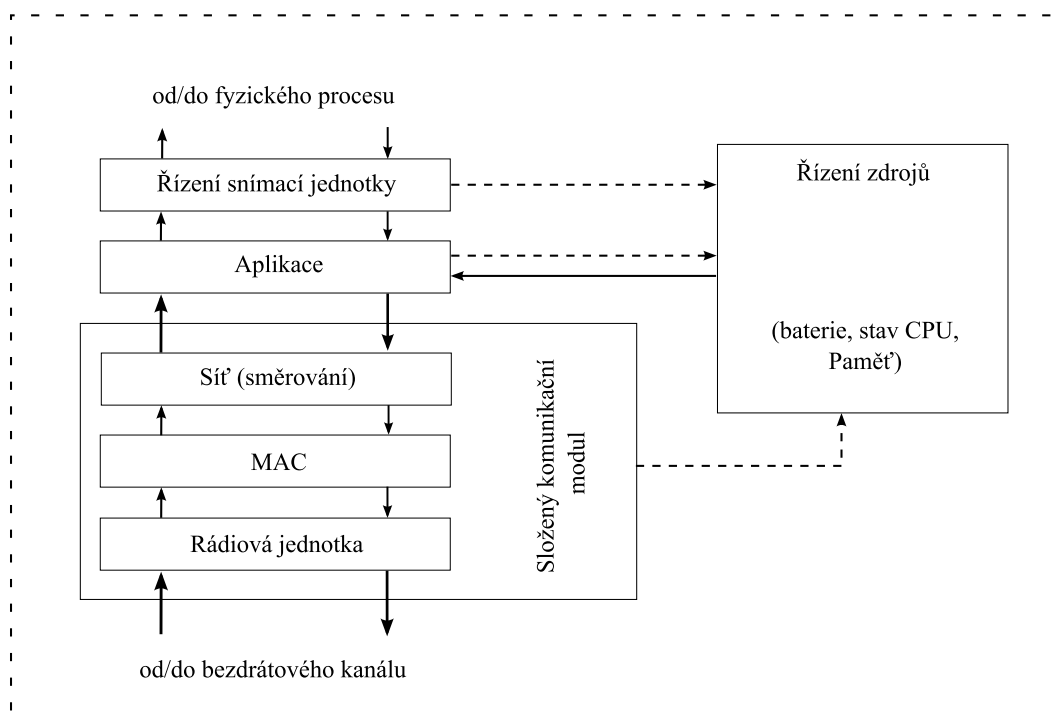
Základními stavebními prvky OMNeTu jsou moduly a zprávy. Základem pro vykonávání příkazů je jednoduchý modul. Ten přijímá zprávy od ostatních modulů, nebo své vnitřní zprávy a na základě takto přijatých zpráv provede daný kus programu. Moduly mohou být i složené z několika jednoduchých modulů, ty pak nazýváme složené moduly.



Obr. 5.1: Struktura Castalie

Struktura modulů je uvedena na obrázku 5.1. Uzly se navzájem nespojují přímo, ale prostřednictvím modulu bezdrátového kanálu. Šipky ukazují zprávy přicházející od jednoho modulu k druhému. Když má uzel paket k odeslání, pošle jej do bezdrátového kanálu, který poté rozhodne kterému uzlu jej doručí. Uzly jsou dále spojeny prostřednictvím fyzických procesů které monitorují.

Vnitřní struktura modulu sensorového uzlu je zobrazena na obrázku 5.2. Uzly jsou ve skutečnosti složeny z několika modulů, které většinou definují jednotlivé komponenty skutečných uzlů. Přerušované šipky zobrazují rozhraní mezi moduly s voláním jednoduchých funkcí a plné šipky zobrazují přenášené zprávy. Například, většina modulů volá funkci správce zdrojů, aby oznámila že bylo spotřebováno určité množství energie. Uživatel může neomezeně měnit modul aplikace, obvykle vytvořením nového modulu, do kterého poté implementuje vlastní algoritmus. Stejně tak pro komunikační moduly MAC a směrování může uživatel implementováním vlastních směrovacích a MAC modulů vytvořit nové moduly. Všechny moduly obsažené v Castalii jsou laditelné pomocí mnoha parametrů. Tato struktura je v Castalii implementována jazykem OMNeT++ NED. Tímto jazykem můžeme jednoduše definovat parametry modulů jako jméno, rozhraní a pokud je modul složený tak popsat strukturu jeho vnitřních modulů.



Obr. 5.2: Vnitřní struktura uzlu v Castalii

5.2 Simulace v Castalii

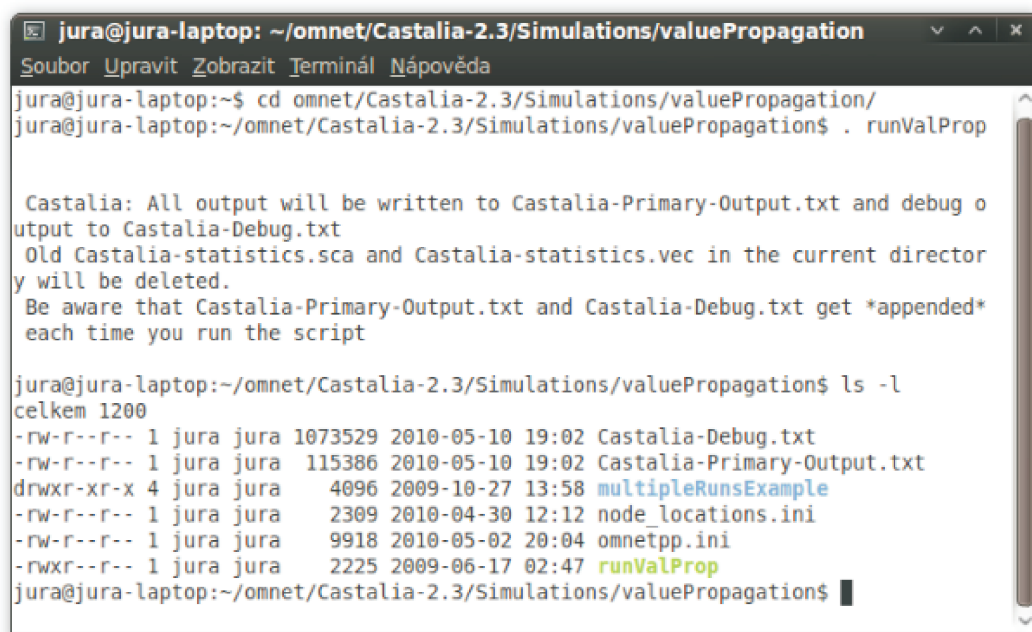
Ve složce `Castalia/Simulations/valuePropagation` jsou umístěny dva soubory `omnetpp.ini` a `runValProp`. Tato složka je pro názornou ukázkou možností simulace v Castalii. `Omnetpp.ini` je konfigurační soubor, který definuje všechny potřebné parametry využívané omnetem i Castalii. `runValProp` je skript, kterým se spouští samotná simulace. Pro její spuštění se v terminálu přepneme do výše zmíněného adresáře a příkazem: `./runValProp` spustíme simulaci. Vypíší se informace o proběhnuté simulaci a název výstupního souboru (obr. 5.3), do kterého se ukládají výstupní hodnoty, nazvaného `Castalia-Primary-Output.txt`. Jak je vidět na obrázku soubor se vytvořil do domovského adresáře simulace, spolu s debugovacím souborem `Castalia-Debug.txt`. Začátek textového souboru obsahuje standardní výstupy z OMNeTu, poté následují informace o načtených `.ned` souborech. Dále jsou již výsledky dané simulace a následuje výpis nastavení jednotlivých parametrů. Výsledky mohou vypadat například takto:

```
finish() at end of Run #1...
Node [0] Value: 0
Node [0] spent energy: 0.249913
Node [1] Value: 0
Node [1] spent energy: 0.249916
Node [2] Value: 0
Node [2] spent energy: 0.249918
Node [3] Value: 0
Node [3] spent energy: 0.249915
Node [4] Value: 0
Node [4] spent energy: 0.249919
Node [5] Value: 0
Node [5] spent energy: 0.249924
Node [6] Value: 40.131
Node [6] spent energy: 0.263053
Node [7] Value: 0
Node [7] spent energy: 0.249909
Node [8] Value: 0
Node [8] spent energy: 0.249918

End run of OMNeT++
```

Výsledky ukazují spotřebovanou energii pro každý uzel a získanou hodnotu naměřeného fyzického procesu. Aplikace sleduje efekt kanálu a parametrů několika MAC úrovní pro přenos hodnot do celé sítě. Uzly začínají snímáním fyzického procesu, každý získá určitou hodnotu (např. teplotu). Pokud tato hodnota překročí nastavený práh (15°C), musí být zaslána. Pokud uzel tuto hodnotu obdrží od jiného uzlu, pokusí se ji zaslat a nastaví svůj praporek oznamující, že splnil svoji úlohu. Každý uzel se řídí podle pravidel na různých MAC úrovních daných jejich parametry. Na základě tohoto chování a na stavu kanálu mezi uzly je hodnota šířena

sítí. Na vzorových výsledcích simulace je vidět, že pouze uzel 6 překročil práh 15°C ($> 40^{\circ}\text{C} + \text{šum}$), proto je jediný zdroj hodnoty pro šíření sítí.



```
jura@jura-laptop: ~/omnet/Castalia-2.3/Simulations/valuePropagation
Soubor Upravit Zobrazit Terminál Nápověda
jura@jura-laptop:~$ cd omnet/Castalia-2.3/Simulations/valuePropagation/
jura@jura-laptop:~/omnet/Castalia-2.3/Simulations/valuePropagation$ . runValProp

Castalia: All output will be written to Castalia-Primary-Output.txt and debug o
utput to Castalia-Debug.txt
Old Castalia-statistics.sca and Castalia-statistics.vec in the current director
y will be deleted.
Be aware that Castalia-Primary-Output.txt and Castalia-Debug.txt get *appended*
each time you run the script

jura@jura-laptop:~/omnet/Castalia-2.3/Simulations/valuePropagation$ ls -l
celkem 1200
-rw-r--r-- 1 jura jura 1073529 2010-05-10 19:02 Castalia-Debug.txt
-rw-r--r-- 1 jura jura 115386 2010-05-10 19:02 Castalia-Primary-Output.txt
drwxr-xr-x 4 jura jura 4096 2009-10-27 13:58 multipleRunsExample
-rw-r--r-- 1 jura jura 2309 2010-04-30 12:12 node_locations.ini
-rw-r--r-- 1 jura jura 9918 2010-05-02 20:04 omnetpp.ini
-rwxr--r-- 1 jura jura 2225 2009-06-17 02:47 runValProp
jura@jura-laptop:~/omnet/Castalia-2.3/Simulations/valuePropagation$
```

Obr. 5.3: Průběh simulace

Parametry jednotlivých modulů se nastavují v souboru `omnetpp.ini`, v němž jsou umístěny jen ty nejdůležitější. Aby všechny parametry nemusely být v jednom velkém souboru, ve kterém by se uživatel jen velmi těžko orientoval, byly vytvořeny repositáře `.ini` souborů umístěné ve složce `Castalia/Simulations/-Parameter_Include_Files`. Každý z těchto souborů definuje parametry jednotlivých modulů, které Castalia obsahuje, jak již bylo zmíněno výše. Soubory se načítají do `omnetpp.ini` pomocí `include` příkazů. Castalia obsahuje více `.ini` souborů pro jednotlivé moduly, ve kterých jsou nastaveny odlišné parametry. Například jsou zde definovány dva soubory pro rádiový modul uzlu a tři MAC moduly. Soubor `omnetpp.ini` se skládá z několika částí. První je označena jako `[General]` a obsahuje cestu k souboru, jenž obsahuje seznam všech `.ned` souborů, které chceme dynamicky načíst. Je nutno se přesvědčit, aby cesty k tomuto souboru byly správně zadané vzhledem k aktuálnímu umístění souboru `omnetpp.ini`. Nakonec jsou definovány názvy dvou výstupních souborů, do kterých jsou ukládány výsledky simulací. Z důvodu používání rozhraní příkazové řádky musí být nastavena část `[Cmdenv]`. Pro tyto potřeby je načten soubor `../Parameter_Include_Files/omnet_cmdenv_reporting.ini`, ve kterém jsou příkazy pro OMNeT, které pro běžné simulace není potřeba měnit. Poslední částí je `[Parameters]`, ve které jsou specifikovány pouze parametry Castalie. Zde jsou parametry k nastavení celé sítě. `SN.field_x` a `SN.field_y` určují rozměry

pole, na kterém budou uzly rozmístěny. Dále můžeme nastavit způsoby rozmístění uzlů. Následují `include` příkazy k načtení jednotlivých souborů z repositáře `Parameter_Include_Files` a parametry jednotlivých modulů. Úplně posledním ze všech parametrů je `[Run]`, který definuje parametry pro vícenásobné kola simulací, označují se jako `[Run1]`, `[Run2]`, `...`, `[Run n]`. Pod příkazy jednotlivých kol se vkládají přímo parametry z konfiguračních souborů modulů, které chceme v daném kole pozměnit.

Jak již bylo zmíněno výše, celá struktura se v Castalii skládá z různých modulů, které se snaží přiblížit svými parametry reálným vlastnostem. Stejně tak uzly jsou poskládané z modulů. Při simulaci můžeme jejich parametry měnit. Nyní budou popsány důležité moduly, které Castalia obsahuje, opět zde bylo čerpáno z uživatelské příručky [2].

5.2.1 Bezdrátový kanál

Při modelaci bezdrátového kanálu je důležité určit přibližný průměrný útlum trasy mezi dvěma body v prostoru, konkrétně mezi dvěma uzly. Vzdálenost dvou uzlů ve WSN se pohybuje od několika metrů do několika stovek metrů. Ukázalo se že nejpřesnější výsledky poskytuje lognormální stínící model. Útlum trasy vyjádřený v decibelech v závislosti na vzdálenosti mezi dvěma uzly je vyjádřen jako:

$$PL(d) = PL(d_0) + 10 \cdot \eta \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (5.1)$$

Kde $PL(d)$ je útlum trasy se vzdáleností uzlů d , $PL(d_0)$ je známý útlum na referenční vzdálenosti d_0 , η je exponent útlumu a X_σ je náhodná proměnná s Gaussovým (normálním) rozložením v oblasti okolo nuly se standardní odchylkou sigma.

Bereme-li v úvahu souvztažnost mezi oběma směry kanálu, není lognormální stínovací model nejpřesnější. Pokud uvažujeme dva směry kanálu jako nezávislé linky, bude odchylka mnohonásobně vyšší než reálné hodnoty. Pro tyto případy je použit model pro určení průměrného útlumu pro oba směry odečtením zvláštní proměnné s Gaussovským rozdělením a standardní odchylkou nastavované parametrem `SN.wirelessChannel.bidirectionalSigma`

Signál přijatý přijímačem lze vypočítat na základě útlumu trasy dané součtem průměrného útlumu a dočasného kolísání a vysílaného výkonu vysílače. Při znalosti šumu nebo rušení přijímače lze spočítat poměr signál/šum (SNR - signal-to-noise ratio, také SIR). Ze SNR a parametrů rádiové části uzlu lze určit pravděpodobnost příjmu paketů na uzlu od konkrétního vysílače. Neměnnou složkou interferencí je tepelný šum. Více současných přenosů může též působit jako rušení. Castalia dynamicky počítá rušení od sousedních uzlů a takto získává výsledný SNR a pravděpodobnost příjmu paketů.

Pro jednoduchost jsou v adresáři `/Simulations/Parameter_Include_Files/WChannel/` tři podadresáře, ve kterých je vždy několik `.ini` souborů s přednastavenými parametry bezdrátového kanálu pro simulaci jeho určitých parametrů. Všechny parametry začínají prefixem `SN.wirelessChannel`, za kterým je název konkrétního parametru a následuje jeho hodnota.

Důležitým parametrem je: `SN.wirelessChannel.collissionModel` - využíván pro jednodušší modelování kolizí. Pokud je tato proměnná nastavena na hodnotu 0, nedochází při přenosu k žádným kolizím. Při 1 je nastaven jednoduchý model kolizí, u nějž pokud dva uzly současně vysílají a přijímač může přijmout jejich signál, dokonce i velice slabý, pak je na přijímači vždy detekována kolize. Při nastavení proměnné na 2 je použit přídavný model kolizí, kde jsou přijaté signály od jiných uzlů počítány jako rušení a lineárně přičteny k SIR přijímače.

Ostatní parametry v těchto konfiguračních souborech udávají hodnoty $PL(d)$, d , $PL(d_0)$, d_0 , η a X_σ dosazované do rovnice 5.1.

5.2.2 Rádiová jednotka

Modul rádiové jednotky se snaží co nejvěrněji napodobit vlastnosti obecné nízkoenergetické rádiové jednotky používané v senzorových sítích. Podporuje několik stavů, ve kterých uzel setrvává - přenos, příjem/poslech a spánek, kde každý stav má odlišnou spotřebu energie a zpoždění, se kterým se přepíná z jednoho stavu do druhého. Modul poskytuje několik úrovní přenosového výkonu.

V adresáři `Simulations/Parameter_Include_Files/Radio` jsou umístěny konfigurační soubory pro dva moduly rádiové jednotky `TelosB_CC2420` a `Mica2_CC1000`. V nich jsou přednastaveny parametry pro tyto jednotky tak, aby co nejvěrněji odpovídaly reálným vlastnostem rádií. Zde lze dle datasheetů vytvořit i vlastní jednotku. Všechny parametry začínají řetězcem `SN.node[*].networkInterface.Radio`. a za ním následuje název konkrétního parametru. Dále budou uváděny pro jednoduchost pouze tyto názvy.

- `dataRate` - udává datovou rychlost.
- `bitsPerSymbol` - slouží k výpočtu bitové rychlosti, což vyžadují některé MAC protokoly.
- `noiseBandwidth` - udávaný v kHz - šířka pásma šumu.
- `noiseFloor` - tepelný šum.

- **modulationType** - nastavuje typ modulace přenášených dat. Možnosti nastavení jsou IDEAL (ideální modulace, bez přenosových chyb), FSK (frequency-shift keying), PSK (phase-shift keying) a alternativy (BSK, QPSK, OQPSK) a CUSTOM, kde se zadá poměr SNR:BER.

Výše zmíněné parametry ovlivňují pravděpodobnost přijetí paketu. Následující parametry popisují hardwarové vlastnosti rádiové jednotky.

- **reciverSensitivity** - citlivost přijímače, využívaná modulem bezdrátového kanálu pro přesnější výpočty.
- **rxPower**, **listenPower**, **sleepPower** - udávají spotřebu energie při různých stavech v mW.
- **txPowerLevels** - udává v dBm různé výkony vysílání.
- **txPowerConsumptionPerLevel** - spotřeba pro výkony nastavené v předchozím parametru, výběr vysílacího výkonu se provede parametrem **txPowerLevelUsed**.

Další parametry určují mód vysílání, velikost bufferu, atd. Poslední částí je skupina parametrů, která udává zpoždění při přechodech rádiové jednotky do jednotlivých stavů (spánek, naslouchání, vysílání).

5.2.3 MAC

Chování uzlů je silně ovlivněno MAC protokolem, proto byl vytvořen oddělený modul, který toto chování definuje. V Castalii jsou tři přednastavené MAC moduly: TunableMAC, ve kterém může uživatel odlaďovat vlastní navržené parametry MAC. T-MAC, ze kterého po jednoduché úpravě získáme S-MAC (zakázáním proměnných aktivních period).

Tunable MAC

Je vysoce laditelný modul pro MAC protokol, který se může po změně parametrů přiblížit k vlastnostem některých skutečných protokolů (B-MAC, LPL). Je vytvořen za účelem broadcastové komunikace, proto nepodporuje potvrzování RTS a CTS kontrolními zprávami. Zde jsou důležité parametry MAC protokolu obsažené v konfiguračním souboru:

- **dutyCycle** - podíl doby, ve které je uzel ve stavu poslouchání a doby spánku. Tento parametr značně ovlivňuje spotřebu energie. Podíl je zadán ve tvaru naslouchání/(naslouchání+spánek).

- `listenInterval` - čas po který zůstává uzel v režimu naslouchání, ovlivňuje `dutyCycle`.
- `BeaconIntervalFraction` - Upozornění spících uzlů před přenosem dat. Standardně se vyšle skupina signálů před přenosem dat. Hodnota vyjadřuje podíl maximálního signálního intervalu (= doba spánku) a nastaveného signálního intervalu. Nižší hodnota je energeticky příznivá, ale je s menší pravděpodobností, že se probudí sousední uzel. Pro garantované probuzení uzlu musíme vyslat signální signál po dobu rovnou délce spánku. Což je nevýhodné z hlediska probuzení ostatních uzlů.
- `probTx` - pravděpodobnost přenosu. V mnoha protokolech se počítá s pravděpodobností 1.
- `numTx` - počet přenosů. Počet pokusů přenosu jedné části dat. Větší hodnota zajišťuje vyšší výkon za cenu vyšší spotřeby
- `randomTxOffset` - náhodný časový interval, ve kterém uzel začne vysílat. Když se uzel rozhodne vysílat data, nezačne je vysílat okamžitě, ale vyšle je v intervalu udaném jako $[0, \text{čas}]$, kde čas je udávám v ms a je horní hranicí intervalu. Tato hodnota pomáhá k předcházení kolizí u broadcastového vysílání.
- `reTxInterval` - Interval mezi opakovaným vysíláním.
- `backoffType` - Mac protokol zajišťuje naslouchání nosné, pokud je potřeba vyslat data, uzel zkontroluje přenosový kanál, zda je volný a lze vysílat. Pokud není kanál volný vysílání je odloženo na určitý čas a MAC protokol přepne rádio do režimu spánku a následující chování ovlivňuje právě parametr `backoffType`. Pokud je jeho hodnota 0, poté je doba odložení rovna době spánku. Při 1 trvá odložení dobu `backoffBaseValue` (následující parametr v konfiguračním souboru). 2 odloží vysílání postupně do časů, kdy kanál nebyl volný.
- `randomBackoff` - parametr určující, jestli je čas odkladu určován náhodně. Hodnota je booleovská - ano/ne.

Zbylé parametry nastavují velikost MAC rámců, velikost signálních rámců, velikost ACK rámců, bufferu a poslední parametr určuje, zda bude uzel naslouchat nosné před samotným přenosem dat.

T-MAC a S-MAC

T-MAC se snaží zachovávat vysoký výkon při co nejnižší spotřebě, čehož dosahuje změnou doby pracovního cyklu podle potřeby. Proto je oblíbeným protokolem ve WSN. S-MAC je jeho předchůdcem, který využívá pevně nastaveného pracovního cyklu. V Castalii je jeden modul nazvaný T-MAC, který obstarává oba tyto protokoly, které se odlišují v nastavení některých parametrů. Konfigurační soubor pro protokol T-MAC se nachází v adresáři `Simulations/Parameter_Include_Files/T_MAC.ini`, kde je i soubor `S_MAC.ini`. Prvními parametry je nastavení velikostí jednotlivých kontrolních zpráv (SYNC, ACK, RTS, CTS). Následuje 13 parametrů definujících chování protokolu. Jejich hodnoty jsou v Castalii nastaveny tak, aby se simulované chování co nejvíce podobalo reálným vlastnostem protokolu. Proto se při vlastních simulacích tyto hodnoty nebudou měnit.

5.2.4 Směrování

Od verze 1.3 jsou v Castalii implementovány dva jednoduché směrovací protokoly. Oba mají své vlastní konfigurační `.ini` soubory. V obou souborech je několik podobných parametrů. Jsou zde parametry pro nastavení velikosti rámců, velikosti kontrolních paketů a bufferu.

Sěrovací technika jednoduchého stromu

Při této technice se využívá jednoduchého stromu. Základová stanice musí iniciovat nastavovací paket, který vyšle do sítě. Uzly, které tento paket přijmou označí jejich odesílatele jako zdroj. Zdrojem může být pouze sousední uzel, rozpoznáný na základě RSSI. Pokud uzel obdrží paket s cílem základové stanice, přepošle jej svému zdroji a ten ho dále přepošle svému zdroji až se data dostanou do základnové stanice. V parametrech lze nastavit maximální počet zdrojů pro jeden uzel, velikost tabulky pro ukládání sousedních uzlů, nebo RSSI (Received Signal Strength Indication) prahovou úroveň pro rozlišení sousedních uzlů.

Sěrování multipath

Zde uzly nemají jednoznačný zdroj dat. Uzel pouze nastavuje číslo úrovně, ve které se nachází. Tyto úrovně tvoří pomyslné kružnice. První nastavovací paket vyslaný základnovou stanicí má úroveň nula. Jakýkoliv uzel jej přijme, nastaví úroveň paketu na 1 a přepošle dále, poté nastaví i svoji úroveň. Takto každý uzel zvyšuje úroveň o 1 do doby, než paket dojde k cíli. Při zpětné komunikaci, když potřebuje uzel zaslat data do základnové stanice, vyšle broadcastově paket a uzel na nižší úrovni jej opět broadcastem vyšle k nižší úrovni, než paket dorazí do ZS. Takto je vytvořeno

mnoho tras k BS. Nevýhodou algoritmu jsou mnohonásobné a překrývající se trasy k dosažení cíle, kde vzniká problém s přeplněním tras.

5.2.5 Fyzický proces

Změřená data se v simulačních nástrojích pro WSN obecně generují tak, že se uzly naplní náhodnými čísly, nebo má každý uzel konstantní hodnotu, nebo jsou uzly naplněny reálnými změřenými daty. Fyzický proces by měl být dostatečně flexibilní a odpovídat co nejvěrněji skutečným procesům. Zdroje hodnot se mohou měnit svou pozici a měřenou hodnotu. Vztah pro model, který určuje hodnotu fyzického procesu v určitém čase a místě je následující:

$$V(p, t) = \sum_{\text{zdroje } i} \frac{V_i(t)}{(K \cdot di(t) + 1)^a} \quad (5.2)$$

Kde $V(p, t)$ je hodnota fyzického procesu v bodě p a v čase t

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

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

K, a - parametry určující jak je hodnota od zdroje vzdálena

Parametry K a a se nastavují přímo v konfiguračním souboru `omnetpp.ini` proměnnými `SN.physicalProcess[0].multiplicative_k` a `SN.physicalProcess[0].attenuation_exp_a`

5.2.6 Modul řízení zdrojů

Tento modul uchovává důležité zdroje uzlů, z nichž nejdůležitější je energie. V konfiguračním souboru je udána počáteční energie napájecího zdroje v Joulech a další méně podstatné parametry. Při komunikaci ostatních modulů s modulem řízení zdrojů se nekomunikuje pomocí zpráv, což je neefektivní, ale přímo se volají funkce manažera. Jeho úkolem je lineárně odečítat objemy spotřebované energie, kterou spotřebovávají ostatní moduly uchovávat tyto hodnoty v paměti. Vývojáři Castalie plánují vyvinout kvalitnější model pro napájení z baterií a sledovat čas výpočtů na základě doby vykonávání operací a režimu procesoru.

6 NÁVRHY SIMULACÍ

Při následujících simulacích budou porovnány dvě komunikační jednotky Mica2 a Imote2 hojně využívané v bezdrátových senzorových sítích. Mica2 má v Castalii přednastavený .ini soubor v základní instalaci. Pro jednotku Imote2 byl využit také přednastavený soubor, ale pro komunikační jednotku TelosB, která využívá stejný čip pro rádiový přijímač (CC2420) jako zmíněná Imote2. Konfigurace jednotlivých parametrů se jen mírně lišila, a proto byly hodnoty parametrů dle datasheetu (lit. [3], [5]) upraveny tak, aby co nejdříve napodobily vlastnosti jednotky Imote2. Tato jednotka je napájena třemi AAA bateriemi. Modul řízení napájení je implementován pouze pro dvě AA baterie a proto byl vytvořen nový modul s upravenými parametry tak, aby byl vhodný pro napájení jednotky Imote2 a byly tak výsledky spotřeby co nejpřesnější. Obě jednotky i se základními parametry jsou popsány v příloze A

6.1 Simulace MAC protokolů

Při simulacích s MAC protokoly byl zvolen časový limit pro běh simulací na 60 s. Všechny naměřené údaje jsou tedy hodnoty pro běh sítě po tomto časovém intervalu, kdyby byl interval větší, byly by například i hodnoty spotřeby energie úměrně vyšší. Do oblasti bylo umístěno 30 senzorových uzlů rozmístěných náhodně rovnoměrně do zvolené oblasti, přičemž Castalia umožňuje několik druhů rozmístění. Náhodné rovnoměrné rozmístění, náhodné do mřížky, rozmístění do mřížky nebo vlastní rozmístění, kde jsou uzly rozmístěny dle souřadnic x a y definovaných v externím .ini souboru. Pro snímací jednotky uzlů byl zvolen modul měřící teplotu. Modul fyzického procesu byl použit `physicalProcess_0_node6_assignedValue40.ini`, který říká, že uzel 6 naměří hodnotu, která je nad nastaveným prahem (40°C) a šíří ji dále sítí, což ukazuje i na konektivitu v síti dle počtu uzlů, které obdrží naměřenou hodnotu od uzlu 6. V reálných aplikacích WSN, kde je využit některý ze směrovacích protokolů není žádoucí, aby všechny jednotlivé uzly mohly komunikovat každý s každým, protože směrovací protokol zajistí trasy pro šíření dat. Bezdrátový kanál byl zvolen realistický, počítající i s rušením. Síťový směrovací modul zde byl přemostěn.

1. experiment – laditelný MAC modul

V prvním experimentu bude porovnána spotřeba energie uzlů pro různé hodnoty parametrů laditelného MAC modulu, který je již implementován. Jeho parametry se mění v konfiguračním souboru `MAC_Tunable.ini`, Jsou zde nastavovány délky

pracovního cyklu (`dutyCycle`) a doby naslouchání (`listenInterval`). `dutyCycle` je bezrozměrná hodnota, daná podílem

$$dutyCycle = \frac{listening}{sleep + listening}.$$

Parametr má v konfiguračním souboru zápis `SN.node[*].networkInterface.MAC.dutyCycle=0,9`. Při hodnotě 1 uzel nikdy nepřejde do režimu spánku. `listenInterval` udává časový interval, po který uzel naslouchá ke komunikaci s ostatními uzly a je udáváný v ms. Zápis parametru se provádí příkazem `SN.node[*].networkInterface.MAC.listenInterval=100`. Lze tedy snadno odvodit dobu, po kterou uzel setrvá v režimu spánku.

$$sleep = \frac{listening}{dutyCycle} - listening \quad [ms]$$

Rozměr oblasti pro rozmístění uzlů byl nastaven na 180×180 m. Výkon rádiové jednotky byl pro každý typ jednotky nastaven rozdílně. Pro Mica2 byl zvolen výkon vysílání na 0 dBm, což je 1 mW. U Imote2 byl zvolen vysílací výkon na -3 dBm (0,50 mW). Parametry laditelného MAC protokolu byly nastavovány pomocí jednotlivých kol simulací.

- **1. kolo**

V prvním kole byly zvoleny hodnoty parametrů tak, aby uzel setrval co nejméně v režimu spánku a co nejdelší dobu naslouchal, což by se značně mělo projevit na spotřebě jednotlivých uzlů.

- `dutyCycle=0,9`
- `listenInterval=100`

Z výše uvedeného vzorce plyne, že doba po kterou uzel setrvá v režimu spánku je 11,11 ms a doba naslouchání na bezdrátovém kanálu je 100 ms.

- **2. kolo**

Při druhém běhu simulace byla změněna pouze hodnota pracovního cyklu. Zde budou intervaly režimů naslouchání a spánku v poměru 1:1.

- `dutyCycle=0,5`
- `listenInterval=100`

Režim spánku i naslouchání jsou dlouhé 100 ms. Spotřeba by měla klesnout, avšak v praxi nejsou tyto intervaly možné z důvodu rušení, velké spotřeby, a také pokud uzly setrvávají polovinu času v režimu naslouchání, není možné doručit všechny hodnoty bez značného zpoždění, což je nevhodné u časově kritických aplikací.

- **3. kolo**

Ve třetím běhu simulace byl zvolen poměr spánek/naslouchání 9:1, což by mělo rapidně snížit spotřebu uzlů. Doba spánku byla 450 ms a doba naslouchání 50 ms.

- `dutyCycle=0,1`
- `listenInterval=50`

- **4. kolo**

V posledním kole simulace byl ponechán stejný poměr spánek/naslouchání jako v předchozím kole, ale doba jednotlivých intervalů byla úměrně zkrácena, abychom mohli na výsledcích zhodnotit vliv této úpravy na spotřebu uzlů a počet uzlů, které obdrží naměřenou hodnotu od uzlu číslo 6. Doba spánku byla rovna 135 ms a doba naslouchání 15 ms.

- `dutyCycle=0,1`
- `listenInterval=15`

Výsledky jsou uvedeny vždy jen pro první tři uzly z celkových třiceti. Kompletní výsledky by byly příliš obsáhlé. Výsledky všech simulací v kompletní formě jsou na příloženém DVD v adresáři **Výsledky**. Pro jednoduchost a lepší čitelnost výsledků byla simulace prováděna pro obě jednotky najednou a jejich rozlišení se opět provedlo pomocí jednotlivých kol. V konfiguračním souboru `omnet.ini` bylo nastaveno 8 kol simulace pro jednodušší čtení a zpracování výsledků. Kola 1-4 jsou pro jednotku Mica2 a kola 5-8 jsou pro Imote2. Parametry MAC modulu jsou nastaveny postupně pro obě jednotky stejně. V druhé části tedy pro běh simulace č. 5 jsou parametry prvního kola, pro běh č. 6 druhého kola atd.

Mica2

Run #1...

Node [0] spent energy: 1.55416
Node [1] Value: 40.236
Node [1] spent energy: 1.55414
Node [2] Value: 40.236
Node [2] spent energy: 1.5543

Run #2...

Node [0] Value: 40.236
Node [0] spent energy: 1.02908
Node [1] Value: 40.236
Node [1] spent energy: 1.0277
Node [2] Value: 40.236

Run #3...

Node [0] Value: 40.236
Node [0] spent energy: 0.52133
Node [1] Value: 40.236
Node [1] spent energy: 0.538985
Node [2] Value: 40.236
Node [2] spent energy: 0.519158

Run #4...

Node [0] Value: 40.236
Node [0] spent energy: 0.498678
Node [1] Value: 40.236
Node [1] spent energy: 0.496197
Node [2] Value: 0
Node [2] spent energy: 0.490607

Imote2

Run #5...

Node [0] Value: 39.945
Node [0] spent energy: 3.71167
Node [1] Value: 39.945
Node [1] spent energy: 3.7113
Node [2] Value: 0
Node [2] spent energy: 3.71183

Run #6...

Node [0] Value: 40.236
Node [0] spent energy: 2.27381
Node [1] Value: 40.236
Node [1] spent energy: 2.26696
Node [2] Value: 40.236
Node [2] spent energy: 2.26297

Run #7...

Node [0] Value: 40.236
Node [0] spent energy: 0.878389
Node [1] Value: 40.236
Node [1] spent energy: 0.881674
Node [2] Value: 40.236
Node [2] spent energy: 0.876911

Run #8...

Node [0] Value: 40.236
Node [0] spent energy: 0.834033
Node [1] Value: 40.236
Node [1] spent energy: 0.841613
Node [2] Value: 40.236
Node [2] spent energy: 0.826378

Výsledky ukazují naměřenou hodnotu (Value) a spotřebovanou energii udávanou v Joulech pro jednotlivé uzly za celou dobu běhu simulace (spent energy). U prvního běhu simulace měly uzly největší spotřebu elektrické energie. Což je způsobeno délkou pracovního cyklu, kdy je uzel většinu času v režimu naslouchání. Komunikační jednotka Imote2 zde měla zhruba dvojnásobnou spotřebu energie. V druhém kole simulace, kde byly oba režimy stejně dlouhé se spotřeba snížila, přičemž pro uzel Imote2 se spotřeba snížila o větší objem energie. Ve zbývajících kolech simulace se spotřeba nadále snižovala, což bylo dáno délkou nastavení pracovního cyklu tak,

že uzly setrvaly většinu doby ve stavu spánku. V posledním kole byl režim naslouchání dosti krátký, což by mohlo mít vliv na počet uzlů, které obdrží změřenou hodnotu.

2. experiment

Ve druhém experimentu budou porovnány protokoly S-MAC a T-MAC. Parametry protokolů S-MAC a T-MAC jsou uvedeny výše viz kap. 2. Jejich nejmarkantnější rozdíl je v době aktivního naslouchání, kdy u S-MAC je nastavena na 61ms a u T-MAC na 15ms. V prvním kole je použit S-MAC a ve druhém kole T-MAC protokol. Opět budou porovnány obě komunikační jednotky. První a druhý běh ukazuje spotřebu jednotky Mica2 a třetí a čtvrtý běh jsou pro jednotku Imote2. Rozměr pole zde byl nastaven na 165×165 m. Síť je záměrně přehušťena, aby všechny uzly obdržely hodnotu a spotřeba tak byla co nejvíce reálná.

Mica2

Run #1...	Run #2...
Node [0] Value: 40.236	Node [0] Value: 40.112
Node [0] spent energy: 0.515948	Node [0] spent energy: 0.428804
Node [1] Value: 40.236	Node [1] Value: 40.112
Node [1] spent energy: 0.516341	Node [1] spent energy: 0.426806
Node [2] Value: 40.236	Node [2] Value: 40.112
Node [2] spent energy: 0.515814	Node [2] spent energy: 0.427805
Node [3] Value: 40.236	Node [3] Value: 0
Node [3] spent energy: 0.515877	Node [3] spent energy: 0.421908

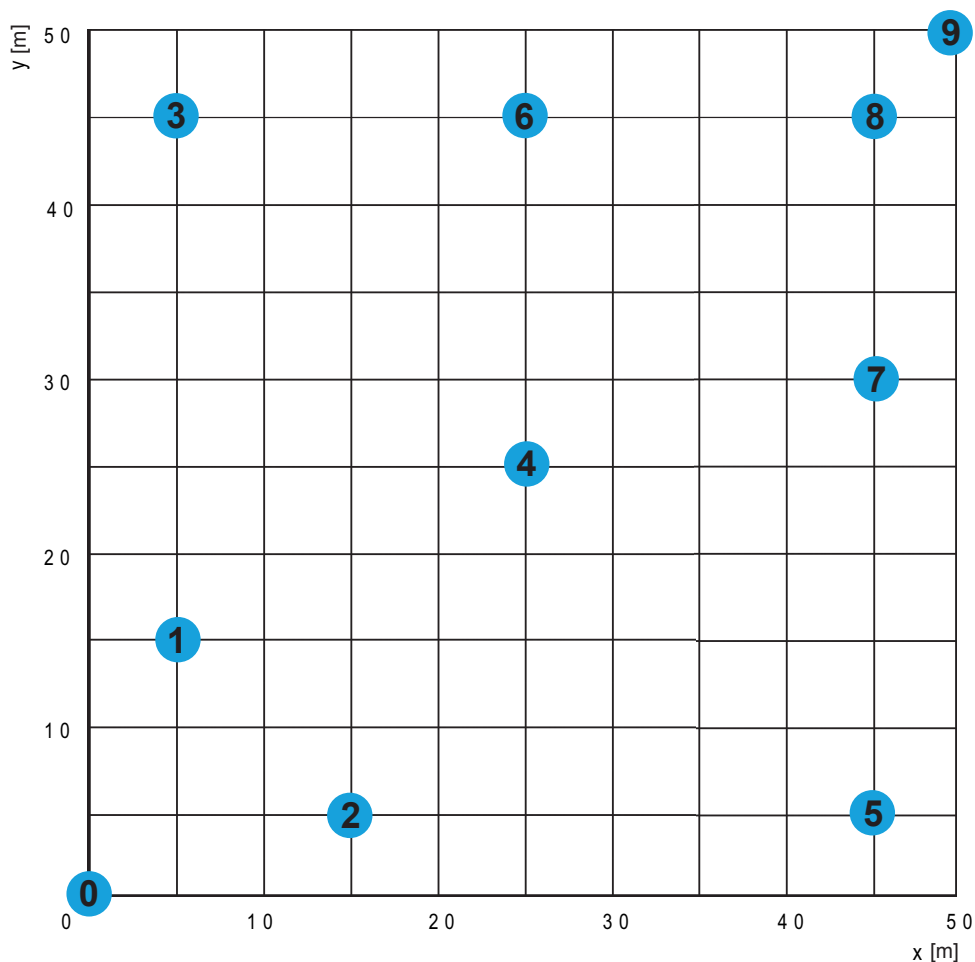
Imote2

Run #3...	Run #4...
Node [0] Value: 40.057	Node [0] Value: 40.236
Node [0] spent energy: 0.873092	Node [0] spent energy: 0.613363
Node [1] Value: 40.057	Node [1] Value: 40.236
Node [1] spent energy: 0.874644	Node [1] spent energy: 0.614139
Node [2] Value: 40.057	Node [2] Value: 40.236
Node [2] spent energy: 0.877316	Node [2] spent energy: 0.614986
Node [3] Value: 40.057	Node [3] Value: 40.236
Node [3] spent energy: 0.87344	Node [3] spent energy: 0.613993

Dle teoretických předpokladů je protokol T-MAC skutečně energeticky úspornější, než S-MAC použitý v prvním a třetím kole simulací. Komunikační jednotka Imote2 zde opět spotřebovává více energie.

6.2 Konektivita sítě

Tato simulace by měla napomoci při určení počtu spojení mezi jednotlivými uzly a ukázat tak celkovou konektivitu sítě. Byl zde použit v Castalii předpřipravený konfigurační soubor v adresáři `Castalia/Simulations/connectivityMap`, kde byl upraven konfigurační soubor `Omnetpp.ini` pro naše účely. Uzly byly rozmístěny do pole o velikosti 50×50 m, kam bylo umístěno 10 sensorových uzlů. Ty byly rozmístěny dle zadaných souřadnic v souboru `node_location.ini`, kde pro každý jednotlivý uzel identifikovaný číslem je určena přesná pozice v kartézské soustavě souřadnic. Rozmístění bylo určeno tak, že uzel [0] je na souřadnicích (0;0) a uzel [9] na (50;50). Tyto uzly jsou od sebe nejvzdálenější, čímž bude sledován maximální dosah při jednotlivých simulacích. Ostatní uzly jsou rozmístěny v oblasti takřka náhodně, aby tak demonstrovaly skutečnou síť a také aby ukazovaly detailně dosah jednotlivých uzlů. Mapa rozmístění uzlů je na obr. 6.1. V prvních dvou kolech simulace jsou



Obr. 6.1: Rozmístění uzlů v síti

porovnány protokoly S-MAC a T-MAC pro komunikační jednotku Mica2. V kolech 3 a 4 jsou opět porovnány zmíněné MAC protokoly, ale pro jednotku Imote2. Obě jednotky měly nastaveny vysílací výkon na 0dBm, díky čemuž budou lépe porovnatelné jejich vlastnosti. Příklad zadání souřadnic pro uzel [0]:

```
SN.node[0].xCoor = 0, SN.node[0].yCoor = 0.
```

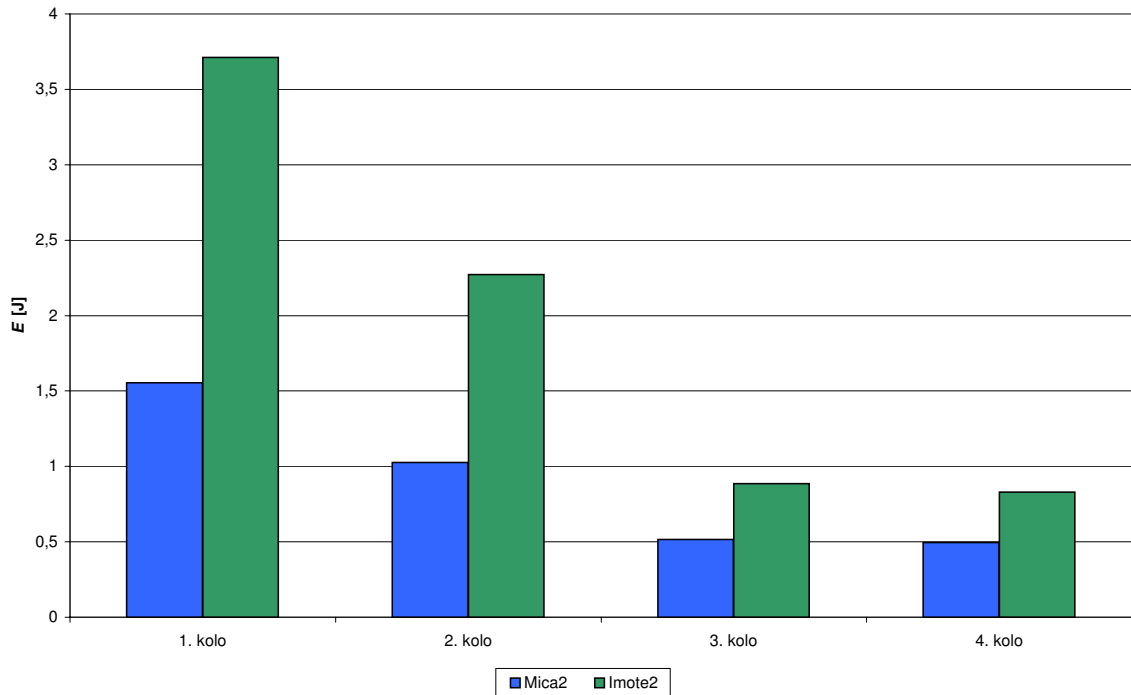
Ve výstupním souboru jsou výsledky v podobě počtu přijatých paketů pro konkrétní uzel od jednotlivých ostatních uzlů. Zde je uvedena jen část výsledků pro uzel [0], které říkají, že obdržel od uzlu [1] 59 paketů ze 100, od uzlu [2] 52 ze 100, od uzlu [3] 32 ze 100, atd.

```
** Node [0] received from:  
[0<--1] --> 59 out of 100  
[0<--2] --> 52 out of 100  
[0<--3] --> 32 out of 100  
[0<--4] --> 32 out of 100  
[0<--6] --> 2 out of 100  
[0<--9] --> 2 out of 100
```

Dále lze pozorovat s kolika uzly bylo navázáno spojení. Zde větší počet nemusí být vždy výhodou, protože síť kde by mohl komunikovat každý s každým není vždy žádoucí. Dále někdy nastal případ, kdy uzly sice komunikovaly, ale byly přijaty např. pouze 2 ze 100 paketů, což je pro komunikaci nevyužitelné.

7 VYHODNOCENÍ VÝSLEDKŮ

Oba experimenty s MAC protokoly se zaměřují na porovnání spotřeby energie uzlů pro různé modifikace laditelného MAC protokolu nebo pro dva známé MAC protokoly. Přičemž srovnává spotřebu pro dvě komunikační jednotky Mica2 a Imote2.



Obr. 7.1: Průměrná spotřeba energie pro jednotlivá kola simulace

První experiment ukazuje spotřebu při různých délkách pracovního cyklu a jeho dvou režimů - spánku a naslouchání. Tyto intervaly jsou jedny z nejdůležitějších vlastností MAC vrstvy co se týče spotřeby uzlů. Jednotlivé časy byly navrženy od nejkratších intervalů režimu spánku až po nejdelší.

Tab. 7.1: Průměrná spotřeba komunikačních jednotek pro jednotlivá kola simulace

kolo	Mica2	Imote2
	spotřeba [J]	
1	1,5541	3,7124
2	1,0268	2,271
3	0,5166	0,887
4	0,4946	0,8305

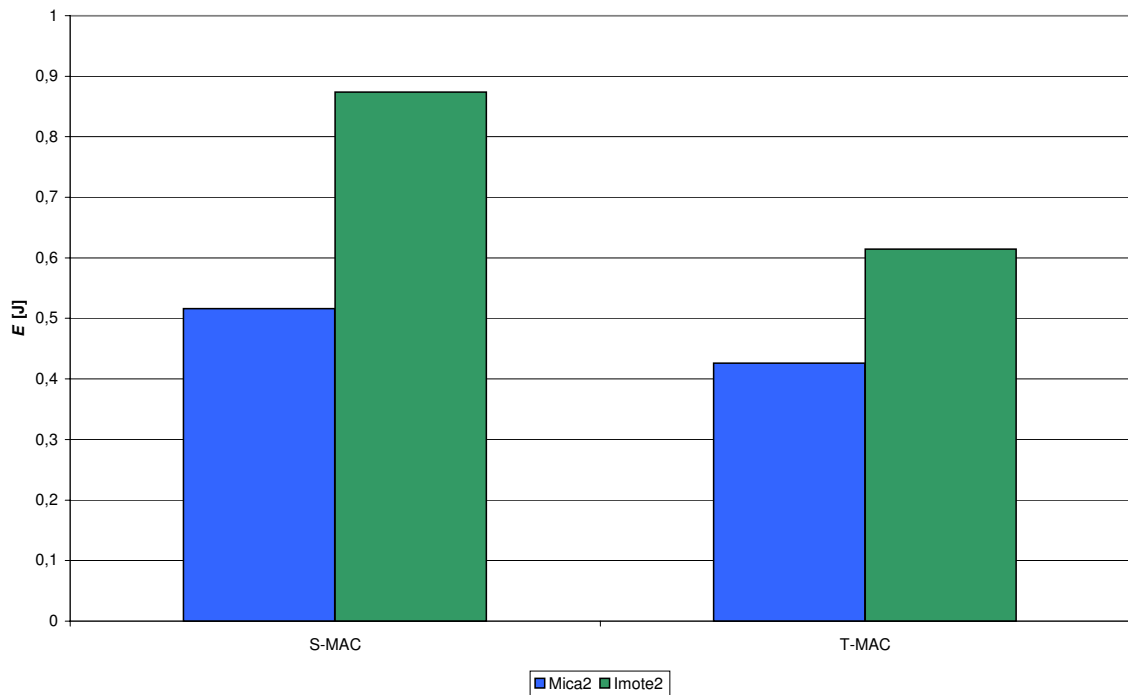
Parametry se nastavovaly v konfiguračním souboru MACtunable, jak již bylo uvedeno v podkapitole 6.1. Výsledky tohoto experimentu jsou na obr. 7.1. Hodnoty spotřebované energie jsou aritmetickým průměrem spotřeby jednotlivých uzlů pro každé kolo simulace. V prvním kole byla nastavená doba režimu naslouchání 9x delší než režimu spánku. Uzly zde měly největší spotřebu energie. Ve druhém kole byly parametry nastaveny tak, aby byly oba režimy stejně dlouhé, oba režimy měly 100 ms. Spotřeba zde klesla zhruba o jednu třetinu. Ve třetím a čtvrtém kole byl režim spánku 9x delší než režim naslouchání s rozdílem, že ve čtvrtém kole byly oba režimy nastaveny co nejkratší. Tento poměr se značně projevil na spotřebě, která je pro tyto kola srovnatelná, ale v porovnání s prvním kolem se snížila více než o polovinu. Při čtvrtém kole byly režimy dost krátké, takže v méně hustých sítích by si uzly nemusely předat všechny potřebné hodnoty, což je nežádoucí.

Tab. 7.2: Průměrná spotřeba komunikačních jednotek pro MAC protokoly

protokol	Mica2	Imote2
	spotřeba [J]	
S-MAC	0,5165	0,8735
T-MAC	0,4262	0,6144

Druhý experiment se zaměřuje na protokoly S-MAC a T-MAC, jejichž detailní popis je v kap. 2. Graf s průměrnou spotřebou uzlů je na obr. 7.2. Zde se potvrdily teoretické předpoklady a protokol T-MAC je skutečně energeticky šetrnější. S-MAC, jenž je jeho předchůdcem, ovšem také nevykazuje špatné hodnoty. Zde je třeba si uvědomit, že síť je přehustěná a na malém prostranství. V sensorové síti o rozloze několik kilometrů čtverečních, kde je rozmístěno několik stovek sensorových uzlů by se tyto rozdíly navyšovaly z důvodu větší vzdálenosti mezi uzly, vznikajících interferencí, atd. Spotřeba uzlů Imote2 je vyšší než při použití Mica2. Imote2 v kombinaci s T-MAC protokolem vykazuje menší rozdíl spotřeby, což opět upřednostňuje tento protokol před S-MAC protokolem. V dalším experimentu jsou dále porovnány oba protokoly z hlediska spojitosti sítě.

Ve třetím experimentu, který se soustředí na výzkum konektivity mezi jednotlivými uzly a tak i celé sítě, byly zvoleny odlišné parametry sítě než při předchozích dvou pokusech. Uzly byly rozmístěny ručně, aby byly výsledky více vypovídající. Konektivita byla porovnána pro výše prověřované MAC protokoly S-MAC a T-MAC a dále srovnána i pro obě komunikační jednotky. V tabulce 7 jsou vyobrazeny výsledky pro jednotlivé uzly. Pro každý uzel je zde počet připojených ostatních uzlů v síti. Dále aritmetický průměr počtu přijatých paketů od jednotlivých uzlů. Kompletní výsledky jsou na příloženém DVD ve výstupním souboru Castalie. Protokol



Obr. 7.2: Průměrná spotřeba energie pro MAC protokoly

T-MAC znovu potvrdil svoje vlastnosti, které jsou značně lepší než pro S-MAC. Počet přijatých paketů byl pro obě komunikační jednotky vždy mnohonásobně vyšší, což přispívá k celkové spojitosti sítě. Ke všemu se v předchozím pokusu ukázal jako energeticky méně náročný, což je dále zvýhodňuje oproti staršímu S-MAC protokolu. K jeho výhodám patří zejména výborná dosažitelnost ostatních uzlů. Při větší pravděpodobnosti přenosu se také nemusí jednotlivé přenosy opakovat, což vede k dalším úsporám.

Ve všech třech simulacích byly porovnávány komunikační jednotky. V prvním a druhém experimentu se projevila jednotka Imote2 jako více energeticky náročná, ale při nastavení parametrů MAC vrstvy na vhodnější hodnoty spotřeba klesala. Mica2 vždy měla nižší spotřebu. Vysílací výkony nebyly u prvních dvou pokusů nastaveny stejně, ale u Imote2 byl zvolen o 3dB nižší vysílací výkon, přičemž naměřenou hodnotu od uzlu 6 obdržel stejný počet uzlů pro obě jednotky. Z toho plyne, že je tato jednotka výkonnější při přenosech dat i při nastaveném nižším vysílacím výkonu. Výhodou nižšího výkonu při zachování dobrých vlastností přenosu je, že v síti nedochází k interferencím. V posledním experimentu se po porovnání přijatých paketů pro jednotlivé jednotky opět potvrdila Imote2 jako výkonnější. Uzly 0 a 9, které jsou na okraji sítě se při využití jednotky Mica2 spojily s více ostatními uzly, ale počet přijatých paketů od vzdálených uzlů byl vždy téměř zanedbatelný, takže

Tab. 7.3: Výsledky třetího experimentu

jednotka:		Mica2		Imote2	
protokol	uzel	počet připojených uzlů	počet přijatých paketů	počet připojených uzlů	počet přijatých paketů
S-MAC	0	8	4,125	5	19
	1	7	4,857	6	15,66
	2	6	5	6	11,17
	3	8	4,875	7	10,86
	4	9	3,44	9	11,78
	5	8	7,25	7	11,14
	6	9	5,55	7	14,43
	7	8	6	8	15,125
	8	7	4,28	5	11,6
	9	6	4,83	5	11,4
T-MAC	0	8	30	5	62
	1	7	38,43	6	56
	2	6	41,83	6	65
	3	7	43,285	8	52,125
	4	8	42,5	9	60,55
	5	8	36,875	8	52,75
	6	8	38,125	8	62,875
	7	7	47,71	8	72,375
	8	7	34	6	48,83
	9	6	41	5	63

tuto vlastnost nelze považovat za výhodu oproti druhé komunikační jednotce. Počet takto přijatých paketů se pohyboval od 2 do 3. Při porovnávání jednotek Castalie nebere ohled na jejich výpočetní výkon. Imote2 je technologicky vyspělejší, tudíž zpracovává data s větší rychlostí a tím může rychleji přecházet do režimu spánku. Dále jednotka sama reguluje frekvenci procesoru podle náročnosti prováděné operace a režimu, ve kterém se nachází. Odběry proudu pro různé režimy jsou uvedeny v tab. A.2. Další nespornou výhodou je možnost připojení externí antény pro zvýšení dosahu jednotlivých uzlů, což je výhodné při aplikaci do rozsáhlých oblastí s menší hustotou rozmístění.

8 ZÁVĚR

Cílem této bakalářské práce bylo nastudovat teorii bezdrátových sensorových sítí a simulovat komunikační schémata v simulačním nástroji OMNeT doplněném o program Castalia, který je určen přímo pro simulace WSN.

Práce začíná popisem obecných vlastností bezdrátových sensorových sítí a porovnává tyto vlastnosti s klasickými bezdrátovými sítěmi. Dále je rozebrána problematika kontroly přístupu k médiu a požadavky sensorových sítí na této úrovni. Jsou zde uvedeny dva protokoly řešící přístup k médiu. S-MAC byl jeden z prvních protokolů pro tyto účely a jeho spotřeba elektrické energie je na velice dobré úrovni. Druhým zmíněným protokolem je T-MAC, který vylepšuje S-MAC ve zkracování pracovního cyklu při nečinnosti na bezdrátovém kanálu, čímž ještě více snižuje spotřebu energie.

V další kapitole je úvod do směrovacích technik pro sensorové sítě a jsou zde uvedena úskalí při návrhu protokolů pro tyto účely. Pokračování je v podobě popisu směrovacích protokolů dělených dle struktury sítě.

Poslední část práce se zabývá simulacemi v simulačním programu Castalia, který ke svému běhu využívá OMNeT. Byly provedeny tři experimenty. První dva se zabývají porovnáním různých parametrů na linkové vrstvě. V prvním experimentu se pomocí laditelného modulu nastavovaly různé délky pracovních cyklů, podle kterých uzly přecházely do režimu spánku nebo naslouchání. Při kratší době režimu spánku uzly vykazovaly vyšší spotřebu, která se snižovala zkrácením aktivního režimu. Ve druhém experimentu byly porovnány dva využívané protokoly na linkové vrstvě. Zde se potvrdily teoretické předpoklady a protokol s aktivním nastavováním délky pracovního režimu byl skutečně úspornější. V posledním pokusu byly tyto protokoly znova otestovány, ale z hlediska spojitosti sítě. Opět se jako lepší projevil protokol T-MAC.

Ve všech simulacích byly vždy porovnány dvě komunikační jednotky. Moderní Imote2, kterou lze využít jako základnovou stanici i jako měřící uzel, se ukázala lepší v mnoha ohledech. Sice při mírně vyšší spotřebě než Mica2, ale její výhody stále převyšují.

LITERATURA

- [1] AL-KARAKI J.N.; KAMAL A.E. *Routing Techniques in Wireless Sensor Networks: A Survey* [online]. Dept. of Electrical and Computer Engineering Iowa State University, 2004 [cit. 2009-15-11]. Dostupné z URL: <<http://www.ece.iastate.edu/~kamal/Docs/kk04.pdf>>.
- [2] BOULIS A. *Castalia A simulator for Wireless Sensor Networks Version 2.3 User's Manual* [online]. NICTA, říjen 2009 [cit. 2010-20-4]. Dostupné z URL: <<http://castalia.npc.nicta.com.au/pdfs/Castalia - User Manual.pdf>>.
- [3] *CC2420, 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver* [online]. Texas Instruments, 2006 [cit. 2010-15-4]. Dostupné z URL: <http://enaweb.eng.yale.edu/drupal/system/files/CC2420_Data_Sheet_1_4.pdf>.
- [4] *Installation directions for Castalia in Linux - How to Install Castalia* [online]. ©2004 Dostupné z URL: <<http://castalia.npc.nicta.com.au/installationLinux.php>>.
- [5] Crossbow Technology, Inc. *Imote2 datasheet* [online]. Document Part Number: 6020-0117-02 Rev A, [cit. 2010-4-23]. Dostupné z URL: <http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/Imote2_Datasheet.pdf>.
- [6] Crossbow Technology, Inc. *MICA2 datasheet* [online]. Document Part Number: 6020-0042-08 Rev A, [cit. 2009-3-12]. Dostupné z URL: <http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf>.
- [7] IBRIQ J.; MAHGOUB I. *Cluster-based routing in wireless sensor networks: issues and challenges* [online]. Department of Computer Science and Engineering Florida Atlantic University [cit. 2009-15-11]. Dostupné z URL: <<http://www.scs.org/getDoc.cfm?id=2888>>.
- [8] INTANGONWIWAT C. *Directed Diffusion for Wireless Sensor Networking* [online]. 2002 [cit. 2009-15-11]. Dostupné z URL: <<http://www.isi.edu/~johnh/PAPERS/Intanagonwiwat02b.pdf>>.
- [9] KREDO K.; MOHAPATRA P. *Medium Access Control in Wireless Sensor Networks* [online]. Electrical and Computer Engineering Department, University of California, Davis. 29.6.2006 [cit. 2009-15-11]. Dostupné z URL: <<http://www.cs.ucdavis.edu/~prasant/pubs/journal/SensorMac.pdf>>.

- [10] LANGENDOEN K. *Medium Access Control in Wireless Networks, Volume II: Practice and Standards* [online]. 2008 [cit. 2009-15-11]. Dostupné z URL: <<http://www.st.ewi.tudelft.nl/~koen/papers/mac4wsn.pdf>>.
- [11] *OMNeT++ 3.x and 4.x on Linux and other Unix-like systems* [online]. Page last modified on May 26, 2009, at 11:44 AM Dostupné z URL: <<http://www.omnetpp.org/pmwiki/index.php?n=Main.InstallingOnUnix>>.
- [12] PERILLO M.A.; HEINZELMAN W. B. *Wireless Sensor Network Protocols* [online]. Department of Electrical and Computer Engineering University of Rochester, 6.2. 2005 [cit. 2009-15-11]. Dostupné z URL: <<http://www.ece.rochester.edu/courses/ECE586/readings/perillo.pdf>>.
- [13] SZYNKIEWICZ E.; KWAŚNIEWSKI P.; WINDYGA I. *Comparative Study of Wireless Sensor Networks Energy-Efficient Topologies and Power Save Protocols* [online]. 2009 [cit. 2009-15-11]. Dostupné z URL: <<http://www.itl.waw.pl/czasopisma/JTIT/2009/3/68.pdf>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ACK	Acknowledgement - potvrzení
ADV	Advertise – oznámení
APTEEN	Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol – směrovací protokol
BER	bit error rate – bitová chybovost
BS	Base Station – základová stanice
CPU	Central Processing Unit – procesor
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance – metoda s vícenásobným přístupem
CSMA/CA	Code Division Multiple Access – mnohonásobný přístup pomocí kódování
$D_i(t)$	vzdálenost bodu p od i -tého zdroje v čase t
η	exponent útlumu
FSK	Frequency-shift keying – frekvenční klíčování
GAF	Geographic Adaptive Fidelity – směrovací protokol
GEAR	Geographic and Energy Aware Routing – směrovací protokol
GPS	Global Positioning System – globální polohový systém
LEACH	Low Energy Adaptive Hierarchy – směrovací hierarchický protokol
MAC	Medium Access Control – Kontrola přístupu k médiu
MIMO	multiple-input and multiple-output – použití více vysílačích a přijímacích antén u bezdrátové komunikaci
PEGASIS	Power-Efficient Gathering in Sensor Information Systems – směrovací protokol
$PL(d)$	útlum trasy se vzdáleností uzlů d
$PL(d_0)$	známý útlum na referenční vzdálenosti d_0
PSK	Phase-shift keying – fázové klíčování

QoS	Quality of Service – kvalita služeb
REQ	Request – žádost
RSSI	Received Signal Strength Indication – kvalita přijatého signálu
SNR	signal-to-interference ratio – poměr signál/rušení
S-MAC	Sensor-MAC – MAC protokol
SNR	Signal-to-noise ratio – poměr signál/šum
SPIN	Sensor Protocols for Information via Negotiation – směrovací protokol
T-MAC	Timeout-MAC – MAC protokol
TDMA	Time Division Multiple Access – časový multiplex
TEEN	Threshold-sensitive Energy Efficient sensor Network Protocol – směrovací protokol
$V_i(t)$	hodnota i -tého zdroje v čase t
$V(p, t)$	hodnota fyzického procesu v bodě p a v čase t
WSN	bezdrátová senzrorová síť – Wireless sensor network
X_σ	náhodná proměnná s Gaussovým rozložením v okolí nuly se směrodatnou odchylkou σ
ZH	Změřená Hodnota

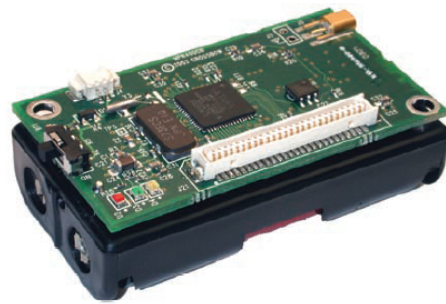
SEZNAM PŘÍLOH

A	Komunikační jednotky	58
A.1	Mica2	58
A.2	Imote2	60
B	Obsah přiloženého DVD	62

A KOMUNIKAČNÍ JEDNOTKY

A.1 Mica2

Mica2 (obr. A.1) je již třetí generací jednoduchých bezdrátových komunikačních uzlů pro senzorové sítě. Tento model je navržen speciálně pro hustě osazené sítě a sítě s počtem uzlů nad 1000. Obsahuje 868/916 MHz multikanálový rádiový přijímač s datovým tokem 38,4 kbps. Modul obsahuje rozšiřující konektor pro připojení rozhraní pro měření mnoha veličin (tlak, teplota, světlo, seismické změny). Uzly pracují na základech operačního systému TinyOS s otevřeným kódem. MPR400 platforma procesoru a vysílací části je založena na nízko energetickém mikrokontroléru Atmel ATmega128L. Jednotlivé parametry jsou uvedeny v tabulce A.1. Kterýkoliv uzel Mica2 může pracovat jako základová stanice, pokud je osazen rozšiřujícím rozhraním s USB portem. [6]



Obr. A.1: Mica2

Tab. A.1: Vlastnosti Uzlu Mica2 [6]

Processor/Radio Board	MPR400CB	Remarks
Processor Performance		
Program Flash Memory	128K bytes	
Measurement (Serial) Flash	512K bytes	>100000 Measurements
Configuration EEPROM	4K bytes	s
Serial Communications	UART	0-3V transmission levels
Analog to Digital Converter	10 bit ADC	8 channel, 0-3V input
Other Interfaces	DIO,I2C,SPI	
Current Draw	8 mA	Active mode
	< 15 μ A	Sleep mode
Multi-Channel Radio		
Center Frequency	868/916 MHz	ISM bands
Number of Channels	4/50	Programmable, country specific
Data Rate	38.4 Kbaud	Manchester encoded
RF Power	-20 to +5 dBm	Programmable, typical
Receive Sensitivity	-98 dBm	Typical, analog RSSI at AD Ch. 0
Outdoor Range	500 ft	1/4 Wave dipole, line of sight
Current Draw	27 mA	Transmit with maximum power
	10 mA	Receive
	< 1 μ A	Sleep
Electromechanical		
Battery	2X AA batteries	Attached pack
External Power	2.7 - 3.3 V	Connector provided
User Interface	3 LEDs	User programmable
Size [mm]	58 x 32 x 7	Excluding battery pack
Weight [g]	18	Excluding batteries

A.2 Imote2

Imote2 (obr. A.2) je pokročilá jednotka pro široké využití ve WSN. Je postavena na nízko energetickém PXA271 XScale procesoru od výrobce Intel. Takt procesoru může být v rozsahu od 13 MHz do 416 MHz s dynamickým nastavovaným napájením. Při nižších takttech procesoru jsou prováděny operace s minimální spotřebou energie, což přispívá k úsporám energie. Procesor obsahuje 256 kB SRAM, 32 MB SDRAM a 32 MB FLASH paměti. Rádiový přijímač CC2420 od Texas Instruments vyhovuje standardu IEEE 802.15.4, datový tok je 250 kbps a přijímač pracuje v pásmu 2,4 GHz při využití 16 kanálů. Imote2 má integrovanou anténu, která zajišťuje dosah přibližně 30 m. Pro větší dosah je možnost napájet SMA konektor přímo na desku plošných spojů pro připojení externí antény, čímž se dosah rapidně zvětší. Jednotka je napájena třemi AAA bateriemi, které jsou umístěny v držáku na napájecí desce připojené k jednotce. [5]



Obr. A.2: Imote2

Tab. A.2: Vlastnosti Uzlu Imote2 [5]

Processor/Radio Board	IPR2400	Remarks
CPU		
Processor	Intel PXA271	
SRAM Memory	256 kB	
SDRAM Memory	32 MB	
FLASH Memory	32 MB	
POWER CONSUMTION		
Current Draw In Deep Sleep Mode	390 μ A	
Current Draw in Active Mode	31 mA	13 MHz, radio off
Current Draw in Active Mode	44 mA	13 MHz, Tx/Rx
Current Draw in Active Mode	66 mA	104 MHz, radio Tx/Rx
Radio		
Transceiver	TI CC2420	
Frequency Band (ISM)	2400.0 – 2483.5 MHz	
Data Rate	250 kb/s	
Tx Power	-24 – 0 dBm	
Rx Sensitivity	30 m	(integrated antenna)
Power		
Battery board	3x AAA	
USB Voltage	5.0 V	
Battery Voltage	3.2 – 4.5 V	
Mechanical		
Dimensions Imote2 Board	36 mm x 48 mm x 9 mm	
Weight	12 g	

B OBSAH PŘILOŽENÉHO DVD

Přiložené DVD obsahuje mimo elektronické verze této práce i následující dva adresáře:

- **Vysledky** – obsahuje tři podadresáře pro jednotlivé simulace. V každém adresáři je vždy výstupní soubor z Castalie s názvem `Castalia-Primary-Output.txt`. Dále debugovací soubor `Castalia-Debug.txt` a soubor `omnetpp.ini`, kterým se celá simulace konfigurovala. V adresáři třetí simulace je navíc soubor, který definuje rozmístění uzlů v síti s názvem `node_locations.ini`.
- **Parameter Include Files** – v této složce jsou všechny konfigurační soubory programu Castalia. Jsou zde i ty, které nebyly při simulacích využity.